Prisoners of their own device: Trojan attacks on device-independent quantum cryptography

Jonathan Barrett,[1,*] Roger Colbeck,[2,†] and Adrian Kent[3,2,‡]

[1]*Department of Mathematics, Royal Holloway, University of London, Egham Hill, Egham, TW20 0EX, U.K.*
[2]*Perimeter Institute for Theoretical Physics, 31 Caroline Street North, Waterloo, ON N2L 2Y5, Canada.*
[3]*Centre for Quantum Information and Foundations, DAMTP, Centre for Mathematical Sciences, University of Cambridge, Wilberforce Road, Cambridge, CB3 0WA, U.K.*

(Dated: 20th January 2012)

Device-independent cryptographic schemes aim to guarantee security to users based only on the output statistics of any components used, and without the need to verify their internal functionality. Since this would protect users against untrustworthy or incompetent manufacturers, sabotage or device degradation, this idea has excited much interest, and many device-independent schemes have been proposed. We point out here a critical weakness of device-independent quantum cryptography for tasks, such as key distribution, that rely on public communication between secure laboratories. Untrusted devices may record their inputs and outputs and reveal encoded information about them in their outputs during later runs. Reusing devices thus compromises the security of a protocol and risks leaking secret data. Possible solutions include securely destroying used devices or isolating them until previously generated data need no longer be kept secret. However, such solutions are costly and impose severe constraints on the practicality of many device-independent quantum cryptographic schemes.

INTRODUCTION

Quantum cryptography aims to exploit the properties of quantum systems to ensure the security of various tasks. The best known example is quantum key distribution (QKD), which can enable two parties to share a secret random string and thus exchange messages secure against eavesdropping, and we mostly focus on this task for concreteness. While all classical key distribution protocols rely for their security on assumed limitations on an eavesdropper's computational power, the advantage of quantum key distribution protocols (e.g. [1, 2]) is that they are provably secure against an arbitrarily powerful eavesdropper, even in the presence of realistic levels of losses and errors [3]. However, the security proofs require that devices function according to particular specifications. Any deviation – which might arise from a malicious or incompetent manufacturer, or through sabotage or degradation – can introduce exploitable security flaws (see e.g. [4] for practical illustrations).

The possibility of devices with deliberately concealed flaws, introduced by an untrustworthy manufacturer or saboteur, is particularly concerning, since (i) it is easy to design devices that appear to be following a secure protocol but are actually completely insecure, and (ii) there is no general technique for identifying security loopholes in standard cryptography devices. This has led to much interest in device-independent protocols [5], which aim to guarantee security *on the fly* by testing the device outputs: no specification of their internal functionality is required.

Known provably secure schemes for device-independent quantum key distribution are inefficient, as they require either independent isolated devices for each entangled pair to ensure full device-independent security [6–9], or a large number of entangled pairs to generate a single bit [6, 10]. Finding an efficient secure device-independent quantum key distribution scheme using two (or few) devices has remained an open theoretical challenge. Nonetheless, in the absence of tight theoretical bounds on the scope for device-independent quantum cryptography, progress to date has encouraged widespread optimism (e.g. [11]) about the prospects for device-independent QKD as a practical commercial technology, as well as for device-independent quantum randomness expansion [12–14] and other applications of device-independent quantum cryptography (e.g. [15]).

However, one key question has been generally neglected in work to date on device-independent quantum cryptography, namely what happens if and when devices are reused. Specifically, are device-reusing protocols *universally composable* – i.e. do individually secure protocols of this type remain secure when combined? We present below attacks that highlight a generic problem in producing universally composable protocols with device-independent security for reusable devices, and show that for many protocols universal composability fails in the strong sense that purportedly secret data becomes completely insecure. While these attacks can be countered by not reusing devices, this solution is so costly that we query whether it is generally practical.

