Protection of Information in Quantum Databases

Y. Ozhigov*
Department of Applied Mathematics,
Moscow State University of Technology Stankin,
Vadkovsky per. 3a, 101472, Moscow, Russia

All modern cryptography is based on secret keys. But such methods of protecting information make no sense if the key can be quickly discovered by an unauthorized person. This way of penetrating protected systems was made possible by quantum computers due to results in [1, 2]. This paper presents a method for protecting information in a database from a spy which knows all about its control system and has a quantum computer. Typically a database cannot distinguish between the operations of a spy and a legal user.

Such a database with quantum mechanical memory plays the role of a probabilistic oracle for some boolean function f. It returns the value $f(a)$ for query a in time $O(N^2 \log^3 n)$, afterwards its initial state is restored also in the same time, where N is a cardinality of $\mathrm{Dom} f$. Software of the database is independent of a function f. A classical state of such a database must contain a list of pairs $(a, f(a))$, $a \in \mathrm{Dom} f$, taken in some order. Quantum mechanical principles allow mixing all these lists with different orders and equal amplitudes into one quantum state. We call such a state *normal*. All user operations extracting $f(a)$ can be fulfilled only in states of such a sort. Now if somebody S tries to learn $f(b)$ for $b \neq a$ the normal state is ruined such that a legal user with high probability will not obtain a pair of the form $a, \ldots$ and hence the presence of S will be detected. It is proved that for a large N the probability that a spy S learns $f(b)$ does not exceed the probability of its exposure.

Here advantage is taken of relative diffusion transformations (RDT), which make it possible to fulfill all operations in normal states. Such transformations look like diffusion transforms used in [2] but RDT are defined in a different manner. A classical database with this property does not exist.

1. Main definitions

All known models of quantum computers: quantum Turing machines [3, 4], quantum gate arrays [5], and quantum cellular automata [6] can simulate each other with a polynomial slowdown and have the same computational power as classical computers. It is unknown if it is possible to simulate absolute (without oracles) quantum computations by

*Electronic mail address: y@oz.msk.ru.

classical computers with a polynomial slowdown or not. Such a simulation is known only with exponential slowdown [4]. As for relativized (with oracles) computations, the classical simulation with a polynomial slowdown is impossible [7]. There is much evidence that quantum computers are substantially more effective than any classical device for particular problems (e.g., [1, 2, 7, 8]). Nevertheless there are known limits to the speed of quantum algorithms. For example, it has been proved that the bulk of short classical computations cannot be sped up by quantum computers [9, 10].

Another field for quantum computer applications is cryptography. It is well known how the famous Shor's result can be used to break classical RSA codes [1]. In this paper we show how a quantum computer can be used to lock a database. To create such a database we use a natural quantum computer model with two parts: a classical part which transforms by classical laws (e.g., as a Turing machine or cellular automaton), and a quantum part which transforms by quantum mechanical principles. We proceed with exact definitions.

A *memory* (quantum part) is a set $\mathcal{E}$ having elements that are called *qubits*. $\mathcal{E}$ may be designed as a discrete lattice $\mathcal{E} \subseteq Z^m$, or as a tree, and so forth. Each qubit takes values from the complex one-dimensional sphere of radius 1: $\{z_0\mathbf{0} + z_1\mathbf{1} \mid z_1, z_2 \in \mathbb{C}, |z_0|^2 + |z_1|^2 = 1\}$. Here $\mathbf{0}$ and $\mathbf{1}$ are referred to as *basic states* of the qubit and form a basis of $\mathbb{C}^2$. It will be convenient to divide $\mathcal{E}$ into registers of two neighboring qubits each so that each register takes values from $\omega = \{0, 1, 2, 3\}$.

A basic state of the quantum part is a function of the form $e : \mathcal{E} \to \{0, 1\}$. If we fix some order on $\mathcal{E} = \{\nu_1, \nu_2, \ldots, \nu_r\}$ (r even) then a basic state e may be encoded as $|e(\nu_1), e(\nu_2), \ldots, e(\nu_r)\rangle$. Such a state can naturally be identified with a corresponding word in alphabet ω.

