KEY AGREEMENT SCHEME BASED ON QUANTUM NEURAL NETWORKS

Nguyen Nam Hai*

Abstract: *In quantum cryptography, the key is created during the process of key distribution, where as in classical key distribution a predetermined key is transmitted to the legitimate user. The most important contribution of quantum key distribution is the detection of eavesdropping. The purpose of this paper is to introduce an application of QNNs in construction of key distribution protocol in which two networks exchange their outputs (in qubits) and the key to be synchronized between two communicating parties. This system is based on multilayer qubit QNNs trained with back-propagation algorithm.*

Keywords: Neural networks, Quantum neural networks, Cryptography.

1. INTRODUCTION

In cryptography, key is the most important parameter that determines the functional output of a cryptographic algorithm. For encryption algorithms, a key specifies the transformation of plaintext into cipher text, and vice versa for decryption algorithms. Keys also specify transformations in other cryptographic algorithms, such as digital signature schemes and message authentication codes. The security of cryptosystems based on encryption keys. In the network information era, one of the most interesting problems is keys transformation that ensures the privacy of them. It is important to structure group key agreement schemes which are designed to provide a set of players, and communicating over a public network with a session key to be used to implement secure multicast sessions, e.g., video conferencing, collaborative computation, file sharing via internet, secure group chat, group purchase of encrypted content and so on.

A key-agreement protocol or key agreement scheme is a protocol whereby two or more parties can agree on a key in such a way that both influence the outcome. If properly done, this precludes undesired third parties from forcing a key choice on the agreeing parties. Protocols that are useful in practice also do not reveal to any eavesdropping party what key has been agreed upon.

Many key exchange systems have one party generate the key, and simply send that key to the other party - the other party has no influence on the key. Using a key-agreement protocol avoids some of the key distribution problems associated with such systems. Protocols where both parties influence the final derived key are the only way to implement perfect forward secrecy. The first publicly known public key agreement protocol that meets the above criteria was the Diffie - Hellman key exchange, in which two parties jointly exponentiate a generator with random numbers, in such a way that an eavesdropper cannot feasibly determine what the resultant value used to produce a shared key is.

Exponential key exchange in and of itself does not specify any prior agreement or subsequent authentication between the participants. It has thus been described as an anonymous key agreement protocol.

Many key agreement protocol use public key cryptosystems to encrypt and send the key via public channel. But, with the development of quantum computation, many public key cryptosystems are not secure [10]. In quantum cryptography, the key is created during the process of key distribution, where as in classical key distribution a predetermined key is transmitted to the legitimate user. The most important contribution of quantum key distribution is the detection of eavesdropping.

In this paper, we introduce a key agreement scheme based on quantum neural network that can ensure the security of the key exchange via public channel. In section 2, we introduce some knowledge about the quantum neural network. Section 3 presents our contributions about the key agreement scheme based on quantum neural network. Section 4, we provide the analysis of our proposed scheme. Section 5 is conclusion.

2. MODELING DETERMINING THE PARAMETERS OF MATERIAL

Quantum Computation

At the beginning of the twentieth century, most people believed that physical phenomena in nature were subject to the laws of Newton and Maxwell. However, in the 1930s, when experiments on subatomic objects were scrutinized, it was found that the laws of classical physics of Newton and Maxwell were no longer valid. Since then a mathematical model for the new physics was called *quantum mechanics* and new theories about quantum physics were developed. Quantum physics includes theoretical physics of quantum electrodynamics and quantum field theory. The idea of computers in terms of physical objects and calculations made on physical processes is of interest and research by some notable scientists such as Richard Feyman and David Deutsch. In [4], Feyman introduces the theory of physical phenomena emulation on computers based on quantum physics principles, and calculations on quantum aspects. In [5], Deutsch explains the basic concepts of Quantum Turing Machines (QTM) and Universal Quantum Computing. Quantum computers build on the principle of quantum phenomena, such as overlapping and quantum entanglements, in order to perform calculations. Electronic calculators usually perform calculations based on pure mathematical logic on computational units, which are bits that receive values 0 and 1 and after each calculation step there is a primary measured value in the form of 0 or 1, but not both. Quantum computers based on computational units are quantum bits related to quantum states. Quantum computing makes direct use of quantum-mechanical phenomena, such as superposition and entanglement, to perform operations on data [4]. Quantum computers are different from binary digital electronic computers based on transistors. Whereas common digital computing requires that the data be encoded into binary digits (bits), each of which is always in one of two definite states (0 or 1), quantum computation uses quantum bits, which can be in superposition of states. A quantum Turing machine is a theoretical model of such a computer, and is also known as the universal quantum computer. The field of quantum computing was initiated by the work of Paul Benioff [2] and Yuri Manin [3], Richard Feynman [4] and David Deutsch [5]. As of 2017, the

development of actual quantum computers is still in its infancy, but experiments have been carried out in which quantum computational operations were executed on a very small number of quantum bits [7]. Both practical and theoretical research continues, and many national governments and military agencies are funding quantum computing research in an effort to develop quantum computers for civilian, business, trade, environmental and national security purposes, such as cryptanalysis [8].

