Quantum Private Queries

Vittorio Giovannetti,[1] Seth Lloyd,[2] and Lorenzo Maccone[3]

[1]*NEST-CNR-INFM & Scuola Normale Superiore, Piazza dei Cavalieri 7, I-56126, Pisa, Italy*

[2]*MIT, RLE and Department of Mechanical Engineering, MIT 3-160, 77 Mass. Avenue, Cambridge, Massachusetts 02139, USA*

[3]*QUIT, Dip. Fisica "A. Volta", Universita Pavia, via Bassi 6, I-27100 Pavia, Italy*

(Received 25 October 2007; revised manuscript received 18 December 2007; published 10 June 2008)

We propose a cheat sensitive quantum protocol to perform a private search on a classical database which is efficient in terms of communication complexity. It allows a user to retrieve an item from the database provider without revealing which item he or she retrieved: if the provider tries to obtain information on the query, the person querying the database can find it out. The protocol ensures also perfect data privacy of the database: the information that the user can retrieve in a single query is bounded and does not depend on the size of the database. With respect to the known (quantum and classical) strategies for private information retrieval, our protocol displays an exponential reduction in communication complexity and in running-time computational complexity.

DOI: 10.1103/PhysRevLett.100.230502 PACS numbers: 03.67.Lx, 03.67.Dd, 03.67.Mn

Privacy is a major concern in many information transactions. A familiar example is provided by the transactions between web search engines and their users. On one hand, the user (say Alice) would typically prefer not to reveal to the server the item she is interested in (*user privacy*). On the other hand, the server (say Bob) would like not to disclose more information than that Alice has asked for (*data privacy*). User and data privacy are apparently in conflict: the most straightforward way to obtain user privacy is for Alice to have Bob send her the entire database, leading to no data privacy whatsoever. Conversely, techniques for guaranteeing the server's data privacy typically leave the user vulnerable [1]. At the information theoretical level, this problem has been formalized by Gertner *et al.* as the symmetrically private information retrieval (SPIR) [1]. This is a generalization of the private information retrieval (PIR) problem [2,3] which deals with user privacy alone. (SPIR is closely related to oblivious transfer [4], in which Bob sends to Alice N bits, out of which Alice can access exactly one–which one, Bob does not know.) No efficient solutions in terms of communication complexity [5] are known for SPIR. Indeed, even rephrasing them at a quantum level [6,7], the best known solution for the SPIR problem (with a single database server) employs $O(N)$ qubits to be exchanged between the server and the user [8] and ensures data privacy only in the case of honest users (here N is the number of items contained in the database, while an honest user is defined as one who does not want to compromise her chances of getting the information about the selected item in order to get more). PIR admits protocols that are more efficient in terms of communication complexity [2,3]. As will be seen below, however, both PIR and SPIR necessarily require $O(N)$ computational complexity on the part of the database.

In this Letter we present a new quantum cryptographic primitive [9], the quantum private query (QPQ), which allows an exponential reduction in the communication and computational complexity with respect to the best (quantum or classical) SPIR protocol proposed so far. QPQ ensures perfect data privacy and it exploits a cheat sensitive strategy [10] that allows Alice to determine whether Bob has been trying to cheat to obtain information about her query. In other words, Alice can ask Bob's database a question and obtain the answer, together with a quantum certificate that Bob retains no record of what question she asked. With respect to (classical or quantum) SPIR and oblivious transfer protocols QPQ presents an

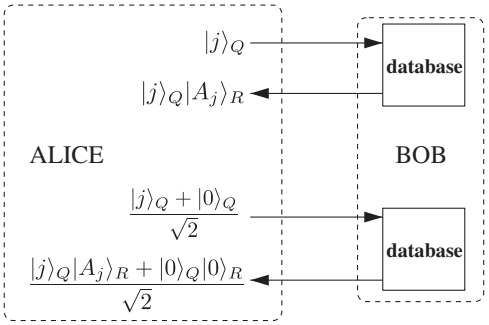

FIG. 1. Scheme of the QPQ protocol. Alice wants to find out the jth record of Bob's database. She then prepares two n-qubit registers, one contains the state $|j\rangle_Q$, the other contains the quantum superposition $(|j\rangle_Q + |0\rangle_Q)/\sqrt{2}$. (She knows that the 0th record of Bob's database contains the fixed value $A_0 = 0$). She then sends, in random order, these two registers to Bob, waiting for his first reply before sending the second. Bob uses each of the two registers to interrogate his database using a qRAM, which records the reply to her queries in a register R. At the end of their exchange, Alice possesses the states $|j\rangle_Q|A_j\rangle_R$ and $(|j\rangle_Q|A_j\rangle_R + |0\rangle_Q|0\rangle_R)/\sqrt{2}$, where the A_j is the content of the jth record in the database. By measuring the first she obtains the value of A_j, with which she can check whether the superposition in the second state was preserved. If this is not the case, she knows Bob has cheated.

