entropy

MDPI

Article

Quantum Private Query Protocol Based on Two Non-Orthogonal States

Yan Chang *, Shibin Zhang, Guihua Han, Zhiwei Sheng, Lili Yan and Jinxin Xiong

College of Information Security Engineering, Chengdu University of Information Technology, Chengdu 610225, China; cuitzsb@cuit.edu.cn (S.Z.); alenhan@cuit.edu.cn (G.H.); shengziwei@cuit.edu.cn (Z.S.); yanlili@cuit.edu.cn (L.Y.); xiongjinxin@yahoo.com (J.X.)

* Correspondence: cyttkl@cuit.edu.cn; Tel.: +86-28-8596-6648

Academic Editor: Gerardo Adesso

Received: 15 December 2015; Accepted: 20 April 2016; Published: 3 May 2016

Abstract: We propose a loss tolerant quantum private query (QPQ) protocol based on two non-orthogonal states and unambiguous state discrimination (USD) measurement. By analyzing a two-point attack by a third party, we find that our protocol has a stronger ability to resist external attacks than G-protocol and Y-protocol. Our protocol requires a smaller number of compressions than that in G-protocol (Gao *et al.*, Opt. Exp. 2012, 20, 17411–17420) and Y-protocol (Yan *et al.* Quant. Inf. Process. 2014, 13, 805–813), which means less post-processing. Our protocol shows better database security and user privacy compared with G-protocol.

Keywords: quantum private query; two-point attack; conclusive bits

1. Introduction

Nowadays, communications are omnipresent. The problems of security and privacy have come to assume an unprecedented importance. Cryptography is an effective way to ensure data security in communication. Both classical and quantum cryptography can be used to ensure security. However, a higher advantage in security has been shown for quantum physical principles. Therefore, in recent years, more and more scholars have been paying attention to quantum cryptography.

In communications, the safety of common privacy is usually considered. However, in communications among distrustful users, both common privacy and individual privacy need to be protected. Private information retrieval (PIR) [1] is an application in such research areas, which guarantees the security of private database query. Another similar application is called symmetrically private information retrieval (SPIR) [2], which finishes the following task: user (Alice) obtains a record in a database that she has paid for it, however, the database provider (Bob) should not know which record Alice obtains. On the other hand, Alice should not know about other records that she does not pay for. That is, SPIR protects both Alice's privacy and Bob's database privacy. However, classical cryptosystems and using physical principles cannot ideally realize the task of SPIR [3].

Quantum Private Query (QPQ) is the quantum scheme for the SPIR problem. Bennett [4] and Brassard [5] proposed quantum protocols to solve the similar tasks of SPIR, and it was found to be difficult to offer complete protection for both sides. In 2008, the pioneer QPQ scheme (GLM protocol, where GML denotes the first letter of the name of the three authors) has been put forward by Giovannetti *et al.* [6]. In the scheme, oracle operations are used to denote records in the database and are operated on the incoming query states. Query state and decoy state are needed in the protocol. The query state is used to obtain the needed record from the database. While possible attacks from Bob are checked with the decoy state, in the GLM scheme, Alice's privacy is protected by the non-signaling principle, which means that the spiteful behavior of Bob may lead to wrong answers to Alice. The behavior of Bob will be found by Alice later which will destroy her trust in him, that is called being cheat sensitive for user privacy. On the other hand, GLM protocol shows good database privacy. That is, no more than two records are obtained through dishonest queries. GLM protocol displays better performance in communication complexity and computational complexity compared with existing schemes. After that, the proof-of-principle experimental realization and the security analysis of GLM protocol were done in Refs. [7,8], respectively. Similar with GLM protocol, Olejnik *et al.* put forward O-protocol [9] (another QPQ protocol) based on oracle operation. O-protocol reduces communication complexity further by using only one query state to obtain the needed record from the database and detect eavesdropping from Bob.

Although the existing schemes show high quality theoretically, they are hard to realize for large databases because of the difficulty of high-dimensional oracle operation. Jakobi *et al.* provides a new way for solving the difficulty, in which Alice and Bob share an oblivious key based on SARG04 QKD, which is proposed by Scarani *et al.* in 2004 [10]. This scheme is a pioneer practical QPQ protocol (J-protocol) [11]. In this practical scheme, Bob knows the whole key, which is for encrypting the whole database, and Alice knows only limited bits of the key, which safeguards the database privacy. Oracle operations and other complex operations are not included in J-protocol; therefore, it is easy to realize for a large database.

In 2012, Gao *et al.* put forward a flexible QPQ scheme (G-protocol) [12] based on J-protocol. G-protocol shows better performance in flexibility, communication complexity and security than J-protocol. In G-protocol, non-orthogonal states $\{|0>, |1>, |0'>, |1'>\}$ are selected as carrier states. (Here, $|0'> = \cos\theta|0> + \sin\theta|1>$, $|1'> = \sin\theta|0> - \cos\theta|1>$, and θ is polarization angle). By adjusting the value of θ, the length of Alice's key bits is limited to a certain reasonable value. When $\theta < \pi/4$, G-protocol displays better database security, but poor user privacy.

In 2013, Yang *et al.* put forward another QPQ scheme (Y-protocol) [13] based on a two-particle entangled state and non-orthogonal projective measurements. Y-protocol has all the features of G-protocol, such as being flexible, loss tolerant and practical. What's more, it displays a better user privacy.