Large-scale quantum computers would theoretically be able to solve certain problems much quicker than any classical computers that use even the best currently known algorithms, like integer factorization using Shor's algorithm or the simulation of quantum many-body systems. There exist quantum algorithms, such as Simon's algorithm, that run faster than any possible probabilistic classical algorithm [9]. A classical computer could in principle (with exponential resources) simulate a quantum algorithm, as quantum computation does not violate the Church - Turing thesis [10]. On the other hand, quantum computers may be able to efficiently solve problems which are not practically feasible on classical computers. A quantum computer maintains a sequence of qubits. A single qubit can represent a one, a zero, or any quantum superposition of those two qubit states; a pair of qubits can be in any quantum superposition of 4 states and three qubits in any superposition of 8 states. In general, a quantum computer with n qubits can be in an arbitrary superposition of up to 2^n different states simultaneously (this compares to a normal computer that can only be in one of these 2^n states at any one time). A quantum computer operates by setting the qubits in a perfect drift that represents the problem at hand and by manipulating those qubits with a fixed sequence of quantum logic gates. The sequence of gates to be applied is called a quantum algorithm. The calculation ends with a measurement, collapsing the system of qubits into one of the 2^n pure states, where each qubit is zero or one, decomposing into a classical state. The outcome can therefore be at most n classical bits of information. Quantum algorithms are often probabilistic, in that they provide the correct solution only with a certain known probability.

A quantum computer with a given number of qubits is fundamentally different from a classical computer composed of the same number of classical bits. For example, representing the state of an n-qubit system on a classical computer requires the storage of 2^n complex coefficients, while to characterize the state of a classical n-bit system it is sufficient to provide the values of the n bits, that is, only n numbers. Although this fact may seem to indicate that qubits can hold exponentially more information than their classical counterparts, care must be taken not to overlook the fact that the qubits are only in a probabilistic superposition of all of their states. This means that when the final state of the qubits is measured, they will only be found in one of the possible configurations they were in before the measurement. It is in general incorrect to think of a system of qubits as being in one particular state before the measurement, since the fact that they were in a superposition of states before the measurement was made directly affects the possible outcomes of the computation.

Qubit

The qubit is a two-state quantum system. It is typically realized by an atom, with an electronic spin with its up state and down one, or a photon with its two polarization states. These two states of a qubit are represented by the computational basis vectors |0⟩ and |1⟩ in a two-dimensional Hilbert space.

$$\left|0\right\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{1}$$

and

$$\left|1\right\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2}$$

An arbitrary qubit state |φ⟩ maintains a coherent superposition of the basis states |0⟩ and |1⟩ according to the expression:

$$\left|\varphi\right\rangle = c_0\left|0\right\rangle + c_1\left|1\right\rangle; \left|c_0\right|^2 + \left|c_1\right|^2 = 1 \tag{3}$$

where c_0 and c_1 are complex numbers called the probability amplitudes. When one observes the |φ⟩, this qubit state |φ⟩ collapses into either the |0⟩ state with the probability $|c_0|^2$, or the |1⟩ state with the probability $|c_1|^2$. These complex-valued probability amplitudes have four real numbers; one of these is fixed by the normalization condition. Then, the qubit state (3) can be written by:

$$\left|\varphi\right\rangle = e^{i\lambda}(\cos\theta\left|0\right\rangle + e^{i\chi}\sin\theta\left|1\right\rangle), \tag{4}$$

where λ, χ, and θ *are real-valued parameters.* The global phase parameter λ usually lacks its importance and consequently the state of a qubit can be determined by the two phase parameters χ and θ:

$$\left|\varphi\right\rangle = (\cos\theta\left|0\right\rangle + e^{i\chi}\sin\theta\left|1\right\rangle) \tag{5}$$

Thus, the qubit can store the value 0 and 1 in parallel so that it carries much richer information than the classical bit. The states |0> and |1> are the basis state; the combinations of them are called *superpositions*.

