

An Efficient Key Distribution Protocol Based on BB84

Parag K. Lala *

Department of Electrical Engineering, Texas A&M University-Texarkana, Texarkana, USA
 *Corresponding author: plala@tamut.edu

Received May 07, 2014; Revised May 26, 2014; Accepted May 26, 2014

Abstract Private key cryptography suffers from a major weakness - it requires sharing of a secret key between two parties. An intruder can copy the secret key as it is being exchanged, thereby severely compromising the security of the system. Thus a private key cryptographic system depends entirely on secrecy of the key. Public key cryptography does not have a key distribution problem but its security relies on the fact that determining the factors of a number that is the product of two very large prime numbers is not computationally feasible. It has been shown that a quantum computer can solve the prime factors of very large numbers in polynomial time which would otherwise take millions of years. Public key cryptography will therefore become insecure if quantum computing becomes a reality. Quantum cryptography, originally presented in BB84 protocol, avoids all these issues by encrypting the shared key using a series of photons. In this paper a key distribution protocol based on the concepts of BB84 is proposed. It provides an additional layer of security by sending the key data bits twice; during the second transmission the original key bits or their complements are randomly chosen for transmission. The sender informs the receiver about the orientation of the key bits during the second round of transmission only after the data has been sent out.

Keywords: quantum key, BB84, RSA, one-time pad

Cite This Article: Parag K. Lala, "An Efficient Key Distribution Protocol Based on BB84." *American Journal of Computing Research Repository*, vol. 2, no. 2 (2014): 33-37. doi: 10.12691/ajcrr-2-2-2.

1. Introduction

The need for secure communication and transfer of data has become extremely important in recent years. Intruders can access such transmitted information through various means. The prevention of unauthorized access to private data or communication is of utmost importance in both commercial and defense applications. A major weakness of many systems is the physical channel used by a system for interconnecting users and the system. An unauthorized user must not have access to the data transmitted via the channel.

The main goal of secure communication is to encrypt data before transmission so that only its intended users can decrypt the data. The encryption and decryption is done using a key. If the encryption and decryption of data is done using the same key, then only the sender and the receiver of the data should have access to the key. The sharing of a private key is the main concept of symmetric key cryptography.

Vernam [1] proposed a symmetric key cryptographic scheme known as *one-time pad*, that encrypts data using a random key. The term one-time pad indicates that the key is used one time only and never used again. In topics of cryptographic communication the sender is identified as Alice, the receiver as Bob and the intruder as Eve. The key must have the same number of bits as the data to be transmitted and must also consist of completely random bits that are kept secret from everyone except the sender

and the receiver. The keys are used only once as indicated above; both the sender and the receiver must destroy their keys after use. The principle of operation of one-time pad is as follows:

Encryption by Alice : $c_i = d_i \oplus k_i$ $i = 1, 2, 3, \dots$

where

d_i : data bits.

k_i : key bits

c_i : encrypted data bits.

Decryption by Bob : $d_i = c_i \oplus k_i$ $i = 1, 2, 3, \dots$

Thus Alice encrypts the data she sends to Bob by EX-ORing it with randomly generated key bits. Bob retrieves the encrypted data by EX-ORing the received data with the same key bits. Figure 1 illustrates the scheme assuming key bits are 100010000101100

	Alice	Bob	
d_i	000010010110000	100000010011100	c_i
\oplus			\oplus
k_i	<u>100010000101100</u>	<u>100010000101100</u>	k_i
c_i	100000010011100	000010010110000	d_i

Figure 1. An example of one-time pad

A major drawback of symmetric key cryptography is that the sender and the receiver must somehow exchange the secret key that they use. A third party might intercept the communication between the sender and the receiver and access the key, thereby compromising the security of the data transmission. The key distribution in a secure and

efficient manner remains as the major weakness of symmetric key cryptography such as one-time pad scheme. An additional problem with symmetric key cryptography is the number of keys needed. For example if each pair of n number of individuals exchanges private data then $n!/2x(n-2)!$ i.e. $n(n-1)/2$ keys are needed. Thus the total number of keys for large n becomes unrealistic.; this is a fundamental limitation of every *perfectly secure* cipher.