*Electronic address: jon.barrett@rhul.ac.uk
†Electronic address: rcolbeck@perimeterinstitute.ca
‡Electronic address: a.p.a.kent@damtp.cam.ac.uk

TOJAN MEMORY ATTACKS

We describe here a new type of attack on device-independent cryptosystems using device memories, which suggests that a serious reappraisal of the potential practicality of such schemes is required. In short, the problem is that an adversary can program devices to store data in one protocol and leak it in subsequent protocols, in ways that are hard or impossible to counter if the devices are indeed reused.

To illustrate this, consider a device-independent scheme that allows two users (Alice and Bob) to generate and share a purportedly secure cryptographic key. A malicious manufacturer can design devices so that they act as Trojan spies in Alice's and Bob's secure laboratories, recording and storing all their inputs and outputs. Although a well designed device-independent protocol can prevent the devices from leaking information about the generated key *during that protocol*, data about these inputs and outputs, and hence about the secure key, can be leaked, using output data discussed over a public channel *whenever the devices are later used*. Moreover, in many existing protocols, such leaks can be surreptitiously hidden in the noise. This allows the devices to operate indefinitely as Trojan spies, apparently complying with security tests, but actually eventually leaking all the purportedly secure data.

No existing security definitions address attacks of the type we describe. A theoretically simple way to prevent these attacks is to dispose of – i.e. securely destroy or isolate – untrusted devices after a single use. However, while this "toxic device disposal" strategy is secure and relies only on standard cryptographic assumptions, and may conceivably be worthwhile for sufficiently high value data in some scenarios, it is clearly costly, and its use would severely limit the practicality of device-independent cryptography.

We proceed by introducing the device-independent scenario we are considering, before describing Trojan memory attacks in more detail, using concrete examples of attacks on device-independent quantum key distribution protocols. As we explain, the attacks also apply to other device-independent quantum cryptographic tasks.

Cryptographic scenario for device independent QKD

We use the standard cryptographic scenario for key distribution between Alice and Bob, each of whom has a secure laboratory. These laboratories may be partitioned into secure sub-laboratories, and we assume Alice and Bob can prevent unwanted communication between their sub-laboratories as well as between their labs and the outside world.

We also assume Alice and Bob each have access to (or can generate) their own string of private random bits. They are connected by an authenticated, but insecure, classical communication channel as well as an insecure quantum channel, and have trusted classical information processing devices in their laboratories. However, all quantum devices they use for the protocol are assumed to be supplied by an untrusted adversary, Eve. These devices effectively function as black boxes for Alice and Bob, receiving classical inputs from them and returning classical outputs. Eve has access to the classical and quantum communication channels between Alice's and Bob's laboratories, and complete knowledge of the protocol. She cannot directly eavesdrop on the classical random data that Alice and Bob generate within their labs and use for the protocol, but she may be able to deduce information about those data from their public communications.

Trojan memory attacks on two-device QKD protocols

The device-independent QKD protocols that have been proven unconditionally secure [6, 8, 9] require separate devices for each measurement performed by Alice and Bob. The reason is that the security proofs – in addition to the usual assumption that no signals can pass from Alice's or Bob's devices directly to Eve – need to assume that no signals can be sent between the separate devices that Alice is using to measure each of her particles, and similarly for Bob. Within the scenario set out above, this can be achieved by having each device isolated in a separate sub-laboratory. The protocols in [8, 9] are at least modestly noise-tolerant and would be considered relatively efficient at generating secure keys, were it not for the requirement of many devices and sub-laboratories.

For practical device-independent QKD, though, Alice and Bob should only use a small number of devices. We look first at implementations of protocols which are of the form of those in [8, 9], except that Alice and Bob use one device each, i.e., Alice (Bob) uses the same device to perform each of her (his) measurements. The memory of a device can then act as a signal from earlier to later measurements, hence the security proofs of [8, 9] do not apply (see also [16] where a different two-device setup is discussed). It is an open question whether a secure key can be efficiently generated in this scenario. Here we demonstrate that even if a key can be securely generated, repeat implementations of the protocol using the same devices render an earlier generated key insecure.

