Technical Report No. 2007-531
Authenticated Quantum Key Distribution without Classical Communication*

Naya Nagy and Selim G. Akl
School of Computing
Queen's University
Kingston, Ontario K7L 3N6
Canada
Email: {nagy,akl}@cs.queensu.ca

Abstract

The aim of quantum key distribution protocols is to establish a secret key among two parties with high security confidence. Such algorithms generally require a quantum channel and an authenticated classical channel. This paper presents a totally new perception of communication in such protocols. The quantum communication alone satisfies all needs of array communication between the two parties. Even so, the quantum communication channel does not need to be protected or authenticated whatsoever. As such, our algorithm is a *purely* quantum key distribution algorithm. The only certain identification of the two parties is through public keys.

Keywords: quantum key distribution, authentication, entanglement

1 Introduction

In cryptography, key distribution is the process whereby two parties reach an agreement on the value of a secret key. Several protocols exist in the quantum cryptography literature for the distribution of quantum keys. These protocols achieve a higher confidence in the key's secrecy than classical methods. To date, quantum key distribution algorithms have used two communication

*This research was supported by the Natural Sciences and Engineering Research Council of Canada.

media: a quantum channel, with quantum bits, and a classical channel, carrying classical information. The classical channel needs to be authenticated.

The algorithm presented here improves the quantum key distribution in two ways. First, there is no classical communication channel. Communication between the two parties is done solely via one insecure quantum channel. Secondly, authentication is done by the quantum algorithm itself, using two public keys. This is essentially different from previous algorithms, where authentication was done exclusively by classical means or by a trusted authority. It was believed that authentication is impossible by quantum means only [9]. This paper proves the opposite.

The rest of the paper is organized as follows: Section 2 reviews the notion of public keys, as used in classical cryptographic algorithms and then describes the particularities of the public keys required by our quantum authentication scheme. Section 3 defines entanglement and describes the particular entanglement based on phase incompatibility used by our algorithm. Section 4 describes the algorithm with authentication and security checking. Section 5 explores eavesdropping and masquerading scenarios and evaluates the security of the algorithm. Section 6 concludes the paper and offers some future directions for investigation.

2 Public Keys: Classical versus Quantum

There is no doubt that public key cryptosystems dominate cryptographic applications today. Their aim is to allow exchanging secret messages reliably and secretly. Public key cryptosysems offer commercially satisfactory security levels. Formally, the problem to be solved cryptographically can be formulated as two entities, Alice and Bob, that want to exchange secret messages on a classical insecure channel. A malevolent third party, Eve, may take advantage of the insecurity of the channel and listen to the message or tamper with its content. The security of the public key cryptosystem relies on the difficulty of inverting particular algebraic functions, also called “one-way” functions.

2.1 Protected Public Keys

Secure communication is achieved using two types of keys: a public key and a private key. If Bob wants to send a secret message to Alice, he uses the public key of Alice to encrypt the message. Alice then reads the message after using her private key for decryption. There are a few very important characteristics of the two keys implied in this communication. Alice’s private key is secret, and not shared with anybody else. In particular, Bob does not need to know Alice's private key. This is a major advantage, as the private key is never seen on any communication channel and therefore, its secrecy is ensured. By contrast, Alice's public key is available to anybody. Bob needs to know it, and also the eavesdropper, Eve, has access to it. In order for the protocol to work, the public key is guaranteed to be protected. This means, there is a consensus about the public key value. Both Bob and Alice are sure that they use the correct, same public key. Eve cannot masquerade as Alice and change the value of Alice's public key, making Bob use a false public key to encrypt his message. This feature is crucial for a public key cryptosystem to work. The public key cryptosystems need the public key to be protected, and accept it as given that such a protection of the public key is practically possible. Current public key algorithms, such as the RSA [8], need to continuously increase the length of the protected public key in order to maintain acceptable security levels.

Our quantum key distribution protocol also relies on the protectedness of public keys. The public keys used in our algorithm are regular binary numbers, but differ in meaning from the conventional public key, such as the RSA key. We will call the public keys used in our quantum algorithm *quantum generated public keys*. Alice has a protected quantum generated public key and Bob has another protected quantum generated public key. In fact these two public keys are the only protected information exchange between Alice and Bob. Exactly as in the case of classical public key cryptosystems, our algorithm requires that such public keys can be published protectedly, with the guarantee that the keys' values *are and remain* protected from masquerading. As will be seen from the algorithm itself, besides having public keys, Alice and Bob share only an insecure quantum channel.

