Memory Attacks on Device-Independent Quantum Cryptography

Jonathan Barrett,[1,2,*] Roger Colbeck,[3,4,†] and Adrian Kent[5,4,‡]

[1]*Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom*
[2]*Department of Mathematics, Royal Holloway, University of London, Egham Hill, Egham TW20 0EX, United Kingdom*
[3]*Institute for Theoretical Physics, ETH Zurich, 8093 Zurich, Switzerland*
[4]*Perimeter Institute for Theoretical Physics, 31 Caroline Street North, Waterloo, Ontario N2L 2Y5, Canada*
[5]*Centre for Quantum Information and Foundations, DAMTP, Centre for Mathematical Sciences, University of Cambridge, Wilberforce Road, Cambridge CB3 0WA, United Kingdom*

(Received 7 March 2012; revised manuscript received 9 October 2012; published 2 January 2013)

Device-independent quantum 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. Here we identify a critical weakness of device-independent protocols that rely on public communication between secure laboratories. Untrusted devices may record their inputs and outputs and reveal information about them via publicly discussed outputs during later runs. Reusing devices thus compromises the security of a protocol and risks leaking secret data. Possible defenses include securely destroying or isolating used devices. However, these are costly and often impractical. We propose other more practical partial defenses as well as a new protocol structure for device-independent quantum key distribution that aims to achieve composable security in the case of two parties using a small number of devices to repeatedly share keys with each other (and no other party).

DOI: 10.1103/PhysRevLett.110.010503 PACS numbers: 03.67.Dd, 03.67.Hk, 03.67.Mn

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., Refs. [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 quantum 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., Ref. [4] for practical illustrations).

The possibility of quantum devices with deliberately concealed flaws, introduced by an untrustworthy manufacturer or saboteur, is particularly concerning, since (i) it is easy to design quantum devices that appear to be following a secure protocol but are actually completely insecure [5], and (ii) there is no general technique for identifying all possible security loopholes in standard quantum cryptography devices. This has led to much interest in device-independent quantum protocols, which aim to guarantee security on the fly by testing the device outputs [6–16]: 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 device-independent security [7,11–13,17], or a large number of entangled pairs to generate a short key [7,17,18]. 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 optimism (e.g., Ref. [19]) about the prospects for device-independent QKD as a practical technology, as well as for device-independent quantum randomness expansion [14–16] and other applications of device-independent quantum cryptography (e.g., Ref. [20]).

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 composable—i.e., do individually secure protocols of this type remain secure when combined? It is clear that reuse of untrusted devices cannot be universally composable; i.e., such devices cannot be securely reused for completely general purposes (in particular, if they have memory, they must be kept secure after the protocol). However, for device-independent quantum cryptography to have significant practical value, one would hope that devices can at least be reused for the same purpose. For example, one would like to be able to implement a QKD protocol many times, perhaps with different parties each time, with a guarantee that all the generated keys can be securely used in an arbitrary environment so long as the devices are kept secure. We focus on this type of composability here.

We describe a new type of attack that highlights pitfalls in producing protocols that are composable (in the above sense) with device-independent security for reusable devices, and show that for all known protocols such composability fails in the strong sense that purportedly secret data become completely insecure. The leaks do not exploit new side channels (which proficient users are assumed to block), but instead occur through the device choosing its outputs as part of a later protocol.

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 (Eve) can design devices so that they record and store all their inputs and outputs. A well designed device-independent protocol can prevent the devices from leaking information about the generated key during that protocol. However, when they are reused, the devices can make their outputs in later runs depend on the inputs and outputs of earlier runs, and, if the protocol requires Alice and Bob to publicly exchange at least some information about these later outputs (as all existing protocols do), this can leak information about the original key to Eve. Moreover, in many existing protocols, such leaks can be surreptitiously hidden in the noise, hence allowing the devices to operate indefinitely like hidden spies, apparently complying with security tests, and producing only data in the form the protocols require, but nonetheless actually eventually leaking all the purportedly secure data.

We stress that our results certainly do not imply that quantum key distribution *per se* is insecure or impractical. In particular, our attacks do not apply to standard QKD protocols in which the devices' properties are fully trusted, nor if the devices are trusted to be memoryless (but otherwise untrusted), nor necessarily to protocols relying on some other type of partially trusted devices. Our target is the possibility of (full) device-independent quantum cryptographic security, applicable to users who purchase devices from a potentially sophisticated adversarial supplier and rely on no assumption about the devices' internal workings.