We first consider QKD protocols with the following standard structure. Although this structure is potentially restrictive, most protocols to date are of this form (we later discuss modifications). Note that we do not need to specify the precise sub-protocols used for error correction or privacy amplification.

1. Entangled quantum states used in the protocol are either supplied to Alice and Bob by Eve, or generated by Bob's device and then shared over an insecure quantum channel with Alice's device. Once

the states are received, the quantum channel is closed.

2. Alice and Bob each pick a random input A_i and B_i to their device, ensuring they receive an output bit (X_i and Y_i respectively) before making the next input (so that the i-th output cannot depend on future inputs). They repeat this M times.

3. Either Alice or Bob (or both) publicly announces their measurement choices, and the relevant party checks that they had a sufficient number of suitable input combinations for the protocol. If not, they abort.

4. (*Sifting.*) Some output pairs may be discarded according to some public protocol.

5. (*Parameter estimation.*) Alice randomly and independently decides whether to announce each remaining bit to Bob, doing so with probability μ (where $M\mu \gg 1$). Bob uses the communicated bits and his corresponding outputs to compute some test function, and aborts if it lies outside a desired range. (For example, Bob might compute the CHSH value [17] of the announced data, and abort if it is below 2.5.)

6. (*Error correction.*) Alice and Bob perform error correction using public discussion, in order to (with high probability) generate identical strings. Eve learns the error correction function Alice applies to her string.

7. (*Privacy amplification.*) Alice and Bob publicly perform privacy amplification [18], producing a shorter shared string about which Eve has virtually no information. Eve similarly learns the privacy amplification function they apply to their error-corrected strings.

Consider now a scenario in which a protocol of this type is run on day 1, generating a secure key for Alice and Bob, while informing Eve of the functions used by Alice for error correction and privacy amplification (for simplicity we assume their protocol has no sifting procedure (Step 4)). The protocol is then rerun on day 2, to generate a second key, using the same devices. Eve can instruct the devices to proceed as follows. On day 1, they follow the protocol honestly. However, they keep hidden records of all the raw bits they generate during the protocol. At the end of day 1, Eve knows the error correction and privacy amplification functions used by Alice and Bob to generate the secure key.

On day 2, since Eve either distributes a new set of quantum states to Alice and Bob, or else has access to the insecure quantum channel over which they are distributed, she can surreptitiously modulate these quantum states to carry new classical instructions to the device in Alice's lab. These instructions tell the device the error correction and privacy amplification functions used on day 1, allowing the device to compute the secret key generated on day 1. They also tell the device to deviate from the honest protocol for randomly selected inputs, by producing as outputs specified bits from this secret key. (For example, "for input 17, give day 1's key bit 5 as output".) If any of these selected outputs are among those announced in Step 5, Eve learns the corresponding bits of day 1's secret key.

Furthermore, if she follows this cheating strategy for $N\mu^{-1} < M$ input bits, Eve is likely to learn roughly N bits of day 1's secret key. Moreover, only the $\approx \mu N \mu^{-1} = N$ output pairs from this set that are publicly compared give Alice and Bob statistical information about Eve's cheating. Alice and Bob cannot a priori identify these cheating output pairs among the $\approx \mu M$ they compare. Thus, if the tolerable noise level is comparable to $N\mu^{-1}M^{-1}$, Eve can (with high probability) masquerade her cheating amongst it (note that in unconditional security proofs it is assumed that eavesdropping is the cause of all noise; even if not, Eve can supply less noisy components than she claims and use the extra tolerable noise to cheat).

Even in the hypothetical case where Alice and Bob have noise-free devices, so that their protocol can abort at any perceivable noise level, Eve learns at least one bit of day 1's string before her cheating is detected on day 2. Note that standard security definitions aim to protect every bit of Alice's and Bob's key from an adversary. Although this may seem an unduly strong requirement (particularly in the case of very long generated strings), there are many practical scenarios in which leaking a single bit can be detrimental.