2.2 Quantum Key Distribution Algorithms

The security of the classical public key RSA cryptosystem relies on the theoretically unproven assumption that factoring large numbers is intractable on classical computers. As described in [7], quantum computers can break some of the best public key cryptosystems.

Quantum cryptography aims to design mechanisms for secret communication with higher security than protocols based on the public key approach. Privacy of a message and its credibility is well satisfied in a private key cryptosystem setting. Alice and Bob share one and the same secret key, k_s. Bob uses the secret key for encryption and Alice consequently decrypts the message with the same key. As long as k_s is unknown to anybody else, the secrecy of the communication is satisfied. There exist various encryption / decryption functions using k_s, such that the encrypted message reveals no information whatsoever about the content of the message, provided the key

k_s is unavailable.

Quantum key distribution protocols establish secret keys via insecure quantum and/or classical channels. Existing quantum key distribution algorithms generally use two communication channels between Alice and Bob: a quantum channel which transmits qubits and a classical channel for classical binary information. The classical channel is used to communicate measurement strategy, or the basis for measurement, and to check for eavesdropping.

Quantum key distribution protocols may derive their efficiency from different quantum properties. The first protocol developed by Charles Bennett and Gilles Brassard, known as the BB84 protocol [2], relies on measuring qubits in two different orthonormal bases. The same idea applies to any two nonorthogonal bases [1]. In [5] the quantum key distribution algorithm is derived from the quantum Fourier transform. Based on the property of entanglement, Artur Ekert [4] gave a quantum key distribution solution using entangled qubits to be shared by Alice and Bob. A simpler version with qubits entangled in the same way, namely in the Bell states, is described in [3].

Note that all quantum key distribution algorithms mentioned above require that the classical channel be authenticated. Authentication is supposed to be done by classical means. The authenticated classical channel prevents Eve from masquerading as someone else and tamper with the communication. It was claimed by Lomonaco [9] that authentication is not possible in quantum computation, that for any secure quantum communication a classical authentication scheme needs to be used.

As will be clear from the algorithm described in this paper, authentication of a quantum communication protocol is not only possible by quantum means only, but in fact a classical channel is superfluous. The general authentication scheme has been developed in [6]. In this previous paper classical communication was still needed, though the classical channel was not authenticated. In the present improved version of the protocol, the classical channel is removed completely. The robustness of the algorithm comes also from the simplicity of the communication support available. Alice and Bob share an insecure quantum channel and two quantum generated public keys. They have an authentication step at the end of the protocol, with the help of the quantum generated public keys. Note that authentication in our algorithm is done at the end of the protocol and is derived from the quantum algorithm itself.

Shi et. al [10] also describe a quantum key distribution algorithm that does not use a classical channel. Authentication is done by a trusted authority, that provides the entangled qubits to Alice and Bob. In our paper, such a trusted authority is not needed. The entangled qubits may come from an insecure source.

3 Entangled Qubits

The key distribution algorithm we present in the following sections relies on entangled qubits. Alice and Bob, each possess one of a pair of entangled qubits. If one party, say Alice, measures her qubit, Bob's qubit will collapse to the state compatible with Alice's measurement.

The algorithms mentioned in section 2 [4, 3, 10], all rely on Bell entangled qubits. The qubit pair is in one of the four Bell states:

$$\frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle)$$

$$\frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle)$$

Suppose Alice and Bob share a pair of entangled qubits described by the first Bell state:

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Alice has the first qubit and Bob has the second. If Alice measures her qubit and sees a 0, then Bob's qubit has collapsed to $|0\rangle$ as well. Bob will measure a 0 with certainty, that is, with probability 1. Again, if Alice measures a 1, Bob will measure a 1 as well, with probability 1. The same scenario happens if Bob is the first to measure his qubit.

Note that any measurement on one qubit of this entanglement collapses the other qubit to a *classical* state. This property is specific to all four Bell states and is then exploited by the key distribution algorithms mentioned above: If Alice measures her qubit, she *knows* what value Bob will measure. The entanglement employed by the algorithm proposed in this paper, however, does not have this property directly.