Let $e_0, e_1, \ldots, e_{K-1}$ be all basic states, taken in some fixed order, and let $\mathcal{H}$ be the K-dimensional Hilbert space with orthonormal basis $e_0, e_1, \ldots, e_{K-1}$, $2^r = K$. This Hilbert space can be regarded as a tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_r$ of two-dimensional spaces, where $\mathcal{H}_i$ is generated by the possible values of $e(\nu_i)$, $\mathcal{H}_i \cong \mathbb{C}^2$. A (pure) state of the quantum part is an element $x \in \mathcal{H}$ such that $|x| = 1$. Thus, in contrast to classical devices, a quantum device may be not only in basic states, but also in their superpositions, and this imparts surprising properties to such devices.

Put $\mathcal{K} = \{0, 1, \ldots, K-1\}$. For elements $x = \sum_{s\in\mathcal{K}} \lambda_s e_s$, $y = \sum_{s\in\mathcal{K}} \mu_s e_s \in \mathcal{H}$ their dot product $\sum_{s\in\mathcal{K}} \lambda_s \bar{\mu}_s$ is denoted by $\langle x|y\rangle$, where $\bar{\mu}$ means complex conjugation of $\mu \in \mathbb{C}$, hence $\langle x|y\rangle = \overline{\langle y|x\rangle}$.

Let $\{1, \ldots, r\} = \cup_{i=1}^{l} L_i^s$, $L_i^s \cap L_j^s = \emptyset$ $(i \neq j)$, and the *unitary transformations* U_i^s act on $\otimes_{j\in L_i^s} e_j$. Then $U^s = \otimes_{i=1}^{l} U_i^s$ acts on $\mathcal{H}$, $s = 1, 2, \ldots, M$. We require that all U_i^s belong to some finite set of transformations independent of $\mathcal{E}$ which can be easily performed by physical devices.

A *computation* is a chain of such unitary transformations:

$$\chi_0 \overset{U^1}{\to} \chi_1 \overset{U^2}{\to} \cdots \overset{U^M}{\to} \chi_M.$$

The passages $U^s \to U^{s+1}$ $s = 1, \ldots, M$ and the value M are determined by the classical algorithm which points to the partition $\cup L_i$ and chooses the transformations U_i^s sequentially for each s. This algorithm is performed by the classical part of the computer.

Let $\chi = \sum_{s \in \mathcal{K}} \lambda_s e_s$ be some fixed state of the computer, often $\chi = \chi_M$. If $A \in \{0,1\}^k$ is the list of possible values for the first k qubits, then we put $B_A = \{i \mid \exists a_{k+1}, a_{k+2}, \ldots, a_r \in \{0,1\} : e_i = A a_{k+1} a_{k+2} \ldots a_r\}$. A (quantum) result of this observation is a new state $\chi^A = \sum_{i \in B_A} \lambda_i / \sqrt{p_a} e_i$, where $p_A = \sum_{i \in B_A} \lambda_i^2$. An *observation* of the first register in state χ is a procedure which gives a pair: ⟨classical word A, quantum state χ^A⟩ with probability p_A for any possible $A \in \{0,1\}^k$. The only way to learn the result of a quantum computation is to receive such words A.

2. Diffusion transformation

In this section we recall some notions and ideas from [2] and [11].

Every unitary transformation $U : \mathcal{H} \to \mathcal{H}$ can be represented by its matrix $U = (u_{ij})$ where $u_{ij} = \langle U(e_j) | e_i \rangle$. Thus for $x = \sum_{p \in \mathcal{K}} \lambda_p e_p$, $U(x) = \sum_{p \in \mathcal{K}} \lambda'_p e_p$ we have $\bar{\lambda}' = U\bar{\lambda}$, where $\bar{\lambda}, \bar{\lambda}'$ are columns with elements λ_p and λ'_p respectively. The following significant *diffusion* transformation D (introduced in [2]) is defined by $D = -WRW^{-1}$, where $W = U_1 \otimes U_2 \otimes \cdots \otimes U_r$, each U_i acts on $\mathcal{H}_i$ and has the matrix

$$J = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix},$$

and R is the phase inversion of e_0. For every state $x = \sum_{p \in \mathcal{K}} \lambda_p e_p$ an average amplitude is taken as $x_{\text{av}} = \sum_{p \in \mathcal{K}} \lambda_p / K$.

