# Experimental relativistic zero-knowledge proofs

Pouriya Alikhani,[1] Nicolas Brunner,[2] Claude Crépeau,[1] Sébastien Designolle,[2,*]
Raphaël Houlmann,[2] Weixu Shi,[2,3] and Hugo Zbinden[2]

[1]*School of Computer Science, McGill University, Montréal, Québec, Canada*
[2]*Department of Applied Physics, University of Geneva, 1211 Genève, Switzerland*
[3]*Department of Electronic Science, National University of Defense Technology, 410073 Changsha, China*

(Dated: December 19, 2020)

Protecting secrets is a key challenge in our contemporary information-based era. In common situations, however, revealing secrets appears unavoidable, for instance, when identifying oneself in a bank to retrieve money. In turn, this may have highly undesirable consequences in the unlikely, yet not unrealistic, case where the bank's security gets compromised. This naturally raises the question of whether disclosing secrets is fundamentally necessary for identifying oneself, or more generally for proving a statement to be correct. Developments in computer science provide an elegant solution via the concept of zero-knowledge proofs: a prover can convince a verifier of the validity of a certain statement without facilitating the elaboration of a proof at all. In this work, we report the experimental realisation of such a zero-knowledge protocol involving two separated verifier-prover pairs. Security is enforced via the physical principle of special relativity, and no computational assumption (such as the existence of one-way functions) is required. Our implementation exclusively relies on off-the-shelf equipment and works at both short (60 m) and long distances (400 m) in about one second. This demonstrates the practical potential of multi-prover zero-knowledge protocols, promising for identification tasks and blockchain-based applications such as cryptocurrencies or smart contracts.

*Introduction.*— In a foreign city where you know absolutely no one, you go to an automatic teller machine to obtain a handful of local cash. You have never heard of the bank owning that teller machine, yet when requested for your Personal Identification Number to obtain money you blindly provide it. No joke, you give away that super unique information to a complete stranger. But why? Because of the cash you get in return? There is actually zero solid reason to trust that teller machine. You should never have to give away this private information to anyone at all! But how could we prove who we are without giving away such a secret piece of data?

The idea behind zero-knowledge proofs was born in the middle of the 1980's [1] and formalises the possibility to demonstrate knowledge of a secret information without divulging it. A natural application is the task of identification, where a user can demonstrate their identity via the knowledge of a secret proof of a mathematical statement they created and published. A well-known example is the RSA cryptosystem [2] in which the mathematical secret is the factorisation into two huge prime numbers of an even larger number. In this work we consider the problem of three-colouring of graphs: an instance is a graph (nodes and edges attaching some of them to one another) and a proof of three-colourability assigns to each vertex one out of three possible colours in a way that any two vertices connected by an edge have different colours, see Fig. 1(a). Some graphs are three-colourable, some are not, and the general problem of deciding whether a graph is three-colourable has no known efficient solution. However, given a colouring it is extremely easy to efficiently check whether the end points of every edge are assigned different colours. For this reason, three-colourability is a problem in **NP**, the class of all problems that are efficiently verifiable given a solution. Moreover, it is also **NP**-complete because an instance of any problem in **NP** can be efficiently simulated by an instance of three-colourability, so that if this latter were in **P**, the class of all problems efficiently solvable, then we would have **P** = **NP**, an equality which has been the most famous challenge of theoretical computer science for the last half century and which remains unsolved.

A zero-knowledge proof for three-colourability has been introduced in Ref. [3] by assuming the existence of one-way functions, that is, functions that can be efficiently computed but for which finding a preimage of a particular output cannot. The zero-knowledge proof guarantees that upon participation to such an interaction, a prover would convince a verifier of the validity of the statement when it is indeed valid (completeness), would not convince the verifier when it is invalid (soundness), while not allowing the latter to improve their ability to find a three-colouring (zero-knowledge), but this is under the assumption that one-way functions exist. It is widely believed that a zero-knowledge proof for any **NP**-complete problem such as three-colourability is not possible without this extra computational assumption. If not, this would lead to vast implications in the world of complexity [4]. However, this feature is generally undesirable as it significantly weakens the long-term security of such zero-knowledge protocols, which are used, e.g., in certain crypto-currencies [5]. This may have important consequences, as security would be fully compromised if the specific one-way function used in the protocol is (later) found to be efficiently invertible. This aspect is particularly relevant given recent advances on quantum computing [6, 7].

