Quantum Information and Computation, Vol. 12, No. 5&6 (2012) 0448–0460
© Rinton Press

LONG DISTANCE QUANTUM CRYPTOGRAPHY MADE SIMPLE

IORDANIS KERENIDIS
CNRS, Laboratoire de Recherche en Informatique, Univ Paris 11, Orsay
Centre for Quantum Technologies, National University of Singapore, 2 Science Drive 3
117543 Singapore

STEPHANIE WEHNER
Centre for Quantum Technologies, National University of Singapore, 2 Science Drive 3
117543

Received September 10, 2011
Revised February 3, 2012

Any two-party cryptographic primitive can be implemented using quantum communication under the assumption that it is difficult to store a large number of quantum states perfectly. However, achieving reliable quantum communication over long distances remains a difficult problem. Here, we consider a large network of nodes with only neighboring quantum links. We exploit properties of this cloud of nodes to enable any two nodes to achieve security even if they are not directly connected. Our results are based on techniques from classical cryptography and do not resort to technologically difficult procedures like entanglement swapping. More precisely, we show that oblivious transfer can be achieved in such a network if and only if there exists a path in the network between the sender and the receiver along which all nodes are honest. Finally, we show that useful notions of security can still be achieved when we relax the assumption of an honest path. For example, we show that we can combine our protocol for oblivious transfer with computational assumptions such that we obtain security if either there exists an honest path, or, as a backup, at least the adversary cannot solve a computational problem.

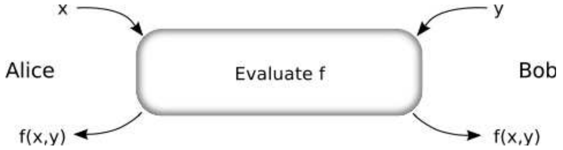
Keywords: quantum cryptography, oblivious transfer, two-party protocols

Communicated by: S Braunstein & H Zbinden

1 Introduction

Quantum communication allows us to achieve cryptographic security without relying on unproven computational assumptions. Two nodes, Alice and Bob, can establish a secure key using quantum key distribution [3, 7], and, moreover, solve any two-party cryptographic problem even if they do not trust each other in the noisy-storage model [32, 17, 27]. Well-known examples of such problems include secure identification [5], as well as electronic voting and secure auctions. More generally, Alice and Bob wish to solve problems where Alice holds an input x (eg. the amount of money she is willing to bid for an item sold by Bob) and Bob holds an input y (e.g. his minimum asking price), and they want to obtain the value of some function $f(x, y)$ (e.g. output no if $x < y$, and x otherwise) as depicted below. In this setting, there is no outside eavesdropper but Alice or Bob themselves may be dishonest.

Security thereby means that Alice should not learn anything about y and Bob should not learn anything about x, apart from what can be inferred from the value of $f(x, y)$ [33].



Unfortunately, quantum communication over long distances poses a formidable problem. At present, quantum key distribution has been achieved over a distance of at most 145km in fiber [25] or 144km in freespace [28, 31]. In addition, having a direct communication link between any two nodes that may wish to communicate is an infeasible problem even when it comes to classical communication. Instead, we have networks of nodes, such as the present day internet, in which only some nodes are directly connected, but are willing to relay communication for other nodes who do not share a direct link. Typically it is easy to connect two nodes who are physically close. In order to achieve longer distances, many forms of quantum repeaters have been proposed in order to extend the range of quantum communication and obtain a quantum version of the internet [18, 16]. Broadly speaking, quantum repeaters used in key distribution come in two variants: in the first, the nodes along the path between Alice and Bob are trusted, and we perform quantum key distribution between each two neighbours. This form of repeater is known as trusted relay and was for example used in the network of SECOQC [1], which is similar to classical methods used in wireless networks [6].[a] The second method is to have the intermediary nodes create entanglement, allowing Alice and Bob to create entanglement between them using the concept of entanglement swapping [13]. This is clearly more desirable than relying on trusted relays, but technologically very difficult to achieve especially when there are many intermediary nodes. Many experiments have been done over the last twelve years [22, 24, 11], but still we are far from using this technology for QKD [1], and similarly for the case of two-party computation in the noisy-storage model. What both of these approaches have in common is that they first try to create the analog of a point-to-point link between Alice and Bob to solve the final cryptographic task.