The attacks we present raise new issues of composability and point towards the need for new protocol designs. We discuss some countermeasures to our attacks that appear effective in the restricted but relevant scenario where two users only ever use their devices for QKD exchanges with one another, and propose a new type of protocol that aims to achieve security in this scenario while allowing device reuse. Even with these countermeasures, however, we show that security of a key generated with Bob can be compromised if Alice uses the same device for key generation with an additional party. This appears to be a generic problem against which we see no complete defense.

Although we focus on device-independent QKD for most of this work, our attacks also apply to other device-independent quantum cryptographic tasks. The case of randomness expansion is detailed in Part VII of the Supplemental Material [21].

Cryptographic scenario.—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 sublaboratories, and we assume Alice and Bob can prevent communication between their sublaboratories as well as between their labs and the outside world, except as authorized by the protocol. The setup of these laboratories is as follows. Each party has a trusted private random string, a trusted classical computer, and access to two channels connecting them. The first channel is an insecure quantum channel. Any data sent down this can be intercepted and modified by Eve, who is assumed to know the protocol. The second is an authenticated classical channel which Eve can listen to but cannot impersonate; in efficient QKD protocols this is typically implemented by using some key bits to authenticate communications over a public channel. Each party also uses a sublaboratory to isolate each of the untrusted devices being used for today's protocol. They can connect them to the insecure quantum channel, as desired, and this connection can be closed thereafter. They can also interact with each device classically, supplying inputs (chosen using the trusted private string) and receiving outputs, without any other information flowing into or out of the secure sublaboratory.

As mentioned before, existing device-independent QKD protocols that have been proven unconditionally secure [7,12,13] require separate devices for each measurement performed by Alice and Bob with no possibility of signaling between these devices [22], or are inefficient [18] (in terms of the amount of key per entangled pair). For practical device-independent QKD, we would like to remove both of these disadvantages and have an efficient scheme needing a small number of devices.

Since the protocols in Refs. [12,13] can tolerate reasonable levels of noise and are reasonably efficient, we look first at implementations of protocols taking the form of those in Refs. [12,13], except that Alice and Bob use one measurement device each; i.e., Alice (Bob) uses the same device to perform each of her (his) measurements. We call these two-device protocols. (Bob also has a separate isolated source device; see below.) The memory of a device can then act as a signal from earlier to later measurements; hence, the security proofs of Refs. [12,13] do not apply (see also Ref. [23] where a different two-device setup is discussed). It is an open question whether a secure key can be efficiently generated by a protocol of this type in this scenario. Here we demonstrate that, even if a key can be securely generated, repeat implementations of the protocol using the same devices can render an earlier generated key insecure.

Attacks on two-device protocols.—Consider a QKD protocol with the standard structure shown in Table I. We imagine 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 the 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 has access to the insecure quantum channel over which the new quantum states are distributed, she can surreptitiously modulate these quantum states to carry new classical instructions to the device in Alice's lab, for example using additional degrees of freedom in the states. These instructions tell the device the error correction and privacy amplification functions used on day 1, allowing it 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. We call this type of attack, in which Eve attempts to gain information from the classical messages sent in step 5, a parameter estimation attack.

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 roughly 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) mask her cheating as noise. (Note that in unconditional security proofs it is generally assumed that eavesdropping is the cause of all noise. Even if in practice Eve cannot reduce the noise to zero, she can supply less noisy components than she claims and use the extra tolerable noise to cheat.)

In addition, Alice and Bob's devices each separately have the power to cause the protocol to abort on any day of their choice. Thus—if she is willing to wait long enough—Eve can program them to communicate some or all information about their day 1 key, for instance by encoding the relevant bits as a binary integer $N = b_1, \ldots, b_m$ and choosing to abort on day $(N + 2)$ [26]. We call this type of attack an abort attack. Note that it cannot be detected until it is too late.

As mentioned above, some well known protocols use many independent and isolated measurement devices. These protocols are also vulnerable to memory attacks, as explained in Part VI of the Supplemental Material [21].

Modified protocols.—We now discuss ways in which these attacks can be partly defended against.

Countermeasure 1.—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 a separate isolated device held by Bob (as in the protocol in Table I) and Bob (rather than Alice) sends selected output data over a public channel in

TABLE I. Generic structure of the protocols we consider. Although this structure is potentially restrictive, most protocols to date are of this form (we discuss modifications later). Note that we do not need to specify the precise subprotocols used for error correction or privacy amplification. For an additional remark, see Part I of the Supplemental Material [21].

