PHYSICAL REVIEW A **83**, 022301 (2011)

# Practical private database queries based on a quantum-key-distribution protocol

Markus Jakobi,[1,2] Christoph Simon,[1,3] Nicolas Gisin,[1] Jean-Daniel Bancal,[1] Cyril Branciard,[1] Nino Walenta,[1] and Hugo Zbinden[1]

[1]*Group of Applied Physics, University of Geneva, CH-1211 Geneva 4, Switzerland*

[2]*Humboldt-Universität zu Berlin, D-10117 Berlin, Germany*

[3]*Institute for Quantum Information Science and Department of Physics and Astronomy, University of Calgary, Calgary T2N 1N4, Alberta, Canada*

(Received 23 February 2010; published 2 February 2011)

Private queries allow a user, Alice, to learn an element of a database held by a provider, Bob, without revealing which element she is interested in, while limiting her information about the other elements. We propose to implement private queries based on a quantum-key-distribution protocol, with changes only in the classical postprocessing of the key. This approach makes our scheme both easy to implement and loss tolerant. While unconditionally secure private queries are known to be impossible, we argue that an interesting degree of security can be achieved by relying on fundamental physical principles instead of unverifiable security assumptions in order to protect both the user and the database. We think that the scope exists for such practical private queries to become another remarkable application of quantum information in the footsteps of quantum key distribution.

DOI: 10.1103/PhysRevA.83.022301 PACS number(s): 03.67.Dd, 03.67.Hk

## I. INTRODUCTION

As telecommunication gains steadily in importance, questions of security and privacy naturally arise. Indeed, private data are stored on a grand scale and have become a precious commodity. Unfortunately, as a matter of principle, classical information theory is not able to secure privacy in telecommunication against an unlimited adversary. It was hence found all the more extraordinary that quantum key distribution (QKD) allows such "unconditionally" private communication, provided that the two parties trust each other. However, the more general case of communication between distrustful parties, who wish to protect not only their *common* privacy against eavesdropping but also their *individual* privacy against each other, is maybe of even greater interest.

Private queries are an important problem of this type. Imagine that a user, Alice, wants to know an element of a database held by a database provider, Bob, but does not want him to know which element she is interested in. Bob in turn wants to limit the amount of information that she can gain about the database. In particular, he does not want to just hand over the whole database, which would trivially allow Alice to learn her bit of interest without giving any information on her choice away. It is not hard to imagine scenarios (e.g., in the financial world) where the capability of implementing such private queries would be useful. The information stored in the database may be both valuable and sensitive, such that Bob would like to sell it piece by piece, whereas the mere fact of being interested in an element of the database might already reveal something important about Alice (e.g., that she is thinking about buying a certain company). Of course if there were a cheap way of realizing the task, it would also be useful for protecting privacy in online bargaining and web search, for example, as well as to construct other interesting cryptographic primitives from it [1].

The described task is also known as symmetrically private information retrieval and as 1 out of $N$ oblivious transfer [2]. It has attracted much attention both in computer science [3,4] and in quantum information. Classically, the problem seems like a logical contradiction. How could a database provider answer a question, which he is not supposed to know, without giving any additional information? One might hope that quantum mechanics could solve this dilemma. Several quantum protocols were proposed (see, for example, Refs. [5,6]), none of which were found to offer complete protection for both sides. Indeed, it was subsequently proven in Ref. [7] that the described task cannot be implemented ideally, not even using quantum physics. The essential assumption in the impossibility proof is that the protocol is *perfectly concealing*, i.e., that Bob has no information whatsoever about which database element Alice has retrieved. Rephrased at the quantum level this is understood as the condition that the density matrix of Bob's subsystem must be completely independent of Alice's choice. Reference [7] shows that under this condition Alice can always implement an attack based on the Schmidt decomposition which allows her to read the entire database. This argument is closely linked to the well-known impossibility proofs for quantum bit commitment [8,9].

Recently, Giovannetti, Lloyd, and Maccone [10] pointed out that very interesting degrees of privacy are achievable for protocols that are not perfectly concealing, because of the possibility to catch dishonest parties due to the errors they introduce (see also Refs. [11,13,14]. In the protocol of Ref. [10] Alice encodes her question in a quantum state, which she sends to Bob. She also sends a decoy state, which gives her a chance to detect if Bob is cheating. The security relies on the impossibility to perfectly discriminate the nonorthogonal question and decoy states and on the changes Bob's measurement will introduce as a consequence. Unfortunately the protocol is very vulnerable in realistic situations where there are significant transmission losses, such that Alice has to send the same question multiple times. If some of the losses are in fact due to Bob tapping the line, then he can learn Alice's question without being detected.