Proposition 1. (From [2].) For every state x

$$\langle e_p | x \rangle - x_{\text{av}} = x_{\text{av}} - \langle e_p | D(x) \rangle. \tag{1}$$

This means that D is an inversion about the average.

One step of Grover's algorithm is the unitary transformation $G = DR_t$ where R_t is a phase rotation of some target state e_t. Proposition 1 implies that the amplitude of e_t grows approximately on $1/\sqrt{K}$ after application of G to the state $x_0 = (e_0 + e_1 + \cdots + e_{K-1})/\sqrt{K}$.

The following two notes about this algorithm have been done in [11].

1. Given a set T of target states of cardinality $|T| = K/4$, then one step of the corresponding transformation G makes all amplitudes of the states $e \notin T$ equal to zero.

2. Let W' be any unitary transformation satisfying $W'(e_0) = 1/\sqrt{K}\sum_{i\in\mathcal{K}} e_i$. Then Proposition 1 remains true if W' is taken instead of W in the definition of D.

3. Relative diffusion transformations

In this section we introduce a generalization of diffusion transformation, the relative diffusion transformation (RDT), which is the key notion in the construction of the control system for our database.

In what follows, for the set $C = \{0, 1, \ldots, N-1\}$, $N < K$, C' denotes $\{e_i \mid 0 \le i \le N-1\}$.

We use the notation $\chi_C = 1/\sqrt{N}\sum_{i\in C} e_i$. Let M be some unitary transformation of the form $\mathcal{H} \to \mathcal{H}$ such that $M(e_0) = \chi_C$. Such a transform M is called *C-mixing*. We do not assume that a subspace spanned by C' is M-invariant.

Definition 1. RDT(B) is the transformation $D_C = -MRM^{-1}$ where R is defined as previously and M is C-mixing.

We now generalize Proposition 1.

Lemma 1. D_C does not change an amplitude μ_s of e_s if $s \notin C$ and makes it $2A/N - \mu_s$ if $s \in C$, where $A = \sum_{s\in C}\mu_s$, $N = |C|$.

Proof. First note that for every $s \in \mathcal{K}$ $M^{-1}(e_s) = \sum_{i\in\mathcal{K}}\alpha_s^i e_i$, $\alpha_s^0 = 1/\sqrt{N}$ for any $s \in C$ and $\alpha_s^0 = 0$ for other s. Indeed, $\alpha_s^0 = \langle M^{-1}(e_s) \mid e_0\rangle = \langle e_s \mid M(e_0)\rangle = 1/\sqrt{N}$ because M is unitary. Now for $x = \sum_{s\in\mathcal{K}}\mu_s e_s$ we have the following equations:

$$\begin{aligned} D_C(x) &= -MR_0\left(\sum_{s\in\mathcal{K}}\mu_s\sum_{i\in\mathcal{K}}\alpha_s^i e_i\right) \\ &= -M\left(\sum_{s\in\mathcal{K}}\mu_s\sum_{i\in\mathcal{K}}\alpha_s^i e_i - 2\sum_{s\in C} e_0/\sqrt{N}\right) \\ &= -\sum_{s\in\mathcal{K}}\mu_s e_s + \frac{2}{\sqrt{N}}\sum_{s\in C}\mu_s\frac{1}{\sqrt{N}}\sum_{j\in C} e_j \\ &= -\sum_{s\notin C}\mu_s e_s + \sum_{j\in C} e_j\left(\frac{2A}{N} - \mu_j\right). \blacksquare \end{aligned}$$

Corollary 1. Let $C \subset \mathcal{K}$, $|C| = N$, $T \subseteq C$, $|T| = N/4$, and M be C-mixing. Then $-MR_0M^{-1}R_T(\chi_C) = \chi_T$.