[*] sebastien.designolle@unige.ch

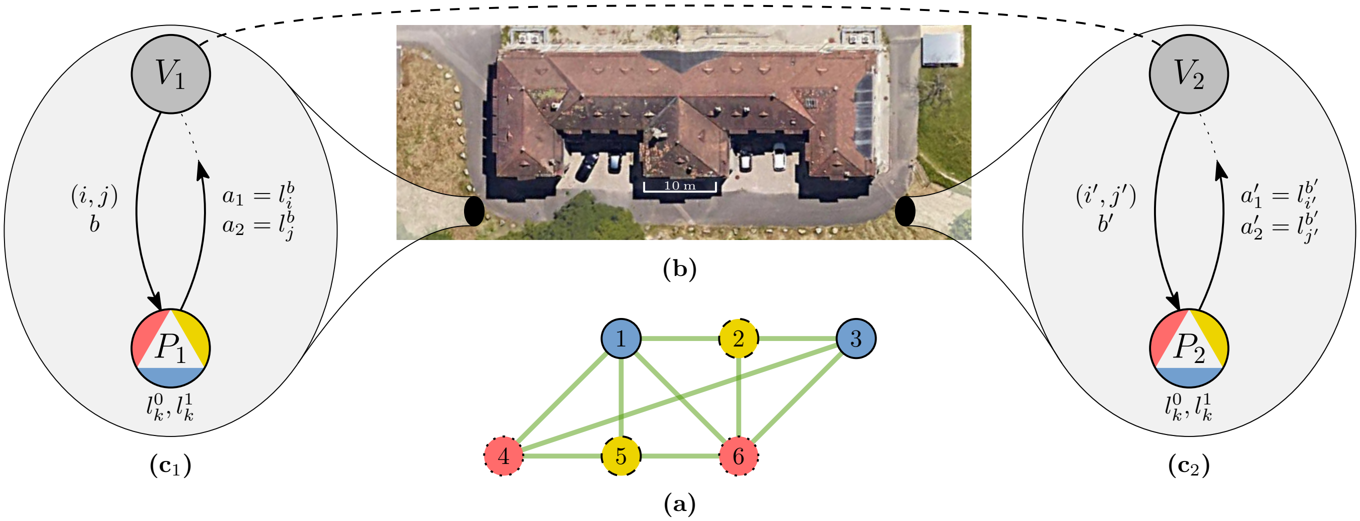

FIG. 1. **Relativistic zero-knowledge protocol for three-colourability on a short distance.** Two separated provers try to convince a verifier that they know a given graph is three-colourable without facilitating the elaboration of a three-colouring. **(a)** A three-colourable graph with six vertices and ten edges. The three-colouring depicted here is such that $c_1 = c_3 = 0$ (full blue), $c_2 = c_5 = 1$ (dashed yellow), $c_4 = c_6 = 2$ (dotted red); vertices linked by an edge are indeed of different colours. **(b)** Satellite view [8] of the building of the experiment. The distance between the two parties involved is 60 m, that is, 200 ns at the speed of light. This separation makes the communication between the two provers impossible at this time scale due to special relativity. The verifiers simultaneously trigger their questions to their provers by means of an optical fibre (dashed line). **(c)** Illustration of a round of the protocol on both verifier-prover pairs. Each verifier sends (downward arrow) an edge and a bit $b$ to their prover, who should answer (upward arrow) their $b$th labellings at the end points of the edge: for all vertex $k$ the provers have indeed pre-agreed on two labellings $l_k^0, l_k^1 \in \{0, 1, 2\}$ that should sum up to a three-colouring, namely, $l_k^0 + l_k^1 \equiv c_k \pmod 3$. When asking the same edge on both sides and opposite bits, the verifiers can check, thanks to the definition of the labellings, that the provers know that the graph is three-colourable. To make sure that the provers are not cheating the verifiers can also send the same bit with edges sharing (at least) one vertex; the consistency of the provers' answers can then be tested. By repeating this procedure many times the verifiers can make the probability for dishonest provers to pass the protocol arbitrarily small (soundness). However, even with all the provers' answers at hand, the verifiers are not more efficient at elaborating a three-colouring than initially (zero-knowledge).

Remarkably, it is possible to devise zero-knowledge protocols without the need of any computational assumption. The key idea, as developed by Ben-Or, Goldwasser, Kilian and Wigderson [9], is to generalise the interactive proof model such that *several* provers are now trying to convince a verifier of the three-colourability of a graph in perfect zero-knowledge without the need of any further assumption. Intuitively, this approach reflects the strategy used by police investigators when interrogating suspects in separate rooms in order to discern the truth more easily: it is harder to collectively lie about the validity of a statement when interrogated *separately*. The key difference between the multi-prover scenario and the original definition of interactive proof rests in the possibility to prevent several provers from talking to each other, a single prover always being able to talk to themself. This naturally suggests the use of spatial separation to enforce the impossibility to communicate [10, 11], at least for some short period of time: assuming the principle of special relativity (nothing can signal faster than the speed of light) and sending queries to the different provers simultaneously, there is a short time window during which they are physically unable to signal between each other. So far, these ideas have been mainly of purely theoretical interest, as known protocols required extremely large information transfer between the provers and verifiers, which prohibited their implementation.

In this work, we report experimental realisations of relativistic zero-knowledge proofs for an **NP**-complete problem (three-colourability). Specifically we develop an efficient implementation of the protocol recently established in Ref. [12] for two separated verifier-prover pairs. In practice, key challenges involve the generation of adequate large three-colourable graphs, as well as an efficient management of the randomness shared between the provers, achieved via suitable error-correcting codes. We report on two experiments: first, using Global Positioning System (GPS) clocks to synchronise the two verifiers, we performed the protocol at a distance of 400 m; second, using a triggering fibre between the two verifiers, we conducted the same test at a shorter distance of 60 m. In both cases, the full running time was about one second. The first implementation shows that the protocol at large distances is rather effortless since the wide relativistic separation only demands a moderate speed on the provers' side; the second one demonstrates a clear potential for serviceable applications. Importantly, the security is enforced by relativistic constraints, and does not rely on any computational hypothesis such as the existence of one-way functions. Note that the aforementioned **NP**-completeness guarantees that any application based on a problem in **NP** can be (polynomially) cast into an instance of our protocol. For example, if you trust the Advanced Encryption Standard (AES) as a secure cryptographic primitive, you can transform AES instances into three-colourable graphs. Our implementation achieves security against classically correlated provers and we discuss the prospects of extending the security to the general case of quantum-mechanically correlated provers below.