 

## II. CLAIM

In this paper we present a new approach to the private query problem. Our protocol is explicitly not perfectly concealing in the above sense, so that the impossibility proof of Ref. [7] does not apply. We show that the following statements hold for our protocol.

(1) *Database security* is very good. Even for relevant multiqubit joint measurements Alice's accessible information is restricted to a well-defined small percentage of the database elements. The concrete limits for different attacks are shown in the security discussion. Moreover the additional elements Alice learns are randomly distributed over the database and therefore of little use to her. In general, database security is ensured by the impossibility of perfectly distinguishing nonorthogonal quantum states.

(2) *User privacy* is also very high. We study several natural attacks and derive a simple limit on the information Bob can obtain. In general, we show that the no-signaling principle implies that every malicious action of Bob's will introduce errors and can hence be detected by Alice—systematic cheating is impossible.

The protocol relies on QKD with changes only in the post-processing and can hence profit from many of the advantages of this well understood and commercially available technology. In comparison to Ref. [10] it offers the advantage of practical feasibility, in particular, loss tolerance and scalability to large databases.

Note that the incorporation of security assumptions such as the bounded storage model [15] could make the protocol completely secure, under the condition that those assumptions are fulfilled. However, even in the absence of such assumptions, our protocol's basic security is guaranteed by fundamental physical principles, namely, the impossibility of perfectly discriminating nonorthogonal quantum states and the impossibility of superluminal communication.

It should be underlined that we do not propose an ideal cryptographic primitive, which would furthermore allow one to construct other ideal cryptographic primitives such as user identification, bit commitment, and coin flipping [1], but rather a new practical and potentially very useful application of quantum communication.

Our protocol is similar to the proposal of Bennett *et al.* [5], which can be interpreted to rely on the Bennett-Brassard 1984 (BB84) QKD [16]. It is well known that the proposal of Ref. [5] is susceptible to a quantum memory attack by the user, which corrupts database security entirely. The crucial point is that Ref. [5] is perfectly concealing, hence Lo's impossibility proof [7] implies that the user can learn the entire database—in this case with the help of a quantum memory. We show that this type of attack can be forestalled by using the Scarani-Acin-Ribordy-Gisin 2004 (SARG04) QKD scheme [17] instead of the BB84 protocol. Then user privacy is slightly weakened, but the quantum memory attack is no longer feasible. Moreover the errors a cheating provider introduces largely guarantee user privacy.

## III. APPROACH

In order to better understand our approach it is very useful to compare it to QKD. In general QKD consists of a first phase, where a large number of quantum states are prepared, exchanged, and measured, and then a second phase, where Alice and Bob extract a key from the quantum communication part with the help of an *a priori* chosen coding and interpretation process. The key is then known to both Alice and Bob entirely and can be used to encrypt the actual message, which is sent via a classical channel. The quantum states and the postprocessing procedure are chosen such that the key cannot be eavesdropped on without introducing errors, thus protecting Alice's and Bob's common privacy.

The basic idea of our protocol is to use QKD in combination with adequate postprocessing to generate an $N$-bit string $K^f$ that will serve as an *oblivious key* [18] for a database of $N$ bits. For this purpose, $K^f$ must be distributed in such a way that (1) Bob knows the key entirely, (2) Alice knows only a few bits of $K^f$—ideally exactly one (database security), and (3) Bob does not know which bits are known to Alice (user privacy). In order to use $K^f$ to encrypt the database, Bob adds key and database bitwise with a relative shift chosen by Alice and sends her the encrypted database. The relative shift is needed in order to ensure that Alice's bit of interest is encoded with an element of $K^f$ she knows, so that she can decipher the bit and thus receive the answer to her private query.