3.1 Entanglement Caused by Phase Incompatibility

Let us look now at an unusual form of entanglement. Consider the following ensemble of two qubits:

$$\phi = \frac{1}{2}(-|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

The ensemble has all four components, $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$, in its expression. And yet, this ensemble is entangled.

Consider the following proof. Suppose the ensemble ϕ is not entangled. This means ϕ can be written as a scalar product of two independent qubits:

$$\phi = \frac{1}{2}(\alpha_1|0\rangle + \beta_1|1\rangle)(\alpha_2|0\rangle + \beta_2|1\rangle)$$

Matching the coefficients from each base vector, we have the following conditions:

1. $\alpha_1\alpha_2 = -1$
2. $\alpha_1\beta_2 = 1$
3. $\alpha_2\beta_1 = 1$
4. $\beta_1\beta_2 = 1$

The multiplication of conditions 1 and 4 yields: $\alpha_1\alpha_2\beta_1\beta_2 = -1$. On the other hand, from conditions 2 and 3, we have: $\alpha_1\alpha_2\beta_1\beta_2 = 1$. This is a contradiction. The product $\alpha_1\alpha_2\beta_1\beta_2$ cannot have two values, both $+1$ and -1. It follows that ϕ cannot be decomposed and thus the two qubits are entangled.

The entanglement of the ensemble is caused by the *signs* in front of the four base vector components. Thus, it is not that some vector is missing in the expression of the ensemble, rather it is the phases of the base vectors that keep the two qubits entangled.

3.2 Measurement

Let us investigate what happens to the ensemble ϕ, when the entanglement is disrupted through measurement.

If the first qubit q_1 is measured and yields $q_1 = |0\rangle = 0$ then the second qubit collapses to $q_2 = \frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle)$. This is not a classical state, but a simple Hadamard gate transforms q_2 into a classical state. The Hadamard gate is defined by the matrix

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}$$

Applying the Hadamard gate to an arbitrary qubit, we have $H(\alpha|0\rangle + \beta|1\rangle) = \alpha\frac{|0\rangle+|1\rangle}{\sqrt{2}} + \beta\frac{|0\rangle-|1\rangle}{\sqrt{2}}$. For our collapsed q_2, we have $H(q_2) = H(\frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle)) = -|1\rangle$. This is a classical 1.

The converse happens when qubit q_1 yields 1 through measurement. In this case q_2 collapses to $q_2 = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Applying the Hadamard gate transforms q_2 to $H(q_2) = H(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)) = |0\rangle = 0$. Again this is a classical state 0.

It follows that by using the Hadamard gate, there is a clear correlation between the measured values of the first and second qubit. In particular, they always have opposite values.

A similar scenario can be developed, when the second qubit q_2 is measured first. In this case, the first qubit q_1, transformed by a Hadamard gate, yields the opposite value of q_2.

4 The Algorithm

The goal of the key distribution algorithm described below is to establish a secret key, known only to Alice and Bob. Subsequently, when Alice and Bob exchange messages, they will use this key to encrypt / decrypt their messages. One session is required to establish a binary secret key, called *secret*, such that Alice and Bob are in consensus about the value of the secret key. The secret key *secret* consists of n bits, $secret = b_1 b_2 ... b_n$. Technically, to perform the algorithm, Alice and Bob need an array of entangled qubit pairs, and two protected public keys. Note that Alice and Bob do not communicate on *any* classical channel.

The array of the entangled qubits has length l, it consists of l qubit pairs denoted $(q_{1A}, q_{1B}), (q_{2A}, q_{2B}), ..., (q_{lA}, q_{lB})$. The array is split between Alice and Bob. Alice receives the first qubit of each entangled qubit pair, namely $q_{1A}, q_{2A}, ..., q_{lA}$, and Bob receives the second half of the qubit pairs, $q_{1B}, q_{2B}, ..., q_{lB}$. The entanglement of a qubit pair is of the type described in the previous section, namely, phase incompatibility. The array of qubits is unprotected. There is no guarantee that the qubits of a pair are indeed entangled; indeed, Eve may have disrupted the entanglement. Also, Eve may have masqueraded as either Alice or Bob, modifying the entangled qubits, such that Alice's qubit is actually entangled with a qubit in Eve's possession rather than Bob's, and the same holds for Bob. In case Eve has disrupted the entanglement or has masqueraded, any result of the algorithm is discarded and the key distribution is attempted all over again, from the beginning.