exponential reduction in communication complexity. This comes from the fact that information-theoretic SPIR protocols require the exchange of the whole database [8], $O(N)$ qubits, while QPQ requires the exchange of only two database elements, identified by $O(\log N)$ qubits. Quantum private queries also provides an exponential reduction in computational complexity over all classical PIR schemes, whether symmetric or not. In both cryptographic and information-theoretic PIR protocols, the owner(s) of the database(s) must perform $O(N)$ "internal" database calls in response to Alice's query. That is, as part of the protocol, Bob must perform operations that access *every* entry in his database, using some cryptographic primitive such as a public key supplied by Alice. If the PIR protocol requires Bob to perform fewer than N internal database calls, then he obtains information about Alice's query simply by monitoring which database entries were and were not called in the course of executing the protocol. That is, a classical PIR protocol for an unencoded database necessarily has database computational complexity $O(N)$ per query. In contrast, QPQ require only two internal database calls per use, each using only $O(\log N)$ time steps [11].

Quantum private queries achieve two competing goals: Bob can provide the service of private searching without having to give up his database, and Alice can test his honesty without having to trust him. It is designed only to protect Alice's privacy and Bob's information: its goals are to prevent him from reading her queries without risking capture, and to prevent her from obtaining more than a few (one or two) answers for each database query. We then suppose that Bob has no interest in providing false answers (e.g., Alice can easily check them). The basic idea is simple: Bob, as a sign of his discretion, returns not only the answer to Alice's query, but the original query itself, retaining no copy. Alice, in addition to performing normal queries, can perform also quantum superpositions of different queries. This means that in addition to being able to request the jth or the kth records in the database, she can also request both records in a quantum superposition. To find out whether Bob is trying to discover her queries, she just has to send proper superpositions of queries and check Bob's answer to see whether the superposition has been altered. In this case, she can be confident that Bob has been cheating. The user security rests on Bob's impossibility of discovering the generic quantum state of Alice's query. Two basic elements of quantum theory enforce this: the no-cloning theorem [12] which forbids the discovery of the state starting from a single copy of it [13], and the inability fully to characterize a composite system using only local operations. The database security of QPQ is ensured by the finite number of signals Bob is sending back to Alice. As we will see these can be as low as two. This automatically implies that in the QPQ a dishonest Alice will be able to recover at most two items from the database to be compared with the $O(\log N)$ bits of information a dishonest user will be able to acquire in the quantum SPIR protocols [7].

The rest of this Letter is devoted to making the previous ideas rigorous and to providing the details of the protocols. We start by describing the quantum communication protocol that Alice and Bob must follow, and give a security analysis. We then conclude with a discussion on how Bob can interrogate his database preserving Alice's superposed queries.

To submit her query on the jth record of Bob's database, Alice uses an n qubit memory register Q. It allows her to interrogate a database of up to $N = 2^n$ elements. To test whether Bob is cheating and is trying to find out what her query is, she needs to submit a superposition of queries. So she prepares two copies of the register Q, one is initialized as $|j\rangle_Q$, the other as $(|j\rangle_Q + |0\rangle_Q)/\sqrt{2}$ (we suppose that the 0th record in Bob's database contains a fixed reference value known to her). She then *randomly* chooses one of these two registers and sends it to Bob; see Fig. 1. He interrogates his database using it as an index register employing the qRAM algorithm described below [see Eq. (4)]. It returns a second register R which contains the answer to the query, and which may be entangled with the register Q if the latter was in the superposition state (without loss of generality we can assume R to be a single qubit). Bob sends back the Q and R registers to Alice. She then sends him her second Q register, which, again, is employed by Bob to interrogate his database and sent back to Alice together with a new R register containing the answer to her second query. It is important to stress that Bob never knows if the register he receives from Alice is the one containing the quantum superposition or the other one: this means he does not know which measurement could extract information on j without disturbing the register. The number of exchanged qubits is $2(n + 1) = 2(\log N + 1)$ (of these only 2 contain information on the database). We see that, in attempting to obtain information about Alice's state, Bob must try to distinguish between two possible states that have overlap $1/\sqrt{2}$. That is, Bob's position is isomorphic to that of Eve in conventional quantum cryptography, and any attempt on his part to gain information can be detected by Alice: the trade-off between the information that Bob can obtain and his probability of being detected by Alice are essentially the same as in quantum cryptography (see, e.g., [14]) as we now show.