Within our approach, the case of Alice knowing exactly one bit cannot be realized deterministically. So in general Alice will know a few bits of $K^f$, which means that database privacy is good but not perfect. As the number of Alice's elements is Poisson distributed, there is also a small probability of Alice having no bit in the end. The protocol then needs to be repeated. This can be done without loss of privacy for either party: The created string $K^f$ does not contain any information on the database, so database security is not touched, and likewise the shift (which maps Alice's known key element onto the database element she needs) is only communicated once a correct key has been established. Of course, Alice could claim to have obtained no element of $K^f$ with the hope of having more elements after a repetition. However, this strategy can be made ineffective by choosing the parameters of the protocol such as to make the case of Alice having no element very unlikely (cf. also Sec. V).

As already mentioned, the generation of $K^f$ can be based on QKD techniques. Consider for instance four-state BB84-type QKD. After Bob has sent the states (without further information), Alice, choosing measurement bases at random, will measure half of the bits she receives in the correct basis—without yet knowing for which ones her choice was correct. When Bob subsequently announces the bases, we have the situation that (I) Bob knows the entire "raw key," (II) Alice knows half of the bits, and (III) Bob cannot know which ones Alice has measured correctly. Alice's limited information on the raw key can now be further diluted by adequate processing in order to generate the oblivious key $K^f$, and this is indeed the way Ref. [5] essentially works. However, if Alice has a quantum memory this protocol is no longer secure. She can then store the received states and postpone all measurements until after Bob's announcement. By doing so, she can learn $K^f$ entirely—there is hence actually no database security at all.

Fortunately this attack can be largely forestalled rather easily if one uses a SARG-QKD scheme instead of the BB84

protocol. The SARG04 protocol uses the same states as the four-state-BB84 protocol. The main difference lies in the attribution of bit values to the quantum states. Whereas in the BB84 protocol one state from each of the two bases codes for 0 and the other one for 1, in the SARG04 protocol it is the basis itself that codes for the bit value. That is, if Bob sends a state in the "up-down" basis $\updownarrow$ this signifies a 0, and a state from the "left-right" basis $\leftrightarrow$ means 1. During the postprocessing Bob does not announce which basis he has used for each qubit. Instead Bob announces the state he has sent plus one state from the other basis (in random order). Alice is thus faced with a state discrimination problem that cannot be solved perfectly, i.e., unambiguously and deterministically at the same time. This slight change has profound implications for SARG04 QKD [19]. Here we show that it is also very useful for implementing private queries. A simple protocol based on this approach consists of the following steps.

## IV. PROTOCOL

(1) Bob sends a long random sequence of qubits (e.g., photons) in states $|\uparrow\rangle$, $|\rightarrow\rangle$, $|\downarrow\rangle$, and $|\leftarrow\rangle$. States $|\uparrow\rangle$ and $|\downarrow\rangle$ code for 0, and states $|\leftarrow\rangle$ and $|\rightarrow\rangle$ correspond to bit value 1. For instance, to send a bit 1 Bob can prepare a qubit in the state $|\rightarrow\rangle$.

(2) Alice measures each state in the $\updownarrow$ or the $\leftrightarrow$ basis at random. This alone does not allow her to infer the sent bit value.

(3) Alice announces in which instances she has successfully detected the qubit; lost or not detected photons are disregarded. The possibility to discard bits does not allow Alice to cheat, because after step 2 she still has no information whatsoever on the sent bit values (cf. step 5). As a consequence, the protocol is completely loss independent.

(4) For each qubit that Alice has successfully measured, Bob announces a pair of two states: the one that has actually been sent and one from the other basis, so $\{|\uparrow\rangle,|\rightarrow\rangle\}$, $\{|\rightarrow\rangle,|\downarrow\rangle\}$, $\{|\downarrow\rangle,|\leftarrow\rangle\}$, or $\{|\leftarrow\rangle,|\uparrow\rangle\}$. If $|\rightarrow\rangle$ has been sent, Bob could announce, for instance, $\{|\uparrow\rangle,|\rightarrow\rangle\}$. This is exactly as in the SARG04 QKD protocol [17].

(5) Alice interprets her measurement results of step 4. Depending on which basis she has chosen and which result she has obtained she will be able to decipher the sent bit value or not. For instance, if $|\rightarrow\rangle$ has been sent and $\{|\uparrow\rangle,|\rightarrow\rangle\}$ has been announced, Alice can rule out $|\uparrow\rangle$ only if she has measured in the $\updownarrow$ basis and obtained the result $|\downarrow\rangle$. She can then conclude that the state was $|\rightarrow\rangle$ and the bit value is 1. Direct measurement as under step 2 will yield 1/4 of conclusive results and 3/4 of inconclusive ones. Both conclusive and inconclusive results are kept. Alice and Bob now share a string which is known entirely to Bob and in a quarter to Alice.

