Journal of Computer Science 6 (10): 1094-1100, 2010
ISSN 1549-3636

Protocol of Secure Key Distribution Using Hash Functions and Quantum Authenticated Channels Key Distribution Process Six-State Quantum Protocol

[1,2]Mohammed Munther A. Majeed, [1]Khalid A.S. Al-Khateeb,
[1,2]Mohamed R. Wahiddin and [2]Magdy M. Saeb
[1]Department of Electrical and Computer Engineering, Faculty of Engineering,
International Islamic University Malaysia, P.O. Box 10, 50728, Kuala Lumpur, Malaysia
[2]Department of Quantum Cryptography, Advanced Information Security Cluster, MIMOS, Berhad,
Technology Park Malaysia, 57000, Kuala Lumpur, Malaysia

Abstract: Problem statement: In previous researches, we investigated the security of communication channels, which utilizes authentication, key distribution between two parties, error corrections and cost establishment. In the present work, we studied new concepts of Quantum Authentication (QA) and sharing key according to previous points. **Approach:** This study presented a new protocol concept that allows the session and key generation on-site by independently applying a cascade of two hash functions on a random string of bits at the sender and receiver sides. This protocol however, required a reliable method of authentication. It employed an out-of-band authentication methodology based on quantum theory, which uses entangled pairs of photons. **Results:** The proposed quantum-authenticated channel is secure in the presence of eavesdropper who has access to both the classical and the quantum channels. **Conclusion/Recommendations:** The key distribution process using cascaded hash functions provides better security. The concepts presented by this protocol represent a valid approach to the communication security problem.

Key words: Quantum Authentication Process (QAP), Virtual Private Network (VPN), Key Distribution Process (KDP)

INTRODUCTION

Since the publication of the BB84 protocol (Bennett and Brassard, 1984), Quantum Key Distribution (QKD), much study has been devoted to application of quantum mechanics in cryptography. QKD schemes however, typically depend on authentication of classical communications by classical methods and relatively little work has been done on the quantum authentication and authenticated QKD. Some quantum authentication proposals were made (Barnum, 1999; Huttner *et al*., 1996), which are variations of the BB84 protocol. They use classical methods of cryptography for authentication. An early quantum protocol, which uses quantum oblivious transfer, (Bennett *et al*., 1995); (Barnum, 1999) and (Zeng and Zhang, 2000) as well as a quantum protocol based on entanglement theory (Barnum *et al*., 2003) all were shown to be insecure (Dusek *et al*., 1999).

In the current proposal, cascaded hash functions are employed to generate a shared secret key locally by the communicating parties. A shared entangled pair is used in the authentication according to the deterministic six-state quantum protocol (6DP) (Shaari *et al*., 2006). Hence, a Quantum Authentication Process (QAP) is established.

The protocol is described as following in the materials and methods. In the first part, we provide a description of the Quantum Authentication Process (QAP) with 6DP approach. The second part describes the Virtual Private Networks (VPN) and the third describes the Key Distribution Process (KDP) in several modes. The discussion segment discusses the specifications of the hash functions utilized in (KDP-6DP) protocol and its various modes. Finally, we provide conclusions.

MATERIALS AND METHODS

Quantum Authentication Process (QAP): The general task of authentication is about verifying the identity of each one of two communicating parties (Alice and Bob), employing a quantum channel or a classical channel. We adopt a Quantum Authentication

Corresponding Author: Mohammed Munther A. Majeed, Department of Electrical and Computer Engineering, Faculty of Engineering, International Islamic University Malaysia, P.O. Box 10, 50728, Kuala Lumpur, Malaysia

Process (QAP) under ideal conditions. That is temporarily neglecting, for sake of clarity, the impact of transmission losses, detection rates and other limiting factors of the physical implementation.

The QAP task is to verify the mutual identification of the two parties. Contrary to the usual convention, the sender in this protocol is Bob and the receiver is Alice. Once the sender and receiver in the quantum channel complete one successful authentication process in 6DP, they would be convinced of each other's identities and that the communication channel is secure.