In this paper, we presents another QPQ scheme based on two non-orthogonal states and unambiguous state discrimination (USD) measurement. Our protocol can resist two-point attacks by a third party. In our protocol, a smaller number of compressions is required than G-protocol and Y-protocol under similar conditions, which means that less post-processing is needed in our protocol. Our protocol shows better database security and user privacy compared with G-protocol. Furthermore, our protocol requires a bigger polarization angle to achieve similar conditions (the number of compressions, the average bits that Alice will know in the final key, the probability that Alice can not know any bits at all and the length of Bob's final key) than G-protocol, which means that our protocol is easier to realize technically compared with Gao's protocol.

2. The QPQ Protocol Based on Two Non-Orthogonal States

On assumption that N records are included in Bob's database, one of them is bought by Alice, and Alice intends to obtain it in secret. The scheme is to help them to complete the task safely. Our idea is to distribute a pair of oblivious secret keys between Alice and Bob which is known completely to Bob and partly to Alice, through a series of steps. To try to reduce the bits Alice knows in the raw key, Bob generates a raw key with length kN, where k is a natural number.

(1) Bob randomly prepares some non-orthogonal states $|\varphi_0\rangle$ or $|\varphi_1\rangle$ forming quantum sequence S, where

$$\begin{aligned} |\varphi_0\rangle &= \cos\tfrac{\theta}{2}\,|0\rangle + \sin\tfrac{\theta}{2}\,|1\rangle, \\ |\varphi_1\rangle &= \cos\tfrac{\theta}{2}\,|0\rangle - \sin\tfrac{\theta}{2}\,|1\rangle. \end{aligned} \quad (1)$$

The range of parameter θ is between 0 and $\pi/2$. Bob inserts some decoy states $|+>, |->, |0>$ or $|1>$ in sequence S randomly, which forms new sequence S′. Bob tells Alice the length of sequence S′ through an authenticated channel. Then, Bob sends new sequence S′ to Alice.

(2) When Alice receives each particle that Bob sends to her, she stores it in quantum memory firstly. After Alice has received the whole sequence S′, Alice informs Bob about this. Then, Bob tells Alice which particles are used as decoy states and the basis of the corresponding decoy state. Alice extracts decoy states and measures them. Alice tells Bob the measurement results. Bob determines whether there is eavesdropping by comparing the information Alice reports and the decoy states Bob prepares. If there is eavesdropping, Bob stops the protocol.

(3) Alice measures the rest particles with unambiguous state discrimination (USD) method [14] to distinguish which state the qubit is in. The success probability of this USD measurement is bounded (from above) by $p = 1 - F(|\varphi_0\rangle\langle\varphi_0|, |\varphi_1\rangle\langle\varphi_1|)$ =1 $- |\langle\varphi_0|\varphi_1\rangle| = 1 - \cos\theta$, where $F(|\varphi_0\rangle\langle\varphi_0|, |\varphi_1\rangle\langle\varphi_1|) = Tr(\sqrt{|\varphi_0\rangle\langle\varphi_0|\,|\varphi_1\rangle\langle\varphi_1|\,|\varphi_0\rangle\langle\varphi_0|})$ is the fidelity between the two states to be discriminated. That is, Alice obtains the state of the qubit she measures with probability $p = 1 - \cos\theta$. Thus, Alice knows the corresponding bit that the qubit is carrying with a certain probability $p = 1 - \cos\theta$. For Bob, he doesn't know which particle Alice measures successfully. Alice represents $|\varphi_0\rangle$ as "0" and $|\varphi_1\rangle$ as "1". Alice publishes which qubits in sequence S she has successfully received.

(4) Bob also obtains a binary sequence according to the quantum sequence S that Bob prepares and the rule: $|\varphi_0\rangle$ denotes "0" and $|\varphi_1\rangle$ denotes "1".

(5) For the lost particles Alice publishes, Bob flips his corresponding bits randomly. Then Bob randomly adds a bit "0" or "1" behind his data; by doing so, Bob doubles his key. Obviously, Bob's bits corresponding to lost particles are all used as parts of the raw key. Therefore, Alice will not falsely declare lost particles, because Alice will not benefit from reporting a lost particle for any unsuccessful USD measurement, such as increasing the probability of conclusive measurements and knowing a larger fraction of bits that are expected for a giving θ. On the contrary, Bob will obtain more information about Alice's bits if Alice reports a lost particle for any unsuccessful USD measurement. In addition, by flipping Bob's bits corresponding to lost particles randomly, Eve can not obtain database information by measuring some lost particles.

Tables 1 and 2 are examples of sharing oblivious raw keys between Alice and Bob by using all particles in sequence S (the lost particles and received particles). Table 1 shows the case that Alice honestly reports all lost particles. Table 2 shows the case that Alice dishonestly reports the lost particles; that is, Alice will report some measuring failure particles as lost particles. In Tables 1 and 2 "#" denotes the honest reporting of lost particle, "?" denotes measurement failure of Alice , "$" denotes measurement failure of Alice but reporting particles as lost particles, "*" denotes knowing nothing about the bit. In Tables 1 and 2 the bits in Bob's final bits corresponding to reporting as lost particles (including the honest reporting of lost particles and measurement failure by Alice, but reporting particles as lost particles) are the results of flipping the original Bob's bits randomly.

Table 1. The case that Alice honestly reports all lost particles.