*Protocol.—* We start by presenting the type of zero-knowledge proof that we used in the experiment. Let $(V, E)$ be a finite undirected graph, namely, a finite set $V$ of vertices and a collection $E$ of edges, that is, unordered pairs of (distinct) vertices. We further assume that this graph is three-colourable, see Fig. 1(a). In the following we denote the three different colours by 0, 1, and 2.

The protocol involves two verifiers and two provers located in a suitable way discussed below. Initially the two provers pre-agree on random three-colourings $c_k(n) \in \{0, 1, 2\}$ for $k \in V$ and $n$ identifying the round. In the following, the dependency in $n$ will be omitted for conciseness. For all vertex $k$ they also choose two labellings $l_k^0$ and $l_k^1$ such that the equality $l_k^0 + l_k^1 \equiv c_k \pmod 3$ holds. Note that, contrary to a proper colouring, the labellings $l^0$ and $l^1$ do not need to have different values on adjacent vertices. A round is then illustrated in Fig. 1(c) and consists in (i) each verifier providing their prover with an edge $(i, j) \in E$ and a bit $b \in \{0, 1\}$, (ii) each prover answering $l_i^b$ and $l_j^b$, and (iii) the two verifiers checking the provers' answers as described in the next two paragraphs. If none of the parties abort the protocol, then we repeat rounds until a certain security level is reached, see below. The verifiers' tests follow two different paths.

On the one hand, the verifiers can check that the provers do indeed know that the graph is three-colourable. This test is done when both verifiers send the same random edge $e = (i, j) = (i', j') = e' \in E$ and when $b \neq b'$. Then the answers $(a_1, a_2)$ and $(a_1', a_2')$ of the two provers are accepted if and only if $a_1 + a_1' \not\equiv a_2 + a_2' \pmod 3$. Clearly this aims at ensuring that the provers know that the graph is three-colourable.

On the other hand, the verifiers can test the consistency of the provers' answers. When the edges sent share at least one vertex (say, $i = i'$) and when the bits sent are equal ($b = b'$), then the verifiers accept if and only if the corresponding answers of the two provers are equal ($a_1 = a_1'$). This test typically prevents the provers from answering in a way that would ignore the edges asked but would only aim at passing the previous check.

For honest verifiers and honest provers (when the graph is three-colourable), it is easy to see that following the protocol will always lead to acceptance. This property of the protocol is referred to as *completeness.*

For honest verifiers and dishonest provers (when the graph is not three-colourable), the *soundness* refers to the verifiers being able to reveal the cheat with very high probability when performing many rounds. Intuitively, if the answers of a prover (say, $P_2$) reach their corresponding verifier ($V_2$) before the question of the other one ($V_1$) could have, by any means, made its way there, then this prover ($P_2$) must have answered without knowing what the other one ($P_1$) has been asked. By separating the verifier-prover pairs by a sufficient distance and by timing the protocol carefully, we can use special relativity to create this separation in order to make the protocol sound against classically correlated provers. We discuss the case of quantum provers below.

For dishonest verifiers (trying to get any knowledge of a three-colouring) and honest provers, the *zero-knowledge* property amounts to the verifiers getting no knowledge whatsoever upon interaction with the provers. The above protocol satisfies this property [12].

From the cases described above, we get the main features of a good strategy for the verifiers to detect cheating provers. Typically, asking edges with no vertex in common is of no interest and the two tests described above should be somehow balanced. When we fix the strategy adopted by the provers, the probability for cheating provers to pass one round can be computed and from there the number of rounds required to reach a given security level. For the protocol that we implemented, that is, the one from Ref. [12] (of which the one presented above is a pedagogical variant), a good strategy is given in Methods 1 together with a reminder of the details of the protocol. With this strategy, when the number of rounds is $9|E|k$, where $|E|$ is the number of edges in the graph, classically correlated provers can dishonestly pass the soundness tests with probability at most $\mathrm{e}^{-k}$.

*Graphs.—* From a theoretical perspective, the three-colourability problem is **NP**-complete. This means that *finding* a solution (a three-colouring) is not known to be possible in polynomial complexity with respect to the size of the problem (typically the number of vertices $|V|$) but *checking* a solution can be made in polynomial complexity. Moreover, the fact that this problem is **NP**-complete, means that it can simulate in polynomial complexity any other **NP** problem.

For the implementation we need a concrete graph together with a three-colouring of it. Here comes a complication: though the general problem is "hard", there exist algorithms efficient in many cases. In order to overcome this difficulty we use sufficiently large *critical* graphs. A four-critical graph is not three-colourable but is such that the deletion of any edge gives rise to a valid three-colouring for the resulting graph. Ref. [13] proposes an algorithm to build large critical graphs corresponding to very hard instances of three-colourability, a fact corroborated by extensive experimental evidence. However, no method for generating a three-colouring on the way is provided therein, so that we adapted the technique so that is suits our needs; see Methods 2. The graph used in the following experiments has $|V| = 588$ vertices and $|E| = 1097$ edges, so that reaching a security parameter of $k = 100$, widely considered safe [14], takes about one million rounds.