Figure 1 shows the infinite data state diagram of the QAP. Any eavesdropping attack can be detected routinely by a sequence of events based on Random Time Interval Generation (RTIG), which triggers the Start State for the QAP within the KDP-6DP.

Initially, the sender prepares six quantum states randomly. Every state contains two different entangled pairs of photons. If any of the two photons is missing or has a changed state of polarization in two forward and backward paths due to eavesdropping, it will deterministically detect. If nothing wrong detected, continuously the two states will send to the receiver. The receiver flips them by one of the four prepared quantum operators I, X, iY or Z and sends them back to the sender. The sender checks and compares all these states with the previous polarization states according to Table 1.

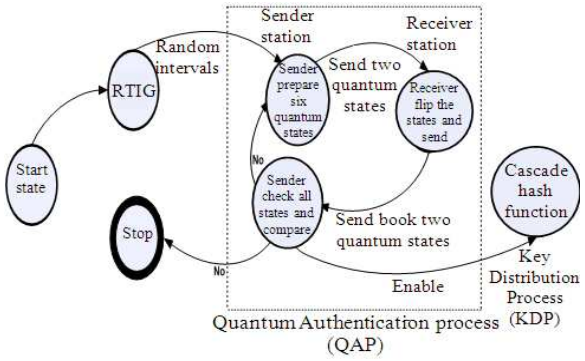

Fig. 1: The infinite data state diagram of the Quantum Authentication Process (QAP)

Table 1: A possible combination of the qubits sent by the sender, the operations performed and the number of bits flipped as result of the measurements by the sender

Qubits Combinations	X	iY	Z	I
$\hat{x}\ \hat{y}\ \hat{y}\ \hat{x}$	1	1	2	0
$\hat{x}\ \hat{z}\ \hat{z}\ \hat{x}$	1	2	1	0
$\hat{y}\ \hat{z}\ \hat{z}\ \hat{y}$	2	1	1	0
Boolean value	10	1	11	0

After checking and comparison of the six states, if it is found that any state is missing or changed, the whole process is thus stopped and an investigation is carried out. Otherwise, the communication will continue as usual by giving an enable instruction to start another phase in the KDP-6DP protocol. This phase, called key distribution process KDP performs key-sharing by using cascaded hash functions.

The Quantum Authentication Process (QAP) starts for example, let us say that the sender sends qubits in the $\hat{x}$, $\hat{y}$ combination by choosing the states $|x\pm\rangle$ and $|y\pm\rangle$. If the sender's final measurement results in none or both of the qubits flipped, the now sender would infer that the operation done by the receiver was I (Z). On the other hand if it results in only the state $|x\pm\rangle$ ($||y\pm\rangle$) flipped, the operator must have been iY (X) (Shaari *et al*., 2006).

In order to guarantee the security of the QAP against an attack by an eavesdropper (Eve), the sender and the receiver must sacrifice some of the runs to perform a control of this quantum channel. They test both the forward and the backward paths of the channel with a procedure equivalent to the one adopted in the BB84 protocol. Upon receiving the two qubits from the sender, the receiver makes a projective measurement of them along a basis randomly chosen among $\hat{x}$, $\hat{y}$ or $\hat{z}$. After that, the receiver forwards both the projected qubits to the sender who himself measures them again. When the bases chosen by the sender and the receiver are the same, then they expect the outcomes of their measurements to be correlated on both the forward and the backward paths. Any deviation from this expected scenario is considered an error. If the detected errors are below a certain security threshold, established in advance by the two legitimate users the communication goes on with the usual error correction and the privacy amplification stages. If the security threshold is exceed, the whole Quantum Authentication Process (QAP) aborts.

After the sender conformation step is complete in this phase from our protocol, it means the sender is enabled to decide deterministically that he has authenticated the receiver. He will send enable signal to the key generation phase to commence hash-based key computations. This key, which is producing by cascade hash functions according to KDP, will be generating locally at the sender and receiver stations.