It is readily seen that applying $D_C R_T$ for $|T| = N/4$ doubles amplitudes of states of the form χ_C, whereas Grover's algorithm increases them by a constant $O(1/\sqrt{K})$. Note that generally speaking, RDT(C) cannot

be implemented on a quantum computer for an arbitrary subset $C \subset \mathcal{K}$ [12]. In section 4 we show how RDT can be implemented in our peculiar case for databases.

4. Control system with relative diffusion transformation for quantum databases

Let f be a function of the form $\{0,1\}^n \to \{0,1\}^n$. A *presentation* of f is a basic state of the form

$$a_0, f(a_0), a_1, f(a_1), \ldots, a_{N-1}, f(a_{N-1}), \gamma_1, \gamma_2, \ldots,$$

where $a_0, a_1, \ldots, a_{N-1}$ are all different strings from $\{0,1\}^n$ taken in some order, $N = 2^n$.

Values of the ancillary qubits $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_{2nN})$. There are $M = N!$ forms of presentations which differ only in their ancillary qubits, we denote them by $P_0^\gamma, P_1^\gamma, \ldots, P_{M-1}^\gamma$, where P_0 corresponds to the lexicographic order on $\{0,1\}^n$. We use the notation $P_i = a_0^i, f(a_0^i), \ldots$, and omit $\gamma = \bar{0}$. A string $B_a = (a, f(a))$ is called a *block*. Put $\mathcal{M} = \{0, 1, \ldots, M-1\}$.

A *control system* for the database consists of the following two parts.

1. *Preparation of the main state.* This is a unitary transformation

$$P_0 \;\to\; \frac{1}{\sqrt{M}} \sum_{i \in \mathcal{M}} P_i \stackrel{\text{def}}{=} \chi_0.$$

2. *Extracting and restoring procedures.* Given query a an extracting procedure consists of two parts.

   (a) The unitary transformation

$$\text{Ex} \;:\; \chi_0 \to \chi_a \stackrel{\text{def}}{=} \frac{1}{\sqrt{(N-1)!}} \sum_{i \in \zeta(a)} P_i$$

   where $\zeta(a) = \{i \mid a_0^i = a\}$.

   (b) Observation of the first $2n$ qubits.

   This gives required information $a, f(a)$ with certainty and does not change the observed state because χ_a has the form $|a, f(a)\rangle \otimes \chi_a'$. The restoring procedure is $(\text{Ex})^{-1}$. It gives χ_0 and the database is ready for the following query.

We only describe Ex because the main state can be prepared along similar lines. If $a = \sigma_1\sigma_2 \ldots \sigma_{n/2}$ is some query to the database and all $\sigma_i \in \omega$, then C_j denotes the set of all basic states of the form P_j where $a_0^j = \sigma_1\sigma_2 \ldots \sigma_j\delta_{j+1}\delta_{j+2} \ldots \delta_{n/2}$, all $\delta_k \in \omega$, and n is even. Given some RDT(C_j):

D_j, the sequential applications of $D_j R_{C_{j+1}}$ for $j = 1, 2, \ldots, n/2 - 1$ result in χ_a by Corollary 1. Now to complete the construction of Ex it would suffice to implement some C_j-mixing transformation M_j on a quantum computer. We construct M_j in three steps. Note that we cannot apply the Walsh–Hadamard transform here as in [2] because P_j does not exhaust all basic states of $\mathcal{H}$.

Step 1. Given $e_0 = P_0 = B_0, B_1, \ldots, B_{N-1}, 0, 0, \ldots, 0$, we first create the state $\xi = |B_0, \ldots, B_{N-1}\rangle \otimes \chi_{H_0} \otimes \chi_{H_{N-1}} \otimes \chi_{H_{N-2}} \otimes \chi_{H_1}$, where $H_l = \{0, 1, \ldots, l-1\}$ if $l \neq 0$ and $H_0 = \{0, 1, \ldots, 2^{n-2j}\}$. This can be done by independently applying the transformations $|0\rangle \to \chi_{H_l}$ to the ancillary registers which are built as in [13].