After the double exchange with Bob, Alice is in possession of the two states $|\psi_1\rangle = |j\rangle_Q|A_j\rangle_R$ and

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|j\rangle_Q|A_j\rangle_R + |0\rangle_Q|A_0\rangle_R), \tag{1}$$

where A_m is the content of the mth record in the database (we can suppose that $A_0 = 0$). She can recover the value of A_j by measuring $|\psi_1\rangle$. This value answers her query, and can be used to construct a measurement to test whether the second state is really of the form $|\psi_2\rangle$. (Her measurement is

a two-valued POVM, whose first element is the projector on $|\psi_2\rangle$. If she obtains the result connected to the second element, she is certain Bob has cheated.) We will show that if Bob is acquiring information on j, he will be perturbing the superposition state $|\psi_2\rangle$ and Alice has a nonzero probability of finding it out. We stress that this probability is not unit (i.e., Bob can avoid detection if he is lucky), but no matter what he does, this probability is different from zero and Bob will get caught cheating sooner or later. The only assumption (it may be dropped by complicating the protocol slightly) is that the value A_j is uniquely determined by j; i.e., there cannot be two different answers to one query.

The simple protocol described here can be easily modified to increase its performance. First of all, in place of the fixed superposition $(|j\rangle_Q + |0\rangle_Q)/\sqrt{2}$, we can allow Alice to employ any arbitrary superposition $\alpha|j\rangle + \beta|0\rangle$ with complex amplitudes α and β unknown to Bob. In this way Bob's ability of masking his actions is greatly reduced. More generally, instead of creating a superposition with the reference query $|0\rangle_Q$, she could superimpose two (or more) different queries. In this case, in addition to the query j which she is interested in, she randomly chooses another query (say the kth). Now she prepares three n-qubits registers in the state $|j\rangle$, $|k\rangle$, and $(|j\rangle + |k\rangle)/\sqrt{2}$. As in the case discussed previously, she sends the registers to Bob in random order and one-by-one (i.e., she waits for Bob's reply before submitting the next). At the end of their exchange, if Bob has not cheated, Alice is in possession of three states: i.e., $|j\rangle|A_j\rangle$, $|k\rangle|A_k\rangle$, and $(|j\rangle|A_j\rangle + |k\rangle|A_k\rangle)/\sqrt{2}$. She starts by measuring the first two, in order to find out the values of A_j and A_k: the former is the answer she was looking for, the latter will be used to prepare a measurement (see above) to test the third state to check the superposition. If the test fails, she can conclude that Bob has cheated. Notice that, in contrast to the classical strategies where she hides her query among randomly chosen ones, the security of the QPQ does not rest on the classical randomness of the queries. However, this randomness is a useful resource also for QPQ: Alice can increase her probability of catching a cheating Bob by choosing a high number of random queries in her superposition.

The user security of the protocol rests on two key features, namely, the fact that Alice is sending her queries in random order, and the fact that she is sending them one by one. The first feature prevents Bob from knowing which kind of query (superposed or plain) he is receiving at each time: otherwise he would just let the superposed queries through, and measure the plain ones, finding out j and evading detection. The second feature prevents Bob from employing joint measurements on the queries, which would allow him to find out the value of j: the subspaces spanned by the joint states of Alice's queries are orthogonal for different choices of j.

To discuss the user security of the protocol it is worth starting from a simple ''intercept-resend'' attack strategy. Suppose for instance that Bob performs projective measurements on both of Alice's queries. By doing so he will always recover the value of j. Moreover, with probability $1/2$, one of his two measurement results will return 0 in correspondence to Alice's superposed query. In this case, Bob's attempt at cheating is successful, as he can correctly reprepare both of Alice's queries. However, with probability $1/2$, Bob gets j from both measurements, and it will impossible for him to determine which was the order of Alice's queries. In this case, no strategy of his has more than $1/2$ probability of passing Alice's test. In fact, this is the probability that a state of the form $|j\rangle_Q|A_j\rangle_R$ passes the test of being of the form $(|j\rangle_Q|A_j\rangle_R + |0\rangle_Q|0\rangle_R)/\sqrt{2}$. If Bob uses this cheating strategy, Alice can find it out with probability $1/4$ (this number can be easily increased using the modified QPQ protocols discussed above).

