International Journal of Engineering Research and General Science Volume 3, Issue 2, Part 2, March-April, 2015
ISSN 2091-2730

Efficient High Capacity Quantum Cryptography Based Key Distribution in WI-FI Network

Ajish S , Assistant Professor
Department of Computer Science and Engineering
College of Engineering, Perumon, Kollam, India, ajishs2014@gmail.com

Abstract— There are a large variety of kinds of mobile wireless networks, Wi-Fi, based on the IEEE 802.11 standard, is a wireless local area network, mainly used in offices and campus at universities or in meeting rooms. For such limited coverage area, IEEE 802.11 standard may be observed as building oriented environment, which potentially offers a chance to let quantum key distribution (QKD) play a role in the security of wireless communications. In fact, secured data transmission is one of the prime aspects of wireless networks as they are much more vulnerable to security attacks. We can explore the possibility of using Quantum Key Distribution (QKD) for authentication and data encryption for IEEE 802.11 standard. Quantum key distribution is based on the laws of quantum physics which ensure that nobody can measure a state of an arbitrary polarized photon carrying information without introducing disturbances which will be detected by the legitimate user. In the new protocol (BB84) the existing 4-way handshake of IEEE 802.11 has been replaced with the QKD based 4-phase handshake. In the BB84 protocol each photons carry only one bits. So to improve the efficiency in Efficient High Capacity Quantum Cryptography based Key Distribution in WI-FI Network each photons carry two bits. The channels used in the QKD are quantum channel and classical channel. The polarized photons are transmitted through the quantum channel and the classical channel deals with recovering identical secrete keys at both ends. The classical channel comprises of four stages, namely Sifting, Error Estimation, Reconciliation and Privacy Amplification. The Quantum Bit Error Rate (QBER) is the measurement of the error probability in the key distributed via the quantum channel. This value allows the users to estimate the maximum amount of information that an eavesdropper could have on the key.

Keywords— IEEE 802.11; Quantum Key Distribution; 4-phase Handshake Protocol; BB84 Protocol; Quantum Channel; Classical Channel.

I. INTRODUCTION

While wireless networks and their applications are becoming popular every day, associated security issues have become a great concern. It could be seen that the key used for data encryption plays a major role to the security of the wireless communication. The amended version IEEE 802.11i of 802.11[1] standard has been using the process known as 4- way handshake to exchange the key between the two parties. It was shown that 4-way hand shake is subject to security issues. The main key obtain through 4-way handshake is the Pair-wise Transient Key (PTK). PTK is used to build the key hierarchy containing few other keys that are needed for other encryptions of 802.11. Thus it is essential to have the PTK distributed safely.

It is well known that from the laws of physics, a key distributed via Quantum Key Distribution (QKD) offers unconditional security between two communication parties. Instead of using 4-way handshake, QKD has been used to distribute the secrete key (PTK) in Wi-Fi network. Since the key obtained via QKD [11] provide unconditional security, 802.11 key hierarchies will inherit the same level of protection enabling secure communications. The QKD comprises of two channels: Quantum Channel and Classical Channel. Quantum channel is used to transmit series of polarized photons representing the key bits that are to be sent to the receiver. Quantum transmission normally associated with errors that are introduced as a result of atmospheric conditions, dark counts of photon apparatus etc, and most importantly eavesdropping. The foundation of quantum cryptography lies in the Heisenberg uncertainty principle, which states that certain pairs of physical properties are related in such a way that measuring one property prevents the observer from simultaneously knowing the value of the other. Thus any intervention of an eavesdropper can easily be recognized by the alterations introduced to the measurements of polarized photons. The Quantum Bit Error Rate (QBER) [2] is the measurement of the error probability in the key distributed via the quantum channel. The QBER can be defined as the ratio of an error rate to the key rate and contains information on the existence of an eavesdropper and how much such eavesdropper knows. This value allows the users to estimate the maximum amount of information that an eavesdropper could have on the key. It serves as an input to the key distillation protocol that transforms raw keys into the secret key.

The classical channel, being the 802.11 wireless network, is used to recover the final key after removing the errors introduced during transmission. In order to make all the system works well, the classical channel comprises of four stages, namely Sifting, Error Estimation, Reconciliation and Privacy Amplification. During the sifting phase Supplicant (STA) and Access Point (AP) communicates to keep the bits that recorded against the correct bases that used to polarize the bits. In the error estimation phase they estimate possible error level by comparing a sample of bits obtained from their key. In the reconciliation phase, STA and AP remove all the errors present in their keys. Finally, in privacy amplification, they apply a chosen hash function to eliminate possible information that may have leaked to a third party. The modifications are mainly focus on the Counter-Mode/CBCMAC Protocol (CCMP) data confidentiality protocol. The existed 4-way handshake has been replaced by the QKD based 4-phase handshake.