Here, we take a novel approach using techniques from classical cryptography to bridge the potentially large physical distance between Alice and Bob. Concretely, we consider for the first time the case where any two nodes that are directly connected by a (quantum) communication link can securely solve the universal cryptographic problem of oblivious transfer (OT), which in turn enables them to solve any two-party cryptographic problem [15]. Implementations of such protocols (*link*-OT) can be found in the noisy-storage model [32, 17, 27]. We thereby assume that the network topology is fixed from the start. Any node in the network may behave honestly, or be dishonest in the sense that it will collaborate with the dishonest Alice or Bob. Note that this implies that all dishonest nodes work together with one of the two parties. A dishonest node also has full control over the communication links attached to it, making it more powerful than for example the eavesdropper in QKD who only has access to the communication link and not to any of the individual labs. Moreover, since we assumed that any two nodes that are directly connected by a quantum communication link can solve any two-party cryptographic problem, this in particular implies that two honest parties can

[a] We would like to emphasize that the problems solved in SECOQC aim for secure communications, rather than secure multi-party computations.

securely exchange messages, without leaking information to an eavesdropper. Another way to achieve this is by performing QKD and then encrypting the messages via a one-time pad encryption scheme. Note that our analysis differs from investigations done in multi-party secure computation for the case that the network graph connecting the players is not fully connected. There it is typically assumed that the players can still perform secure broadcast, even if the graph is not fully connected (see [9] and references therein). We make no such assumption here, but instead start out from pairwise oblivious transfer to implement oblivious transfer between two remote nodes.

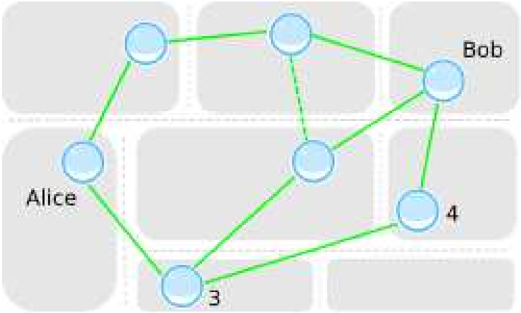


Fig. 1. If any two nodes with a direct link can perform oblivious transfer, then Alice and Bob can solve any two-party cryptographic problem as long as there exists a path from Alice to Bob (e.g., 3 and 4) along which all intermediary nodes are honest, or the cheating party cannot solve a computational problem efficiently.

Results We first provide a simple protocol for oblivious transfer between Alice and Bob who do not share a direct quantum link (*path*-OT), that is secure for both parties, as long as all nodes along one of the paths from Alice to Bob are honest (we provide the definition of oblivious transfer and the notion of security in the following section). We will refer to this path as an *honest path* from Alice to Bob, which is in flavor similar to recent extensions to the idea of trusted relays for QKD [26]. Whereas this may seem like a strong assumption, we prove that this is in fact all we can hope to achieve without any additional resources: Without an honest path no protocol between Alice and Bob can be secure when we are only given the resource of link-OT and classical communication. Furthermore, without link-OT we also cannot hope to obtain a secure protocol, i.e., having access to a large network of nodes does not allow us to solve the problem of oblivious transfer on its own [b].

We then successively relax the assumption of an honest path. First, since providing security only for the case when a honest path exists may still be rather unsatisfactory [c], we show that we can add a security backup. More precisely, our protocol can be made secure if there is either an honest path, or at least the dishonest node cannot break a computational assumption. Note that this also means that even with the assumption of an honest path, quantum communication (allowing for link-OT) only increases the security over classical computational assumptions. We then show that the assumption of the honest path can be relaxed if each

[b] Note that there could be only a single node on the honest path from Alice to Bob, which means we cannot simply solve this problem by assuming only very few of the nodes are dishonest as in secure multi party computation.

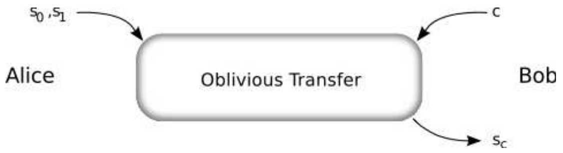
[c] When there are only few paths connecting Alice and Bob it is less likely for such a path to exist

pair of nodes are given a classical shared key for free. Finally, we look at the case where an honest Alice (similarly for an honest Bob) performs an oblivious transfer protocol in a network where everyone else is dishonest. We show that even in this case, where we have made no assumptions, a non-trivial notion of security is still achievable for Alice and Bob. More precisely, we will consider a variant of Oblivious Transfer (defined in more detail below) which is still impossible to achieve classically but becomes possible in the presence of quantum information.

Our results open the door for extending implementations of oblivious transfer in the noisy-storage model to large distances similar to the case of QKD [1].

2 The protocol

Let us first explain the problem of oblivious transfer (OT) [23]; a formal definition can be found in [17]. Alice (the sender) holds two input strings $s_0, s_1 \in \{0,1\}^\ell$ [d]and Bob (the receiver) holds a choice bit $c \in \{0,1\}$. If both nodes are honest, Bob should receive the input of his choosing, s_c, at the end of the protocol.