Linear superposition is closely related to the familiar mathematical principle of linear combination of vectors. Quantum systems are described by a wave function ψ that exists in a Hilbert space. The Hilbert space has a set of states, $|\varphi_i>$, that form a basis, and the system is described by a quantum state. A postulate of quantum mechanics is that if a coherent system interacts in any way with its environment (by being measured, for example), the superposition is destroyed. This loss of coherence is governed by the wave function ψ. The coefficients *ci* are called probability amplitudes, and $|c_i|^2$ gives the probability of $|\psi>$ being measured in the state $|\varphi_i>$. Note that the wave function ψ describes a real physical system that must collapse to exactly one basis state. Therefore, the probabilities governed by the amplitudes *ci* must sum to unity. A two state quantum system is used as the basic unit of quantum computation. Such a system is referred to as a quantum bit or qubit and naming the two states |0> and |1>, it is easy to see why this is so.

Interference is a familiar wave phenomenon. Wave peaks that are in phase interfere constructively while those that are out of phase interfere destructively. This is a phenomenon common to all kinds of wave mechanics from water waves to optics. The well-known double slit experiment demonstrates empirically that at the quantum level interference also applies to the probability waves of quantum mechanics. The wave function interferes with itself through the action of an operator the different parts of the wave function interfere constructively or destructively according to their relative phases just like any other kind of wave.

Entanglement is the potential for quantum systems to exhibit correlations that cannot be accounted for classically. From a computational standpoint, entanglement seems intuitive enough it is simply the fact that correlations can exist between different qubits for example if one qubit is in the |1> state, another will be in the |1> state. However, from a physical standpoint, entanglement is little understood. The questions of what exactly it is and how it works are still not resolved. What makes it so powerful (and so little understood) is the fact that since quantum states exist as superposition, these correlations exist in superposition as well. When coherence is lost, the proper correlation is somehow communicated between the qubits, and it is this communication that is the crux of entanglement. Mathematically, entanglement may be described using the density matrix formalism. The density matrix ρ_ψ of a quantum state $|\psi>$ is defined as $\rho_\psi = |\psi_i h_\psi|$.

No-Cloning Theorem The most common function with digital media is copying. This cannot be done in quantum information theory.

Theorem 1.1. (Wootters and Zurek [27], Dieks [28]) An unknown quantum system cannot be cloned by unitary transformations.

Proof. Suppose there would exist a unitary transformation U that makes a clone of a quantum system. Namely, suppose U acts, for any state $|\varphi\rangle$, as

$$U : \left|\varphi 0\right\rangle \to \left|\varphi\varphi\right\rangle$$

Let $|\varphi\rangle$ and $|\phi\rangle$ be two states that are linearly independent. Then we should have $U|\varphi 0\rangle \to |\varphi\varphi\rangle$ and $U|\phi 0\rangle \to |\phi\phi\rangle$ by definition. Then the action of U on $|\psi\rangle = \frac{1}{\sqrt{2}}(|\varphi\rangle + |\phi\rangle)$ yields,

$$\mathrm{U}|\psi 0\rangle = \frac{1}{\sqrt{2}}(\mathrm{U}|\varphi 0\rangle + U|\phi 0\rangle) = \frac{1}{\sqrt{2}}(\mathrm{U}|\varphi\varphi\rangle + U|\phi\phi\rangle) \cdot$$

If U were a cloning transformation, we must also have

$$\mathrm{U}|\psi 0\rangle = |\psi\psi\rangle = \frac{1}{2}(|\varphi\varphi\rangle + |\varphi\phi\rangle + |\phi\varphi\rangle + |\phi\phi\rangle),$$

which contradicts the previous result. Therefore, there does not exist a unitary cloning transformation.

Clearly, there is no way to clone a state by measurements. A measurement is probabilistic and non-unitary, and it gets rid of the component of the state which is in the orthogonal complement of the observed subspace.

Quantum Gates

In quantum computing, the logical operations are realized by reversible, unitary transformations on qubit states. Here, we denote the symbols for the logical universal operations, i.e., the single-qubit rotation gate U_θ shown in Figure 1 and the two-qubit controlled NOT gate U_{CNOT} 2 qubit shown in Figure 2.

First we sketch the single-qubit rotation gate U_θ. We can represent the computational basis vectors $|0\rangle$ and $|1\rangle$ as vectors in a two- dimensional Hilbert space as follows:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{6}$$

In such a case we have the representation of $|\varphi\rangle = (\cos\theta_i|0\rangle + e^{i\chi}\sin\theta_i|1\rangle)$ and the matrix representation of U_θ operation can be described:

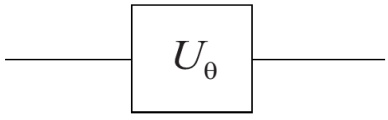
$$U_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{7}$$

This gate changes the phase of the probability amplitudes from θ_i to $\theta_i + \theta$ as follows:

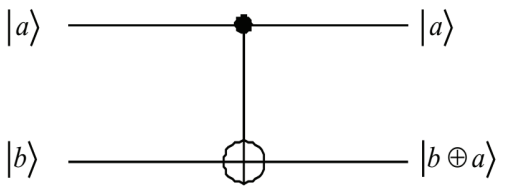
$$\begin{aligned} |\varphi'\rangle = U_\theta|\varphi\rangle &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} \cos\theta_i \\ \sin\theta_i \end{pmatrix} \\ &= \begin{pmatrix} \cos\theta\cos\theta_i - \sin\theta\sin\theta_i \\ \sin\theta\cos\theta_i + \cos\theta\sin\theta_i \end{pmatrix} = \begin{pmatrix} \cos(\theta_i + \theta) \\ \sin(\theta_i + \theta) \end{pmatrix} \end{aligned} \tag{8}$$

From Figure 2 we see the UCNOT gate operates on two-qubit states. These are states of the form $|a\rangle\otimes|b\rangle$ or simply $|ab\rangle$, a tensor product of two vectors $|a\rangle$ and $|b\rangle$. It is usual to represent these states as follows:

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \tag{9}$$



Figures 1. *Single-qubit rotation gate.*



Figures 2. *The two-qubit controlled NOT gate (⊕: XOR).*

This standard representation is one of several important bases in quantum computing. When the U_{CNOT} gate works on these two-qubit states as vectors (9) in

a four-dimensional Hilbert space, the matrix representation of the U_{CNOT} operation can be described by:

$$U_{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{10}$$

This controlled NOT gate has a resemblance to a XOR logic gate that has |a⟩ and |b⟩ inputs. As shown in Figure 4, this gate operation regards the |a⟩ as the control and the |b⟩ as the target. If the control qubit is |0⟩, then nothing happens to the target one. If the control qubit is |1⟩, then the NOT matrix is applied to the target one. That is, |ab⟩ → |a, b ⊕ a⟩. The symbol ⊕ indicates the XOR operation.

An arbitrary quantum logical gate or quantum circuit is able to be constructed by these universal gates.

Complex-valued description of qubit neuron state

Our qubit neuron model is a neuron model inspired by the quantum logic gate functions: its neuron states are connected to qubit states, and its transitions between neuron states are based on the operations derived from the two quantum logic gates. To make the connection between the neuron states and the qubit states, we assume that the state of a firing neuron is defined as a qubit basis state |1⟩, the state of a non-firing neuron is defined as a qubit basis state |0⟩ and the state of an arbitrary qubit neuron is the coherent superposition of the two:

$$neuronstate = \alpha|0\rangle + \beta|1\rangle; |\alpha|^2 + |\beta|^2 = 1 \tag{11}$$

corresponding to Equation (3). In this qubit-like description, the ratio of firing and non-firing states is represented by the probability amplitudes α and β. These amplitudes are generally complex-valued. We, however, consider the following state, which is a special case of Equation (5) with $\chi = 0$.

$$neuronstate = \cos\theta|0\rangle + \sin\theta|1\rangle \tag{12}$$

as a qubit neuron state in order to give the complex-valued representation of the functions of the single-qubit rotation gate U_θ and the two-qubit controlled NOT gate U_{CNOT}. We introduce the following expression instead of Equation (12):

$$f(\theta) = \cos\theta + i\sin\theta = e^{i\theta}, \tag{13}$$

where i is the imaginary unit and θ is defined as the quantum phase. The complex-valued description (13) can express the corresponding functions to the operations of the rotation gate and the controlled NOT gate.

Phase rotation operation as a counterpart of U_θ

The rotation gate is a phase shifting gate that transforms the phase of qubit neuron state. Since the qubit neuron state is represented by Equation (13), the following relation holds:

$$f(\theta_1 + \theta_2) = f(\theta_1) f(\theta_2) \tag{14}$$

Phase reverse operation as a counterpart of U_{CNOT}

This operation is defined with respect to the controlled input parameter γ corresponding to the control qubit as follows:

$$f(\frac{\pi}{2}\gamma - \theta) = \begin{cases} \cos\theta - i\sin\theta & (\gamma = 0) \\ \sin\theta + i\cos\theta & (\gamma = 1) \end{cases} \tag{15}$$

Cryptography system based on neural network

Cryptography system based on neural network structure (Neural cryptography) is based on the synchronization between two neural networks when mutual learning (Ruttor et al. 2006). In each step of this process we receive a sample of the input signals and compute the output values.

Then, both neural networks use the output provided by the other network to adjust their weights. This process synchronizes the weight vector. The synchronization of the neural network is a complex process. The weights of the networks in each implementation step are based on a random walk and a probabilistic selection. Two objects A and B want to exchange a secret message over a public channel. To protect the contents of the message against the attacker T from eavesdrop the traffic, A encrypt the message, and B needs to know the secret key that transmitted over a public channel.