Virtual Private Network (VPN): VPNs create the ideal infrastructure for the exchange of data and network resources with clients without sacrificing the security and integrity of the data. Figure 2 shows our proposed setup for the Virtual Private Network (VPN).

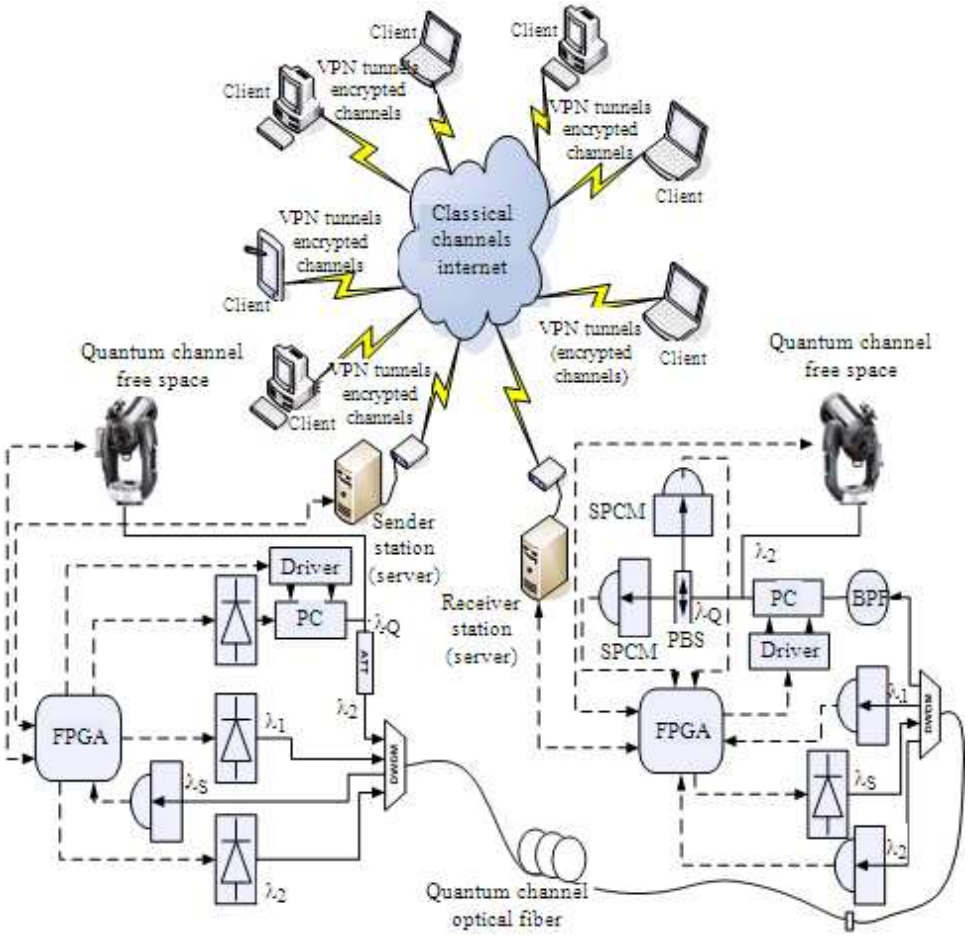

Fig. 2: The virtual private networks VPN (classical channels) and the quantum channel

A VPN is a virtual network, built on top of existing physical networks, which can provide a secure communications mechanism for data and IP information transmitted between networks. Since a VPN can be use over existing networks, such as the Internet, it can facilitate the secure transfer of sensitive data across public networks. This is often less expensive than other alternatives such as dedicated private telecommunications lines between organizations or branch offices. VPNs can also provide flexible solutions, such as securing communications between remote telecommuters and the organizations servers, regardless of where the telecommuters are located. A VPN can even be establishing within a single network to protect particularly sensitive communications from other parties on the same network (Frankel *et al*., 2005).