Attacks on modified protocols

The security loophole discussed above can be partly closed by redesigning QKD protocols so that all quantum data and all public communication of output data in the protocol come from one party, say Bob. Thus, the entangled states used in the protocol are generated by Bob's device (as allowed above) and Bob (rather than Alice) sends selected output data over a public channel in Step 5. If Bob's device is forever kept isolated from incoming communication, Eve has no way of sending it instructions to calculate and leak secret key bits from day 1 (or any later day).

Even protocols modified in this way are insecure if reused, however. Eve can still communicate with Alice's device, and Alice needs to be able to make some public communication to Bob, if only to abort the protocol. Eve can thus obtain at least one secret key bit from day 1 on day 2 by programming Alice's device to abort or not depending on a particular bit value. Alternatively, Eve can pre-program Bob's device to leak raw key data from day 1 via output data on subsequent days, at a low enough rate (compared to the background noise level) that this cheating is unlikely to be detected. If the actual noise level is lower than the level tolerated in the protocol, and Eve knows both (a possibility Alice and Bob must allow for), she can thereby eventually obtain all Bob's raw key data from day 1, and hence the secret key.

Note too that these last attacks apply even if Bob has separate isolated state preparation and measurement devices. Eve can still communicate with Alice's measurement device, and can still pre-program Bob's measurement device to leak raw day 1 key data on subsequent days.

One way of temporarily countering device memory attacks is for Alice and Bob to share a small initial secret key and to use part of this key to choose the privacy amplification function in Step 7 of the protocol, which may then never become known to Eve. However, even in this case, Eve can pre-program Bob's measurement device to leak raw data from day 1 on subsequent days. While Eve cannot obtain bits of the secret key so directly in this case, provided the protocol is composed sufficiently many times, she can eventually obtain all the raw key. This means that Alice and Bob's residual security ultimately derives only from the initial shared secret key: their QKD protocol produces no extra permanently secure data.

Finally, note that Alice's and Bob's devices each separately have the power to cause the protocol to abort on any day of their choice. Thus – if Eve is willing to wait long enough – she can program them to communicate some or all information about their day 1 raw outputs, for instance by encoding the relevant bits as a binary integer $N = b_1 \ldots b_m$ and choosing to abort on day $(N+2)$. This version of the attack seems unavoidable in any standard cryptographic model.

To reiterate, the essential point is that if any devices know crucial secrets, using those devices in future protocols potentially compromises security. Although we have considered two-device QKD protocols so far, the Trojan device memory attacks we describe apply far more generally. We illustrate their application to some well known multi-device QKD protocols and to quantum randomness expansion protocols in the Appendix.

Toxic device disposal

As noted above, standard cryptographic models postulate that the parties can create secure laboratories, within which all operations are shielded from eavesdropping. Device-independent cryptographic models also necessarily assume that devices within these laboratories cannot signal to the outside – otherwise security is clearly impossible. Multi-device protocols assume that the laboratories can be divided into effectively isolated sub-laboratories, and that devices in separate sub-laboratories cannot communicate. In other words, Alice and Bob must be able to build arbitrary configurations of screening walls, which prevent communication among Eve and any of her devices, and allow only communications specified by Alice and Bob.

Given this, there is no problem *in principle* in defining protocols which prescribe that devices must be permanently isolated: the devices simply need to be left indefinitely in a screened sub-laboratory. While this could be detached from the main working laboratory, it must be protected indefinitely: screening wall material and secure space thus become consumed resources. And indeed in some situations, it may be more efficient to isolate devices, rather than securely destroy them, since devices can be reused once the secrets they know have become public by other means. For example, one may wish to securely communicate the result of an election before announcing it, but once it is public, the devices used for this secure communication could be safely reused.