If Bob is honest, then our goal is to ensure that whatever attack Alice may mount, she can nevertheless not gain any information about c. Conversely, if Alice is honest, we want that a dishonest Bob is unable to gain any information about at least one of Alice's inputs, s_{1-c}. Whereas oblivious transfer by itself may seem like a rather obscure task, it has in fact been shown that Alice and Bob can use it to solve any other cryptographic problem securely [15]. Below we use $\mathrm{OT}((s_0, s_1), c)$ to indicate that we use a link-OT protocol as a black box.

The above definition corresponds to the so-called 1-out-of-2 OT, where Alice has two strings and Bob learns one of them. One can generalize it to the k-out-of-n OT, where Alice holds n different strings and Bob learns k of them.

We now provide two protocols, where the first is unconditionally secure for the sender Alice [e]and secure for the receiver Bob provided there is an honest path. The second has exactly opposite security properties: it is unconditionally secure for Bob and secure for Alice provided there is an honest path.

2.1 A protocol unconditionally secure against cheating Bob

Let E be the number of links in the network and let N be the number of paths connecting Alice to Bob that will be used in the protocol. We denote by $v_1, \ldots, v_N$ the nodes adjacent to Alice on the N possible paths. We use '+' and '$\cdot$' to indicate bitwise addition and multiplication modulo 2 respectively [f], and $\in_R$ to denote that a variable has been chosen uniformly and independently at random.

[d] The two input strings are chosen uniformly at random, unknown to Bob.

[e] It is secure even if there is no honest path.

[f] The bitwise addition of two strings $a, b \in \{0,1\}^\ell$ is defined as the string $(a_1 + b_1 \mod 2, \ldots, a_\ell + b_\ell \mod 2)$.

Protocol 1:
Input: $s_0, s_1 \in \{0,1\}^\ell$ for Alice, $c \in \{0,1\}$ for Bob Output: s_c to Bob.

Bob: Chooses N bits $c_1, \ldots, c_N \in_R \{0,1\}$ such that $c = c_1 + \ldots + c_N$. He encrypts and sends c_j to node v_j along the j-th path.

Alice: Chooses N keys $r_1, \ldots, r_N \in_R \{0,1\}^\ell$ such that $r_1 + \ldots + r_N = 0$.

Performs $\mathrm{OT}((t_{0,1}, t_{1,1}), c_1)$ with with node v_1 where $t_{0,1} = s_0 + r_1$ and $t_{1,1} = s_1 + r_1$. Performs $\mathrm{OT}((t_{0,j}, t_{1,j}), c_j)$ with nodes $v_2, \ldots, v_N$ where $t_{0,j} = r_j$ and $t_{1,j} = s_0 + s_1 + r_j$.

Intermediary Nodes: Node v_j sends $t_{c_j,j}$ to Bob along the j-th path.

Bob: Computes $s_c = t_{c_1,1} + \ldots + t_{c_N,N}$.

Let us first detail what we mean by "Bob encrypts and sends c_j to node v_j along the j-path“. Without loss of generality, we have assumed that each pair of adjacent nodes can create a secret key and communicate encrypted messages (this follows form the assumption that adjacent nodes can perform OT or by performing QKD). Hence, Bob sends to his adjacent node in the j-th path, the name of the path j and an encryption of the bit c_j (by using one bit of a secret key). The adjacent node, decrypts the bit c_j, encrypts it using one bit of the shared key with the following node on the j-path and sends it, along with the name of the path j. This process continues until the bit c_j reaches the node v_j. Note that the bit c_j remains secure against any eavesdropper, as long as all nodes along the path j are honest.

Note that the intermediate nodes send $t_{c_j,j}$ to Bob along the j-th path without having to encrypt it.

We now study the correctness and security of the protocol.

First, the protocol is correct when both players are honest, since Bob computes

$$s_c = t_{c_1,1} + \ldots + t_{c_N,N} = s_0 + (s_0 + s_1) \cdot c \tag{1}$$

On every one of the N paths, we use one link-OT and secret keys of length (in bits) equal to the length of the path (which cannot be larger than E). Hence, the rate is given by

$$R_{\mathrm{OT}} = \frac{1}{N \cdot E} \; . \tag{2}$$

Note that since there are no errors in the link-OTs themselves, there will be no errors in the resulting protocol.

We now argue that the protocol is also secure. We provide a brief sketch of our argument here; details can be found in the appendix. Suppose first that Bob is dishonest. Note first that even if Bob is working together with *all* intermediary nodes, he can only learn at most one of $(t_{0,j}, t_{1,j})$ from each of the N link-OTs. However, since Alice uses fresh keys $\{r_j\}_j$ in each round, and s_0 and s_1 are themselves completely unknown to Bob, one can show that Bob would need to retrieve at least $N+1$ entries $t_{c_j,j}$ in order to compute both s_0 and s_1. Hence, Bob learns nothing about one of s_0 or s_1 as desired.