*Implementation.*— For the realisation of this zero-knowledge protocol, we used two verifier-prover pairs and we implemented the protocol from Ref. [12], recalled in Methods 1. The critical part dwells on the speed of the answer on the provers' side; therefore they were operated on field-programmable gate-arrays (FPGA) to reduce the communication latency, speed up the computation, and improve its time reliability. On the verifiers' side, FPGAs were also used for communication, together with standard computers for global monitoring and checking of the answers; see Methods 3 for details of the hardware, which builds upon techniques developed for the implementation of bit commitment [15].

As the protocol involves a significant number of rounds and requires the provers to share in advance some randomness, this resource must be used sparingly. For instance, it is easy to see that, starting from a shared three-colouring, storing a single permutation of three elements is enough to draw a random three-colouring in each round. Regarding the remaining shared randomness needed in the protocol, it requires at first sight one random trit per vertex and per round, which is not affordable. On second thought only four (two per prover) may suffice but for this the provers should know which question their partner was asked, i.e., which trits were "consumed", which is not possible. Drawing a connection with error-correcting codes [16, 17] we could nonetheless overcome this difficulty; see Methods 4.

Note that two timescales are involved in the experiment: the speed of the exchange between the verifiers and the provers and the repetition rate of the rounds. The former fixes the minimum distance required between the two locations and is limited by the speed of computation on the provers' side; the latter determines the time that the protocol takes to reach a given security parameter.

In the next sections, we explain our implementations of the protocol in two complementary spatial domains.

*Long-distance.*— The two verifier-prover pairs are placed in different buildings on the campus at 390 m from one another, corresponding to a time separation of 1.3 µs. The synchronisation relies on GPS clocks as in Ref. [18]. Both verifiers send to their neighbouring prover a stream of challenges at a frequency of 0.3 MHz. As soon as they receive a challenge the provers compute their answers based on their shared data, see Fig. 1(c). Taking into account the imprecision of the system used, the total time elapsed between the emission of the verifiers' challenges and the reception of the provers' answers is 840 ns, which is below the 1.3 µs time separation between the parties, thus fulfilling the soundness requirement. The whole protocol with one million rounds runs in about 3 s.

Note that the theoretical minimum distance between the verifiers is fixed by the 840 ns in which the provers respond and is thus about 250 m. Also there is no upper bound for this distance since the two verifier-prover pairs are disconnected in this case. Applications involving faraway actors may be designed based on this simple protocol [15]: as the distance between the verifier-prover pairs increases, the one between the verifier and the prover within a pair becomes less constrained. Typically verifier-prover pairs widely separated would allow the provers to be anywhere in the verifiers' cities.

*Short-distance.*— The two verifier-prover pairs are placed on two tables outside of the university building at 60 m from one another, corresponding to a time separation of 200 ns, see Fig. 1(b). A trigger signal is exchanged between the two verifiers. When the trigger is captured by the second verifier, it sends a challenge to the prover it is connected to. The first verifier delays the emission of their challenge by the time the trigger will take to be transferred to the second verifier. So both verifiers send to their neighbouring prover a stream of challenges. Again, as soon as they receive a challenge the provers compute their answers based on their shared data and send them back to the verifiers. Altogether a round is achieved in a maximum 192 ns, thus constraining the verifier to be at a minimal distance of 57.6 m; see Methods 3 for details. With a repetition rate of 3 MHz the whole protocol with one million rounds runs in less than a second. Note that with the hardware used and its time and memory limitations, it would still be possible to gain an order of magnitude in the size of the graphs, namely, $|V| \sim 5000$ and $|E| \sim 10000$, while keeping the same security parameter $k = 100$ and a reasonable total time (about 3 s).

In our implementation the time needed for an exchange between the provers and the verifiers is mostly constrained by the latency of the hardware, primarily the one of the multi-gigabit transceivers used for the optical links. The computation of the provers' answers (memory look-up and calculation) is done in a single clock cycle (here 8 ns). A parallel communication with dedicated input/outputs could reduce the transfer time from and to the physical pins of the provers' FPGAs, adding no more than another clock cycle of delay, hence bringing the exchange time down to 16 ns. Moreover, implementing the scheme on a state-of-the-art application-specific integrated circuit (ASIC) technology would further reduce the clock cycle, thus the overall delay. Therefore it seems possible to run a full exchange in only a few nanoseconds so that the two verifier-prover pairs could eventually be placed about a meter away from each other.

*Quantum provers.*— So far we have only considered the case of classically correlated provers. However, in general, it would be desirable to extend the security to the case of quantum provers. This is because they could establish stronger correlations compared to classically correlated ones, a phenomenon known as quantum nonlocality [19, 20]. In principle there exist protocols that are secure against such quantum provers [21]; they typically rely on a third verifier-prover pair [22] or on the extension of the graph under study [23]. Concerning our protocol, it is at the moment unclear whether it remains secure in the case of two quantum provers, but the adaptations just mentioned are out of reach for now. On the one hand, building on the results from Ref. [22] on the addition of a third verifier-prover pair, it was shown in Ref. [12] that the number of rounds for which security can currently be proven is about $(21|E|)^4k$, completely unpractical for graphs of reasonable size. On the other hand, Ref. [23] gives a method to inflate the graph into an only quadratically bigger one for which security against quantum provers can be demonstrated; however, classical and quantum security are not linked therein so that the number of rounds required remains unknown. Therefore, in all cases, improving theoretical proofs clearly represents the key challenge.