Modifications are proposed in two places of the existing 802.11 protocol: firstly, during the association process to negotiate QKD specific parameters to be used, secondly, the 4-way handshake protocol. These changes only require few field level modifications thus the existing frame structure remains intact.

II. 4-WAY HANDSHAKE

The 4-way handshake is started by the Authenticator by sending a value ANonce (Authenticator Nonce) to the Supplicant. Upon receiving the value ANonce, the Supplicant generates the value SNonce (Supplicant Nonce) and has all materials to build the key hierarchy. To build the Pair-wise key hierarchy, the Supplicant uses a Pseudo Random Function (PRF) to derive the PTK of 384 bits (for CCMP) or 512 bits (for TKIP) from the PMK, the MAC(Medium Access Control) address of the Authenticator (AMAC), the MAC address of the Supplicant (S-MAC), the ANonce, and the SNonce. The PTK is then split into a KEK of 128 bits, a KCK of 128 bits, and a TK of 128 bits (for CCMP) or 256 bits (for TKIP). However, this key hierarchy is not used until the Authenticator is authenticated and ready to use these keys.

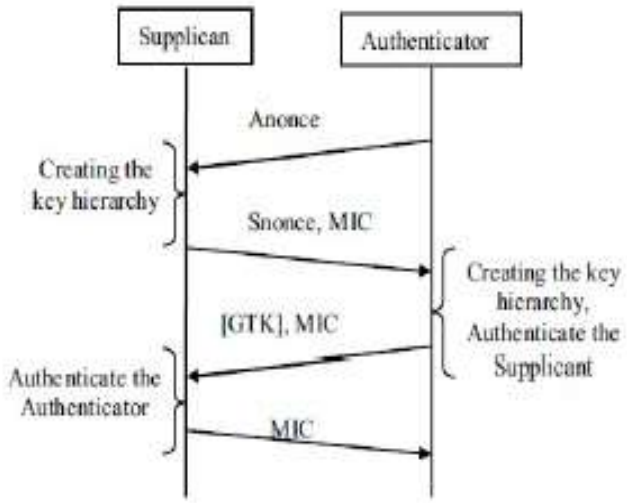

Figure 1. 4-way Handshake.

In the second message of the 4-way handshake, the Supplicant sends to the Authenticator the value Snonce and a MIC calculated based on the content of the message and the KCK which has just derived. The algorithm used to calculate the MIC is HMAC-MD5 or HMAC-SHA1-128 [4] depending on the cipher suite chosen for the system. Upon receiving this message, the Authenticator has all materials to build the same key hierarchy. Then it uses the KCK to check the MIC. If the MIC is correct, that means that the Supplicant obtains the PMK, and thus the Supplicant is authenticated.

In the third message of the 4-way handshake, the Authenticator tells the Supplicant that it has finished the derivation of the key hierarchy. It also sends a MIC calculated based on the content of the message and the KCK which has just derived. Upon receiving this message, the Supplicant checks the MIC in order to verify that the Authenticator obtains the PMK, and thus authenticates the Authenticator. Then, the key hierarchy can be used without the doubt about the authenticity of the access point. The third message of the 4-way handshake can be used by the access point as a means to distribute the GTK to the mobile terminal. In this case, the GTK is sent encrypted using the KEK in the key hierarchy just derived.

The last message of the 4-way handshake is for the purpose of synchronization. The Supplicant tells the Authenticator that the 4-way handshake is now successfully completed and both can turn on the encryption of user data. This message also includes a MIC to assure the Authenticator that this message is sent by the Supplicant and that it is not modified. After the 4-way handshake, the Temporal Key (TK) is used by the encryption algorithm to provide confidentiality and the integrity of user data.

III. MODIFIED BB84 PROTOCOL

BB84 was introduced by Bennet and Brassard [3] in 1984, thus it was named BB84. BB84 is a nondeterministic protocol, which means that it is useful only for the distribution of a random sequence. BB84 is a four state protocol. Other protocols can be a two-state protocol (e.g. the B92), a three-state protocol or a six-state protocol. The BB84 and B92 protocols are nowadays widely used. These protocols are securely proven and largely experimented.

The BB84 protocol is used for the integration of quantum cryptography in 802.11 networks. The operating mode of BB84 protocol consists on two main steps : Quantum transmission as presented in Figure 2, and public discussion.