Suppose now that Bob is honest, and there exists an honest path between Alice and Bob.

Note that Bob effectively performs a secret sharing of his input along all paths [g] so that Alice needs all shares $\{c_j\}_j$ in order to recover c [29]. However, the share on the honest-path remains unknown to Alice since it has been securely transmitted (using the secret keys between adjacent pairs of nodes). The security of the link-OT ensures she cannot use it to gain any information about c either.

Clearly, there is a tradeoff between the rate and the security of the protocol. By reducing the number of paths used in the protocol, we can increase the rate but at the same time there may no longer exist an honest path among the now smaller set of possible paths. This means that security for honest Bob could no longer be guaranteed, even though an honest path exists in the network.

2.2 A protocol unconditionally secure against cheating Alice

Our second protocol is similar, but with Alice performing a secret sharing of her inputs. Let $w_1, \ldots, w_N$ be the nodes adjacent to Bob on the N possible paths.

Protocol 2:
Input: $s_0, s_1 \in \{0,1\}^\ell$ for Alice, $c \in \{0,1\}$ for Bob Output: s_c to Bob.

Alice: Chooses N strings $s_{01}, \ldots, s_{0N} \in_R \{0,1\}^\ell$ such that $s_0 = s_{01} + \ldots + s_{0N}$ and similarly $s_{11}, \ldots, s_{1N} \in_R \{0,1\}^\ell$ such that $s_1 = s_{11} + \ldots + s_{1N}$. She encrypts and sends l-bit strings s_{0j}, s_{1j} to node w_j, i.e. the j-th neighbour of Bob via the j-th path.

Bob: Performs OT$((s_{0j}, s_{1j}), c)$ with nodes w_j for all j. Computes $s_c = s_{c1} + \ldots + s_{cN}$.

Clearly, the protocol is correct if both parties are honest. On every one of the N paths, we use one link-OT and l bits of a secret key for every link of the path (which cannot be larger than E). Hence, the rate is given by

$$R_{\mathrm{OT}} = \frac{1}{N \cdot l \cdot E} \; . \tag{3}$$

The security of the link-OT for the receiver ensures that even if a dishonest Alice controls all nodes adjacent to Bob, she nevertheless cannot learn c. Finally, the protocol is secure against a dishonest Bob, assuming that there exists an honest path: In this case, at least one of the shares s_{0j} or s_{1j} remains unknown to Bob, since they are securely transmitted to node w_j via the honest path, and the link-OT protocol between w_j and Bob is secure for the sender. Hence, he cannot learn both inputs s_0, s_1.

One may wonder whether we could have constructed a path-OT protocol without relying on the existence of a link-OT protocol, which is impossible to obtain without assumptions [19]. However, it is easy to see that the existence of *any* path-OT protocol would imply a secure link-OT protocol between two directly connected parties, Anne and Bill: First, Anne picks a path from Alice to Bob in the original setting. Then Bill picks a path from Bob to Alice. The remaining paths they split arbitrarily. Now Anne acts as Alice would and in addition

[g] Informally, a secret sharing is a procedure where a secret c is split up into so-called shares, here $\{c_j\}_j$, such that c cannot be reconstructed without obtaining all shares. Relaxed schemes exist where also a smaller number of shares is enough to reconstruct the secret.

simulates the action of all nodes in the paths assigned to her. Bill also simulates the actions of Bob together with all nodes in the paths assigned to him. Clearly, no matter who will be dishonest, we are always in the setting where there is an honest path in the original protocol, as one path is always simulated by someone being honest. This means that we cannot hope to achieve OT in the honest-path model without additional assumptions either [19].

3 Security without an honest path

However, one might still hope that given such a strong primitive as link-OT we might be able to achieve security using only classical communication, even without the assumption of an honest path. Unfortunately, it turns out that an honest path is indeed a necessary condition for security: If there is no honest path, then there exists a subset of corrupted nodes M, such that any communication between Alice and Bob goes through them. Intuitively this means that either M can gain information about c, or else must know enough about s_0 and s_1 to be able to supply Bob with the desired output. In the first case, dishonest Alice can learn c from M, and in the second dishonest Bob can break security by learning information about both of Alice's inputs. This holds even for weak forms of oblivious transfer where we allow an error in the security (see appendix).