In another class of cryptographic system known as public key cryptographic system communicating parties use two separate keys—a *public key* and a *private key*. The public key as the name suggests may be made accessible to anybody. The private key on the other hand is kept secret. Figure 2 shows the encryption and decryption process in a public key cryptographic system.

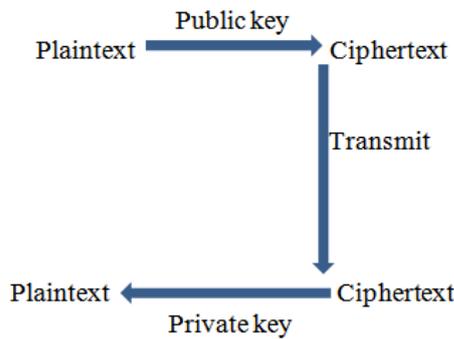


Figure 2. Public key cryptographic system

Public-key cryptography uses a method of encoding and decoding that employs a special case of the one-way function known as a *trapdoor one-way function* [2]. A function $f(x)$ is considered to be a one-way function if it is easy to compute $f(x) = y$ for any input x , but it is hard to invert $f(x)$ i.e. it is difficult to compute x from a known y . A trapdoor one-way function is one which is easy to invert if some piece of information (the trapdoor) is known. For example it is relatively easy to multiply two prime numbers to generate a composite number, but it is extremely difficult to factor a composite number (especially a very large integer) into a product of two prime numbers unless one of the numbers is known.

The public key is used to encrypt the data to be sent out. Anybody can have access to the public key, however the encrypted data can only be decrypted by a party who knows the corresponding private key. Thus public-key cryptosystems are essentially trapdoor functions; encryption is the one-way operation, and the private key is the trapdoor information that allows the user to invert the function and thus decrypt the received data. The distribution of the private key is avoided, thus preventing any unauthorized party from accessing the key.

The widely used RSA technique is a public key cryptographic system [3]. It facilitates the generation of public and private keys by choosing two large prime numbers p and q , and making $N = p \cdot q$. Next a random positive integer e is chosen such that it is relatively prime to $(p-1)(q-1)$; e is called the encryption constant. Then the decryption constant d is derived such that $e \cdot d = 1 \text{ mod } (p-1)(q-1)$. The public key is (N, e) , and the private key is d .

It should be mentioned that although N is revealed to all, the factors p and q of N are kept secret. Obviously if an intruding party can factor N to find p and q , then it can use e of the public key to derive the private key d from the

expression $e \cdot d = 1 \text{ mod } (p-1)(q-1)$. Figure 3 shows the steps of the RSA algorithm:

1. Generate two large prime numbers, p and q , and let $n = p \cdot q$;
2. Let $\phi = (p-1)(q-1)$;
3. Choose another number e which is relatively prime to ϕ ; two numbers a and b which have no common factors other than 1 are said to be co-prime or relatively prime.
4. Select e , $1 < e < \phi$ such that $\text{gcd}(e, \phi) = 1$; gcd (the greatest common divisor) of two integers a and b is the largest integer that divides both numbers..
5. Find d , such that $de \equiv 1 \text{ mod } \phi$. The notation $a \equiv b \text{ (mod } n)$ means a is congruent to b that is. a and b have the same remainder when divided by n .
6. Encryption: compute $c = m^e \text{ mod } n$, where m is message block represented as a number $0 < m < n - 1$ and c is the encrypted message.
7. Decryption: compute: $m = c^d \text{ mod } n$

Figure 3. RSA algorithm

The security of RSA system is based on the fact that currently no algorithm is available for factoring a large number into a product of two prime numbers in a reasonable amount of time, especially if these prime numbers are roughly the same size. However it has been known for some time that a quantum computer is capable of factoring very large numbers efficiently [4]. Thus the security of public key cryptographic system can be guaranteed only till quantum computers become technologically feasible

2. Quantum Cryptography

As indicated previously the distribution of keys is a major weakness of private key cryptography. Quantum mechanics overcomes this drawback by providing a secure way of sharing a random key between two separate parties. An additional advantage of quantum keys is that the sender and receiver can easily verify whether the key has been tampered with. It should be emphasized here that quantum cryptography is not a technique for encryption and decryption of data, it allows only secure distribution of private keys. Thus symmetric key cryptography such as one-time pad in conjunction with quantum key distribution can guarantee secure generation and transmission of private keys.