Given a pair of sequences $\bar{\imath} = i_0, i_1, \ldots, i_{N-1}$ and $\bar{r} = r_0, r_1, \ldots, r_{N-1}$, where $r_s \in H_{N-s}$ $s = 1, \ldots, N-1$, $r_0 \in H_0$, we define the pair of sequences $k_0, k_1, \ldots, k_s$ and $h^s_{s+1}, \ldots, h^s_{N-1}$ by induction on s, where k_i and h^s_i depend on $\bar{\imath}$ and $\bar{r}$.

Basis, $s = 0$: $k_0 = j_s$, $h^0_1, \ldots, h^0_{N-1}$ is obtained from $\bar{\imath}$ by a cancellation of j_{r_0}.

Step, $s > 0$: All $k_0, \ldots, k_{s-1}$ are already defined. Put $k_s = h^{s-1}_{s+r_s}$, a new sequence $h^s_{s+1}, \ldots, h^s_{N-1}$ is obtained from $h^{s-1}_s, \ldots, h^{s-1}_{N-1}$ by a cancellation of k_s. Denote $0, 1, \ldots, N-1$ by $\bar{1}$ and $0, 0, \ldots, 0$ by $\bar{0}$.

Let $T = 2^{n-2j}$, $j_1 < j_2 < \cdots < j_T$ and let $B_{j_1}, b_{j_2}, \ldots, B_{j_T}$ be all the blocks from C_j.

Step 2. Perform a chain of classical transformations (with unitary matrices containing only ones and zeroes): $\xi \to \xi_0 \to \cdots \xi_{N-1}$, where

$$\xi_s = B_{k_0}, B_{k_1}, \ldots, B_{k_{s-1}}, B_{h^{s-1}_s}, B_{h^{s-1}_{N-1}}, \rho_0, \ldots, \rho_{s-1}, r_s, \ldots, r_{N-1}.$$

(a) The passage $\xi \to \xi_0$ is the replacement of r_0 by the number q such that $i_q = j_{r_0}$, where $i_{r_0} = j_t$. This can be done because the mapping $q \to j_q$ is reversible.

(b) The passage $\xi_s \to \xi_{s+1}$ with $s = 0, 1, \ldots, N-2$ finds the block B_{k_s} and establishes it immediately after $B_{k_{s-1}}$, the order of all other blocks remains unchanged. This can be done because of the definition of k_s by means of classical unitary transformations independent of the contents of the blocks. The replacement $r_s \to \rho_s$ ensures reversibility, that is, this transformation is unitary.

Step 3 (Optional). The transformation $\rho_s(\bar{\imath}, \bar{r}) \to \rho_s(\bar{\imath}, \bar{r}) - \rho_s(\bar{1}, \bar{0})$ with $s = 0, 1, \ldots, N-1$ results in zeroes in the ancillary qubits if an initial state is P_0. Applying these steps to P_0 gives all states from C_j with the same amplitudes, therefore it gives χ_a.

Detailed analysis shows that Steps 1 trough 3 take time $O(N^2 \log^2 N + T(N))$ on a quantum Turing machine where $T(N)$ is the time required

for Step 1 when a precision is fixed. Hence the procedure Ex takes time $O((N^2 \log^2 N + T(N)) \log N)$.

We have described the procedure for extracting $a, f(a)$. The reverse procedure restores the main state of the database. The main state χ_0 can be prepared along similar lines, which takes time $O(N^3 \log^3 N + NT(N) \log N)$.

Note that observation of the first block as described gives $a, f(a)$ only in an ideal case, that is, if the following effects can be neglected.

1. Precision of the transformations is not absolute, especially for the procedures in Step 1.
2. The presence of noise, spontaneous transformations of the forms: $0 \to 1$, $0 \to -0$, which touch a sufficiently small part of each block B_i.
3. Unauthorized actions that are aimed at learning a value $f(b)$ when the control system works at a query $a \neq b$.

Section 5 deals with point 3 and in section 6 we briefly run through point 2.

5. Protection of information against unauthorized actions

We presume that the aim of unauthorized actions is to learn $f(b)$ for $b \neq a$ with high probability p. Suppose that some blocks g are inaccessible for these actions. The first block, where the control system observes the result, is among them. To learn $f(b)$ would require dealing with Np blocks of memory. This is because the value $f(b)$ is distributed among all of the blocks except the first with the same probability at any instant of time.