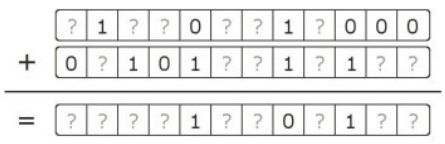


FIG. 1. How to reduce Alice's information: her information on a sum string is lower than that on the initial strings. Question marks symbolize bits whose value is unknown to Alice.

(6) The created string must be of length $k \times N$ (with $k$ being a security parameter). It is cut into $k$ substrings of length $N$. These strings are added bitwise in order to reduce Alice's information on the key to roughly one bit (cf. Fig. 1).

(7) If Alice is left with no known bit after step 6, the protocol has to be restarted. The probability for this to occur can be kept small. See also the discussion in the previous and following sections.

(8) If $K^f$ has been established correctly, Alice will know at least one element of it. Suppose she knows the $j$th bit $K_j^f$ and wants the $i$th bit of the database $X_i$. She then announces the number $s = j - i$ in order to allow Bob to encode the database by bitwise adding $K^f$, shifted by $s$. So Bob announces $N$ bits $C_n = X_n \oplus K_{n+s}^f$ where Alice can read $C_i = X_i \oplus K_j^f$ and thus obtain $X_i$. The shift will hence make sure that Alice's bit of interest is coded with a key element she knows so that the private query can be completed.

## V. DISCUSSION

Steps 1 to 5 of the above protocol are completely identical to SARG04 QKD with the only difference that every bit is kept, regardless if it is conclusive or not for Alice. SARG04 QKD was initially conceived to make QKD more resistant to photon number splitting attacks when weak pulses are used instead of single photons for the sake of practical feasibility. In our case the use of SARG04 QKD not only provides us with the benefits of loss tolerance, technological practicability, and conceptual closeness to well-understood QKD, but it also prevents the quantum memory attack that destroyed the security of the protocol of Ref. [5]. Even using a quantum memory Alice is always confronted with the problem of discriminating two nonorthogonal quantum states and will hence always have incomplete knowledge on the raw key. This lack of information is subsequently further amplified by step 6.

Note that following the "honest" way of measuring and interpreting her results Alice will also gain probabilistic information on nonconclusive bits. If Alice obtains no result it is with probability 2/3 because she has chosen the same basis for measurement as Bob has chosen for state preparation (which will never yield a conclusive result). Considering the example of step 5, Alice can obtain the result $|\rightarrow\rangle$ when measuring in $\leftrightarrow$ both if Bob sent $|\rightarrow\rangle$ (then with probability 1) and if Bob sent $|\uparrow\rangle$ (then with probability 1/2 only). So, although $|\rightarrow\rangle$ is not a conclusive result, Alice can infer that the sent state was $|\rightarrow\rangle$ (bit 1) with probability 2/3 and $|\uparrow\rangle$ (bit 0) with probability 1/3. This additional information can be diluted to a negligible level by the postprocessing of step 6.

After creation of the raw key of $k \times N$ bits, the string is divided into $k$ substrings of length $N$. Following the protocol, after adding the substrings, Alice will on average know $\bar{n} = N(\frac{1}{4})^k$ bits, where the number $n$ follows approximately a Poisson distribution. On the other hand, the probability $P_0$ that she does not know any bits at all and that the protocol must be restarted is $P_0 = [1 - (\frac{1}{4})^k]^N \approx e^{-\bar{n}}$. For large $N$, which is the most interesting case in practice, it is therefore possible to ensure both $\bar{n} \ll N$ and small $P_0$ by choosing an appropriate value of $k$. For instance, for a database of $N = 50\,000$ elements

TABLE I. Example of possible choices of $k$ for different database sizes $N$. We show the failure probability $P_0$ and the expected number of elements $\bar{n}$ an honest Alice will obtain.