The alternative, securely destroying devices and then eliminating them from the laboratory, preserves laboratory space but raises new security issues: consider, for example, the problems in disposing of a device programmed to change its chemical composition depending on its output bit.

That said, no doubt there are pretty secure ways of destroying devices, and no doubt devices could be securely isolated for long periods. However, the costs and problems involved, together with the costs of renewing devices, make us query whether these are really viable paths for practical device-independent cryptography.

DISCUSSION

A malicious manufacturer who wishes to mislead users or obtain data from them can equip devices with a memory and use it in programming them. The full scope and seriousness of this threat seems to have been overlooked in the quantum cryptographic literature to date. A task is potentially vulnerable to our attacks if it involves secret data generated by devices and if Eve can learn some function of the device outputs. Since even causing a protocol to abort communicates some information to Eve, the class of tasks potentially affected is large indeed. In particular, for the most important application, device-independent QKD, every protocol so far proposed (as far as we are aware) is acutely vulnerable.

One can think of the problems our attacks raise as a new issue of cryptographic composability. One way of thinking of standard composability is that a secure output from a protocol must still have all the properties of an ideal secure output when combined with other outputs from the same or other protocols. The device-independent key distribution protocols examined above fail this test because the reuse of devices causes later outputs to depend on earlier ones. In a sense, the underlying problem is that the *usage of devices* is not composably secure. This applies too, of course, for devices used in different protocols: devices used for secure randomness expansion cannot then securely be used for key distribution without potentially compromising the generated randomness, for example.

We should stress that our attacks do not apply to all device-independent quantum tasks. For example, even devices with memories cannot mimic nonlocal correlations in the absence of shared entanglement [19, 20], and so device-independent entanglement testing remains viable. In addition, in applications that require only short-lived secrets, devices may be reused once such secrets are no longer required. Partially secure device-independent protocols for bit commitment and coin tossing [15] in which the committer supplies devices to the recipient are also immune from our attacks so long as the only data entering the devices comes from the committer. Nonetheless, in our view, the attacks are generic and problematic enough to merit a serious reappraisal of the scope for device-independent quantum cryptography as a practical technology.

Acknowledgements

AK was partially supported by a Leverhulme Research Fellowship, a grant from the John Templeton Foundation, and the EU Quantum Computer Science project (contract 255961). This work was supported by the CHIST-ERA DIQIP project. Research at Perimeter Institute is supported by the Government of Canada through Industry Canada and by the Province of Ontario through the Ministry of Research and Innovation.

[1] Bennett, C. H. & Brassard, G. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, 175–179. IEEE (New York, 1984).

[2] Ekert, A. K. Quantum cryptography based on Bell's theorem. *Physical Review Letters* **67**, 661–663 (1991).

[3] Renner, R. *Security of Quantum Key Distribution*. Ph.D. thesis, Swiss Federal Institute of Technology, Zurich (2005). Also available as `quant-ph/0512258`.

[4] Gerhardt, I. *et al.* Full-field implementation of a perfect eavesdropper on a quantum cryptography system. *Nature Communications* **2**, 349 (2011).

[5] Mayers, D. & Yao, A. Quantum cryptography with imperfect apparatus. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS-98)*, 503–509 (IEEE Computer Society, Los Alamitos, CA, USA, 1998).

[6] Barrett, J., Hardy, L. & Kent, A. No signalling and quantum key distribution. *Physical Review Letters* **95**, 010503 (2005).

[7] Masanes, L., Renner, R., Christandl, M., Winter, A. & Barrett, J. Unconditional security of key distribution from causality constraints. e-print `quant-ph/0606049v4` (2009).

[8] Hänggi, E. & Renner, R. Device-independent quantum key distribution with commuting measurements. e-print `arXiv:1009.1833` (2010).

[9] Masanes, L., Pironio, S. & Acín, A. Secure device-independent quantum key distribution with causally independent measurement devices. *Nature Communications* **2**, 238 (2011).