We shall regard the following scenario. Let somebody S (say, spy) be equipped with a quantum computer having its own memory. The database is preparing to answer on a query a. When the database is in state χ_{C_j} S does the following.

1. Observes any Np accessible blocks of the database at one instant of time.
2. Performs unitary transformations with an accessible part of the database and the memory of the intruding computer with the aim of covering up all traces of the observation.

After that the control system continues its work as usual. Denote by P_{ex} the probability that the control system observing the first block will not receive a word of the form a, A, thus exposing S.

Theorem 1. There exists a function $\alpha(g, N)$ such that $\forall \varepsilon > 0 \exists g$: $\alpha(g, N) > 1 - \varepsilon$, $N = 1, 2, \ldots$ with the following property. For every choice of a block observed by S and the unitary transformations

$$P_{\text{ex}} \geq p\alpha.$$

Sketch of the proof. Memory of the computer used by S can be considered as an ancillary part of memory in the database.

We use χ_i, R_i instead of χ_{C_i}, R_{C_i}. Denote by Q_0 a state of the computer after an unauthorized action with the state χ_j. Then the control system performs the transformations $D_{i+j}R_{i+j+1}$, $i = 1, 2, \ldots, t-j-1$, and $t = n/2$ sequentially. Denote by Q_{i+1} the results: $Q_{i+1} = D_{i+j}R_{i+j+1}(Q_i)$. Put $\varepsilon = \langle Q_0 \mid \chi_j \rangle$. Because of the unitarity of all transformations at hand $\forall i = 1, \ldots, t-j-1$ $\langle Q_i \mid \chi_{i+j} \rangle = \varepsilon$. Denote by S_{suc} the set of basic states with the first block of the form a, A for some A. For any final state Q_{t-j-1} the probability to expose S is $1 - \sum_{e \in S_{\text{suc}}} |\langle e \mid Q_{t-j-1} \rangle|^2$. We have

$$1 - P_{\text{ex}} = pP_1 + (1-p)P_2,$$

where P_1 (P_2) is the probability that the control system receives a, A on condition that a block $a, f(a)$ was observed by S (was not observed by S respectively).

Case 1. A block $a, f(a)$ was observed by S.

Let $L_i = (N-1)!2^{t-(i+j)}$ be the cardinality of C_i. Denote by q_i^{av} the average amplitude of all basic states from C_{i+j} when the database is in state Q_i. We thus have $|q_i^{\text{av}}| \le |\varepsilon|/\sqrt{L_i}$. Let δ_{norm} and δ_{S} denote an absolute growth of average amplitudes of basic C_t-states in cases without S and with S respectively. It follows from Lemma 1 that $\delta_{\text{norm}} \ge \varepsilon\delta_{\text{S}}$ and in state Q_{t-j-1} all basic states from S_{suc} with nonzero amplitudes are contained in C_t. Therefore $P_1 \le |\varepsilon|^2$.

Case 2. A block $a, f(a)$ was not observed by S.

Here we roughly estimate $P_2 \le 1$.

Joining these cases we conclude that $1 - P_{\text{ex}} \ge p\varepsilon^2 + 1 - p$. Finally, in view of the assumed conditions, ε can be estimated as $|\varepsilon| \le 2(1-p)^g$. ■

6. Error correcting procedure for the database

Random errors in the database are transformations on the basic states induced by changes of qubit values of the form $0 \to 1$ or *vice versa* and changes of phases $0 \to -0$ or $1 \to -1$ that affect only a small number of the qubits in each block of memory. Note that the phase errors $0 \to -1$, $1 \to -1$ can be reduced to the change of values as shown in [14]. Error correcting code (ECC) is the conventional tool to correct errors of such a sort. Let each block contain n qubits.

Encoding is an injection of the form $E : \{0,1\}^n \to \{0,1\}^{n_1}$, where $n_1 > n$. If $w_n(A) = \sum_{i=1}^n a_i$ is the Hamming weight of the word $A = a_1 a_2 \ldots a_n \in \{0,1\}^n$, the distance between two such words A, B is $d_n(A,B) = w_n(A \oplus B)$ where $\oplus$ denotes a bitwise addition modulo 2. Put $d(E) = \min_{A,B \in \{0,1\}^n} d_{n_1}(E(A), E(B))$.