| | $N$ | | | | | |
|---|---|---|---|---|---|---|
| | $10^3$ | $5\times10^3$ | $10^4$ | $5\times10^4$ | $10^5$ | $10^6$ |
| $k$ | 4 | 5 | 6 | 7 | 7 | 9 |
| $P_0$ | 0.020 | 0.008 | 0.087 | 0.047 | 0.002 | 0.022 |
| $\bar{n}$ | 3.91 | 4.88 | 2.44 | 3.05 | 6.10 | 3.81 |

$k = 7$ is a choice providing Alice with $\bar{n} \approx 3$ elements of the final key on average whereas the probability of failure is only about 5% (see also Table I). The case of many repetitions (which might allow Alice to wait until she obtains a large value of $n$ by chance) is hence very unlikely. This is important for the protocol's security. Since the states sent by Bob do not contain any information about the database, and since Alice only chooses and communicates the shift $s$ to Bob once she knows at least one bit of the final key, a few repetitions will not compromise anybody's security. Note that even if Alice knows $n > 1$ bits of the oblivious key, she has to pick a single shift $s$, which means that in general she can only learn one *chosen* element of the database, since the other $n-1$ bits known to her will be at random positions in the key and thus in the database.

However, the fact that Alice normally obtains additional, less interesting bits should not be seen only as a drawback of the protocol, as it also offers an interesting possibility to enhance her security: Alice can buy the extra bits in question publicly (as opposed to privately), in order to compare them with Bob's answers. As explained in detail in the security section, a cheating Bob will always lose knowledge on $K^f$. The errors he thus introduces will then be detectable for Alice. This way what seems to be a flaw in the protocol can be used to strengthen user privacy.

## VI. SECURITY

We now turn to the question of which degree of privacy our protocol offers precisely. We study the most evident attacks and clarify the way in which two fundamental physical principles provide the basis for the protocol's security. While basic attacks are studied and the essential intuition is given, a complete security analysis remains work for the future.

### A. Database security

Let us first discuss database security. In general one must assume that Alice disposes of a quantum memory and is hence not forced to measure directly as in step 2. Instead she can keep the photon and, once Bob has announced the state pair, apply the optimal *unambiguous state discrimination* (USD) measurement [20,21] that will correctly tell her which of the two announced states has actually been sent. The success probability of the USD measurement is, for the case of two equally likely states, bounded by $1 - F(\rho_0, \rho_1)$, where $F(\rho_0, \rho_1)$ is the fidelity between the two quantum states one seeks to discriminate. Here, Alice's measurement will hence only work with a success probability of $1 - |\langle\uparrow|\rightarrow\rangle| = 1 - 1/\sqrt{2} \approx 0.29$, only slightly more than the 0.25 of the direct measurement. In the above example with $N = 50\,000$ and $k = 7$ this will provide her with $\bar{n} = 9.3$ elements on average—only a small gain compared to $\bar{n} = 3$ and very little in relation to $N = 50\,000$ for such a complex attack. So even using a quantum memory, individual measurements will not substantially increase her information on $K^f$. The reason for this is precisely the fact that our protocol is based on SARG04 coding rather than on BB84 coding.

A more general attack is to store the received photons in a quantum memory and to postpone all measurements until the very end of the protocol after step 6, so that she knows which $k$ qubits contribute to an element of the final key. The individual bit values of the raw key are actually of no interest to her. So, instead of performing the optimal individual measurement on each of the $k$ qubits constituting an element of $K^f$, Alice should perform a joint measurement. An example for this is Helstrom's minimal error-probability measurement, i.e., the measurement that distinguishes two quantum states with the highest information gain [22,23]. In the case of two equally likely quantum states $\rho_0$ and $\rho_1$, the probability to guess the state at hand correctly is bounded by $P_{\text{guess}} = \frac{1}{2} + \frac{1}{2}D(\rho_0, \rho_1)$, where $D(\rho_0, \rho_1)$ is the trace distance. For a joint Helstrom measurement on a bit of $K^f$ one finds this probability to scale with the number $k$ of added qubits as $P_{\text{guess}} = \frac{1}{2} + \frac{1}{2\sqrt{2^k}}$. So the more substrings are added to generate the final key, the harder it is for her to guess the bit value, i.e., the parity of the $k$ qubits. For example, for $k = 7$ Alice will guess a key element correctly with 54.4% instead of 50% for a random guess. Likewise, the success probability of unambiguously discriminating the two $k$-qubit mixed states corresponding to odd and even parity declines rapidly with the number of qubits $k$ (see Fig. 2). In conclusion, it is clear that the *impossibility to perfectly distinguish nonorthogonal quantum states* can effectively protect the database's security and prevent Alice from knowing a substantial part of it, even when she uses perfect storage technology and realizes the theoretically optimal joint measurements. We see that incorporating a SARG04 state discrimination problem as a vital part of the protocol, the Schmidt attack of Lo's impossibility proof can be averted. The price to pay is a protection of the user that is not total. We now turn to the question of user privacy.