We know the VPNs can use both symmetric and asymmetric forms of cryptography. Symmetric cryptography uses the same key for both encryption and decryption. While asymmetric cryptography uses separate keys for encryption and decryption, or to digitally sign and verify a signature. Symmetric cryptography is generally more efficient and requires less processing power than asymmetric cryptography. That is why it is typically uses to encrypt the bulk of the data being sending over a VPN. One problem with symmetric cryptography is with the key exchange process; keys must be exchange in an out-of-band fashion to ensure confidentiality. Common algorithms that implement symmetric cryptography include Digital Encryption Standard (DES), Triple DES (3DES), Advanced Encryption Standard (AES), Blowfish, RC4, International Data Encryption Algorithm (IDEA) and the Hash Message Authentication Code (HMAC) versions of Message Digest 5 (MD5) and Secure Hash Algorithm (SHA-1) (Frankel *et al*., 2005). In our case, a key exchange protocols based on the use of hash functions by selecting or a cascading two hash

functions and a long-term shared secret, is used to construct the key. Consequently, the session key is generated on-site by independently applying a hash function on the random string sent by the servers.

Although there are numerous ways in which IPSec can be implemented, most implementations use both symmetric and asymmetric cryptography. Asymmetric cryptography is used to authenticate the identities of both parties, while symmetric encryption is used for protecting the actual data because of its relative efficiency (Frankel *et al*., 2005).

It is essential to realize that VPNs do not remove all risks from networking. While VPNs can greatly reduce risk, particularly for communications that occur over public networks, they cannot eliminate all risks for such communications. One potential problem is the strength of the implementation. For example, flaws in an encryption algorithm or the software implementing the algorithm could allow attackers to decrypt intercepted traffic; random number generators that do not produce sufficiently random values could provide additional attack possibilities. Another issue is encryption key disclosure; an attacker who discovers a key could not only decrypt traffic, but potentially also poses as a legitimate user. Another area of risk involves availability. A common model for information assurance is based on the concepts of confidentiality, integrity and availability. Although VPNs are designed to support confidentiality and integrity, they generally do not improve availability, the ability for authorized users to access systems as needed. In fact, many VPN implementations actually tend to decrease availability somewhat because they add more components and services to the existing network infrastructure (Frankel *et al*., 2005).

In our case, the protocols of Virtual Private Network (VPN) can negotiate with a Trust Security System (TSS). TSS supports a safe communication channel between security nodes in the internet. It furnishes authentication, confidentiality, integrity and access control to secure nodes in order to transmit data packets with IPSec protocol. Our TSS consists of Key Distribution Protocol (KDP) block, Security Involvement (SI) block and IPSec engine block. The KDP block negotiates hash function and key used in IPSec engine block. SI blocks setting-up and manages security association information. IPSec engine block treats IPSec packets and consists of networking functions for communication. The IPSec engine block should embodied by the hardware and in-line mode transaction for high speed IPSec processing. Our concept is based on the high speed security processing that supports our protocol for key distributions and in-line mode. This effort is one of method to provide safe data communication in a network environment. However, it is a very difficult problem to provide data communication of high speed and at the same time is safe on network environment such as present.

The Key Distribution Process (KDP) using hash functions: In this phase of KDP-6DP protocol is used to generate the shared key locally on-site at the servers' stations. To make the sharing key locally in KDP-6DP more secure we added the value of j as the function in sharing key and this value can be used to choose random combinations of two cascade hash functions stored. The selected hashes, for example, can show below in the Table 2.

The KDP-6DP protocol employs long-term and short-term distribution keys. This phase of this protocol can be use in three different techniques or modes. Also for more understanding this phase of our protocol, we performing table for atomic actions summary see it in Appendix. These modes we describes as follows.