Order Number	**1**		**2**		**3**		**4**		**5**		**6**		**7**		**8**		**9**	
The states Bob prepares	$\vert\varphi_0\rangle$		$\vert\varphi_1\rangle$		$\vert\varphi_1\rangle$		$\vert\varphi_1\rangle$		$\vert\varphi_0\rangle$		$\vert\varphi_0\rangle$		$\vert\varphi_0\rangle$		$\vert\varphi_0\rangle$		$\vert\varphi_0\rangle$	
Bob's original bits	0	1	1	0	1	1	1	0	0	1	0	1	0	0	0	1	0	1
Lost particles and Alice's measuring result	#	*	$\vert\varphi_1\rangle$	*	$\vert\varphi_1\rangle$	*	?	*	?	*	$\vert\varphi_0\rangle$	*	?	*	?	*	#	*
Alice's bits			1		1						0							
Bob's final bits	0	1	1	0	1	1	1	0	0	1	0	1	0	0	0	1	1	1

Notes: $|\varphi_0\rangle$ and $|\varphi_1\rangle$ denotes non-orthogonal states in Equation (1).

Table 2. The case that Alice dishonestly reports the lost particles.

Order Number	**1**		**2**		**3**		**4**		**5**		**6**		**7**		**8**		**9**	
The states Bob prepares	$\|\varphi_0\rangle$		$\|\varphi_1\rangle$		$\|\varphi_1\rangle$		$\|\varphi_1\rangle$		$\|\varphi_0\rangle$		$\|\varphi_0\rangle$		$\|\varphi_0\rangle$		$\|\varphi_0\rangle$		$\|\varphi_0\rangle$	
Bob's original bits	0	1	1	0	1	1	1	0	0	1	0	1	0	0	0	1	0	1
Lost particles and Alice's measuring result	#	*	$\|\varphi_1\rangle$	*	$\|\varphi_1\rangle$	*	$	*	?	*	$\|\varphi_0\rangle$	*	$	*	$	*	#	*
Alice's bits			1		1						0							
Bob's final bits	0	1	1	0	1	1	0	0	0	1	0	1	1	0	0	1	1	1

In Table 1, Alice honestly reports all lost particles (1,9) and measurement failure of particles (4,5,7,8). In Table 2, Alice dishonestly reports the lost particles (1,4,7–9), where particles 4,7,8 are the case of measurement failure by Alice but reporting particles as lost particles. By comparing Alice's bits and Bob's final bits in Tables 1 and 2 we find that the dishonest reporting of lost particles will not impact Alice's bits. In addition, we still find that the bits corresponding to dishonest reporting of lost particles in Bob's final bits are the results of flipping Bob's corresponding original bits randomly; however, Alice knows nothing about these bits. From Tables 1 and 2 we find that if Alice honestly reports lost particles, Bob knows that Alice may successfully measure particles 2–8; if Alice reports failure measurements as lost particles, Bob knows that Alice may successfully measure particles 2,3,5,6. That is, the probability that Bob knows Alice's bits increases. Therefore, dishonest reporting of lost particles will make Bob know more information of Alice's bits; however, Alice can not know more information about Bob's final bits.

For Alice, with the exception of those two-bit-groups corresponding to lost particles, she owns the first bit of each of Bob's two-bit-groups corresponding to received photons, but nothing about the second bit. Whatever the measurement result of Alice, the second bit is 0 or 1 with equal probability. Therefore, Alice knows the first bit of Bob's each two-bit-group corresponding to received photons with probability $p = 1 - \cos\theta$.

In this way, Alice and Bob have shared a raw key K^r known completely to Bob and partly to Alice. Obviously, for each two-bit-group corresponding to received photons, Alice should know only one bit with probability $1 - \cos\theta$. Therefore, Alice knows only $p \leqslant (1 - \cos\theta)/2$ of the raw key K^r on average. That is, $(1 - \cos\theta)/2$ is the upper bound of the bits that Alice knows of the raw key K^r.

(6) To reduce the bits by Alice known in the raw key they shared in the above steps, Alice and Bob execute post-processing on the raw key. Because the length of K^r is kN, where k is a natural number, Alice and Bob break K^r up into k parts, thus the length of each parts is N. By adding the k parts bitwise, the raw key becomes a final key with length N. Bob knows the whole key, while Alice only knows several bits. For example, Bob's bits are "011011100000" and Alice's bits are "**1*1***0***", where "*" denotes that Alice knows nothing about the bit. If $k = 2$, Bob and Alice divide their bits into two parts, respectively: Bob's bits: "011011, 100000"; Alice's bits: "**1*1*, **0***". Then, Bob and Alice perform bitwise exclusive-OR (XOR) operation on the top six bits and the post six bits, respectively: Bob: 011011 XOR 100000 = 111011; Alice: **1*1* XOR **0*** = **1***. Therefore, Bob's and Alice's key is 111011 and **1***, respectively. The process is similar to that in J-protocol, G-protocol and Y-protocol. If Alice knows nothing of the final key after this post-processing, the protocol should be restarted. However, this condition can be avoided by choosing appropriate parameters, which will be analyzed later.