*Conclusion.*— We have demonstrated that a relativistic zero-knowledge proof for the **NP**-complete problem of three-colourability is possible in practice, even for small distances. For the example mentioned in the introduction, one could thus conceive a teller machine with two separate ports; customers may then simply spread their arms and insert a pair of chips to identify themselves by proving they know their (public) graph is three-colourable. Given the simplicity of the operations on the provers' side in our protocol, these chips may furthermore be integrated in (two) cell phones. More generally, these ideas may find applications in a wide range of areas where the concept of zero-knowledge is relevant, such as blockchain systems and smart contracts [24], electronic voting and auctions [25, 26], as well as nuclear warhead verification [27].

## ACKNOWLEDGMENTS

Financial supports by the Swiss National Science Foundation (Starting grant DIAQ, NCCR-QSIT) and the European project OpenQKD are gratefully acknowledged. C. C. is supported in part by Québec's FRQNT.

[1] S. Goldwasser, S. Micali, and C. Rackoff. *The knowledge complexity of interactive proof systems*. Siam J. Comput. **18**(1), 186–208 (1989).

[2] R. L. Rivest, A. Shamir, and L. Adleman. *A method for obtaining digital signatures and public-key cryptosystems*. Commun. ACM **21**(2), 120–126 (1978).

[3] O. Goldreich, S. Micali, and A. Wigderson. *Proofs that yield nothing but their validity or all languages in* **NP** *have zero-knowledge proof systems*. J. ACM **38**(3), 690–728 (1991).

[4] L. Fortnow. *The complexity of perfect zero-knowledge*. Proc. STOC'87 pp. 204–209 (1987).

[5] E. Ben Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. *Zerocash: Decentralized anonymous payments from bitcoin*. Proc. IEEE Symp. Secur. Priv. pp. 459–474 (2014).

[6] D. J. Bernstein and T. Lange. *Post-quantum cryptography*. Nature **549**(7671), 188–194 (2017).

[7] F. Arute et al. *Quantum supremacy using a programmable superconducting processor*. Nature **574**(7779), 505–510 (2019).

[8] Google Maps. *Group of Applied Physics*. Chemin de Pinchat 22, 1227 Carouge, Switzerland (2020).

[9] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. *Multi-prover interactive proofs: how to remove intractability assumptions*. Proc. STOC'88 pp. 113–131 (1988).

[10] J. Kilian. *Strong separation models of multi prover interactive proofs*. DIMACS Workshop on Cryptography (1990).

[11] A. Kent. *Unconditionally secure bit commitment*. Phys. Rev. Lett. **83**, 1447–1450 (1999).

[12] C. Crépeau, A. Massenet, L. Salvail, L. Stinchcombe, and N. Yang. *Practical relativistic zero-knowledge for* **NP**. Proc. Inf.-Theor. Cryptogr. (ITC'20) **4**, 1–18 (2020).

[13] K. Mizuno and S. Nishihara. *Constructive generation of very hard 3-colorability instances*. Discret. Appl. Math. **156**(2), 218–229 (2008).

[14] J. Katz and Y. Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 3rd ed. (2020).

[15] E. Verbanis, A. Martin, R. Houlmann, G. Boso, F. Bussières, and H. Zbinden. *24-Hour relativistic bit commitment*. Phys. Rev. Lett. **117**, 140506 (2016).

[16] N. Li, C. Li, T. Helleseth, C. Ding, and X. Tang. *Optimal ternary cyclic codes with minimum distance four and five*. Finite Fields their Appl. **30**, 100–120 (2014).

[17] T. Tassa and J. L. Villar. *On proper secrets, (t, k)-bases and linear codes*. Des. Codes, Cryptogr. **52**(2), 129–154 (2009).

[18] T. Lunghi, J. Kaniewski, F. Bussières, R. Houlmann, M. Tomamichel, S. Wehner, and H. Zbinden. *Practical relativistic bit commitment*. Phys. Rev. Lett. **115**, 030502 (2015).

[19] J. S. Bell. *On the Einstein–Podolsky–Rosen paradox*. Physics Physique Fizika **1**, 195–200 (1964).

[20] N. Brunner, D. Cavalcanti, S. Pironio, V. Scarani, and S. Wehner. *Bell nonlocality*. Rev. Mod. Phys. **86**, 419–478 (2014).

[21] J. Kempe, H. Kobayashi, K. Matsumoto, B. Toner, and T. Vidick. *Entangled games are hard to approximate*. Siam J. Comput. **40**(3), 848–877 (2011).

[22] A. Chailloux and A. Leverrier. *Relativistic (or 2-prover 1-round) zero-knowledge protocol for* **NP** *secure against quantum adversaries*. Adv. Cryptol. — Eurocrypt'17 pp. 369–396 (2017).

[23] Z. Ji. *Binary constraint system games and locally commutative reductions*. arXiv:1310.3794 (2013).

[24] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev. *Scalable, transparent, and post-quantum secure computational integrity*. IACR ePrint Arch. (2018).

[25] J. Groth. *Non-interactive zero-knowledge arguments for voting*. Proc. Appl. Cryptogr. Netw. Secur. (ACNS'05) pp. 467–482 (2005).