[10] Barrett, J., Colbeck, R. & Kent, A. in preparation (2012).

[11] Ekert, A. Less reality, more security. *Physics World* September (2009).

[12] Colbeck, R. *Quantum and Relativistic Protocols For Secure Multi-Party Computation*. Ph.D. thesis, University of Cambridge (2007). Also available as `arXiv:0911.3814`.

[13] Colbeck, R. & Kent, A. Private randomness expansion with untrusted devices. *Journal of Physics A* **44**, 095305 (2011).

[14] Pironio, S. *et al.* Random numbers certified by Bell's theorem. *Nature* **464**, 1021–1024 (2010).

[15] Silman, J. *et al.* Fully distrustful quantum bit commitment and coin flipping. *Physical Review Letters* **106**, 220501 (2011).

[16] Hänggi, E., Renner, R. & Wolf, S. The impossibility of non-signalling privacy amplification. e-print `arXiv:0906.4760` (2009).

[17] Clauser, J. F., Horne, M. A., Shimony, A. & Holt, R. A. Proposed experiment to test local hidden-variable theories. *Physical Review Letters* **23**, 880–884 (1969).

[18] Bennett, C. H., Brassard, G. & Robert, J.-M. Privacy amplification by public discussion. *SIAM Journal on Computing* **17**, 210–229 (1988).

[19] Barrett, J., Collins, D., Hardy, L., Kent, A. & Popescu, S. Quantum nonlocality, Bell inequalities, and the memory loophole. *Physical Review A* **66**, 042111 (2002).

[20] Gill, R. D. Accardi contra Bell (cum mundi): The impossible coupling. In Moore, M., Froda, S. & Léger, C. (eds.) *Mathematical Statistics and Applications: Festschrift for Constance van Eeden*, vol. 42 of *IMS Lecture Notes – Monograph Series*, 133–154 (2003).

[21] Fehr, S., Gelles, R. & Schaffner, C. Security and composability of randomness expansion from Bell inequalities. e-print `arXiv:1111.6052` (2011).

[22] Vidick, T. & Vazirani, U. Certifiable quantum dice or, testable exponential randomness expansion. e-print `arXiv:1111.6054` (2011).

[23] Pironio, S. & Massar, S. Device-independent randomness expansion secure against quantum adversaries. e-print `arXiv:1111.6056` (2011).

APPENDIX

TROJAN MEMORY ATTACKS ON MULTI-DEVICE QKD PROTOCOLS

To illustrate further the generality of our attacks, we now turn to multi-device protocols, and show how to break iterated versions of two well known protocols.

Attacks on compositions of the BHK protocol

The Barrett-Hardy-Kent (BHK) protocol [6] requires Alice and Bob to share MN^2 pairs of systems (where M and N are both large with $M \ll N$), in such a way that no measurements on any subset can effectively signal to the others. In a device-independent scenario, we can think of these as black box devices supplied by Eve, containing states also supplied by Eve. Each device is isolated within its own sub-laboratory of Alice's and Bob's, so that Alice and Bob have MN^2 secure sub-laboratories each. The devices accept integer inputs in the range $\{0, \ldots, N-1\}$ and produce integer outputs in the range $\{0, 1\}$. Alice and Bob choose random independent inputs, which they make public after obtaining all the outputs. They also publicly compare all their outputs except for those corresponding to one pair randomly chosen from among those in which the inputs differ by ± 1 or 0 modulo N. If the publicly declared outputs agree with quantum statistics for specified measurement basis choices (corresponding to the inputs) on a singlet state, then they accept the protocol as secure, and take the final undeclared outputs (which are almost certainly anticorrelated) to define their shared secret bit.