FIG. 2. (Color online) The upper bound on the success probability of the joint unambiguous state discrimination (USD) measurement on $k$ qubits declines rapidly with $k$.

### B. User privacy

As we have discussed above, a not perfectly concealing protocol, i.e., a protocol where Bob can gain some information on Alice's choice, is the prerequisite to prevent her from being able to compromise database security entirely [7]. For the given protocol it may not be obvious at first sight how Bob can access information on Alice's choice, in the absence of any classical or quantum communication from her to him. It turns out that he can indeed gather information on a bit's conclusiveness and hence infer if that particular bit is more or less likely to be a key element Alice knows.

The simplest attack for Bob is to send states other than those he announces, for instance, a state $|\nearrow\rangle$ that is exactly intermediate between $|\uparrow\rangle$ and $|\rightarrow\rangle$, while announcing a pair $\{|\uparrow\rangle,|\rightarrow\rangle\}$. Alice's probabilities to measure $|\downarrow\rangle$ or $|\leftarrow\rangle$ are largely reduced. Indeed, she will find a probability of only 14.64% to have such a conclusive result. Likewise sending the state $|\swarrow\rangle$ (orthogonal to $|\nearrow\rangle$) while announcing $\{|\uparrow\rangle,|\rightarrow\rangle\}$ will raise the probability to interpret the result as conclusive to 85.36%. Bob can thus bias the probability of conclusive results for Alice continuously between the above limits. However, every such attack will introduce errors, as Bob cannot predict her outcome with certainty. In the example above, Alice registering $|\downarrow\rangle$ and $|\leftarrow\rangle$, i.e., both bit values, are equally likely events, and Bob's bit error rate will therefore be as high as 50%. This evident example shows that Bob can gain information on the *conclusiveness* of Alice's bits but will then lose information on the *bit values* she has recorded.

The presented attack is closely related to an attack that uses entanglement. Bob prepares a state of two qubits $\frac{1}{\sqrt{2}}\{|\uparrow\rangle_A|R_0\rangle_B + |\rightarrow\rangle_A|R_1\rangle_B\}$, where the first qubit is sent to Alice and the second is kept in Bob's register (with $\langle R_0|R_1\rangle_B = 0$). Bob announces having sent $|\uparrow\rangle$ or $|\rightarrow\rangle$. Once Alice has successfully measured and accepted her qubit, Bob can decide if he wants to measure honestly, i.e., recover the sent bit value, or gain some information on the conclusiveness of Alice's measurement. In order to proceed honestly Bob measures his register in the basis $\{|R_0\rangle,|R_1\rangle\}$, which tells him which of the two announced states has actually been sent [24]. He then knows which bit value Alice will record in case of a conclusive outcome, but has gained no improved estimation of the likelihood for this to happen. In contrast, measuring in the $\{(|R_0\rangle + |R_1\rangle)/\sqrt{2},(|R_0\rangle - |R_1\rangle)/\sqrt{2}\}$ basis provides him with likelihood information on the conclusiveness of a bit, but clearly yields no information at all on the sent bit value.

This second measurement can also be seen from another angle. If Alice has obtained a conclusive result (probability 1/4) Bob's register is in the state