1. Entangled quantum states used in the protocol are generated by a device Bob holds (which is separate and kept isolated from his measurement device) and then shared over an insecure quantum channel with Alice's device. Bob feeds his half of each state to his measurement 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 ith 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 [24] 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 [25], 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.

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).

Existing protocols modified in this way are still insecure if reused, however. For example, in a modified parameter estimation attack, Eve can preprogram 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.

In addition, 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 secret key bits from day 1 on a later day using an abort attack.

Countermeasure 2.—Encrypt the parameter estimation information sent in step 5 with some initial preshared seed randomness (this was suggested to us in Ref. [27]). Provided the seed required is small compared to the size of final string generated (which is the case in efficient QKD protocols [12,13]), the protocol then performs key expansion [28]. Furthermore, even if they have insufficient initial shared key to encrypt the parameter estimation information, Alice and Bob could communicate the parameter estimation information unencrypted on day 1, but encrypt it on subsequent days using generated key.

Note that this countermeasure is not effective against abort attacks, which can now be used to convey all or part of their day 1 raw key. This type of attack seems unavoidable in any standard cryptographic model requiring composability and allowing arbitrarily many device reuses if either Alice or Bob has only a single measurement device.

This countermeasure is also not effective in general cryptographic environments involving communication with multiple users who may not all be trustworthy. Suppose that Alice wants to share key with Bob on day 1, but with Charlie on day 2. If Charlie becomes corrupted by Eve, then, for example by hiding data in the parameter estimation, Eve can learn about day 1's key (we call this an impostor attack). This attack applies in many scenarios in which users might wish to use device-independent QKD. For example, suppose Alice is a merchant and Bob is a customer who needs to communicate his credit card number to Alice via QKD to complete the sale. The next day, Eve can pose as a customer, carry out her own QKD exchange with Alice, and extract information about Bob's card number without being detected.

For discussion of a related countermeasure, in which the privacy amplification function is encrypted, see Part II of the Supplemental Material [21].

Countermeasure 3.—Alternative protocols using additional measurement devices. Suppose Alice and Bob each have m measurement devices, for some small integer $m \geq 2$. They perform steps 1–6 of a protocol that takes the form given in Table I but with countermeasures 1 and 2 applied. They repeat these steps for each of their devices in turn, ensuring no communication between any of them (i.e., they place each in its own sublaboratory). This yields m error-corrected strings. Alice and Bob concatenate their strings before performing privacy amplification as in step 7. However, they further shorten the final string such that it would (with near certainty) remain secure if one of the m error-corrected strings were to become known to Eve through an abort attack. (See Table 2, and Part IV of the Supplemental Material [21] for more details.)

This countermeasure doesn't avoid impostor attacks. Instead, the idea is to prevent useful abort attacks (as well as parameter estimation attacks due to countermeasure 2), and hence give us a secure and composable protocol, provided the keys produced on successive days are always between the same two users. The information each device has about day 1's key is limited to the raw key it produced. Thus, if each device is programmed to abort on a particular day that encodes their day 1 raw key, then after an abort, Eve knows one of the devices' raw keys and has some information on the others (since she can exclude certain possibilities based on the lack of abort by those devices so far). After an abort, Alice and Bob should cease to use any of their devices unless and until such time that they no longer require that their keys remain secret. Intuitively, provided the set of m keys was sufficiently shortened in the privacy amplification step, Eve has essentially no information about the day 1 secret key, which thus should remain secure.

In summary, we have shown how 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 of this threat seems to have been overlooked in the literature on device-independent quantum cryptography 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 in a subsequent protocol. Since even causing a protocol to abort communicates some information to Eve, the class of tasks potentially affected is large indeed. In particular, for one of the most important applications, QKD, none of the protocols so far proposed remain composably secure in the case that the devices are supplied by a malicious adversary.

We have also discussed some possible defenses and countermeasures against our attacks. A theoretically simple one is to dispose of—i.e., securely destroy or isolate—untrusted devices after a single use (see Part III of the Supplemental Material [21]). While this would restore universal composability, it is clearly costly and would severely limit the practicality of device-independent quantum cryptography. Another interesting possibility is to design protocols for composable device-independent QKD guaranteed secure in more restricted scenarios. However, the impostor attacks described above appear to exclude the possibility of composably secure device-independent QKD when the devices are used to exchange key with several parties (at least one of whom may become corrupted).

Many interesting questions remain open. Nonetheless, the attacks we have described merit a serious reappraisal of current protocol designs and, in our view, of the practical scope of universally composable quantum cryptography using completely untrusted devices.