[26] S. Micali and M. O. Rabin. *Cryptography miracles, secure auctions, matching problem verification*. Commun. ACM **57**(2), 85–93 (2014).

[27] A. Glaser, B. Barak, and R. J. Goldston. *A zero-knowledge protocol for nuclear warhead verification*. Nature **510**(7506), 497–502 (2014).

## METHODS

### 1. Protocol

Let us first recall the zero-knowledge proof as proposed in Ref. [12]. The notations are the same as in the main text. Initially the two provers pre-agree on random three-colourings $c_k(n) \in \{0,1,2\}$ and randomisers $b_k(n) \in \{0,1,2\}$ for $k \in V$ and $n$ labelling the round. A round is illustrated in Figs 2 and consists in (i) each verifier providing their prover with an edge $(i,j) = e \in E$ and two randomisers $(r,s) \in \{1,2\}$, (ii) each prover answering $a_1 \equiv b_i \cdot r + c_i \pmod 3$ and $a_2 \equiv b_j \cdot s + c_j \pmod 3$, and (iii) the two verifiers checking the provers' answers as described in the next two paragraphs. If none of the parties abort the protocol, then we repeat rounds until a certain security parameter in reached. As in the main text, the verifiers have two ways of checking the provers' answers.

On the one hand, the verifiers can check that the provers do indeed know that the graph is three-colourable. This test is done when both verifiers send the same random edge $e = (i,j) = (i',j') = e' \in E$ and when $(r',s') = (-r,-s)$. Then the answers $(a_1,a_2)$ and $(a'_1,a'_2)$ of the two provers are accepted if and only if $a_1 + a'_1 \not\equiv a_2 + a'_2 \pmod 3$.

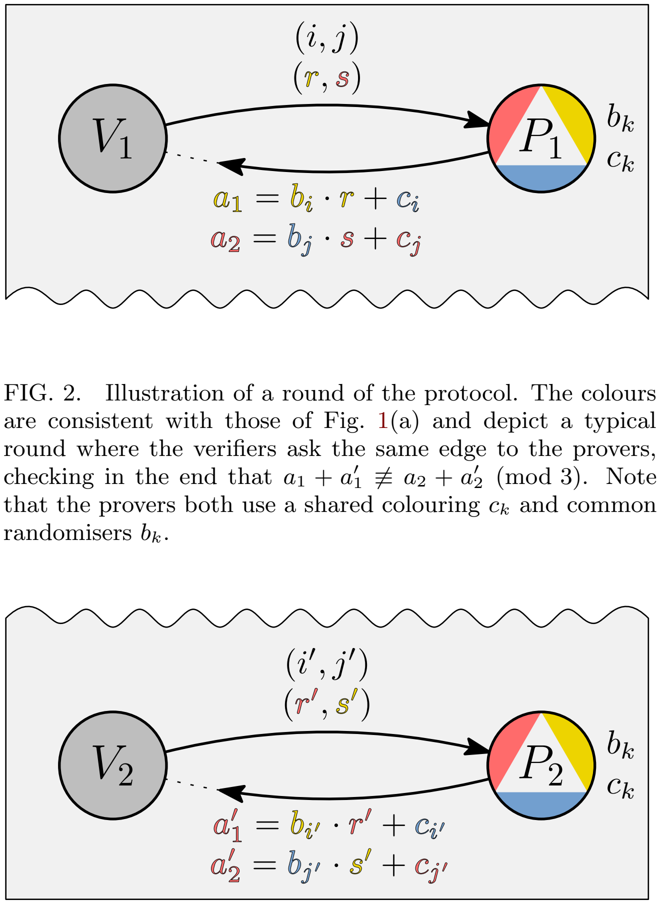
$(i,j)$
$(r,s)$
$V_1$
$P_1$
$b_k$
$c_k$
$a_1 = b_i \cdot r + c_i$
$a_2 = b_j \cdot s + c_j$
$(i',j')$
$(r',s')$
$V_2$
$P_2$
$b_k$
$c_k$
$a'_1 = b_{i'} \cdot r' + c_{i'}$
$a'_2 = b_{j'} \cdot s' + c_{j'}$

FIG. 2. Illustration of a round of the protocol. The colours are consistent with those of Fig. 1(a) and depict a typical round where the verifiers ask the same edge to the provers, checking in the end that $a_1 + a'_1 \not\equiv a_2 + a'_2 \pmod 3$. Note that the provers both use a shared colouring $c_k$ and common randomisers $b_k$.

On the other hand, the verifiers can test the consistency of the provers' answers. When the edges sent do share at least one vertex (say, $i = i'$) and when the corresponding randomisers are equal ($r = r'$), then the verifiers accept if and only if the answers of the two provers are equal ($a_i = a'_{i'}$). This test typically prevents the provers from answering in a way that ignores the edges asked but only aims at passing the previous test.

In Ref. [12] the following strategy for the verifiers is given. First the edge $(i,j)$ and the randomisers $(r,s)$ are chosen (uniformly) at random. Then with probabilities $\frac{1}{5}$, $\frac{2}{5}$, and $\frac{2}{5}$ (respectively), one of the three following options is chosen: (i) the edges are chosen to be equal and the randomisers opposite, that is, $(i',j') = (i,j)$ and $(r',s') = (-r,-s)$; (ii) the first randomiser is the same, the second one is chosen at random, and so is the second edge among those containing $i$, that is, $r' = r$, $i' = i$, and $(i',j') \in E$; (iii) the second randomiser is the same, the first one is chosen at random, and so is the second edge among those containing $j$, that is, $s' = s$, $j' = j$, and $(i',j') \in E$.