This can be achieved by synchronizing data between two machines, one for A and one for B, respectively. After the synchronization, the system will generate a random bit string to check. When any different network is trained on this bit sequence, it cannot extract information based on statistical properties of the chain.

Artificial neural networks are used to construct an effective encryption system to secure key exchange. Neural network structure is an important parameter, because it depends on the purpose of the system. Normally, we usually use multi-layer neural network structure. Neural network provides an extremely strong and popular framework based on nonlinear mappings that compute many different output parameters from many different input parameters. The process of determining the values of these parameters on a provided data set called learning, or training, and the data is often called the training data set. Neural network can be considered an appropriate choice for the encryption and decryption functions.

Two identified systems, derived from different starting conditions, can be synchronized by an identical external signal. Two synchronized networks based on mutual training the weight over time independently. This phenomenon also applies in cryptography. Neural network learns from the input samples. A "teacher" network will perform the first pair of input/output data and the "student" network will be trained based on this data. After the training process, "student" can generalize: it can sort - with a probability - an input without depending on the training set. In this case, A and B do not need to share a secret key for decryption. In the case an attacker neural network E knows all the details of the algorithm and

traffic logs through the channels also cannot synchronize themselves with the object being attacked and thus difficult to calculate the secret key. We assume that the attacker E knows about the algorithms, the input vector sequence and output bit sequence but do not know the network structure. Attackers from the initial weight vector compute weighted vectors based on the input and output sequence. All starting positions are oriented to a final state vector, the only key. However this is proved to be impossible in computational implementation when do not use synchronization process by mutual learning.

Quantum neural network

Classical neuron model

The well-known real-valued conventional neuron model is expressed by the equations:

$$u = \sum_{m=1}^{M} w_m x_m - v, \tag{16}$$

$$y = (1 + e^{-u})^{-1} \tag{17}$$

where, u is the internal state of a neuron y. x_m is the neuron state of the m-th neuron as one of M inputs to y. w_m and v are the weight connection between x_m to y and the threshold value, respectively. These neuron parameters are real numbers.

The complex-valued neuron model is like a real-valued one except that the neuron parameters are extended to the complex numbers Wl as the weight, Xl and Yl as the neuron state, V as the threshold and so on giving rise to the following equations that correspond to Equation (16), (17):

$$U = \sum_{i}^{L} W_l X_l - V, \tag{18}$$

$$Y = (1 + e^{-\mathrm{Re}(U)})^{-1} + i(1 + e^{-\mathrm{Im}(U)})^{-1} \tag{19}$$

Quantum neuron model

We have to observe the transition of the state of the qubit neuron in terms of the unitary transformation as the qubit concept is used for the description of the neuron state. A certain unitary transformation can be realized by the combination of the single-qubit rotation gate U_θ and the two-qubit controlled NOT gate U_{CNOT} corresponding to Equation (16), (17) (or (18), (19)). In this case, the output state of qubit neuron has to be also described by Equation (13). To implement this scheme, we assume the following: we replace the classical neuron weight parameter w_l (or W_l) with the phase rotation operation $f(\theta_l)$ as a counterpart of U_θ and install the phase reverse operation as a counterpart of U_{CNOT} instead of using the non-linear function in Equation (17) (or (19)), and then we consider the following equations:

$$u = \sum_{l}^{L} f(\theta_l) x_l - f(\lambda) = \sum_{l}^{L} f(\theta_l) f(\gamma_l) - f(\lambda), \tag{20}$$

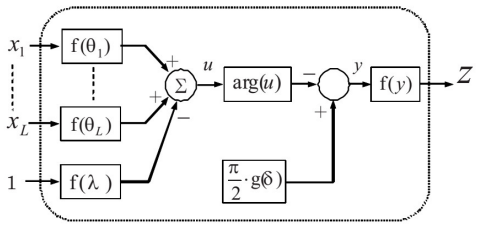
$$y = \frac{\pi}{2} g(\delta) - \arg(u), \tag{21}$$

$$z = f(y) \tag{22}$$

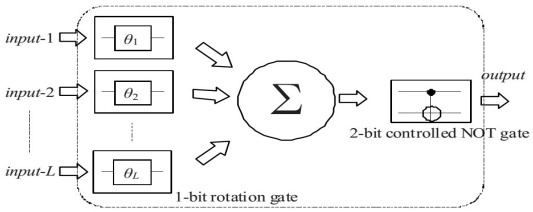
Here, u is the internal state of a quantum neuron z. x_l is the quantum neuron state of the l-th neuron as one of inputs from L other qubit neurons to z. θ_l and λ are the phases regarded as the weight connecting x_l to z and the threshold value, respectively. y and y_l are the quantum phases of z and x_l, respectively. f is the same function as defined in Equation (13) and g is the sigmoid function with the range (0,1):

$$g(\delta) = \frac{1}{1+e^{-\delta}} \tag{23}$$

Two kinds of parameters exist in this neuron model: phase parameters in the form of weight connection θ_l and threshold λ and the reversal parameter δ in Equation (23). The phase parameters correspond to the phase of the rotation gate, and the reversal parameter to the controlled NOT gate. By substituting $\gamma=g(\delta)$ in Equation (15), we obtain the neuron model as shown in Figure 3:



Figures 3. *Quantum neuron model.*



Figures 4. *Quantum gate diagram of quantum neuron.*

Quantum neural network

Now we proceed to construct the multi-layered neural network employing quantum neurons called "quantum neural network".

As shown in Figure 5, QNN has the three sets of neuron elements: $\{I_l\}$ ($l=1,2,...L$), $\{H_m\}$ ($m=1,2,...M$) and $\{O_n\}$ ($n=1,2,...N$), whereby the variables I,H,O indicate the Input, Hidden, and Output layers, and L,M,N are the numbers of neurons in the input, hidden and output layers, respectively. We denote this structure of the three-layered NN by the numbers of L-M-N.

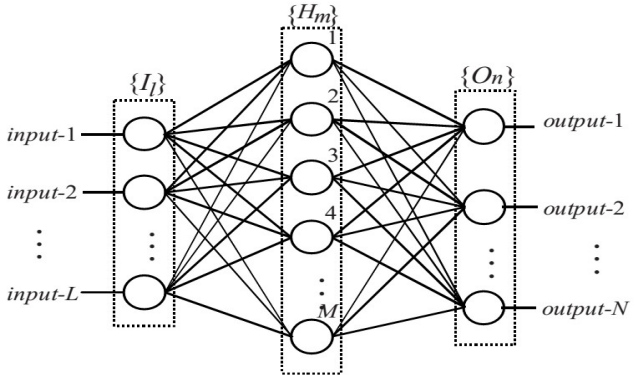
When input data (denoted by $input_l$) is fed into the network, the input layer consisting of the neurons in $\{I_l\}$ converts input values into quantum states with phase values in the range $[0,\pi/2]$.

The output of input neuron I_l becomes the input to the hidden layer:

$$z_l^I = f(\frac{\pi}{2} input_1) \quad 24)$$

The hidden and output layers' neurons respectively, obey Equation (20), (21), (22). We obtain the output to the network, denoted by $output_n$, by calculating the probability for the basic state $|1\rangle$ in the n-th neuron state z_n^O in the output layer:

$$output_n = \left|\mathrm{Im}(z_n^0)\right|^2 \quad (25)$$



Figures 5. *Three layered neural network.*

This output definition is based on the probabilistic interpretation in the way of applying quantum computing to neural network.

Quantum modified back propagation learning

Next, we define a quantum version of the well-known Back Propagation algorithm in order to incorporate learning process in QNN. The gradient-descent method, often used in the Back Propagation algorithm, is employed as the learning rule. This rule is expressed by the following equations:

$$\theta_l^{new} = \theta_l^{old} - \eta \frac{\partial E_{total}}{\partial \theta_l} \quad (26)$$

$$\lambda^{new} = \lambda^{old} - \eta \frac{\partial E_{total}}{\partial \lambda} \quad (27)$$

$$\delta^{new} = \delta^{old} - \eta \frac{\partial E_{total}}{\partial \delta} \quad (28)$$

where η is a leaning rate. E_{total} is the squared error function defined by:

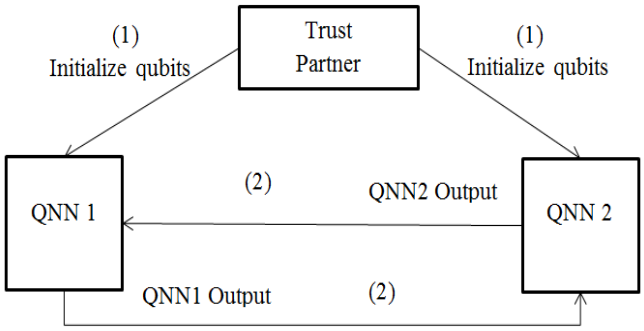
$$E_{total} = \frac{1}{2} \sum_p^N \sum_n^N (t_{p,n} - output_{p,n})^2 \quad (29)$$

This quantity is the cost function to be minimized as part of the learning process. Here, P is the number of learning patterns, $t_{(p,n)}$ is the target signal for the n-th neuron and $output_{(p,n)}$ means $output_n$ of the network when it learns the p-th pattern.