(7) Bob encrypts the database and Alice obtains the record that she needs. The detailed procedure is as follows: if Alice owns the jth bit K_j of Bob's key K, and she needs to obtain the ith record X_i in Bob's database. Then, Alice tells Bob the value $s = j - i$. If s is a negative number, Bob shifts K right circularly with $|s|$ bits; otherwise, Bob shifts K left circularly with s bits, by doing so Bob

can obtain a new key K'. Bob encrypts the database with K' in the way of a one-time pad. Alice decrypts the ith record with her key K_j.

3. Analysis

After step (6) (Alice and Bob add k substrings bitwise), Alice obtains $\overline{n} = NR^k$ bits of the final key K on average, where $R = \frac{1-\cos\theta}{2}$. The probability that Alice can not know any bits at all is $P_0 = (1-R^k)^N = (1-\frac{\overline{n}}{N})^N$. When $N \gg \overline{n}$ is satisfied, $P_0 \approx e^{-\overline{n}}$ is obtained and $\overline{n}$ follows a Poisson distribution approximately. When $\overline{n} \leqslant 2$ is satisfied, we have $P_0 \geqslant 0.135$; when $\overline{n} \geqslant 3$ is satisfied, we have $P_0 \leqslant 0.05$. In order to make Alice get less bits, at the same time, the failure rate is very small, obviously, $\overline{n} = 3$ is an optimal value. We have $P_{\geqslant 4} = 1 - P_3 - P_2 - P_1 - P_0 = 1 - 13e^{-3} = 0.35$ and $P_{\geqslant 5} = 1 - P_4 - P_3 - P_2 - P_1 - P_0 = 0.18$, where $P_{\geqslant 4}$ denotes the probability that Alice knows more than four bits in a distribution and $P_{\geqslant 5}$ denotes the probability that Alice knows more than five bits in a distribution when the average value of $\overline{n}$ is three. That is to say, the success probability of Alice to learn more bits by cheating is very small. The distribution of $\overline{n}$ in G-protocol and Y-protocol also follows a Poisson distribution approximately.

Figure 1 (locating N = 10,000) indicates that a different $\overline{n}$ is reached by adjusting θ and k when N is fixed. Compared with G-protocol (see Figure 2 in [12]), we find that, to achieve similar N and $\overline{n}$, the value of k ranges from 2 to 5 in our protocol, while from 5 to 10 in G-protocol. This means that we need smaller k compared with G-protocol under similar conditions ($\overline{n}$, θ and N). That is, less post-processing is required in our protocol.

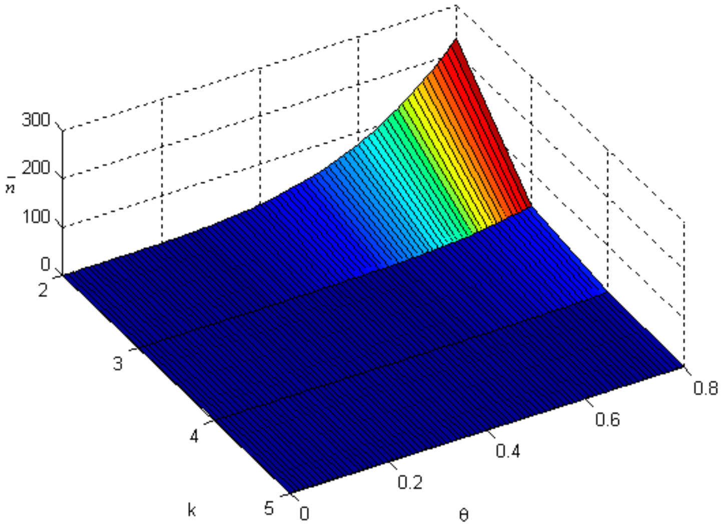

Figure 1. When N = 10,000, $\overline{n}$ << N can be achieved.

3.1. Loss Tolerant

The following two reasons can show that our protocol is loss tolerance. Firstly, Bob's bits corresponding to lost particles are all used as parts of the raw key. Therefore, Alice will not falsely declare lost particles, because Alice will not benefit from reporting a lost photon for any unsuccessful USD measurement, such as increasing the probability of conclusive measurements and knowing a larger fraction of bits than expected for a giving θ. On the contrary, Bob will obtain more information about Alice's bits if Alice reports a lost particle for any unsuccessful USD measurement. Secondly, by flipping Bob's bits corresponding to lost particles randomly and post-processing on the raw key, Eve can not obtain database information by measuring some lost photons.

3.2. The Third-Party Attacks

The intercept resend attack and Trojan Horse attack are two common attack strategies in quantum secure communication. In our protocol, the particles are one-way transmission, and there is no loop (circuit); therefore, there is no Trojan Horse attack. Two-point attack [15] is a specific intercept resend attack strategy aiming to attack the QKD system based on two non-orthogonal states [16]. In two-point attacks, there are two third-party eavesdroppers Eve1 and Eve2. Eve1 is located at a point near Alice's security domain, and Eve2 at a point near Bob's security domain. Eve1 and Eve2 can communicate with each other through a classical channel. Eve1 intercepts the quantum channel and measures every quantum state. Eve2 resends the quantum state according to the measurement result Eve1 tells him. Next, we analyze the influence of two-point attacks on user privacy and database privacy in our protocol.