What if Bob employs a more sophisticated cheating strategy? Bob is presented randomly with one among two possible scenarios (A or B) depending on which state Alice sends first. These scenarios refer to the following joint states of her query $|S_A\rangle = |j\rangle_{Q_1}(|j\rangle_{Q_2} + |r\rangle_{Q_2})/\sqrt{2}$ and $|S_B\rangle = (|j\rangle_{Q_1} + |r\rangle_{Q_1})|j\rangle_{Q_2}/\sqrt{2}$, where Q_1 and Q_2 are her first and second query. The failure of the above cheating strategy stems from Bob's impossibility to determine which scenario Alice is using. This is a common problem to all cheating strategies: it is related to the nonorthogonality of the states $|S_A\rangle$ and $|S_B\rangle$, and to the limit posed by the timing of the protocol (to gain access to Q_2, Bob must first respond to Q_1). Working along these lines, one can show that Alice has a nonzero probability of discovering that Bob is cheating, whatever sophisticated methods he employs. More precisely, following a derivation which is similar to that performed in Ref. [14], it can be shown that his impossibility of performing joint measurements on Q_1 and Q_2 places a bound on the information Bob obtains on j: Alice can enforce the privacy of her queries by requiring that Bob is never caught cheating. Here we just sketch the main idea of the security proof, providing the details elsewhere [15].

Any action by Bob in response to Alice's two queries can be described in terms of two unitary transformations U_1 and U_2. The transformation U_1 acts on the registers Q_1, R_1 and on an ancillary system B which is under Bob's control (it also includes his database). The transformation U_2 acts on Q_2, R_2 and B. If Bob is not cheating, U_1 and U_2 are instances of the qRAM algorithm of Eq. (4) below: they coherently copy the information from the database to the R registers leaving the ancilla B in its initial state. If instead Bob is cheating, at the end of the communication the system B will be correlated with the rest. In this case Alice's final state is the mixture

$$\rho_\ell(j) \equiv \mathrm{Tr}_B[U_2U_1|\Psi_\ell(j)\rangle\langle\Psi_\ell(j)|U_1^\dagger U_2^\dagger], \qquad (2)$$

where the label $\ell = A, B$ refers to the scenario used by Alice to submit her query j, and where $|\Psi_\ell(j)\rangle \equiv |S_\ell\rangle_{Q_1Q_2}|0\rangle_{RB}$ is the corresponding input state ($|0\rangle_{RB}$ being the initial state of the registers $R_{1,2}$ and of the ancilla B).

The probability $1 - P_\ell(j)$ that the state $\rho_\ell(j)$ supplied by Bob will pass Alice's test can be easily computed by considering its overlap with the states corresponding to the answer that a noncheating Bob would provide. On Bob's side, the information I_B that he retains on the query is stored in the final state of the ancilla B, i.e.,

$$\sigma_\ell(j) \equiv \mathrm{Tr}_{Q_1 Q_2 R_1 R_2}[U_2 U_1 |\Psi_\ell(j)\rangle\langle\Psi_\ell(j)| U_1^\dagger U_2^\dagger]. \quad (3)$$

An information-disturbance trade-off [16] can be obtained by noticing that if $1 - P_\ell(j) \simeq 1$, then $\sigma_\ell(j)$ must be independent from j. Specifically, requiring $P_\ell(j) \leq \epsilon$ for all ℓ and j, one can show that $1 - F(\sigma_\ell(j), \sigma_*) \leq O(\epsilon^{1/4})$, where σ_* is a fixed state and F the fidelity [17]. Therefore, in the limit of $P_\ell(j) \to 0$ (i.e., Bob passes the test with high probability), the states he retains are independent from the label j that identifies Alice's query. This can also transformed into an upper bound on the mutual information I_B evaluating the Holevo information [18] associated to the ensemble $\{p_j, \sigma(j)\}$ where $p_j = 1/N$ is the probability that Alice will send the jth query, and where $\sigma(j) = [\sigma_A(j) + \sigma_B(j)]/2$ is the final state of B (from his point of view), since Alice randomly chooses among the scenarios A and B with probability $1/2$. By doing so it can be shown [15] that $I_B \leq O(\epsilon^{1/4} \log N)$: his information on her query is upper bounded by the probability of getting caught.