Quantum cryptography has its origin in a novel idea Stephen Wiesner, a graduate student at Columbia University in 1969. His idea was to use properties of quantum mechanics to create bank notes, “quantum money”, that cannot be counterfeited [5]. Each note would contain certain “light traps”, each of which could be filled with a randomly polarized photon. Photons can be polarized in one of two modes, rectilinear (+) or diagonal (x) using an appropriate filter; a filter allows the transmission of a photon through it only if the polarization of the photon is aligned with the filter. In the rectilinear mode only photons with horizontal or vertical polarization pass through the polarizing filter. In the diagonal mode, on the other hand, only photons with polarization that are at an angle of 45° or 135° to the horizontal axis can pass through the polarizing filter. In the rectilinear mode,

orientations | and — represent 1 and 0 respectively whereas in the diagonal mode orientations \ and / are assumed to be 1 and 0 respectively. For example, assume a bank note with serial number 1234 and polarization sequence \ - / \ | - (not visible). A counterfeiter can easily copy the serial number but the security of the light traps is guaranteed by the Heisenberg’s uncertainty principle which states certain pairs of properties, known as non-commuting properties, are related in a way that it is impossible to measure these simultaneously [6]. Rectilinear and diagonal polarizations constitute such a pair of non-commuting properties. Thus a filter with | and — orientation can correctly detect a rectilinearly polarized photon; similarly a filter with \ and / orientation can detect a diagonally polarized photon. if the counterfeiter uses a diagonally (rectilinearly) polarized photon to detect a rectilinearly (diagonally) polarized photon the outcome will be random with equal probabilities, and the photon will lose all the information of its previous state. For the photon sequence in the bank note assumed above, if the counterfeiter uses a diagonal filter he will correctly identify the orientation of first photon in the sequence. However for the second photon in the sequence the diagonal filter will perturb its orientation as a direct consequence of the uncertainty principle. There is a 50% chance that the photon would rotate up to a \ or a / orientation, or it will be blocked. In any of the above scenarios the measurement of the photon orientation will give erroneous result, thus guaranteeing the prevention of counterfeiting of quantum notes.

Bennet and Basard [7], inspired by Wiesner’s scheme, proposed a quantum key distribution protocol. This protocol known as BB84 allows a sender (Alice) to send photons to a receiver (Bob). Alice and Bob communicate via a one-way quantum channel and a two-way public channel. Alice has a source of single photons and two polarizing filters - one rectilinear and one diagonal. Single photons cannot be copied; this is because the linearity of quantum mechanics does not allow cloning of unknown quantum states [8]. Alice can transmit single photons randomly in either rectilinear (+) or diagonal (X) mode. In each mode one orientation of the photon is used to represent the logic value 0 and the other one to represent 1. Figure 4 shows the mode, angle, polarization and value of single photons. For example in the rectilinear mode (+) orientations | and — represent 1 and 0 respectively. In the diagonal (X) mode orientations \ and / represent 1 and 0 respectively.

Mode	Angle	Polarization	Logic Value
+	0°	—	0
+	90°		1
X	45°	/	0
X	135°	\	1

Figure 4. Characteristics of single photons

Alice generates a random sequence of 0s and 1s. She then randomly selects a polarization mode, rectilinear or diagonal, and replaces each bit in the binary sequence with a photon polarized using the selected mode. She sends the resulting photons for each bit to Bob via the quantum channel while keeping record of the polarizing mode and the logic value of the transmitted photon. In other words

Alice transmits photons to Bob in four different orientations |, —, \ and /.