(1) The influence of two-point attacks on user privacy

In our protocol, if Eve1 intercepts a particle and successfully measures it with a USD method, after Eve1 tells the result to Eve2, Eve2 resends the particle to Alice. Then, Eve1 and Eve2 will know the bit of Alice. That is, user privacy is threat. However, because Bob inserts decoy particles with x-basis and z-basis randomly in quantum sequence S, Eve1 and Eve2 has $1/3$ probability or less to select correct basis (USD, x-basis or z-basis) to measure the intercepted particles successfully. In addition, the eavesdropping of Eve1 and Eve2 will be easily found with probability $2/3$. They only have the probability

$$P_T = \frac{1-\cos\theta}{3} \tag{2}$$

to know the corresponding bit of Alice. While in G-protocol and Y-protocol, the probability that Eve1 and Eve2 are found is 0, and the probability that Eve1 and Eve2 know the corresponding bit of Alice is

$$P_T^G = P_T^Y = \frac{\sin^2\theta}{2}. \tag{3}$$

From Figure 2, we find that the probability that Eve1 and Eve2 know Alice's bits for different θ in our protocol is much lower than that in G-protocol and Y-protocol.

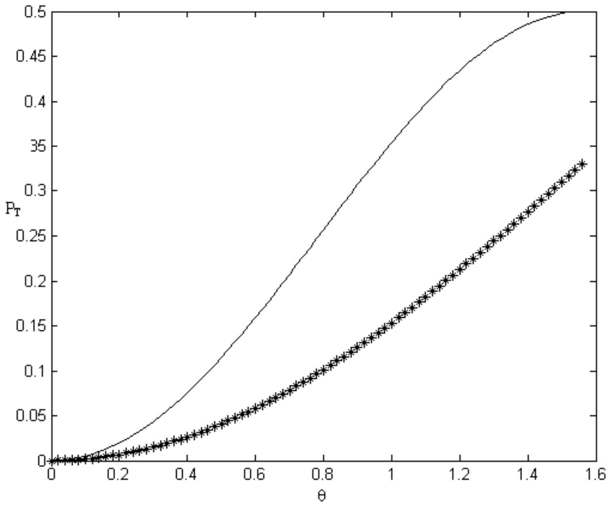

Figure 2. The probabilities that Eve1 and Eve2 know Alice's bits for different θ in our protocol, G-protocol and Y-protocol. The solid line represents the probabilities for G-protocol (${P_T}^G$) and Y-protocol (${P_T}^Y$); the star line represents the probability for our protocol (P_T).

(2) The influence of two-point attacks on database privacy

In our protocol, Bob's bits corresponding to particles that are intercepted by Eve1 without being resend by Eve2 are flipped randomly; therefore, Eve1 and Eve2 know nothing about these bits. For the particles that Eve1 intercepts and Eve2 resends, according to analysis in the influence of two-point attacks on user privacy, we know that Eve1 and Eve2 will be easily found with probability $2/3$, and they only have the probability $(1 - \cos\theta)/3$ to know the corresponding bit. Therefore, it is difficult to obtain database information for Eve1 and Eve2, and their eavesdropping will be found easily.

3.3. Qubit Efficiency Comparison

To estimate the efficiency of QPQ, we define the qubit efficiency as $\eta = \frac{N}{\overline{n}(M+l)}$, where N denotes the final classical bits that can be generated for Bob, $\overline{n}$ denotes the final classical bits that can be generated for Alice, M denotes the total coding particles in each communication, and l denotes the decoy photons to check the presence of eavesdropping in each communication.

Because $\overline{n} = NR^k$, we have $\frac{N}{\overline{n}} = \frac{1}{R^k}$, then

$$\eta = \frac{1}{R^k(M+l)} \tag{4}$$

For G-protocol, $R = \frac{\sin^2\theta}{2}$, $M = kN$ and there is no eavesdropping detecting, thus

$$\eta_G = \frac{1}{(\sin^2\theta/_2)^k kN}. \tag{5}$$

For Y-protocol, $R = \frac{\sin^2\theta}{2}$, $M = 2kN$ and there is no eavesdropping detecting, thus

$$\eta_Y = \frac{1}{(\sin^2\theta/_2)^k 2kN}. \tag{6}$$

For our protocol, $R = \frac{1-\cos\theta}{2}$, $M = kN$, thus

$$\eta_{Our} = \frac{1}{(\frac{1-\cos\theta}{2})^k (kN+l)}. \tag{7}$$

In Table 3, we compare the qubit efficiency of G-protocol [12], Y-protocol [13] and our protocol when N = 10,000.

Table 3. Examples of qubit efficiency comparison of G-protocol, Y-protocol and our protocol when N = 10,000 and l = 15,000.

θ, k	θ = 0.15, k = 2	θ = 0.25, k = 2	θ = 0.36, k = 3	θ = 0.4, k = 3	θ = 0.57, k = 4	θ = 0.62, k = 4
G-protocol	0.4010	0.0534	0.1395	0.0765	0.0556	0.0308
Y-protocol	0.2005	0.0267	0.0698	0.0382	0.0278	0.0154
Our protocol	0.9064	0.1183	0.6749	0.3614	0.4656	0.2424

From Table 3, we find that our protocol has better qubit efficiency than G-protocol [12] and Y-protocol [13].

3.4. Database Security

Database security means that Alice can not get extra records in Bob's database no matter the methods she uses. Suppose that, to obtain extra records, Alice performs more efficient measurements on particles that Bob sends to her.