If for some $A', B' \in \{0,1\}^{n_1}$ $B' \in \mathrm{Im}(E)$, $d_{n_1}(A', B') < d(E)/2$ then such B' are defined for A' uniquely and we obtain the partial functions $A' \to B' \to E^{-1}(B') = A \in \{0,1\}^n$. Their superposition $\mathcal{D}: A' \to A$ is called a *decoding procedure* for encoding E. $\mathcal{D}$ corrects $\leq d(E)/2$ errors that occur in encoding words B'. This procedure is essentially classical because the mapping $A' \to B'$ is not reversible. But if we use additional registers that consist of ancillary qubits and denote by γ their contents we can regard a reversible function $A' \to B'$, $\gamma(A') \to E^{-1}(B'), \gamma(A')$ instead of the classical decoding and perform this procedure on a quantum computer. Simple and convenient quantum linear codes are also proposed in [14].

ECC can be used during computations to correct errors which occur randomly as the result of noise. The size of the ancillary register thus increases with longer computation times. In [15] error correcting procedures are presented which correct errors repeatedly with a constant rate during the course of computation and requires a size of ancillary registers polylogarithmical on the time of computation. This error correcting procedure can be applied to our database which results in basic states of the form $E(e_i), \gamma_i$ instead of e_i considered previously, here all properties of the database will remain unchanged.

7. Acknowledgments

I am grateful to Victor Maslov for his attention and help, to Peter Hoyer for his comments and criticism, and to Lov Grover for his attention to my work.

References

[1] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *Society for Industrial and Applied Mathematics Journal on Computing*, **26** (1997) 1484–1509.

[2] L. K. Grover, "A Fast Qantum Mechanical Algorithm for Database Search," in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, 1996, Philadelphia PA, USA, 212–219.

[3] D. Deutsch, "Quantum Theory, the Church–Turing Principle and the Universal Quantum Computer," *Proceedings of the Royal Society,* London, A, **400** (1985) 97–117.

[4] E. Bernstein and U. Vazirani, "Quantum Complexity Theory," *Society for Industrial and Applied Mathematics Journal on Computing*, **26** (1997) 1411–1473.

[5] A. Yao, "Quantum Circuit Complexity," in *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*, 1993, pp. 352–361.

[6] J. Watrous "On One-Dimensional Quantum Cellular Automata," in *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, 1995.

[7] A. Berthiaume, G. Brassard, "Oracle Quantum Computing," *Journal of Modern Optics*, **41** (1994) 2521–2535.

[8] D. Deutsch, R. Jozsa, "Rapid Solution of Problems by Quantum Computation," *Proceedings of the Royal Society*, London, A, **439** (1992) 553–558.

[9] Y. Ozhigov, "Quantum Computers Speed Up Classical with Probability Zero," *Chaos, Solitons, and Fractals*, **10** (1999) 1707–1714; LANL e-print quant-ph/9803064.

[10] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, "Strengths and Weaknesses of Quantum Computation," *Society for Industrial and Applied Mathematics Journal on Computing*, **26** (1997) 1510–1523; LANL e-print quant-ph/9701001.

[11] M. Boyer, G. Brassard, P. Hoyer, and A. Tapp, "Tight Bounds on Quantum Searching," *Fourth Workshop on Physics and Computation*, Boston University, November 22–24, 1996; LANL e-print quant-ph/9605034.

[12] Y. Ozhigov, "About Quantum Mechanical Speeding Up of Classical Algorithms," LANL e-print quant-ph/9706003.

[13] A. Kitaev, "Quantum Measurements and the Abelian Stabilizer Problem," LANL e-print quant-ph/9511026.

[14] A. R. Calderbank and P. W. Shor, "Good Quantum Error-correcting Codes Exist," *Physical Review*, **54** (1996) 1098–1105.

[15] D. Aharonov and M. Ben-Or, "Fault-Tolerant Quantum Computation with Constant Error," in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, 1997.