Note that the amount of data exchanged is very small compared to previous protocols: in Ref. [22] this quantity is polynomial in the number $|V|$ of vertices while here it is only logarithmic in $|V|$. This feature allows for short distances between the verifier-prover pairs since the communication time is short, even for large graphs.

### 2. Graph generation

In this section we describe how we construct large three-colourable graphs which are hard to colour *together* with a three-colouring.

In Ref. [13] the authors give (i) seven small graphs that are four-critical, that is, not three-colourable but such that any graph obtained by deleting any edge is three-colourable, and (ii) a procedure to assemble two four-critical graphs into a (bigger) four-critical graph. Typically, the method consists in replacing one edge of the first graph by the second one. Importantly, the small and assembled graphs do not contain any near-four-clique, that is, any subgraph with four vertices all connected to one another except for one pair, e.g., ⧄. Such structures indeed appear as weaknesses exploitable by algorithms looking for a three-colouring and should thus be avoided. With their procedure they experimentally demonstrated using various softwares that the complexity of the resulting instances was exponential in the number of vertices.

However, they do not include any algorithm to keep track of the three-colourings that arise upon removal of an edge. We developed such a method to build large graphs that are very hard to three-colour together with a three-colouring.

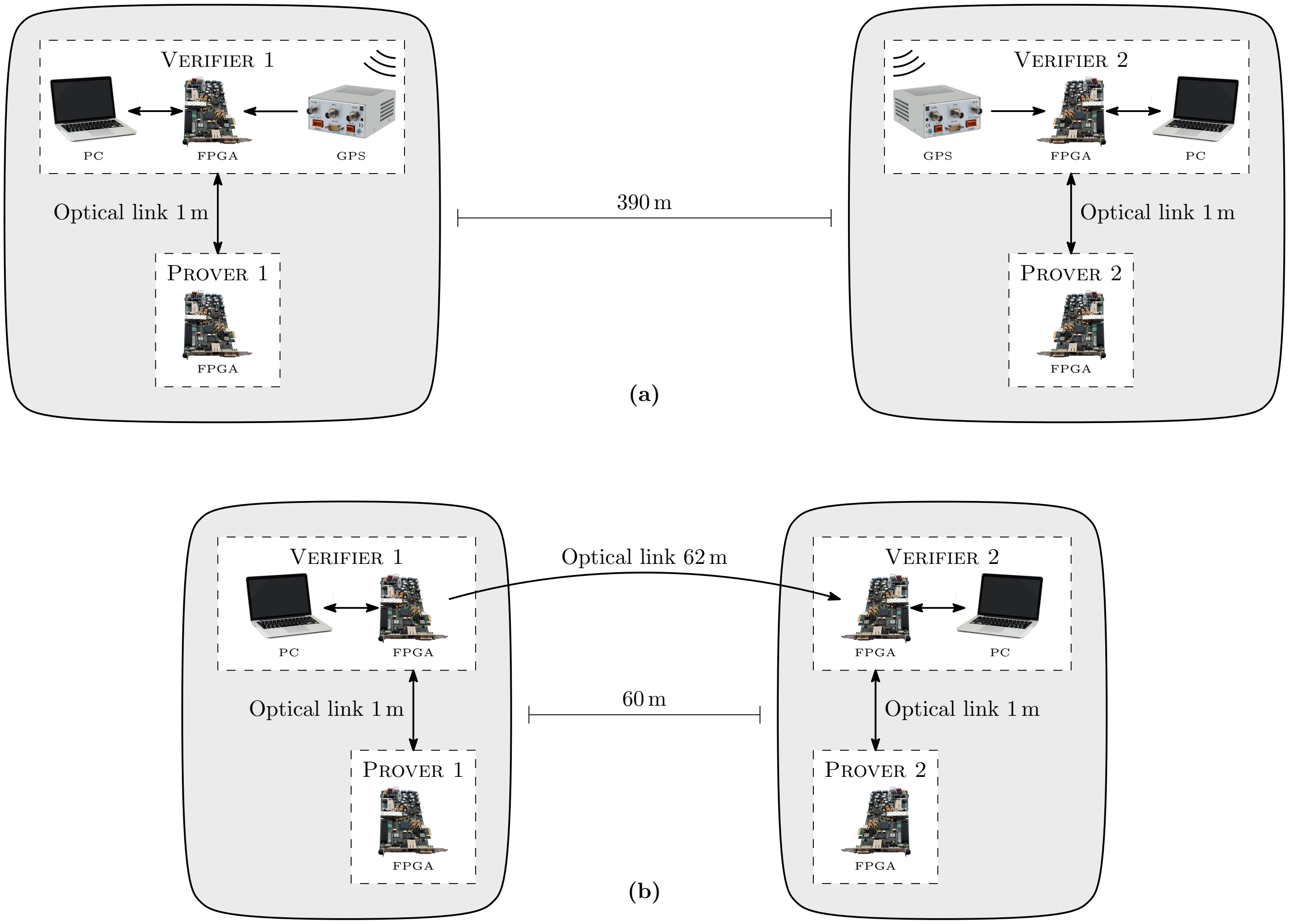

FIG. 3. Illustration of the hardware used in our two implementations: **(a)** the GPS version and **(b)** the triggered version. The essential difference is the method used for synchronising the verifiers' questions. In **(a)** the connection is wireless as it uses communication with satellites at the expense of a higher imprecision thus further verifier-prover pairs. In **(b)** the connection is physical and oriented from the first to the second verifier; the former sends a trigger through the fibre and delays their action by the time needed for this signal to reach the latter. With a better accuracy this second method allows for shorter distances between the verifier-prover pairs, here 60 m but arguably improvable.