A security backup: Nevertheless, the assumption of an honest path may appear quite strong, and it would be useful to have some security guarantees even if this assumption fails. Fortunately, it is straightforward to adopt existing techniques from classical cryptography [12, 20] to extend our protocols to be secure if either the honest-path assumption holds, or else if the dishonest party cannot break a certain computational problem. To this end, we combine our protocol with a protocol for classical oblivious transfer. OT can be achieved classically under a large variety of assumptions. Here, we choose to combine our protocol with the protocol of Naor and Pinkas [21], which is secure against a dishonest sender if he cannot break the decisional Diffie Hellman problem (DDH), and unconditionally secure against a dishonest receiver. Note that this means that just like for our honest-path assumption, we have unconditional security against one party, and security according to either the DDH or the honest-path assumption against the other.[h] Using the $\{3, 2\}$-robust uniform OT-combiner from [20, Theorem 2] we hence immediately obtain that there exists an oblivious transfer protocol that is secure if either the honest path or the DDH assumption holds using two instances of protocol 1, and two instances of the OT protocol of [21]. An explicit protocol can be found in [20].

Secret keys: In the classical model for secure multiparty computation one usually assumes that there exist private links between all nodes and we are trying to show security against subsets of dishonest nodes. Clearly, this is a strong assumption as it requires us to establish keys over potentially long distances. Nevertheless, it is interesting to consider a hybrid-model, where there exists a complete network of classical private links and also a network of quantum links between neighboring nodes allowing them to perform link-OT. It is easy to see that our protocol can be transformed to achieve security as long as one of the neighbours of Alice and Bob is honest, instead of the entire path being honest: we use the private channels to directly

[h] We can similarly construct a protocol that is secure against a dishonest receiver if there exists an honest path or he cannot break the decisional Diffie Hellman problem (DDH), and unconditionally secure against a dishonest sender.

communicate with the immediate neighbours instead of relying on the entire path. This easy example shows that allowing link-OT is indeed more powerful than what one can hope to gain in the classical model of secure multi-party computation.

No assumptions: Finally, let us consider what happens if we allow an *arbitrary* number of network nodes to be dishonest. Curiously, some weak notion of security still remains.

More specifically, we can perform Protocol 1 and Protocol 2 sequentially with different inputs for Alice and Bob in the two executions. We can, hence, construct a 2-out-of-4 path-OT, where Alice has four inputs s_0, s_1, t_0, t_1, and Bob has two choice bits c and d such that: if everyone is honest, then Bob learns the bits s_c and t_d, and Alice learns nothing about Bob's two choice bits. If Alice is honest, but everyone else in the network is dishonest, then Bob learns three bits, but *not* all four of them (the bit s_{1-c} remains unknown). If Bob is honest, but everyone else is dishonest, then Alice learns one of the two choice bits of Bob, but *not* both of them (d remains unknown). These properties follow directly from our previous analysis.

Note that this weak form of security is still impossible classically on a complete network with private links, unless computational assumptions are added. In our model, it becomes possible because we added the neighboring quantum links and assumed that we can perform short distance OT protocols via these quantum links. One can turn this weak OT protocol into some weak bit commitment protocol as well, leading to weak forms of coin tossing over long distances. On the other hand, this variant of OT is not strong enough to be universal. This, of course, should be expected, since OT is known to be impossible in the quantum world without any assumptions.

Another series of works have considered weaker forms of security for OT, where a dishonest player is allowed to gain some information about the other player's inputs but not all of it [4, 30, 10, 14, 8]. These security guarantees are only possible in the quantum world but again are not strong enough to provide unconditional security for general two-party computation. They defer from our variant, since in our case, the dishonest player gets full information about some inputs and no information about the remaining.

4 Conclusions

We have shown security against dishonest Alice (or Bob) whenever there is at least one honest path, or the dishonest party cannot break a computational assumption. To achieve one instance of OT between distant Alice and Bob, we thereby used a number of link-OTs that is equal to the number of all paths (N) connecting Alice and Bob. Note that the same protocol would work had we picked a smaller number of paths. However, note that decreasing the number of paths may make it less likely to ensure that there is indeed an honest one. One can easily extend our protocol to be robust against the case where the intermediary nodes may be dishonest independently of Alice and Bob, and try to alter Alice's or Bob's input. In our present protocols this is of course possible since they could for example flip one of the bits $\{c_j\}_j$. To make the protocol robust we can simply use a more advanced secret sharing scheme that, similar to an error correcting code, protects against 'errors' introduced in the secrets [2]. Note that depending on our choice of secret sharing scheme, we may require more than one honest path to achieve robustness or more communication rounds [i].

Our protocols show that two-party cryptographic primitives can be implemented over long distances in an extremely simple manner. Our result enables us to extend the range of protocols in the noisy-storage model in a similar way as has been done in QKD [1]. Clearly, our protocols still require a considerable amount of classical communication. However, this is technologically much easier to achieve than entanglement swapping which of course still remains the more desirable solution. The quantum operations that the nodes are performing are no harder than the ones necessary in the link-OT protocols, i.e. it suffices that they create and measure BB84 [3] states [17]. No complicated operations like Bell state measurements, or memory are required.