We thank Anthony Leverrier and Gonzalo de la Torre for Ref. [27], Lluís Masanes, Serge Massar and Stefano Pironio for helpful comments. J. B. was supported by the EPSRC, and the CHIST-ERA DIQIP project. R. C. acknowledges support from the Swiss National Science Foundation (Grants No. PP00P2-128455 and No. 20CH21-138799) and the National Centre of Competence in Research "Quantum Science and Technology." A. K. was partially supported by a Leverhulme Research Fellowship, a grant from the John Templeton Foundation, and the EU Quantum Computer Science project (Contract No. 255961). This research is supported in part by Perimeter Institute for Theoretical Physics. 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.

*jonathan.barrett@cs.ox.ac.uk
†colbeck@phys.ethz.ch
‡a.p.a.kent@damtp.cam.ac.uk

[1] C. H. Bennett and G. Brassard, in *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing* (IEEE, New York, 1984), pp. 175–179.

[2] A. K. Ekert, Phys. Rev. Lett. **67**, 661 (1991).

[3] R. Renner, Ph.D. thesis, Swiss Federal Institute of Technology, Zurich, 2005, arXiv:quant-ph/0512258.

[4] I. Gerhardt, Q. Liu, A. Lamas-Linares, J. Skaar, C. Kurtsiefer, and V. Makarov, Nat. Commun. **2**, 349 (2011).

[5] In BB84 [1], for example, a malicious state creation device could be programmed to secretly send the basis used for the encoding in an additional degree of freedom.

[6] D. Mayers and A. Yao, in *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS-98)* (IEEE Computer Society, Los Alamitos, CA, 1998), pp. 503–509.

[7] J. Barrett, L. Hardy, and A. Kent, Phys. Rev. Lett. **95**, 010503 (2005).

[8] A. Acin, N. Gisin, and L. Masanes, Phys. Rev. Lett. **97**, 120405 (2006).

[9] V. Scarani, N. Gisin, N. Brunner, L. Masanes, S. Pino, and A. Acín, Phys. Rev. A **74**, 042339 (2006).

[10] A. Acin, N. Brunner, N. Gisin, S. Massar, S. Pironio, and V. Scarani, Phys. Rev. Lett. **98**, 230501 (2007).

[11] L. Masanes, R. Renner, M. Christandl, A. Winter, and J. Barrett, arXiv:quant-ph/0606049v4.

[12] E. Hänggi and R. Renner, arXiv:1009.1833.

[13] L. Masanes, S. Pironio, and A. Acín, Nat. Commun. **2**, 238 (2011).

[14] R. Colbeck, Ph.D. thesis, University of Cambridge, 2007, arXiv:0911.3814.

[15] S. Pironio, A. Acin, S. Massar, A. B. de la Giroday, D. N. Matsukevich, P. Maunz, S. Olmschenk, D. Hayes, L. Luo, T. A. Manning *et al.*, Nature (London) **464**, 1021 (2010).

[16] R. Colbeck and A. Kent, J. Phys. A **44**, 095305 (2011).

[17] J. Barrett, A. Kent, and S. Pironio, Phys. Rev. Lett. **97**, 170409 (2006).

[18] J. Barrett, R. Colbeck, and A. Kent, arXiv:1209.0435.

[19] A. Ekert, Phys. World **22**, 28 (2009).

[20] J. Silman, A. Chailloux, N. Aharon, I. Kerenidis, S. Pironio, and S. Massar, Phys. Rev. Lett. **106**, 220501 (2011).

[21] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.110.010503 for further remarks and a discussion of privacy amplification, multidevice protocols and randomness expansion.

[22] Within the scenario described above, this could be achieved by placing each device in its own sublaboratory.

[23] E. Hänggi, R. Renner, and S. Wolf, arXiv:0906.4760.

[24] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt, Phys. Rev. Lett. **23**, 880 (1969).

[25] C. H. Bennett, G. Brassard, and J.-M. Robert, SIAM J. Comput. **17**, 210 (1988).

[26] In practice, Eve might infer a day $(N + 2)$ abort from the fact that Alice and Bob have no secret key available on day $(N + 2)$, which in many scenarios might detectably affect their behavior then or subsequently. Note too that she might alternatively program the devices to abort on every day from $(N + 2)$ onwards if this made N more easily inferable in practice.

[27] G. de la Torre and A. Leverrier (private communication).

[28] QKD is often referred to as quantum key expansion in any case, taking into account that a common method of authenticating the classical channel uses preshared randomness.