The size n of the secret key is less than half of the length l of the initial qubit array, $n < \frac{l}{2}$. Indeed, $\frac{l}{2}$ qubits, that is half of the qubits, are discarded because the bases in which Alice and Bob measure are inconsistent 50% of the time. From the remaining half of qubits a further arbitrary number is sacrificed for security checking. The number of qubits thus sacrificed depends on the desired degree of security.

Two public keys are needed by the algorithm. Alice has a public key key_A and Bob has a public key key_B. The two public keys key_A and key_B

are independent. Alice and Bob use these public keys to exchange classical binary information and also, very importantly, for authentication. The keys, as used in this algorithm, have some characteristics that are different from the classical public keys. The keys are established *during* the computation. They are not known prior to the key distribution algorithm and are defined in value during the computation according to the measured values of some of the qubits. This means that the keys are available *after* the key distribution protocol. Consequently, the keys have to be posted after the algorithm, which is unlike the classical case, where a public key is known in advance.

Also, the two public keys key_A and key_B are valid for one session, that is, for one application of the key distribution algorithm. If Alice and Bob want to distribute a second secret key using the same algorithm, they will have to create new public keys, which are different in value from the public keys of the previous session.

The key distribution algorithm, like all quantum key distribution algorithms, develops the value of the secret key during the computation. Implicitly, the values of the public keys as well are developed *during* the computation. There exists no knowledge whatsoever about the values of the keys (secret and public) prior to running the algorithm.

Both Alice and Bob follow the same steps briefly denoted below:

1. **Measure your entangled qubits**
2. **Compute your own public key and post it**
3. **Read your partner's key and check for eavesdropping**
4. **Construct the value of the secret key**

A detailed description of the algorithm follows.

Step 1

Alice works with the array of qubits q_{1A}, q_{2A}, ..., q_{lA}. Binary information is rendered by the results of measuring. All measurements are performed in the standard computational basis. Alice has two options for processing her qubits. She either measures a qubit directly, or she transforms the qubit by a Hadamard gate and measures afterwards. For each qubit, q_{iA}, Alice decides randomly on one of the two processing options. Notably, there is no communication with Bob at this stage. To look at a concrete example, suppose Alice has 10 qubits q_{1A}, q_{2A}, ..., q_{10A}. Qubits q_{iA} transformed by the Hadamard gate are denoted Hq_{iA}; for those measured directly the notation is unchanged. Suppose that by random choice, Alice has processed her qubits as follows:

$$q_{1A}, Hq_{2A}, Hq_{3A}, q_{4A}, q_{5A}, q_{6A}, Hq_{7A}, Hq_{8A}, q_{9A}, q_{10A},$$

and suppose again, she has measured the following binary values:

$$1, 1, 1, 0, 0, 0, 0, 1, 1, 1$$

In the meantime, Bob processes his qubits q_{1B}, q_{2B}, ..., q_{10B} following the same policy. He too, has a random choice on each qubit: to measure directly or to measure after a Hadamard transformation. Suppose again, that by random choice, Bob has obtained the following array:

$$Hq_{1B}, Hq_{2B}, q_{3B}, Hq_{4B}, q_{5B}, q_{6B}, q_{7B}, Hq_{8B}, Hq_{9B}, q_{10B},$$

with the values

$$0, 1, 0, 1, 1, 0, 1, 0, 0, 1$$

We have seen in the previous section that two entangled qubits $q_{iA}q_{iB} = \frac{1}{2}(-|00\rangle + |01\rangle + |10\rangle + |11\rangle)$, consistently render opposite classical bit measurements, if and only if exactly one qubit is measured directly and the other is measured after a Hadamard transformation. It is of no consequence whether the first qubit is Hadamard tranformed or the second. The order of the qubits is irrelevant, the important issue is that exactly one of the qubits is passing a Hadamard gate. Thus, there are two "valid" measurement options:

1. q_{iA}, Hq_{iB} and
2. Hq_{iA}, q_{iB}

These measurement scenarios are valid in the sense that they, and only they, yield opposite classical bits after measurement. Each of Alice and Bob knows with certainty the value the other person has measured. Such qubits are considered valid by Alice and Bob and will be used to form the secret key and to check for eavesdropping.