Acknowledgments

SW was supported by NSF grants PHY-04056720, PHY-0803371, as well as the National Research Foundation and the Ministry of Education, Singapore. IK was supported by ANR grants ANR-09-JCJC-0067-01 and ANR-08-EMER-012. Part of this work was done while SW was at the Institute for Quantum Information, Caltech.

References

1. R. Alleaume, J. Bouda, C. Branciard, T. Debuisschert, M. Dianati, N. Gisin, M. Godfrey, P. Grangier, T. Langer, A. Leverrier, N. Lütkenhaus, P. Painchault, M. Peev, A. Poppe, T. Pornin, J. Rarity, R. Renner, G. Ribordy, M. Riguidel, L. Salvail, A. Shields, H. Weinfurter, and A. Zeilinger. Secoqc white paper on quantum key distribution and cryptography. arXiv:quant-ph/0701168, 2007.
2. S. Micali B. Chor, S. Goldwasser and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *Proceedings of 44th IEEE FOCS*, pages 383–395, 1985.
3. C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179, 1984.
4. A. Chailloux, I. Kerenidis, and J. Sikora. 2010.
5. I. B. Damgård, S. Fehr, L. Salvail, and C. Schaffner. Secure identification and QKD in the bounded-quantum-storage model. In *Advances in Cryptology—CRYPTO '07*, volume 4622 of *Lecture Notes in Computer Science*, pages 342–359. Springer-Verlag, 2007.
6. Y. Desmedt. pages 38–41, 2005.
7. A. Ekert. Quantum cryptography based on Bell's theorem. *Physical Review Letters*, 67:661–663, 1991.
8. F. Gao, B. Liu, Q. Wen, and H. Chen. arXiv:1111.1511, 2011.
9. J. Garay and R. Ostrovsky. In *Proceedings of EUROCRYPT*, pages 307–323, 2003.
10. V. Giovannetti, S. Lloyd, and L. Maccone. *Physical Review Letters*, 100:230502, 2008.
11. A. M. Goebel, C. Wagenknecht, Q. Zhang, Y. Chen, K. Chen, J. Schmiedmayer, and J. Pan. Multistage entanglement swapping. *Physical Review Letters*, 101:080403, 2008.
12. D. Harnik, J. Kilian, M. Naor, O. Reingold, and A. Rosen. On robust combiners for oblivious transfer and other primitives. In *Advances in Cryptology — EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 96–113, 2005.
13. M. Żukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert. "event-ready-detectors" bell experiment via entanglement swapping. *Physical Review Letters*, 71(26):4287–4290, Dec 1993.
14. M. Jakobi, C. Simon, N. Gisin, J. Bancal, C. Branciard, N. Walenta, and H. Zbinden. *Physical Review A*, 83:022301, 2001.

[i] This is, for example, the case when letting Alice and Bob use some runs of the protocol to detect tampering by any intermediary nodes.

15. J. Kilian. Founding cryptography on oblivious transfer. In *Proceedings of 20th ACM STOC*, pages 20–31, 1988.
16. H. J. Kimble. The quantum internet. *Nature*, 453:1023–1030, 2008.
17. R. König, S. Wehner, and J. Wullschleger. Unconditional security from noisy quantum storage. arXiv:0906.1030, 2009.
18. S. Lloyd, J. H. Shapiro, F. N. C. Wong, P. Kumar, S. M. Shahriar, and H.P. Yuen. *ACM SIGCOMM Computer Communication Review*, 34(5):9–20, 2004.
19. H-K. Lo. Insecurity of quantum secure computations. *Physical Review A*, 56:1154, 1997.
20. R. Meier, B. Przydatek, and J. Wullschleger. Robuster combiners for oblivious transfer. In *Theory of Cryptography Conference — TCC*, Lecture Notes in Computer Science, 2007.
21. M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *Proceedings of 12th SODA*, pages 448–457, 2001.
22. J. Pan, D. Bouwmeester, H. Weinfurter, and A. Zeilinger. Experimental entanglement swapping: Entangling photons that never interacted. *Physical Review Letters*, 80:3891–3894, 1998.
23. M. Rabin. How to exchange secrets by oblivious transfer. Technical report, Aiken Computer Laboratory, Harvard University, 1981. Technical Report TR-81.
24. M. Riebe, T. Monz, K. Kim, A. S. Villar, P. Schindler, M. Chwalla, M. Hennrich, and R. Blatt. Deterministic entanglement swapping with an ion-trap quantum computer. *Nature Physics*, 4:839–842, 2008.
25. D. Rosenberg, C.G. Rosenberg, J. Harrington, P. Rice, N. Dallman, K.T. Tyagi, K.P. McCabe, R.J. Hughes, J.E. Nordholt, R.H. Hadfield, B. Baek, and S. Nam. Long distance quantum key distribution in optical fiber. In *Proceedings of the Conference on Optical Fiber communication*, pages 1–3, 2008.
26. L. Salvail, M. Peev, E. Diamanti, R. Alleaume, N. Lütkenhaus, and T. Länger. Security of trusted repeater quantum key distribution networks. *Journal of Computer Security*, 18(1):61–87, 2010.
27. C. Schaffner. Simple protocols for oblivious transfer and secure identification in the noisy-quantum-storage model. arXiv:1002.1495. Workshop on Cryptography from Storage Imperfections, Recent work, Caltech, March 20–22, 2010.
28. T. Schmitt-Manderbach, H. Weier, M. Frst, R. Ursin, F. Tiefenbacher, Th. Scheidl, J. Perdigues, Z. Sodnik, J. G. Rarity, A. Zeilinger, and H. Weinfurter. Experimental demonstration of free-space decoy-state quantum key distribution over 144 km. *Physical Review Letters*, 98:010504, 2007.
29. A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.
30. J. Sikora. arXiv:1009.2735, 2011.
31. R. Ursin, F. Tiefenbacher, T. Schmitt-Manderbach, H. Weier, T. Scheidl, M. Lindenthal, B. Blauensteiner, T. Jennewein, J. Perdigues, P. Trojek, B. Ömer, M. Fürst, M. Meyenburg, J. Rarity, Z. Sodnik, C. Barbieri, H. Weinfurter, and A. Zeilinger. Entanglement-based quantum communication over 144 km. *Nature Physics*, 3:481–486, 2007.
32. S. Wehner, C. Schaffner, and B. M. Terhal. Cryptography from noisy storage. *Physical Review Letters*, 100(22):220502, 2008.
33. A. C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual IEEE FOCS*, pages 160–164, 1982.