### 3. Hardware

For the implementation, the verifiers consist of a standard computer (Intel core i3 processor with 4 GB of RAM) and an FPGA development board (Xilinx SP605 evaluation board featuring Spartan-6 XC6SLX45T), the two being connected together through a PCI-Express link; the provers consist only of the same FPGA development board. Within each verifier-prover pair, FPGA boards are communicating with each other through a 2.5 Gbit/s small form-factor pluggable (SFP) optical link. On the provers' side the main data (graph, colouring) is stored in memories available in the FPGA (Block RAM of about 2 Mbit) and the random data on Flash memories available on the FPGA development board (32 MB), the latter being slower than the former. This shared randomness was generated by means of the quantum random number generator (QRNG) Quantis by IDQuantique.

#### *a. GPS version*

A schematic view of the setup in this case is depicted on Fig. 3(a). The verifiers' FPGAs are synchronised to the Coordinated Universal Time (UTC) by means of a Global Positioning System (GPS) clock, that is, a GPS receiver and an Oven-Controlled Quartz-Crystal Oscillator (OCXO) that creates a sinusoidal wave with a frequency of 10 MHz. This OCXO signal, locked to an electronic pulse per second (PPS) delivered by the GPS with a precision of 150 ns, is sent to the verifiers' FPGAs where its frequency is multiplied to a 125 MHz signal through a phase-locked loop. Eventually this 125 MHz signal is used as a time reference for the computations performed on the FPGAs, which also receive the PPS signal to check the synchronisation with the GPS clock. Specifically, we verified that there were $1.25 \times 10^6 \pm 1$ cycles between two successive PPS signal, fixing the cycle duration to 8 ns.

This shows that the inaccuracy added by the generation of the 125 MHz clock would be below 24 ns. Therefore, since the PPS signals are also labelled with a universal time stamp, the verifiers are able to synchronise their questions with an accuracy of $150 + 24 = 174$ ns.

### *b. Triggered version*

A schematic view of the setup in this case is depicted on Fig. 3(b). The verifiers FPGA's are connected to one another with an SFP optical link, this link is used to synchronise the questions sent to the provers. The FPGAs run at a base clock frequency of 125 MHz. The first verifier generates a stream of triggers at a rate of about 3 MHz. These impulsions are transferred to the second verifier through a fibre channel of 62 m connecting both devices, in order to trigger the challenges sent to the prover. On the first verifier this trigger is delayed by 440 ns to compensate for the delay in the fibre and the latency of the electronics. With an oscilloscope we measured that the imprecision between the delayed trigger and the trigger sent through the optical fibre does not exceed three cycles, i.e., 24 ns. Moreover the total time of the exchange in the verifiers' FPGA is inferior to 35 cycles but we determined that the verifiers internal latency, that is, the time when the data is still in the FPGA plus the time the answer is already back, accounts to at least 14 cycles, thus reducing this time to 21 cycles. Note that this time arises from the conversion from electronical to optical signals. When adding the imprecision of the trigger, we get that a round is achieved in a maximum of 24 cycles, that is, 192 ns, thus constraining the verifier to be at a minimal distance of 57.6 m.

## 4. Shared randomness

In the protocol presented in Methods 1 the two provers need to use the same random colouring and randomisers in each round. Given the high number of rounds needed to reach a satisfying security and the relatively low memory of the FPGAs, a frugal approach is mandatory. In this section we give the details of our implementation with this regard.

For the colouring, it is easy to see that there is a thrifty option: storing a fixed one and only drawing a random permutation of the colours. The "randomness cost" of this part is therefore of one bit and one trit in each round.

For the randomisers, a naive approach would demand one random trit per vertex each time, thus requiring far too much memory given the high number of rounds. Noting that only four of them are actually used in each round (two per prover), we apply a radically more affordable alternative: storing $|V|$ (the number of vertices) fixed trits and drawing $2m+1$ trits, where $m$ is the number of digits of $|V|$ in base three. The idea is to expand randomness by assigning in advance a small ternary vector (of $2m+1$ trits) to each node; then each randomiser is simply chosen by computing the scalar product with a common random ternary vector (of $2m+1$ trits). The subtle point to take care of is the independence of the resulting random variables, which amounts to the linear independence of the vectors assigned to the nodes. As only four randomisers are consumed in each round, we want all sets of four such vectors to be free. The literature luckily offers an elegant solution to this problem via ternary cyclic linear codes with minimum Hamming distance of five [16]. The parity check matrix of a linear code with minimum Hamming distance $d$ is indeed such that all sets of $d - 1$ of its columns are linearly independent; see, e.g., Ref. [17, Lemma 3.5]. Moreover, the cyclicity of the code used allows to store only one trit per node and to use it together with the ones of the next $2m$ nodes (in numerical order) to create its ternary vector.