To illustrate let us assume that Alice decides to send the following bits to Bob

Bits 1 0 0 1 1 0 1 0

and chooses the following polarizing modes to convert the bits

Mode + + + + + + + +

The polarization of the resulting single photons are :

Polarization | / — \ \ / | —

Bob detects the state of each photon he receives by randomly picking one of the polarizing modes of photons. if he makes the correct guess in picking the polarization mode Alice used for sending a particular photon, he obviously detects the correct orientation of the photon and therefore the correct logic value the photon represents. For example if Alice sends a 1 using the rectilinear mode (as in the first bit in the above bit sequence) and Bob chooses the same polarization mode he is guaranteed to receive a 1. On the other hand if Bob picks the diagonal mode the probability of his receiving a 1 is reduced to 50%, and there is a 50% probability of his receiving a 0 instead. Table 1 shows the modes Bob selected and polarizations of the resulting photons.

Mode + + + + + + + +

Polarization | — \ \ \ / | \

A simplified version of the BB84 protocol was proposed in Ref. [9]. This version uses two states, rectilinear and diagonal to represent 0 and 1 respectively, instead of four states in BB84. Pasquinucci et. al. [10] proposed a protocol that uses three orthogonal bases and six states to encode the key bits. Thus an intruder has to correctly choose the base used by the sender and receiver out of three possible bases. This increases the probability of the intruder making more errors in selecting the correct base, thus allowing easier intrusion detection. Scarani et. al. [11] proposed a variation of the BB84 protocol in which during the second round Alice transmits a pair of non-orthogonal polarization states to Bob instead of the base she used to encode a bit; one of these states in this pair is the state used by Alice to encode the key data bit. At the receiving end Bob will correctly measure the polarization state if only he chooses the same base as Alice, otherwise the data bit will have an unpredictable value.

3. Modified Distribution Protocol

In this paper a modified form of BB84 protocol is presented. This protocol significantly reduces the number of key bits to be discarded because of the incorrect guessing of the polarizing mode by the recipient (Bob) of the transmitted photons. The steps of the protocol are listed below:

- i. The encryption key is represented by a series of polarized photons; the modes of polarization are chosen randomly by the sender (Alice).
- ii. The receiver (Bob) measures the received photons using a randomly chosen polarization mode for the photon detectors. (If Bob chooses the same polarization mode as Alice then he measures the correct modes of the photons, and consequently will have the key that Alice wants to share with him.)

iii. Alice resends Bob the original key bits (i.e. key bits she previously sent to him in step i) or the bit-by-bit complement of these key bits. Bob measures the new key bits from Alice using photon detectors that have polarization modes exactly opposite to those he used to identify key bits sent to him in the first round.

iv. Alice sends to Bob the photon polarization modes she used for the key. She also informs Bob whether she sent the actual key bits in step iii, or the complement of these bits.

v. From the two sets of data Bob received from Alice, he keeps only those bits in which the polarization modes of photon detectors he used match those sent by Alice. The resulting bits correspond to the encryption key.

Let us illustrate the application of the revised protocol by considering a situation when Alice and Bob use rectilinear and diagonal orientation modes for both transmission and detection. Suppose Alice's key is

1 1 1 0 0 1 1 0

and she sends this to Bob using the following modes of photon polarization

+ + × × × + × + ×.

Bob is not aware of what polarization mode Alice used, and chooses the following modes for detecting the photons

× + + + × × + × ×

Based on that, Bob receives the following key bits where a “- “ indicates that the bit in this position can be either 0 or 1 because the polarization modes of the corresponding photons do not match:

- 1 - 0 - - 0

Next Alice resends the key (or its bit-by-bit complement) to Bob using the same polarization mode she used previously in step i. However, Bob is not aware of whether Alice is sending the actual key or its bit-by-bit complement. He uses photon detectors with polarizing modes that are exactly opposite to what he used in step ii, in other words he replaces mode + with × and vice versa. So we need to consider the following two cases:

Case i. Alice resends the encryption key to Bob. Bob uses the inverse of the polarization modes he chose to identify the key bits received from Alice in the first round (as discussed above). Thus the polarization modes Bob uses in the second round are.