$$\rho_c = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix};$$

if Alice's measurement was nonconclusive (probability 3/4) he has

$$\rho_n = \begin{pmatrix} 1/2 & \sqrt{2}/3 \\ \sqrt{2}/3 & 1/2 \end{pmatrix}.$$

As $\rho_c \neq \rho_n$ the protocol is not perfectly concealing. Using the criteria of Refs. [20,21] one can show that these two density matrices cannot be discriminated unambiguously for the single-qubit case. The best chance to guess the state correctly is 85.36%, as for the previous attack. The second given measurement basis does indeed constitute Helstrom's minimal error probability measurement [22,23] for the conclusiveness of one of Alice's bits. As a matter of fact, one can show that, given an arbitrary mixed qubit state, the likelihood to measure a conclusive result will be confined by the very same bounds (85.36% and 14.64%). No qubit state can yield only conclusive results upon the above measurement, or yield only inconclusive results. This individual attack is therefore optimal, yields information on the bit's conclusiveness, and completely erases the bit value information from Bob's register. This last point means that Bob will not know $K^f$ correctly—a cheating Bob can then be caught when providing wrong answers [13]. In principle these results can be generalized to joint measurements on several qubits; however, these complicated attacks are beyond the scope of this paper. Instead we now clarify the conceptual reason *why* it is impossible for Bob to have both the correct bit value and conclusiveness information.

Let us suppose that Bob can gain information on the conclusiveness of one of Alice's elements of the raw key, either by construction of the sent state or by some measurement performed on his register at the end of the protocol. Let us characterize this information by $p_c$, the probability with which Bob correctly guesses that Alice has a conclusive result. (Remember that this likelihood is physically bounded by $p_c \leqslant 0.8536$ if a single qubit is sent.) Let us also assume that, either by construction of the state or by some second measurement, Bob can also guess the bit value $b$ Alice has recorded (if her measurement was conclusive) and is correct about it with the probability $p_b$. Recalling the way Alice interprets her measurement results in step 5 of the protocol, it is clear that, if Bob correctly guesses that Alice's result was indeed conclusive and correctly guesses which bit value she has obtained, then he also correctly guesses which measurement basis she has used for this qubit in step 2. However, since there is no communication whatsoever from Alice to Bob about her choice of basis, the *no-signaling principle* dictates that his probability to guess her basis correctly has to be equal to 1/2. Otherwise the procedure would allow Alice to send signals to Bob that are faster than the speed of light. This immediately implies the bound

$$p_c \times p_b \leqslant 1/2.$$

The inequality arises because even for inconclusive results Bob has a chance to guess Alice's basis correctly. This simple upper bound illustrates the crucial point: Whenever Bob tries to alter the conclusiveness probability of certain bits in order to better judge which bits of $K^f$ are (un)known to Alice, he will necessarily lose information on the bit value Alice records in order to comply with the no-signaling principle. This introduces errors in $K^f$ and hence also in the encrypted database; i.e., he will run the risk of giving wrong answers.

This shows that our protocol is cheat sensitive in the spirit of Refs. [10,13]. In our scenario, Bob sells his database bit by bit. Systematic cheating and hence giving wrong answers will ruin his reputation as a database provider. As we already mentioned

above, one can now even make use of the fact that Alice normally obtains additional database elements. If she buys those elements from Bob in a regular, nonprivate way, she can use them to check Bob's honesty [25]. By doing so, Alice has a powerful prompt privacy check at hand. One can thus turn what seems to be a flaw into an advantage, in order to make full use of the privacy, which, as we have seen, is guaranteed by the *impossibility of superluminal communication* in quantum physics.

## VII. OUTLOOK AND CONCLUSIONS

The above discussion has shown that practically very interesting levels of privacy in database queries can be achieved for both sides. The security of the presented protocol relies on fundamental physical principles (the impossibility to deterministically discriminate nonorthogonal states and the impossibility of superluminal communication), rather than on assumptions on quantum storage limitations [15], mathematical complexity [3], or noncommunication between servers in multiserver protocols [4].

We have already emphasized that the protocol is completely loss resistant. We believe that error correction is possible as well. This requires additional classical two-way communication and still needs to be elaborated in more detail. Moreover, it is clear that the protocol can be implemented with weak coherent pulses as well. The acceptable amount of loss then depends on the mean photon number per pulse, in order to safeguard database security. High mean photon numbers largely facilitate unambiguous state discrimination for Alice, if one assumes that she is in control of the transmission line. Finally, it is possible to improve database security by more sophisticated postprocessing, e.g., by taking a couple of strings created in our probabilistic protocol (with $P_0 \ll 1$) and allowing Alice to combine them, i.e., to freely choose relative shifts to add them bitwise. Simulations show that she will be left with knowing exactly one bit of the final key with overwhelming probability. Both error correction and the described way of achieving tighter database security complicate the security analysis due to the necessary two-way communication.