Measurements of the form

3. q_{iA}, q_{iB} and
4. Hq_{iA}, Hq_{iB}

cannot be used by Alice and Bob. For any value measured by Alice, the value measured by Bob is still determined probabilistically. Qubits measured according to these scenarios, will unfortunately have to be discarded. As the scenarios 1, 2, 3, 4 are equally likely, 50% of the initial qubits will be discarded because of probabilistically inconsistent measurements.

As mentioned, half of the l qubits are discarded because of incompatible measurement bases. The size n of the secret key is therefore $n < \frac{l}{2}$. From the remaining qubits, depending on the desired security level, some other qubits are sacrificed for checking.

For the example of the 10 qubits given above, there are five valid qubit-pairs:

$$(q_{1A}, Hq_{1B}), (Hq_{3A}, q_{3B}), (q_{4A}, Hq_{4B}), (Hq_{7A}, q_{7B}), (q_{9A}, Hq_{9B}),$$

carrying the values

$$(1, 0), (1, 0), (0, 1), (0, 1), (1, 0)$$

Step 2

At this point Alice has no idea what measuring option Bob has employed on his qubits. She does not know that qubits 1, 3, 4, 7, and 9 are valid. Bob is in the same situation.

Therefore, Alice will publish her measuring strategy in her public key. Alice has measured $l = 10$ qubits. As such, the first l bits of Alice's public key explain which qubits have been Hadamard transformed and which were measured directly. If Alice has applied the Hadamard gate on qubit q_iA then the i-th qubit of the public key is set to 1, $key_A(i) = 1$. Otherwise, if q_{iA} has been measured directly, then the i-th qubit is 0, $key_A(i) = 0$. For the example of 10 qubits, the first ten bits of Alice's public key are

$$key_A(1..10) = 0110001100$$

The second part of Alice's public key is used for security checking. A certain fraction f, for example $f = 40\%$, of the original qubits are made public for Bob to check for eavesdropping. Alice chooses randomly 40% of her l qubits. For each chosen qubit, Alice publishes the index of the qubit and the binary value she has measured. To continue our example, Alice chooses randomly the indices 1, 2, 9, 10. She will publish index 1 with value 1, index 2 with value 1, index 9 with value 1 and index 10 with value 1. Translated in binary this is

$$(0001)1(0010)1(1001)1(1010)1$$

Alice's final public key is the concatenation of the measuring (Hadamard / no Hadamard) information and the qubit checking information:

$$key_A = 0110001100 \;\; 0001\,1 \;\; 0010\,1 \;\; 1001\,1 \;\; 1010\,1$$

The length of the public key depends on the length l of the qubit array and also on the desired security level given by the fraction f. The following formula computes the length of the key:

$$length(key_A) = l + f(1 + \log l)$$

Here, l, the first term in the sum, refers to the measuring strategy; the second term, $f(1 + \log l)$, represents the part that publishes the qubits for eavesdropping checking.

Bob creates his public key following exactly the same steps. Bob's measuring strategy is encoded at the beginning of his public key. For our example, this means

$$key_B(1..10) = 1101000110$$

Suppose Bob sacrifices qubits 1, 5, 7, 8 for checking. In his public key he will publish (0001)0(0101)1(0111)1(1000)0. Thus, Bob's final key, the one that Alice and indeed everybody can see, is:

$$key_B = 1101000110 \quad 0001 \quad 0 \quad 0101 \quad 1 \quad 0111 \quad 1 \quad 1000 \quad 0$$

Both Alice's and Bob's keys, key_A and key_B are made public and are available to everybody, including Eve.

Step 3

At this stage, Alice and Bob, in full knowledge and consensus of each other's keys, will proceed to check for eavesdropping. Alice is looking at Bob's public key key_B and learns the values Bob has measured on the randomly sacrificed $f = 40\%$ of his qubits, here qubits 1, 5, 7, 8. Because of the various measuring options, only half of the $f = 40\%$ qubits will be useful. In our example, qubits 1 and 7 are measured with correct options, namely exactly one Hadamard gate applied to an entangled pair. Alice can find out the valid qubits by XOR-ing the measuring strategy of Bob with her own:

$$(0110001100) XOR (1101000110) = (1011001010)$$

which means qubits 1, 3, 4, 7, 9 have been measured well. Alice is left only to compare the values of qubits 1 and 7 she has measured with the values posted by Bob. With no malevolent interference, the binary values are opposite. Thus, if these values are opposite, Alice concludes that the protocol was not influenced by Eve. Otherwise, Alice discards all information and starts all over again. Bob performs the same checking. He will find the valid qubits posted by Alice 1 and 9 and will compare Alice's binary measured values with his own. Thus Bob makes his own independent decision concerning eavesdroppping. For reasonably large qubit arrays and a resonably large number of qubits checked, Alice and Bob will reach the same conclusion concerning the validity of the measured binary data. This conclusion effectively implies the absence of eavesdropping/masquerading (assuming, of course, that the qubits were initially entangled).