4. KEY AGREEMENT SCHEME BASED ON QUANTUM NEURAL NETWORKS

Key agreement scheme based on quantum neural network we built here based on the synchronization between the two multilayer quantum neural networks described above. The input bit sequence is converted to the qubit format before performing mutual training two networks. We use the quantum neural network to synchronize the key, the adaptive parameters of both neural networks were used as the key and the network was trained by using the back-propagation algorithm. The topology of each network based on our training set. Each network is trained based on qubits of data representation. The key here is the adaptive parameters of network including topology structure and parameters (weight of phase). The number of output neurons will be equal to the number of input neurons, and the number of the hidden layer neurons is chosen arbitrarily.

Key agreement scheme can be described as Figure 8.



Figures 8. *Key agreement scheme.*

In this key agreement scheme, a trust partner sends an initialize qubits sequence to A (QNN1) and B (QNN2). After calculating the output states QNN1 sends the output to QNN2 and QNN2 simultaneously sends his output to the QNN1. Then, QNN1 and QNN2 synchronize with each other to get the same parameters of quantum neural networks. At the end, QNN1 and QNN2 shared the same key as their parameters after synchronization training phase.

5. SECURITY ANALYSIS

The security of this scheme is unconditional, because follow the ANN property the attacker cannot recover the key without synchronization with one of the target partners. In addition, with quantum property, the attacker cannot get the extract qubit in the traffic between sender and receiver based on the no-cloning theorem.

With the scheme above, we construct a simulation model for two quantum neural networks. Each network has three-layer that fully connected. Parameter values of both QNNs in our experimental study are the following:

- Each input layer consists of 8 nodes, which represents the 8-qubit blocks;
- Each hidden layer consists of 8 nodes;

- Each output layer consists of 8 nodes, used to define the decrypted output message;
- The activation function is a sigmoid function;

After simulating this quantum neural network we have generated an initialize qubit sequence and send it to QNN1 and QNN2. After the first adapted quantum neural network, the output send to the QNN2, he received this output, and try to synchronization process with the QNN1. We found that:

- The quantum neural network works reliably and absolutely no errors in the outputs;
- The quantum neural network can synchronize the key as the parameters with each other.

This model presents an attempt to design an encryption system based on quantum artificial neural networks of the backpropagation type. The simulation results of the proposed QNN have shown very good results.

6. CONCLUSION

The paper gave the study of model neurons cryptography, quantum neural network model and applications to propose a key agreement scheme based on two multilayer quantum neural networks between sender and receiver trained by mutual learning with back-propagation algorithm. The result of the paper is the basis for the construction of research on neural network applications in quantum cryptography. In the future the authors will use this study to build other cryptographic systems based on quantum neural network structure, quantum neural network used in the construction of key exchange protocols, and authentication.

REFERENCES

[1]. Gershenfeld, Neil; Chuang, Isaac L. (June 1998). *"Quantum Computing with Molecules",* (PDF). Scientific American.

[2]. Benioff, Paul (1980). *"The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines"*. Journal of statistical physics. 22 (5): 563–591. Bibcode:1980JSP....22..563B. doi:10.1007/BF01011339.

[3]. Manin, Yu. I. (1980). *"Vychislimoe i nevychislimoe [Computable and Noncomputable]"* (in Russian). Sov.Radio. pp. 13–15. Retrieved 2013-03-04.

[4]. Feynman, R. P.u (1982). *"Simulating physics with computers"*. International Journal of Theoretical Physics. 21 (6): 467–488. Bibcode:1982IJTP...21..467F. doi:10.1007/BF02650179.

[5]. Deutsch, David (1985). *"Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer"*. Proceedings of the Royal Society of London A. 400 (1818): 97–117. Bibcode:1985RSPSA.400...97D. doi:10.1098/rspa.1985.0070.

[6]. Finkelstein, David (1968). *"Space-Time Structure in High Energy Interactions"*. In Gudehus, T.; Kaiser, G. Fundamental Interactions at High Energy. New York: Gordon & Breach.