As analyzed in Ref. [11], there is a measurement called minimal error probability measurement [17], which distinguishes two equally likely qubits. If Alice measures the k qubits forming an element of the final key K with the efficient measurement, she obtains the bits of K directly. By using minimal error probability measurement, the maximal chance of distinguishing the two states ρ_0 and ρ_1 correctly is $P_{guess} = 1/2 + \mathrm{D}(\rho_0, \rho_1)/2$, where $\mathrm{D}(\rho_0, \rho_1)$ denotes trace distance between ρ_0 and ρ_1. In our protocol, the probability is $P_{guess} = (\frac{1}{2} + \frac{1}{2}\sin^k\theta)$ at most. However, even though Alice guesses $|\varphi_0\rangle$ and $|\varphi_1\rangle$ correctly, (the probability is $P_{guess}/3$), she can infer correctly the corresponding two-bit group of Bob with 50% chance, that is, on average, Alice can infer correctly each bit of Bob with a 75% chance. Therefore, Alice can correctly guess the bit of Bob with probability

$$P_{guess} = \frac{1}{4}(\frac{1}{2} + \frac{1}{2}\sin^k\theta) \tag{8}$$

at most. Obviously the probability is much less than that in G-protocol ($P^G_{guess} = \frac{1}{2} + \frac{1}{2}\sin^k\theta$), which means better database security than that in G-protocol. Figure 3 shows that P_{guess} in our protocol is much less than that in G-protocol for a giving $k = 2$.

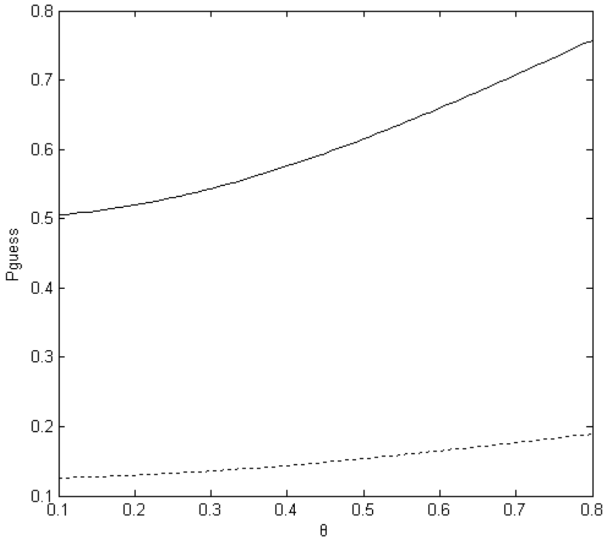

Figure 3. Comparison of probability with which Alice correctly guesses the bit of Bob for different θ between our protocol and G-protocol when $k = 2$. The solid line represents P_{guess} for G-protocol; the dotted line represents P_{guess} for our protocol.

3.5. User Security

We suppose that Bob transmits a false qubit $|\Theta>$ to Alice, where $|\Theta> = \cos\beta|+> + \sin\beta|->$. Alice obtains the conclusive result with probability $(1 - \cos(\theta \pm \beta))/2$. By sending a fake state, Bob biases the probability of measurement results of Alice between $((1 - \cos(\theta + \beta))/2, (1 - \cos(\theta - \beta))/2)$ unless $\beta = 0$. Let $Y = \cos(\theta - \beta) - \cos(\theta + \beta)$, by deducing $\frac{dY}{d\beta} = 0$ and $\frac{d^2Y}{d^2\beta}$, we can get $\beta = \pi/2$ when $\frac{d^2Y}{d^2\beta} < 0$. We construe this result as that optimal probability with which Bob knows Alice's bits is between $((1 - \cos(\theta + \pi/2))/2, (1 - \cos(\theta - \pi/2))/2)$. That is, the bounds on P_{b} (the probability with which Bob knows Alice's bits) is:

$$(1 - \cos(\theta + \pi/2))/2 < P_{\text{b}} < (1 - \cos(\theta - \pi/2))/2. \tag{9}$$

Figure 4 shows that when θ ∈ (0,π/2), the low bound of P_b for our protocol is smaller than that in G-protocol. The upper bound of P_b for our protocol is smaller than that in G-protocol when $0 < \theta < \pi/4$, and bigger than that in G-protocol when $\pi/4 < \theta < \pi/2$. That is, when $0 < \theta < \pi/4$, we can achieve a better user privacy compared with G-protocol.

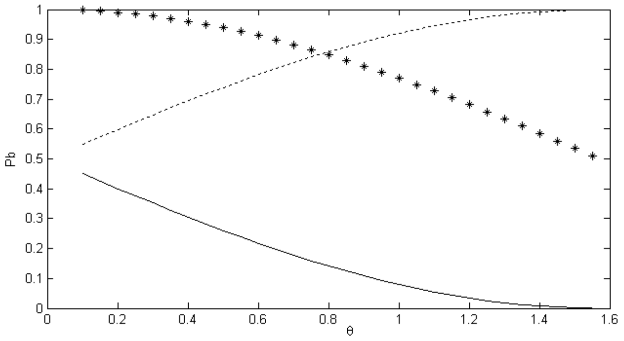

Figure 4. Comparisons of probability with which Bob knowing Alice's bits for different θ between our protocol and G-protocol. The star line represents P_b for G-protocol; the dotted line represents the upper bound of P_b for our protocol; the solid line represents the lower bound of P_b for our protocol.