Step 4

At this stage, the possibility of eavesdropping has already been eliminated. The qubits that have not been published by Alice or Bob in their public keys continue to be unknown to anybody else. These unpublished qubits form the secret key *secret*, that is, *secret* will be formed from Alice's recorded values, and Bob's complementary values. In our ten qubit example, valid unpublished qubits are qubits 4 and 9. Therefore, the secret key will be Alice's qubits 4 and 9:

$$secret = 01$$

Bob has to complement his qubits to reach the same value as Alice.

The size (length) n of the secret key depends on the initial length of the qubit array l, as well as the fraction of discarded qubits f. Alice and Bob have decided randomly which qubits to publish. In the worst case, the set of qubits published by Alice is disjoint from the set published by Bob. Thus, the fraction of unpublished qubits is $1-2f$. From these unpublished qubits, only half (50%) are measured correctly. The length of the secret key is given by the formula

$$n = (1-2f)\frac{1}{2}l$$

For our example

$$n = (1-2\frac{40}{100})\frac{1}{2}10 = 1$$

The length of the secret key is 1 in the worst case. For our particular example we could use 2 bits.

5 Security Evaluation or Catching the Evil Eavesdropper

Let us consider the algorithm described in the previous section, from the point of view of the eavesdropper Eve. Eve wants to ideally gather knowledge about the value of the secret key without being noticed by either Alice or Bob. It is well known that an entangled qubit pair reveals no information whatsoever unless the qubits are measured and the entangled state collapses. Even so, the algorithm presented in this paper supposes that the entanglement is not protected, only the public keys are protected. This means that the qubits are not guaranteed to be entangled. Eve may masquerade and distribute qubit arrays of her own choice. It is of no advantage to Eve to distribute entangled qubits, as she gains no knowledge about the future secret key from unmeasured entangled qubits. The best choice for Eve is to distribute classical bits, or independent qubits in a known state.

The best Eve can do is to give Alice an array of classical 0s:

$$q_{1A}q_{2A}...q_{lA} = 00...0$$

and to Bob an array of $H1$:

$$q_{1B}q_{2B}...q_{lB} = H1\ H1...H1$$

All other possible arrays Eve could send to Alice and Bob are equivalent or less advantegeous than the arrays above. In particular, Eve will want to send any pair (q_{iA}, q_{iB}) that *can* be measured correctly : $(0, H1)$, $(H0, 1)$, $(1, H0)$, or $(H1, 0)$. Any such pair is equally advantageous. For simplicity we will discuss the arrays of 0s and H1s, respectively. For a pair $(0, H1)$, Alice and Bob apply randomly one of the four measurement options (see section 4). The first correct measurement option (q_{iA}, Hq_{iB}) consistently yields complementary correct results, namely $(0, 1)$. The second correct measurement option (Hq_{iA}, q_{iB}) yields all four possible classical bit combinations $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$. Moreover, these combinations are equally likely. In one-half of the cases, measurements will be $(0, 0)$ or $(1, 1)$. This cannot happen, if the qubits are entangled and untouched. This situation reveals the intervention of Eve. Thus, on any qubit checked for eavesdropping, there is a $\frac{1}{4} \times \frac{1}{2} = \frac{1}{8}$ chance of detecting Eve.

As Alice and Bob respectively check a fraction f of the original array, the expected number of times Eve is detected, that is, the *expected detection rate*, is

$$expected\_detection\_rate = \frac{1}{8} \times f \times l$$

For our example, the expected detection rate is

$$expected\_detection\_rate = \frac{1}{8} \times \frac{40}{100} \times 10 = \frac{1}{2}$$

This expected detection rate is rather low given the toy example we have considered, but of course it can be increased arbitrarily by increasing f and/or l.

Suppose we have an array of 1024 qubits and work with the same fraction $f = \frac{40}{100}$. In this case, the length of the final key is

$$n = (1 - 2\frac{40}{100})\frac{1}{2}1024 \approx 100$$

This is a length that can be used in practice.