+ × × × + + × + +

As a result the photon polarization modes Bob uses in the second round are different from those of Alice in positions 2, 5 and 9 from left; he receives the key bits shown below:

1- 1 0- 0 1 1-

Case ii. Alice sends the bit-bit complement of the encryption key to Bob. Bob uses the same polarization modes for the photon detectors as in Case i, and receives the following key bits :

0- 0 1- 1 0 0-

As indicated previously when Alice sends Bob the polarization modes of photons, she also informs him whether during the second round of transmission she used the original key bits or their complements. Bob compares the polarization modes he used with those of Alice. Table 1 shows the comparison of the polarization modes Alice sent and the modes Bob used during round 1 and round 2;

an up-arrow (↑) identifies a match between Bob's mode with that of Alice.

From Table 1 it can be seen that Bob's choice of polarization modes were correct three times in the first round but six times in the second; this is because of the reverse orientation of the photon detectors during the second round. However, if Alice indicated that she had sent the complement of the encryption key in the second time then Bob would have needed to take the bit-by-bit complement of the received bits. In any case the encryption bits for which polarizations used by Alice and Bob match i.e. the bits corresponding to the ↑ symbols above are the used as the encryption key. Based on the matching of polarization modes in two rounds as shown in in Table 1 it is clear Bob can generate the key Alice intended to send. It should be mentioned this approach does not result in any key bit loss because of the incorrect selection of photon polarization modes by Bob, otherwise almost 66% of the key bits might be lost in this particular case.

Table 1. Comparison of photon polarization modes

Alice	+	+	×	×	×	+	×	+	×	
Bob	×	+	+	+	×	×	+	×	×	(Round 1)
		↑			↑					↑
Bob	+	×	×	×	+	+	×	+	+	(Round 2)
	↑		↑	↑		↑	↑	↑		

4. Conclusion

A key distribution protocol based on the concepts of BB84 protocol is presented in this paper. It provides an additional layer of security by resending the key data bits. This is because only after sending the second round of data, the sender informs the receiver whether the data sent was the original key bits or their complements. Thus the intruder even he becomes aware of the key bits are being resent has to wait to determine whether the sent bits are inverted or not.

In the original BB84 protocol in general only about 50% of the key bits can be utilized to form a key. A major advantage of the protocol proposed in this paper is that it allows the receiver to retrieve all bits of the transmitted key unless they are lost during transmission or due to defects in the photon detectors. Thus the proposed approach significantly enhances the efficiency of encryption key generation.

References

- [1] Vernam, G.S., "Cipher Printing Telegraph Systems for secret wire and radio telegraphic communications," J. AIEE 45, pp. 109-115, 1926.
- [2] Diffie, W. and Hellman, M., "New directions in cryptography", IEEE Transactions on Information Theory, vol. IT-22, No. 6, pp. 644-654, Nov. 1976.
- [3] Rivest, R, Shamir, A and Adleman, L, "A method for obtaining digital signatures and, public key cryptosystems," Communications of the ACM, pp. 120-126, 21, 1978.
- [4] Riefel, E. and Polak, W., "An introduction to quantum computing for non-physicists", arXiv: quant-ph/9809016, 1998.
- [5] Wiesner, S. "Conjugate coding", SIGACT News, 15 (1): 78-88, 1983. Original manuscript written circa 1970.
- [6] Yanofsky, N.S. and M.A. Mannucci, *Quantum Computing for Computer Scientists*, Cambridge University Press, 2008.

- [7] Bennett, C.H., and Brassard, G. "Quantum cryptography: public key distribution and coin tossing", International Conference on Computers, Systems & Signal Processing, pp. 175-179, 1984.
- [8] Wootters, W.K., and Zurek, W.H., "A Single Quantum Cannot Be Cloned", Nature 299, pp. 802-803, 1982.
- [9] Bennett, C.H., "Quantum cryptography using any two non-orthogonal states", Phys. Rev. Letts. 68, pp. 3121-3124, 1992.
- [10] H. Bechmann-Pasquinucci and N. Gisin, "Incoherent and coherent eavesdropping in the six state protocol of quantum computing", Phys. Rev. A 59, pp. 4238-4248, 1999.
- [11] A. Scarani, A. Acin, G. Ribordy and N. Gisin, "Quantum cryptography protocols robust against photon number splitting attacks", Physical Review Letters, vol. 92, No. 5, pp. 2004.