[7]. Gershon, Eric (2013-01-14). *"New qubit control bodes well for future of quantum computing"*. Phys.org. Retrieved 2014-10-26.
[8]. Quantum Information Science and Technology Roadmap for a sense of where the research is heading.
[9]. Simon, D.R. (1994). *"On the power of quantum computation"*. Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on: 116–123. doi:10.1109/SFCS.1994.365701. ISBN 0-8186-6580-7.
[10]. Chuang, Michael A. Nielsen & Isaac L. (2001). *"Quantum computation and quantum information"* (Repr. ed.). Cambridge [u.a.]: Cambridge Univ. Press. ISBN 978-0521635035.
[11]. Karras D. A., Zorkadis V. *"On neural network techniques in the secure management of communication systems through improving and quality assessing pseudorandom stream generators"*. Neural networks: the official journal of the International Neural Network Society, 16(5-6), pp. 899–905, 2003.
[12]. Lauria F. E.: *"On Neurocrytology"*. Proceedings of the Third Italian Workshop on Parallel Architectures and Neural Networks, pp. 337–343, 1990.
[13]. Marsaglia G., Zaman A. *"A New Class of Random Number Generators"*. Ann. Applied Prob, pp. 462–480, 1991.
[14]. Nobuyuki Matsui, Haruhiko Nishimura, Teijiro Isokawa, *"Qubit Neural Network: Its Performance and Applications"*, IGI Global, 2009
[15]. Othman K. M. Z., Jammas M. H. A. L.: *"Implementation of Neural-Cryptographic System Using FPGA"*. Journal of Engineering Science and Technology, 6(4), pp. 411–428, 2011.
[16]. Pointcheval D.: *"Neural Networks and their Cryptographic Applications"*. Pascale Charpin Ed. India, 1994.
[17]. Priti Gupta and Chota Madan Markan, *"Exploring a Quantum-Hebbian Approach Towards Learning and Cognition"*, NeuroQuantology, Volume 11, Issue 3, Page 416-425, September 2013
[18]. Rukhin A., Soto J., Nechvatal J., Smid M., Barker E., S. Leigh, Levenson M., Vangel M.,Banks D., Heckert A., Dray J., Vo S.: *"A statistical test suite for random and pseudorandom number generators for cryptographic applications"*. Special Publication 800-22 National Institute Standart Technology, 2010.
[19]. Ruttor A.: *"Neural Synchronization and Cryptography"*. PhD thesis, Bayerischen JuliusMaximilians-Universitat at Wurzburg, 2006.
[20]. Sa˘gıro˘glu S., Ozkaya N.: *"Neural Solutions for Information Security"*. Journal of Polytechnic, 10(1), pp. 21–25, 2007.
[21]. Schneider B.: *"Applied Cryptography. Protocols, Algorithms, and Source codes in C"*, 1996.
[22]. Sivagurunathan G., Rajendran V., Purusothaman T.: *"Classification of Substitution Ciphers using Neural Networks"*. International Journal of Computer Science and Network Security, 10(3), pp. 274–279, 2010.

[23]. Su S., Lin A., Yen J.: *"Design and realization of a new chaotic neural encryption/decryption network"*. IEEE Asia-Pasific Conf.Cir and Syst., pp. 335–338, 2000.

[24]. Tope Komal1, Rane Ashutosh, Rahate Roshan, S.M.Nalawade. *"Encryption and Decryption using Artificial Neural Network"*. International Advanced Research Journal in Science, Engineering and Technology, Vol. 2, Issue 4, April 2015.

[25]. Yayık A., Kutlu Y.: *"Metin i¸c in Yapay Sinir A˘gı Tabanlı Hash Fonksiyonu"*. International Conferance on Cryptology and Information Security, 2013 (in English)..

[26]. Yayık A., Kutlu Y.: *"Sozde Rastsal Sayı Uretecinin Yapay Sinir Agları ile Guclendirilmesi"*. Sinyal Isleme ve ˙ Iletisim Uygulamaları (SIU) Kurultayı (SIU2013), 2013 (in English). ˙

[27]. W.K.. Wootters and W.H. Zurek, *"A single quantum cannot be cloned"*, Nature 299, 802 (1982)

[28]. D.Dieks, *"Communication by EPR devices",* Phys. Lett. A92, 271 (1982).

TÓM TẮT

GIAO THỨC THỎA THUẬN KHÓA DỰA TRÊN MẠNG NƠ RON LƯỢNG TỬ

Trong thuật toán mật mã lượng tử, khoá được tạo ra trong quá trình phân phối khóa, trong đó như trong phân phối khóa cổ điển, một khoá được xác định trước được truyền đến người dùng hợp pháp. Đóng góp quan trọng nhất của phân phối khoá lượng tử là phát hiện nghe trộm. Mục đích của bài báo này là giới thiệu một ứng dụng của QNN trong việc xây dựng giao thức phân phối khóa, trong đó hai mạng trao đổi các kết quả đầu ra của họ (theo qubits) và khóa được đồng bộ giữa hai bên giao tiếp. Hệ thống này được dựa trên QNN qubit đa lớp được đào tạo với thuật toán lan truyền ngược.

Từ khóa: Mạng nơ ron, Mạng nơ ron lượng tử, Mật mã học.

Received date, 13th March, 2017
Revised manuscript, 10th April, 2017
Published, 01st May, 2017

Address: [1] Academy of Cryptography Technique;
[*] Email: nam_haivn@yahoo.com.