The number of qubits checked by Alice (and also by Bob) is

$$checked\_qubits = \frac{1}{2} \times \frac{40}{100} \times 1024 = 204.8$$

On each qubit, Eve can escape being caught with probability $\frac{3}{4}$. Thus Eve can escape with probability $\frac{3}{4}^{204.8} = 3.25 \times 10^{-26}$. This probability is infinitesimal for any practical purposes.

6 Conclusion

The algorithm presented above shows clearly that authentication can be done by quantum means only. Besides an insecure quantum channel, Alice and Bob have only protected quantum generated keys to communicate with. The parallel with the classical authentication scheme is simple. In classical authentication, Alice and Bob have

1. an insecure classical channel and
2. one or two standard protected public keys, posted before any communication on the channel.

In the quantum authentication scheme presented in this paper, Alice and Bob equivalently have two items:

1. an insecure quantum communication channel, and
2. two quantum generated protected public keys.

An important difference concerning the two types of public keys, classical and quantum generated, is that the value of a quantum generated public key is developed during the computation and posted after any communication on the quantum channel is performed. Therefore, the quantum generated public keys depend on the specific communication session. They are not known prior to the execution of the key distribution algorithm and differ in value from one session to the next. This mirrors the behavior of the secret key to be established by the key distribution protocol. In all quantum key distribution protocols, the secret key is developed *during* the execution of the protocol.

If entangled qubits are easily available, the secret key established by the algorithm can be arbitrarily long. Our algorithm can distribute a “one time pad” [11] without Alice and Bob having to meet. To use one time pads, traditionally, Alice and Bob meet in secret and exchange a long list of keys, each as long as the message it is supposed to encrypt, and each to be used exactly once.

The algorithm presented performs quantum key distribution based on entangled qubit pairs. The entanglement type is not of the generally used Bell states, but an unusual entanglement based on phase incompatibility. The advantage of this type of entanglement is that Alice and Bob perform *different* measurement steps: one is measuring the qubit directly, and the other is measuring after applying a Hadamard gate. Therefore, the measurement is not symmetric. This property, combined with random choice on the measurement steps leaves Eve with no knowledge of how to measure a tampered qubit in advance. How other protocols and algorithms may benefit from asymmetric measurement is an open problem.

The principle of checking and authenticating at the end of the protocol with quantum generated public keys, is not restricted to the algorithm described here. The same type of public keys, generated per session, posted after the execution of the main body of the algorithm, can be successfully used in authenticating other types of algorithms. This is also a direction worth investigating.

References

[1] Charles H. Bennett. Quantum cryptography using any two nonorthogonal states. *Physical Review Letters*, 68(21):3121–3124, May 1992.

[2] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179, IEEE, New York, 1984. Bangalore, India, December 1984.

[3] Charles H. Bennett, Gilles Brassard, and David N. Mermin. Quantum cryptography without Bell's theorem. *Physical Review Letters*, 68(5):557–559, February 1992.

[4] Artur Ekert. Quantum cryptography based on Bell's theorem. *Physical Review Letters*, 67:661–663, 1991.

[5] Marius Nagy and Selim G. Akl. Quantum key distribution revisited. Technical Report 2006-516, School of Computing, Queen's University, Kingston, Ontario, June 2006.

[6] Naya Nagy and Selim G. Akl. Quantum authenticated key distribution. In *Proceedings of International Conference on Unconventional Compu-*

tation. Lecture Notes in Computer Science 4618. Springer-Verlag, Heidelberg, 2007.

[7] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, 2000.

[8] Ronald L. Rivest, Adi Shamir, and Len M. Adleman. A method of obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[9] Jr. Samuel J. Lomonaco. A Talk on Quantum Cryptography or How Alice Outwits Eve. In *Proceedings of Symposia in Applied Mathematics*, volume 58, pages 237–264, Washington, DC, January 2002.

[10] Bao-Sen Shi, Jian Li, Jin-Ming Liu, Xiao-Feng Fan, and Guang-Can Guo. Quantum key distribution and quantum authentication based on entangled states. *Physics Letters A*, 281(2-3):83–87, 2001.

[11] Serge Vaudenay. *A Classical Introduction to Cryptography: Applications for Communications Security*. Springer, 2006.