The BHK protocol produces (with high probability) precisely one secret bit: evidently, it is extremely inefficient in terms of the number of devices required. It also requires essentially noise-free channels and error-free measurements. Despite these impracticalities it illustrates our theoretical point well. Suppose that Alice and Bob successfully complete a run of the BHK protocol and then (unauthorised by BHK) decide to use the same $2MN^2$ devices to generate a second secret bit, and ask Eve to supply a second batch of states to allow them to do this.

Eve — aware in advance that the devices may be reused — can design them to function as follows. In the first run of the protocol, she supplies a singlet pair to each pair of devices and the devices function honestly, carrying out the appropriate quantum measurements on their singlets and reporting the outcomes as their outputs. However, they also store in memory their inputs and outputs. In the second run, Eve supplies a fresh batch of singlet pairs. However, she also supplies a hidden classical signal identifying the particular pair of devices that generated the first secret bit. (This signal need go to just one of this pair of devices, and no others.) On the second run, the identified device produces as output the same output that it produced on the first run (i.e. the secret bit generated, up to a sign convention known to Eve). All other devices function honestly on the second run.

With probability $\frac{MN^2-1}{MN^2}$, the output from the cheating device on the second run will be made public, thus revealing the first secret bit to Eve. Moreover, with probability $1 - \frac{3}{2N} + O(N^{-2})$, this cheating will not be detected by Alice and Bob's tests, so that Eve learns the first secret bit without her cheating even being noticed.

There are defences against this specific attack. First, the BHK protocol [6] can be modified so that only outputs corresponding to inputs differing by ± 1 or 0 are publicly shared.[1] While this causes Eve to wait many rounds for the secret bit to be leaked, and increases the risk her cheating will be detected, it leaves the iterated protocol insecure. Second, Alice and Bob could securely destroy or isolate the devices producing the secret key bit outputs, and reuse all their other devices in a second implementation. Since only the devices generating the secret key bit have information about it, this prevents it from being later leaked. While effective, this last defence really reflects the inefficiency of the BHK protocol: to illustrate this, we turn next to a more efficient multi-device protocol.

Attacks on compositions of the HR protocol

Hänggi and Renner (HR) [8] consider a multi-device QKD protocol related to the Ekert [2] protocol, in which Alice and Bob randomly and independently choose one of two or three inputs respectively for each of their devices. If the devices are functioning honestly, these correspond to measurements of a shared singlet in the bases U_0, U_1 (Alice) and V_0, V_1, V_2 (Bob), defined by the following vectors and their orthogonal complements

$$\begin{aligned}
U_1 &\leftrightarrow |0\rangle\,, \\
V_0 &\leftrightarrow \cos(\pi/8)|0\rangle + \sin(\pi/8)|1\rangle\,, \\
U_0, V_2 &\leftrightarrow \cos(\pi/4)|0\rangle + \sin(\pi/4)|1\rangle\,, \\
V_1 &\leftrightarrow \cos(3\pi/8)|0\rangle + \sin(3\pi/8)|1\rangle\,.
\end{aligned}$$

The raw key on any given run is defined by the $\approx 1/6$ of the cases in which U_0 and V_2 are chosen. Information reconciliation and privacy amplification proceed according to protocols of the type described in the main text (in which the functions used are released publicly).

Evidently, our attacks apply here too if (unauthorised by HR) the devices are reused to generate further secret keys. Eve can identify the devices that generate the raw

[1] As originally presented, the BHK protocol requires public exchange of all outputs except those defining the secret key bit. This is unnecessary, and makes iterated implementations much more vulnerable to the attacks discussed here.

key on day 1, and request them to release their key as cheating outputs on later days, gradually enough that the cheating will be lost in the noise. Since the information reconciliation and privacy amplification hash functions were made public by Alice, she can then obtain the secret key. Even if she is unable to communicate directly with the devices for a long time (because they were preinstalled with a very large reservoir of singlets), she can program all devices to gradually release their day 1 outputs over subsequent days, and so can still deduce the raw and secret keys.