4. Discussion

In the above section, we have analyzed the advantages of our protocol in the aspect of less post-processing (smaller k) compared with G-protocol under similar conditions ($\overline{n}$, θ and N), in the aspect of qubit efficiency compared with G-protocol and Y-protocol for fixed N, and θ and k, in the aspect of database security and user security compared with G-protocol. In this section, we give a more comprehensive discussion focusing on security and resources for specific, optimal sets of parameters.

From Table 4, we find that when $\overline{n}$ is fixed, G-protocol, Y-protocol and our protocol have the same P_0. When N and $\overline{n}$ are given , for a specific k, P_b in our protocol is much lower than that in G-protocol and is close to that in Y-protocol, which means that our protocol shows much higher user security than G-protocol and the user security of our protocol is close to Y-protocol. Furthermore, in the same condition, our protocol always needs bigger θ than that in G-protocol and Y-protocol. Because a very small θ might make its realization technically difficult [12], our protocol is easier to realize technically than G-protocol and Y-protocol.

Table 4. Examples of P_0, P_b, θ and $M + l$ comparison of G-protocol, Y-protocol and our protocol for a given N and $\overline{n}$.

***a.* $N = 10^4$, $\overline{n} = 3$ and $k = 3$**				
Protocol	P_0	P_b	θ	$M + l$
Y-protocol	0.05	0.32~0.68	0.375	6×10^4
G-protocol	0.05	0.98	0.375	3×10^4
Our-protocol	0.05	0.25~0.75	0.5234	4.5×10^4
***b.* $N = 10^5$, $\overline{n} = 3$ and $k = 3$**				
Protocol	P_0	P_b	θ	$M + l$
Y-protocol	0.05	0.38~0.62	0.252	6×10^5
G-protocol	0.05	0.98	0.252	3×10^5
Our-protocol	0.05	0.33~0.67	0.3544	4.5×10^5
***c.* $N = 10^5$, $\overline{n} = 3$ and $k = 4$**				
Protocol	P_0	P_b	θ	$M + l$
Y-protocol	0.05	0.31~0.69	0.395	8×10^5
G-protocol	0.05	0.96	0.395	4×10^5
Our-protocol	0.05	0.24~0.76	0.5510	5.5×10^5

Table 5 shows that, for a given N and $\overline{n}$, when three protocols (Y-protocol, G-protocol and our protocol) achieve similar θ, our protocol shows higher qubit efficiency than Y-protocol and better user privacy than G-protocol.

Table 5. Examples of P_0, P_b, θ and $M + l$ comparison of G-protocol, Y-protocol and our protocol for $N = 10^4$ and a given $\overline{n}$.

a. $\overline{n}$ **= 3**					
Protocol	P_0	P_b	θ	k	$M + l$
Y-protocol	0.05	0.24~0.76	0.539	4	8×10^4
G-protocol	0.05	0.93	0.539	4	4×10^4
Our-protocol	0.05	0.25~0.75	0.5234	3	4.5×10^4
b. $\overline{n}$ **= 5**					
Protocol	P_0	P_b	θ	k	$M + l$
Y-protocol	0.0067	0.23~0.77	0.579	4	8×10^4
G-protocol	0.0067	0.92	0.579	4	4×10^4
Our-protocol	0.0067	0.23~0.77	0.5712	3	4.5×10^4

In a practical condition, channel noise is inevitable, which will cause errors in the obvious key shared between Alice and Bob. Therefore, error correction is necessary. We can use the method proposed in Ref. [18] to correct errors. Suppose the kN-bit raw oblivious key is denoted as $O^R = O_1^R O_2^R ... O_{kN}^R$, the final obvious key after dilution is denoted as $.O^F = .O_1^F O_2^F ... O_N^F$. Here, $O_i^F = \bigoplus_{j=0}^{k-1} O_{i+jN}^R$, $1 \leqslant i \leqslant N$, and $\oplus$ denotes the addition module 2. Alice and Bob select a $[k, s]$ error-correcting code [19] which uses k bits codeword to encode s bits word using generator matrix G and can correct one codeword error bits with error-correcting function. Bob chooses a bits word $M = (m_1, m_2, \ldots, m_s)$ and obtains the corresponding bits codeword $W = (w_1, w_2, \ldots, w_k)$ by calculating $W = M \cdot G$. Then, Bob encrypts W with $\{O_{i+jN}^R\}$ as the key, by using a one-time pad, and sends the ciphertext c to Alice. If Alice knows all the k bits $\{O_{i+jN}^R\}$, Alice decrypts c with $\{O_{i+jN}^R\}$ and obtains a k-bit codeword W'. Alice corrects the error in W' and obtains W. By adding the k bits in W bitwise, Alice knows O_i^F. If Alice does not know all the k bits in $\{O_{i+jN}^R\}$, she labels $O_i^F = ?$. Bob also adds the k bits in W bitwise to obtain his corresponding bit O_i^F.

5. Conclusions

We put forward a novel QPQ protocol based on two non-orthogonal states and unambiguous state discrimination (USD) measurement. Our protocol is loss tolerant. Compared with existing QPQ protocols, we have the following differences:

(1) We analyze the influence of two-point attacks by a third party on user privacy and database privacy in our protocol. By comparing, we find that the probability that Eve1 and Eve2 know Alice's bits for different θ in our protocol is much lower than that in G-protocol and Y-protocol, which means that our protocol has a stronger ability to resist external attacks than G-protocol and Y-protocol.