First Mode (Four Exchanges sub-keys in Key Distribution Process (KDP)): In this technique or mode, the sender (here Alice as server of VPN) sends a random string S_A to the receiver (here is Bob acting as server of VPN also). At the same time Bob sends a random string S_B to Alice. In Both Alice and Bob stations the operation $K_s = S_A$ XOR S_B is performed, where K_s is the temporary key used only for verification. In addition, at the same time Both Alice and Bob exchange their sub-keys (K_{sA} send to Bob and K_{sB} send to Alice). After they receive the temporary keys, they start at the same time comparing it with their own temporary keys. If these temporary keys are not equal, stop and restart from the first step of the key generating process. If these temporary keys are equal, the two parties start get to j, which is the value of the selection of the cascade hash functions according Table 2 and get keys (K) in two stations:

$$\text{Key } (K^{AS}) = h_j (\alpha, \beta, \varsigma, K_{sA}) \text{ and Key } (K^{BS}) = h_j ((\alpha, \beta, \varsigma, K_{sB})$$

Table 2: An example of selecting a cascade of two hashes

j	Sample selection of cascaded hash functions
0	h_4, h_7
1	h_3, h_6
2	h_2, h_5
3	h_1, h_4
4	h_0, h_3
5	h_5, h_2
6	h_6, h_1
7	h_7, h_0

Next Bob by using these keys (K) gets the massage after decrypting the cipher by $M = D_K (C_M)$. This technique or mode algorithm we can write it as below:

First mode algorithm:

Start;

Generate random string S_A in Alice station;

Generate random string S_B in Bob station;

Send S_A and S_B to both sides;

Produce temporary keys K_{sA} and K_{sB} by S_A XOR S_B in both stations;

Send K_{sA} and K_{sB} to both sides and compare it with opposite temporary keys;

If K_{sA} and K_{sB} are not equal, stop and go to Start;

Else if K_{sA} and K_{sB} are equals get j in both sides (value of cascade hash functions);

By values of j, K_{sA}, K_{sB} and initial values in both stations can get the final Key (K);

By the final Key (K) can encryption the massage in Alice station and by this Key (K) can decryption the cipher in Bob station;

If new key sharing session, Go to start from the first step and generate random string S_A and S_B in both sides;

Else, Go to end;

End.

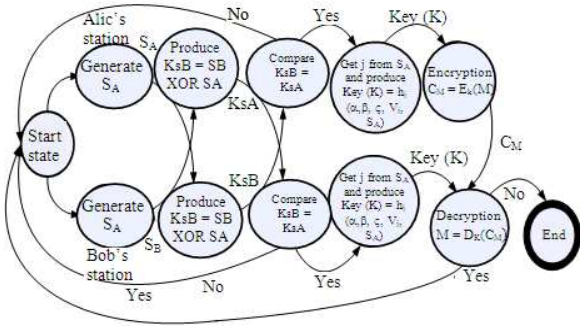

Fig. 3: The infinite data state diagram of the Key Distribution Process (KDP) first mode using random string between Alice and Bob

In addition, the entire process of this mode of key distribution shows in the Fig. 3 by an infinite data state diagram concept.

We notice that Alice and Bob are verifying the correctness of the temporary key and that no transmission errors have occurred while exchanging S_A and S_B, moreover, we can select a different hash every time a new session is start. The hash is selected from a value in a secret field of the temporary key K_s.

Second Mode (one exchange sub-keys in Key Distribution Process (KDP)): In this technique or mode, by assuming error detection and correction supported channel, this mode in our protocol can work in a second mode that is called the short mode. This mode starts when Alice generates and sends a random string S_A to Bob. At the same time both Alice and Bob, get the value of j from S_A. The j value is employed as a pointer to select the cascade of the hash functions. It produces Keys (K) in two stations (K^{AS} and K^{BS}) at the same time. Next Bob by using these keys (K) gets the massage after decrypting the cipher by $M = D_K (C_M)$. This technique or mode algorithm we can write it as below:

Second mode algorithm:

Start;

Generate random string S_A in Alice station and waiting state in Bob station;

Send S_A to both sides;

Get j in both sides (value of cascade hash functions) from S_A;

Get the final Key (K) from values of j, S_A and initial values in both stations;

By the final Key (K) can encryption the massage in Alice station and by this Key (K) can decryption the cipher in Bob station;

If new key sharing session, Go to start from the first step and generate random string S_A in Alice side;

Else, Go to end;

End.