The above analysis shows that the chance of Bob finding the query without being caught is bounded away from one by a quantity which depends on the information he obtains on the query. Once Alice has determined the loss of fidelity due to Bob, she can protect her privacy to any desired degree by the simple expedient of hiding her actual query amongst a group of security devoted queries. In this respect, the guarantee of security of QPQ is closely analogous to the one of quantum cryptography [9]. There, the communicating parties sacrifice some of the transmitted qubits to characterize Eve's intervention in the channel. Once they have ascertained the loss of fidelity due to Eve, they perform privacy amplification to guarantee that their secret key is indeed secret. Analogously, in QPQ Alice may sacrifice some of her queries to determine the loss of fidelity due to Bob's possible cheating. She then dilutes her actual query amongst a set of security devoted queries to obtain her desired degree of privacy. In both quantum cryptography and in QPQ, privacy is obtained by the users probing the system to ascertain the degree of possible cheating, and then taking measures to guarantee their privacy in the face of such cheating. QPQ possesses essentially the same degree of privacy as quantum key distribution.

In closing we comment briefly on the qRAM algorithm [11,16] that Bob uses to interrogate his database. Starting from an array of spatially separated memories, the aim of the qRAM protocol is to read a location (or quantum superposition of locations) specified by an index register Q, and return the contents in a second register R according to the transformation

$$\sum_j \alpha_j |j\rangle_Q \to \sum_j \alpha_j |j\rangle_Q |A_j\rangle_R. \quad (4)$$

Conventional designs [16] require $O(2^n)$ quantum logic operations to perform a qRAM call. However, we have recently exhibited qRAM architectures in which this number can be reduced to $O(n)$ [11].

We acknowledge useful feedback from Micali and Sudan. V. G. acknowledges support from the Quantum Information Research program of Centro di Ricerca Ennio De Giorgi of SNS. S. L. acknowledges fruitful discussions with S. Brin and L. Page.

[1] Y. Gertner *et al.*, J. Comput. Syst. Sci. **60**, 592 (2000).
[2] B. Chor *et al.*, J. ACM **45**, 965 (1998); C. Cachin, S. Micali, and M. Stadler, in *Advances in Cryptology-EUROCRYPT99* (Springer-Verlag, Berlin, 1999); C. Gentry and Z. Ramzan, in *Proc. 32nd ICALP* (Springer-Verlag, Berlin,2005), p. 803; S. Yekhanin, Technical Report No. ECCC TR06-127, 2006.
[3] E. Kushilevitz and R. Ostrovsky, in *Proc. 38th IEEE Symposium FOCS97* (1997), p. 364.
[4] S. Wiesner, *ACM SIGACT News*, (Livermore, 1983), Vol. 15, p. 78; M. O. Rabin, Harvard Aiken Computational Laboratory, Technical Report No. TR-81, 1981; A. Jakoby, M. Liskiewicz, and A. Madry, arXiv:quant-ph/0605150v1.
[5] A. Ambainis, in *Proceedings of the 24th ICALP*, Lect. Notes Comput. Sci. (Springer-Verlag, Berlin, 1997), Vol. 1256, p. 401.
[6] I. Kerenidis and R. de Wolf, arXiv:quant-ph/0208062.
[7] I. Kerenidis and R. de Wolf, arXiv:quant-ph/0307076.
[8] Slightly better performances are obtained for multiple nonmutually communicating servers [2]. Moreover sublinear communication complexity is possible under some computational complexity assumption, e.g. [3].
[9] C. H. Bennett and G. Brassard, *Proc. IEEE Int. Conf. on Computers, Systems and Signal Processing, Bangalore, India* (IEEE, New York, 1984), p. 175.
[10] L. Hardy and A. Kent, Phys. Rev. Lett. **92**, 157901 (2004).
[11] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **100**, 160501 (2008).
[12] W. K. Wootters and W. H. Zurek, Nature (London) **299**, 802 (1982).
[13] G. M. D'Ariano and H. P. Yuen, Phys. Rev. Lett. **76**, 2832 (1996).
[14] M. Christandl and A. Winter, IEEE Trans. Inf. Theory **51**, 3159 (2005).
[15] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum Private Queries: Security Analysis (to be published).
[16] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, England, 2000), p. 586.
[17] A. Uhlmann, Rep. Math. Phys. **9**, 273 (1976).
[18] A. S. Holevo, *Probabilistic and Statistical Aspects of Quantum Theory* (North Holland, Amsterdam, 1982).