(2) Smaller k is required to achieve similar conditions (n, θ and N) than G-protocol and Y-protocol, which means less post-processing and higher qubit efficiency.

(3) For a given N and $\overline{n}$, our protocol shows much higher user security than G-protocol and the user security of our protocol is close to Y-protocol. However, in the same condition, our protocol always needs bigger θ than that in G-protocol and Y-protocol. Because a very small θ might make its realization technically difficult [12], our protocol is easier to realize technically than G-protocol and Y-protocol.

However, because of eavesdropping detection in Step (2), our protocol requires quantum memory on the Alice side. The use of quantum memory will bring our protocol difficulties in practicality and realizability using current technology.

Acknowledgments: This work is supported by the National Natural Science Foundation of China (Grant Nos. 61402058, 61572086), the Fund for Middle and Young Academic Leaders of Chengdu University of Information Technology (Grant No. J201511), the Science and Technology Support Project of Sichuan Province of China (Grant No. 2013GZX0137), and the Fund for Young Persons Project of Sichuan Province of China (Grant No. 12ZB017).

Author Contributions: All of the authors read and approved the final manuscript. Yan Chang and Shibin Zhang conceived and designed the protocol; Yan Chang and Guihua Han performed the experiments; Yan Chang analyzed the data; Yan Chang, Zhiwei Sheng, Lili Yan and Jinxin Xiong contributed modification of paper; Yan Chang wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chor, B.; Goldreich, O.; Kushilevitz, E.; Sudan, M. Private Information Retrieval. *J. ACM* **1998**, *45*, 965–981. [CrossRef]
2. Gertner, Y.; Ishai, Y.; Kushilevitz, E.; Malkin, T. Protecting data privacy in private information retrieval schemes. *J. Comput. Syst. Sci.* **2000**, *60*, 592–629. [CrossRef]
3. Lo, H.-K. Insecurity of quantum secure. *Phys. Rev. A* **1997**, *56*, 1154–1162. [CrossRef]
4. Bennett, C.-H.; Brassard, G.; Crlpeau, C.; Skubiszewska, M.H. Practical quantum oblivious transfer. *Adv. Cryptol.* **1992**, *576*, 351–366.
5. Brassard, G.; Crepeau, C.; Jozsa, R.; Langlois, D. A quantum bit commitment scheme provably unbreakable by both parties. In Proceedings of the 34th Annual Symposium on Foundations of Computer Science, Palo Alto, CA, USA, 3–5 November 1993; Volume 362.
6. Giovannetti, V.; Lloyd, S.; Maccone, L. Quantum private queries. *Phys. Rev. Lett.* **2008**, *100*, 230502. [CrossRef]
7. Martini, F.D.; Giovannetti, V.; Lloyd, S.; Maccone, L.; Nagali, E.; Sansoni, L.; Sciarrino, F. Experimental quantum private queries with linear optics. *Phys. Rev. A* **2009**, *80*, 010302. [CrossRef]
8. Giovannetti, V.; Lloyd, S.; Maccone, L. Quantum private queries: Security analysis. *IEEE Trans. Inf. Theory* **2010**, *7*, 3465–3477. [CrossRef]
9. Olejnik, L. Secure quantum private information retrieval using phase-encoded queries. *Phys. Rev. A* **2011**, *84*, 022313. [CrossRef]
10. Scarani, V.; Acin, A.; Ribordy, G.; Gisin, N. Quantum cryptography protocols robust against photon number splitting attacks for weak laser pulse implementations. *Phys. Rev. Lett.* **2004**, *92*, 057901. [CrossRef] [PubMed]
11. Jakobi, M.; Simon, C.; Gisin, N.; Bancal, J.D.; Branciard, C.; Walenta, N.; Zbinden, H. Practical private database queries based on a quantum-key-distribution protocol. *Phys. Rev. A* **2011**, *83*, 022301. [CrossRef]
12. Gao, F.; Liu, B.; Wen, Q.Y. Flexible quantum private queries based on quantum key distribution. *Opt. Exp.* **2012**, *20*, 17411–17420. [CrossRef] [PubMed]
13. Yang, Y.-G.; Sun, S.J.; Xu, P.; Tian, J. Flexible protocol for quantum private query based on B92 protocol. *Quant. Inf. Process.* **2014**, *13*, 805–813. [CrossRef]
14. Raynal, P. Unambiguous State Discrimination of two density matrices in Quantum Information Theory. **2006**, arXiv: quant-ph/0611133.
15. Yang, L.; Wu, L.-A. Two-point attack on the two nonorthogonal states QKD protocol over a fiber optic channel. **2005**, arXiv: quant-ph/0310080.
16. Bennett, C.-H. Quantum Cryptography Using Any Two Nonorthogonal States. *Phys. Rev. Lett.* **1992**, *68*. [CrossRef]
17. Helstrom, C.W. *Quantum Detection and Estimation Theory*; Academic Press: New York, NY, USA, 1976.
18. Gao, F.; Liu, B.; Huang, W.; Wen, Q.-Y. Postprocessing of the oblivious key in Quantum Private Query. *IEEE J. Sel. Top. Quant. Electron.* **2014**, *21*, 6600111.
19. MacWilliams, F.J.; Sloane, N.J.A. *The Theory of Error-Correcting Codes*; North-Holland Publishing Company: Amsterdam, The Netherlands, 1977.