In addition, the Fig. 4 shows the infinite data state diagram of the Key Distribution Process (KDP) using one exchange.

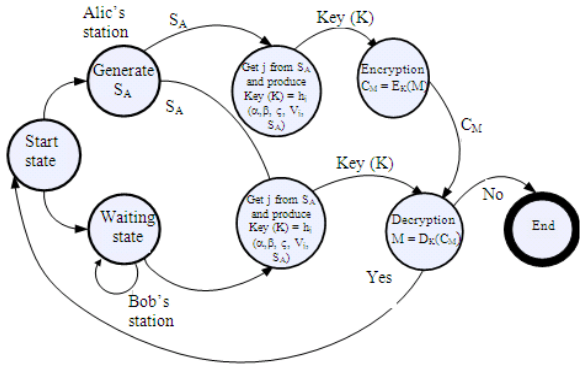

Fig. 4: The infinite data state diagram of key distribution process KDP using one string generated at Alice station

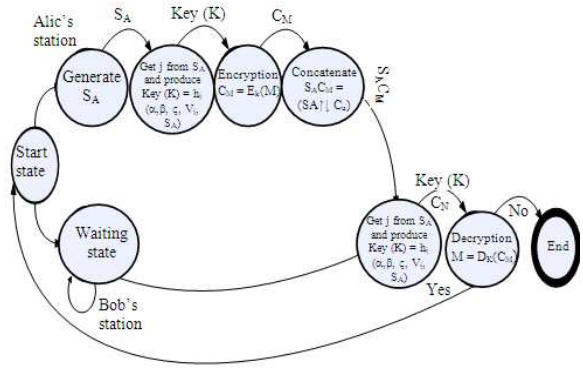

Fig. 5: The infinite data state diagram of Key Distribution Process (KDP) using one string generated, encryption message and concatenate it at Alice Station, after that send Bob station to get message

The Third Mode (one-time in Key Distribution Process (KDP)): In this technique or mode, we called this mode is a one-time mode. It starts when Alice generates a random string S_A locally. From S_A, we get the j value after that the Key (K^{SA}) = h_j ((α, β, ς, S_A) is generated. Then Alice encrypt the massage by using of K^{AS} and the resulting cipher is $C_M = E_K(M)$ is concatenated with S_A to be S_AC_M. The process performed at Alice's station and Bob is waiting in the whole time. After that, Alice sends S_AC_M to Bob. Bob receives S_AC_M and from S_A he gets the value of j, then the Key (K^{BS}) = h_j ((α, β, ς, S_A) is generated. Next Bob gets the massage after decrypting the cipher by $M = D_K$ (C_M). This technique or mode algorithm we can write it as below.

Third mode algorithm:

Start;

Generate random string S_A in Alice station, Bob station waiting;

Get j (value of cascade hash functions) from S_A, Bob station waiting;

Get the final Key (K) from values of j, S_A and initial values, Bob station waiting;

By the final Key (K) can encryption the massage and get cipher C_M, Bob station waiting;

Concatenate with S_A with C_M to be S_AC_M and send to Bob station;

Get j (value of cascade hash functions) from S_A and get the final Key (K) from values of j, S_A and initial values in both stations;

By the final Key (K) and cipher C_M, can Bob decrypt the cipher to get massage M;

If new key sharing session, Go to start from the first step and generate random string S_A in Alice side;

Else, Go to end;

End.

In the Fig. 5 shows the infinite data state diagram of this mode.