Appendix A Security of protocol 1

We can show security of our protocol using the formal definitions for fully-randomized OT [17]. However, here we restrict ourselves to the simple arguments below in order to not obscure our argument. These arguments are sufficient since our setting is very straightforward to analyze.

Claim A. 1 *Protocol 1 forms a secure oblivious transfer scheme with unconditional security against Alice, and security against Bob whenever there exists an honest path.*

Proof. We first show that the protocol is correct when both Alice and Bob are honest. This follows immediately by noting that Bob can compute

$$\begin{aligned} s_c &= t_{c_1,1} + \ldots + t_{c_N,N} && \text{(A.1)} \\ &= (s_0 + (s_0 + s_1) \cdot c_1 + r_1) + \sum_{i=2}^{N} ((s_0 + s_1) \cdot c_i + r_2) \\ &= s_0 + (s_0 + s_1) \cdot c \;=\; s_c \;. \end{aligned}$$

We now show that the protocol is secure if Alice is honest, where we allow all intermediary players and Bob to be dishonest. From the security of the link-OT protocol, it follows that Bob can learn at most one of Alice's inputs from each invocation. In the most general cheating strategy, Bob can arbitrarily choose values as his input bits to the N link-OT protocols. Let $d_1, d_2, \ldots, d_N$ denote these inputs and let $t_{d_1,1}, \ldots, t_{d_N,N}$ be the inputs of Alice that Bob learns. Note that for any choice of Bob's inputs $\{d_i\}_i$ there exists a $c \in \{0,1\}$ such that $t_{d_1,1} + \ldots + t_{d_N,N} = s_c$. Moreover, $t_{d_1,1} + \ldots + t_{d_{N-1},N-1} + t_{1-d_N,N} = s_{1-c}$. Our goal is now to show that Bob cannot gain any information about s_{1-c}. First of all, note that since Alice uses fresh keys $\{r_j\}_j$ in each link-OT, and s_0 and s_1 are themselves randomly chosen bit strings unknown to Bob, the values of $t_{d_1,1}, \ldots, t_{d_N,N}$ and $t_{1-d_N,N}$ are all independent. Hence, Bob would need to retrieve all such $N+1$ entries in order to compute both s_0 and s_1, which contradicts the security of the link-OT.

It remains to prove security if Bob is honest. Note that Bob effectively performs a secret sharing of his input

$$c = \sum_{j \in \{1,\ldots,N\}} c_j \tag{A.2}$$

along all paths such that the bit c can only be recovered if and only if Alice learns all shares $\{c_j\}_j$. However, Alice has no information about the value of c_j on the honest-path as the communication between honest players is secure. Furthermore, the link-OT used between Alice and v_j is secure for the receiver, and hence we conclude that Alice cannot learn c as promised. □

Appendix B Necessity of the honest path

We now prove that an honest-path is a necessary condition for OT, where we use a weaker definition which is implied by the formal ones given e.g. for (fully randomized) oblivious transfer in [17]. Note that this is sufficient to prove the impossibility of the more difficult task as well. More concretely, the following conditions must hold for any protocol that is both correct and secure. Any impossibility proof for a protocol aiming for perfect security is rather unsatisfactory since we would be willing to accept a very small probability of failure. We hence include a security parameter $\varepsilon > 0$ which intuitively corresponds to the error we are willing to accept.