Alice and Bob could counter these attacks by securely destroying or isolating all the devices that generated raw key on day 1 — but this costs them $1/6$ of their devices, and they have to apply this strategy each time they generate a key, leaving $(5/6)^N$ of the devices after N runs, and leaving them able to generate shorter and shorter keys. As the length of secure key generated scales by $(5/6)^N$ (or worse, allowing for fluctuations due to noise) on each run, the total secret key generated is bounded by $\approx 6M$, where M is the secret key length generated on day 1.

Note that, as in the case of the iterated BHK protocol, all devices that generate secret key become toxic and cannot be reused. While the relative efficiency of the HR protocol ensures a (much) faster secret key rate, it also requires an equally fast device depletion rate. This example shows that our attacks pose a generic problem for device-independent QKD protocols of the types considered to date.

ATTACKS ON DEVICE-INDEPENDENT RANDOMNESS EXPANSION PROTOCOLS

Device-independent quantum randomness expansion (DVI QRE) protocols were introduced by two of us [12, 13], developed further theoretically and investigated experimentally in [14], and recently extended further with reported unconditional security proofs [21–23]. The cryptographic scenario here is slightly different from that of key distribution in that there is only one honest party, Alice.

Alice's aim is to expand an initial secret random string to a longer one that is guaranteed secret from an eavesdropper, Eve, even if the quantum devices and states used are supplied by Eve. The essential idea is that seed randomness can be used to carry out nonlocality tests on the devices and states, within one or more secure laboratories, in a way that guarantees (with numerical bounds) that the outcomes generate a partially secret and random string. Privacy amplification can then be used to generate an essentially fully secret random string, which (provided the tests are passed) is significantly longer than the initial seed.

There are already known pitfalls in designing such protocols. For example, although one might think that carrying out a protocol in a single secure laboratory guarantees that the initially secure seed string remains secure, and so guarantees randomness expansion if any new secret random data is generated, this is not the case [13]. Eve's devices may be programmed to produce outputs depending on the random seed in such a way that the length of the final secret random string depends on the initial seed. Protocols with this vulnerability are not composably secure. (To see this can be a practical problem, note that Eve may infer the length of the generated secret random string from its use.)

A corollary of our results is that, if one wants to reuse the devices to generate further randomness, it is crucial to carry out DVI QRE protocols with devices permanently held within a *single* secure laboratory, avoiding any public communication of device output data at any stage. It is crucial too that the devices themselves are securely isolated from classical communications and computations within the laboratory, to prevent them from learning details of the reconciliation and privacy amplification.

Even under these stringent conditions, our attacks still apply in principle. For example, consider a noise-tolerant protocol that produces a secret random output string of variable length, depending on the values of test functions of the device outputs (the analogue of QKD parameter estimation for QRE) that measure how far the device outputs deviate from ideal honest outputs. This might seem natural for any single run, since – if the devices are never reused – the length of the provably secret random string that can be generated does indeed depend on the value of a suitable test function. However, iterating such a protocol allows the devices to leak information about (at least) their raw outputs on the first run by generating artificial noise in later rounds, with the level of extra noise chosen to depend suitably on the output values. Such noise statistically affects the length of the output random strings on later rounds.

In this way, suitably programmed devices could ultimately allow Eve to infer all the raw outputs from the first round, given observation of the key string lengths created in later rounds. This makes the round one QRE insecure, since given the raw outputs for round one, and knowing the protocol, Eve knows all information about the output random string for round one, except that determined by the secret random seed.

One defence against this would be to fix a length L for the random string generated corresponding to a maximum acceptable noise level, and then to employ the Procrustean tactic of always reducing the string generated to length L, regardless of the measured noise level.

Even then, though, the abort attack on QKD protocols described above also applies here. The devices have the power to cause the protocol to abort on any round of their choice, and so – if she is willing to wait long enough – Eve can program them to communicate any or all information about their round 1 raw outputs by choosing the round on which they cause an abort.