The proposed protocol can be realized with any existing QKD system that is compatible with the SARG04 protocol. Besides ensuring loss tolerance, this also makes it easy to scale up to large databases. We hope that our proposal will stimulate further work to clarify the open questions. Besides a more in-depth study of its security, these include the optimal classical procedures for oblivious key generation and error correction. We think that there is the potential for private queries to become a genuine application of quantum information technology in the footsteps of QKD.

## ACKNOWLEDGMENTS

We thank G. Brassard, V. Giovannetti, S. Hastings-Simon, U. Herzog, L. Maccone, S. Pironio, C. Schaffner, D. Stucki, S. Wolf, and J. Wullschleger for useful discussions and insightful comments. Financial support by the Swiss NCCR-QP, the European ERC-AG Qore, and the EU project QESSENCE is gratefully acknowledged. C.S. was supported by an NSERC Discovery Grant.

[1] J. Kilian, in *Proceedings of the 20th STOC* (Assoc. Comput. Mach., New York, 1988), p. 20.
[2] M. O. Rabin, Technical Report TR-81, Aiken Computation Lab, Harvard University, 1981.
[3] E. Kushilevitz and R. Ostrovsky, in *Proceedings of the 38th Annual Symposium on Foundations of Computer Science* (IEEE, New York, 1997), p. 364.
[4] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, in P*roceedings of the 36th Annual Symposium on Foundations of Computer Science* (IEEE, New York, 1995), p. 41.
[5] C. H. Bennett, G. Brassard, C. Crépeau, and M.-H. Skubiszewska, in *Lecture Notes in Computer Science*, Vol. 576 (Springer, London, 1992), p. 351.
[6] G. Brassard, C. Crépeau, R. Jozsa, and D. Langlois, in *Proceedings of the 34th Annual Symposium on Foundations of Computer Science* (IEEE, Washington, 1993), p. 362.
[7] H.-K. Lo, Phys. Rev. A **56**, 1154 (1997).
[8] D. Mayers, Phys. Rev. Lett. **78**, 3414 (1997).
[9] H.-K. Lo and H. F. Chau, Phys. Rev. Lett. **78**, 3410 (1997).
[10] V. Giovannetti, S. Lloyd, and L. Maccone, Phys. Rev. Lett. **100**, 230502 (2008).
[11] This implies that one has to accept a certain trade off between user privacy and database security. This has been studied quantitatively for bit commitment [12].
[12] R. W. Spekkens and T. Rudolph, Phys. Rev. A **65**, 012310 (2001).
[13] A. Jakoby, M. Liśkiewicz, and A. Mádry, in *Lectures Notes in Computer Science*, Vol. 5155 (Springer, Berlin, 2008), p. 121.
[14] F. de Martini, V. Giovannetti, S. Lloyd, L. Maccone, E. Nagali, L. Sansoni, and F. Sciarrino, Phys. Rev. A **80**, 010302 (2009).
[15] I. Damgaard, S. Fehr, L. Salvail, and C. Schaffner, in *Proceedings of the 27th Annual International Conference on Advances in Cryptology* (Springer, Berlin, 2007), p. 342.
[16] C. H. Bennett and G. Brassard, in *Proc. IEEE International Conference on Computers, Systems and Signal Processing* (IEEE, New York, 1984), p. 175.
[17] V. Scarani, A. Acín, G. Ribordy, and N. Gisin, Phys. Rev. Lett. **92**, 057901 (2004).
[18] D. Beaver, *Lecture Notes in Computer Science*, Vol. 963 (Springer, London, 1995), p. 97.
[19] C. Branciard, N. Gisin, B. Kraus, and V. Scarani, Phys. Rev. A **72**, 032301 (2005).
[20] P. Raynal, e-print arXiv:quant-ph/0611133.
[21] U. Herzog and J. A. Bergou, Phys. Rev. A **71**, 050301 (2005).
[22] C. A. Fuchs, e-print arXiv:quant-ph/9601020.
[23] C. W. Helstrom, *Quantum Detection and Estimation Theory* (Academic Press, New York, 1976).
[24] This procedure is equivalent to rolling a quantum die in order to randomly decide which state to send.
[25] Once Bob has sent the encrypted database over a classical channel, he must have measured his quantum register. At this point it is hence no longer possible for Bob to decide to measure certain bits honestly and others not. The cheating is then detectable in the bit error rate of the key, i.e., in the wrong answers he gives.