First of all, for any protocol that is correct we must have that the probability that honest Bob with input c can guess honest Alice's input, s_c, satisfies

$$\text{Correctness: } \Pr[s_c|Bob] \geq 1 - \varepsilon \;. \tag{B.1}$$

Furthermore, if Alice is honest, then for whatever attack Bob may conceive we have that he cannot guess at least one of the two inputs

$$\text{Security against Bob: } \exists b \ \Pr[s_b|Bob] \leq \frac{1}{2^\ell} + \varepsilon \ , \tag{B.2}$$

Finally, if Bob is honest and his input bit is c, then for any strategy of dishonest Alice, she is unable to learn Bob's choice bit

$$\text{Security against Alice: } \Pr[c|Alice] \leq \frac{1}{2} + \varepsilon \ . \tag{B.3}$$

To obtain an impossibility proof, our goal is now to show that (B.1), (B.2) and (B.3) can never be satisfied simultaneously for small values of ε. That is, we can only hope to achieve very imperfect version of oblivious transfer with a large error ε.

Claim B. 1 *There exists no protocol for oblivious transfer based on only link-OT and classical communication that is secure without an honest path between Alice and Bob with security parameter* $\varepsilon < 1/4 - 1/2^{\ell+2}$.

Proof. If there is no honest path, then there exists some subset of potentially dishonest nodes M that separates the network into two disconnected components, one containing Alice and the other Bob. Let us now establish some basic properties of the probabilities that Alice, Bob or M can learn s_0, s_1 and c in an honest execution of any protocol.

Note that in any protocol, Bob cannot gain more information about Alice's inputs than M can, since all information between Alice and Bob runs through M (wlog we can furthermore assume that dishonest Bob would give any shared secret keys with Alice to M for free). Hence, we have that

$$\forall b \ \Pr[s_b|Bob] \leq \Pr[s_b|M] \ . \tag{B.4}$$

Similarly, Alice cannot gain more information about Bob's input than M can, hence

$$\Pr[c|Alice] \leq \Pr[c|M] \ . \tag{B.5}$$

First, suppose that for an honest execution of any protocol the probability that M is able to guess c satisfies $\Pr[c|M] > 1/2 + \varepsilon$. Then, Alice can violate the security condition (B.3) by running the protocol honestly with Bob and then asking M for a guess of c. Hence, it must hold that

$$\Pr[c|M] \leq \frac{1}{2} + \varepsilon \ . \tag{B.6}$$

Second, by the correctness condition (B.1) and equation (B.4), for an honest execution of any protocol, we have

$$\Pr[s_c|M] \geq 1 - \varepsilon \ . \tag{B.7}$$

Third, suppose that for an honest execution of any protocol, $\Pr[s_{1-c}|M] > \frac{1}{2^\ell} + \varepsilon$. Then, Bob can violate the security condition (B.2) by running the protocol honestly with Alice and then asking M for a guess for both s_0 and s_1. Hence, it must hold that

$$\Pr[s_{1-c}|M] \leq \frac{1}{2^\ell} + \varepsilon \ . \tag{B.8}$$

We now show that these conditions imply that whenever Bob is honest, there exists a cheating strategy for Alice. Alice first chooses two random inputs $s_0, s_1 \in \{0,1\}^\ell$, and runs the protocol as an honest Alice would do. Afterwards, she picks a random b and asks M, who by definition will willingly cooperate with any cheating party, to send her a guess $\tilde{s}_b$ for s_b. Note that (B.7) and (B.8) now tell us that the probability that M succeeds is very large for s_c, but extremely small for s_{1-c}. Alice then outputs b as her guess for c if M guessed correctly and $1-b$ if M guessed wrongly. The probability that Alice succeeds using this strategy obeys

$$\begin{aligned}
&\Pr[c|Alice] && \text{(B.9)}\\
&= \Pr[b=c]\Pr[s_c|M] + \Pr[b=1-c](1-\Pr[s_{1-c}|M]) &&\\
&\geq \frac{1}{2}(1-\varepsilon) + \frac{1}{2}(1-\frac{1}{2^\ell}-\varepsilon) && \text{(B.10)}\\
&= (1-\varepsilon-\frac{1}{2^{\ell+1}})\ . && \text{(B.11)}
\end{aligned}$$

Comparing (B.11) with (B.3) concludes our claim. □

Note, however, that OT *is* of course possible if M would be fully quantum, and in particular would be able to perform entanglement swapping between Alice and Bob.