DISCUSSION

In the previous materials, we have established the following: Firstly, the authentication is the most obvious problem in key distribution protocols. In this study, we propose an out-of-band authentication using a non-classical channel. The authentication process is performed at random intervals. Next, once the authentication process using 6DP is completed, a signal enables the key generation process. Therefore, it is difficult for an eavesdropper Eve to estimate the time for her attack on the quantum channel by the commonly known quantum attacks. In addition, she cannot estimate the quantum states between the sender and the receiver.

Secondly, for added security, the VPNs provide an active form of IPSec security concept by:

- Authentication: Verifies that the packet received is actually from the claimed sender

- Integrity: Ensures that the contents of the packet did not change in transit
- Confidentiality: Conceals the message content through encryption

Thirdly, in our three modes (techniques) of Key Distributions Process (KDP) based on hash functions, we analysis some advantages and disadvantage that are related with it. The advantages are:

- The speed in first and second techniques or modes depends on the communication channel (on VPN traffic delay). The second and third modes the speed depends on the clients and servers stations, i.e., microprocessor speed and the capacity of memory
- It is relatively easy detect and correct error in real-time for the first mode only
- The third mode can provide the key and the massage quickly and this technique is more secure. In this mode, just one communication transaction used to transmit the random string S_A and concatenate it with the cipher

The ostensible issues of these techniques are:

- The first and second modes are slow depending on VPN's speed
- The second and third modes are relatively difficult implement error detection and difficult correction
- The authentication for three modes is an issue. Therefore, we propose that the authentication is to be performed employing a quantum channel for the servers and authentication by IPSec concept for the clients using a VPN

CONCLUSION

The comparison between Quantum Authentication (QA) in our protocol and quantum Key Distribution Protocols (QKD) from a cost point of view, we believe the QA is less expensive than QKD. The QKD is employed point-to point between two parties and when we need more nodes, we must establish the complex quantum network and this network need a new infrastructure and it is expensive and it cannot use the current communication infrastructure. In (KDP-6DP) protocol, one can use the current communication infrastructure and using hash functions by cascade form to generate the key distributions. We believe that this protocol is secure even in the presence of an eavesdropper who has access to the classical and the quantum channels.

REFERENCES

Barnum, H.N., 1999. Quantum secure identification using entanglement and catalysis. ArXiv Inc. http://arxiv.org/abs/quant-ph/9910072

Barnum, H., E. Knill, G. Ortiz and L. Viola, 2003. Generalizations of entanglement based on coherent states and convex sets. Phys. Rev. A., 68: 1-21.

Bennett, C.H. and G. Brassard, 1984. Quantum cryptography: Public key distribution and coin tossing. Proceedings of the IEEE International Conference on Computers Systems and Signal Processing, Dec. 10-12, IBM Press, Bangalore India, pp: 175-179. http://www.research.ibm.com/people/b/bennetc/bennettc198469790513.pdf

Bennett, C.H., G. Brassard, C. Crepeau and U.M. Maurer, 1995. Generalized privacy amplification. IEEE Trans. Inform. Theory, 41: 1915-1923.

Dusek, M., O. Haderka, M. Hendrych and R. Myska, 1999. Quantum identification system. Phys. Rev. A., 60: 149-155.

Huttner, B., N. Imoto and S.M. Barnett, 1996. Short distance applications of quantum cryptography. J. Nonlinear Opt. Phys. Mater., 5: 823-832.

Frankel, S., K. Kent, R. Lewkowski, A.D. Orebaugh and R.W. Ritchey *et al*., 2005. Guide to IPsec VPNs. National Institute of Standards and Technology. http://csrc.nist.gov/publications/nistpubs/800-77/sp800-77.pdf

Shaari, J.S., M. Lucamarini and M.R.B. Wahiddin, 2006. Deterministic six states protocol for quantum communication. Phys. Lett. A., 358: 85-90.

Zeng, G.H. and W.P. Zhang, 2000. Identity verification in quantum key distribution. Phys. Rev. A., 61: 22303.