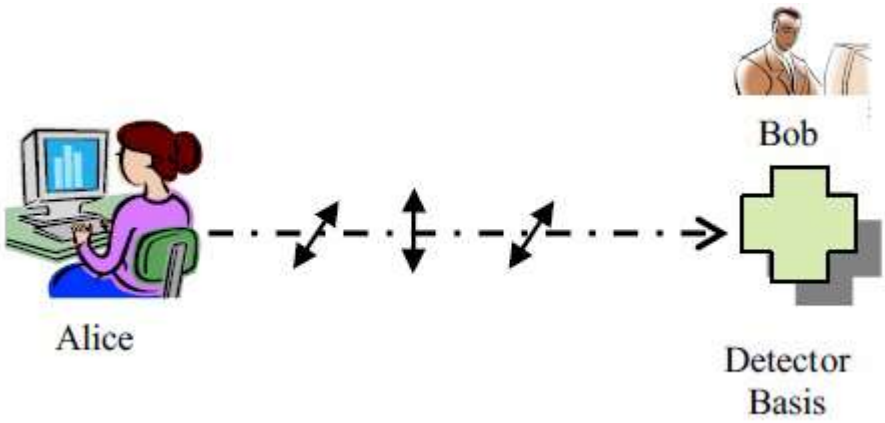

Figure 2. Photon Exchange.

Polarization Direction	Bit Representation
0	00
Π/4	01
Π/2	10
3Π/4	11

Table I
Polarization Direction of Photons and Corresponding Bit Representation

In the phase of quantum transmission, the information is encoded in non-orthogonal quantum states. This could be a single photon with a polarization direction of 0, π/4, π/2 or 3π/4. The sender and the receiver must agree first on the meaning of the photon polarizations for instance 0 for a binary 00, π/4 for binary 01, π/2 for 10 and 3π/4 for a binary 11.The polarization direction of photons and the corresponding bit representation is shown in Table I.

The sender (Alice) generates a random bit string and a random sequence of polarization bases then sends the receiver (Bob) photon by photon. Each photon represents two bit of the generated bit string polarized by the random basis for this bit position. When receiving photons, Bob selects the polarization filters (rectilinear or diagonal) to measure the polarization of the received photon.

In the phase of public discussion, after finishing the quantum transmission Bob reports the bases that he picked for each received photon. Alice checks Bob bases and says which ones were correct as described in Figure 3.

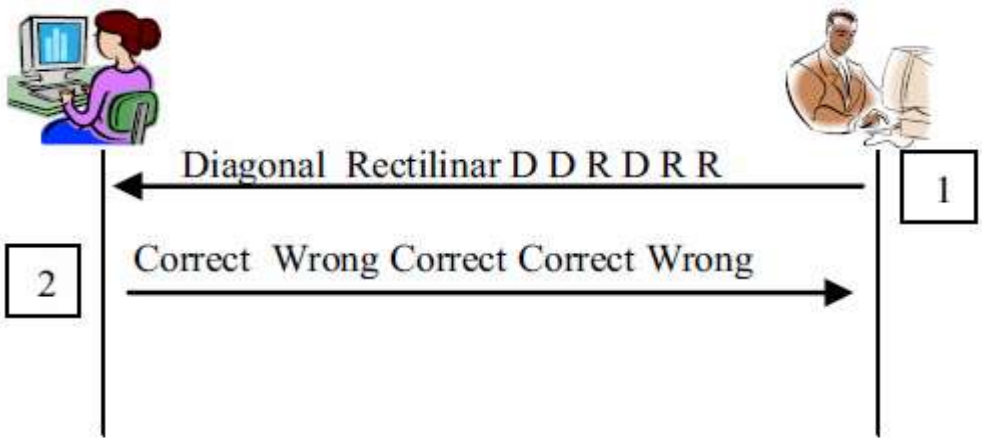

Figure 3. Validation of Bob Bases.

Bob and Alice take the bits resulting from these correct bases, these bits are only known by Alice and Bob. At this moment Alice and Bob share a secret bit string. This exchange is unconditionally secure providing that there is no eavesdrop or active attack and that the quantum channel is perfect. However, as an attack is always possible and the quantum channel is usually imperfect, an additional step is used to estimate the error rate. In this step, Bob chooses a random sequence of testing bits and sends it back to Alice. Alice checks whether these bits are in conformity with those sent by Alice originally. If there is an attack on the quantum channel the error rate will be about 25%or higher. In this case, Alice and Bob detect the eavesdropper. Otherwise, i.e. the error rate is less than 25%, the two parties discard the revealed bits and take the resulting stream as the secret key. The secrecy of this final stream is unconditional.

Other steps could be applied to enhance the secrecy and generalize the unconditional security of key exchange. These steps are done mainly by error correction and privacy amplification.

IV. PROPOSED 4-PHASE HANDSHAKE PROTOCOL

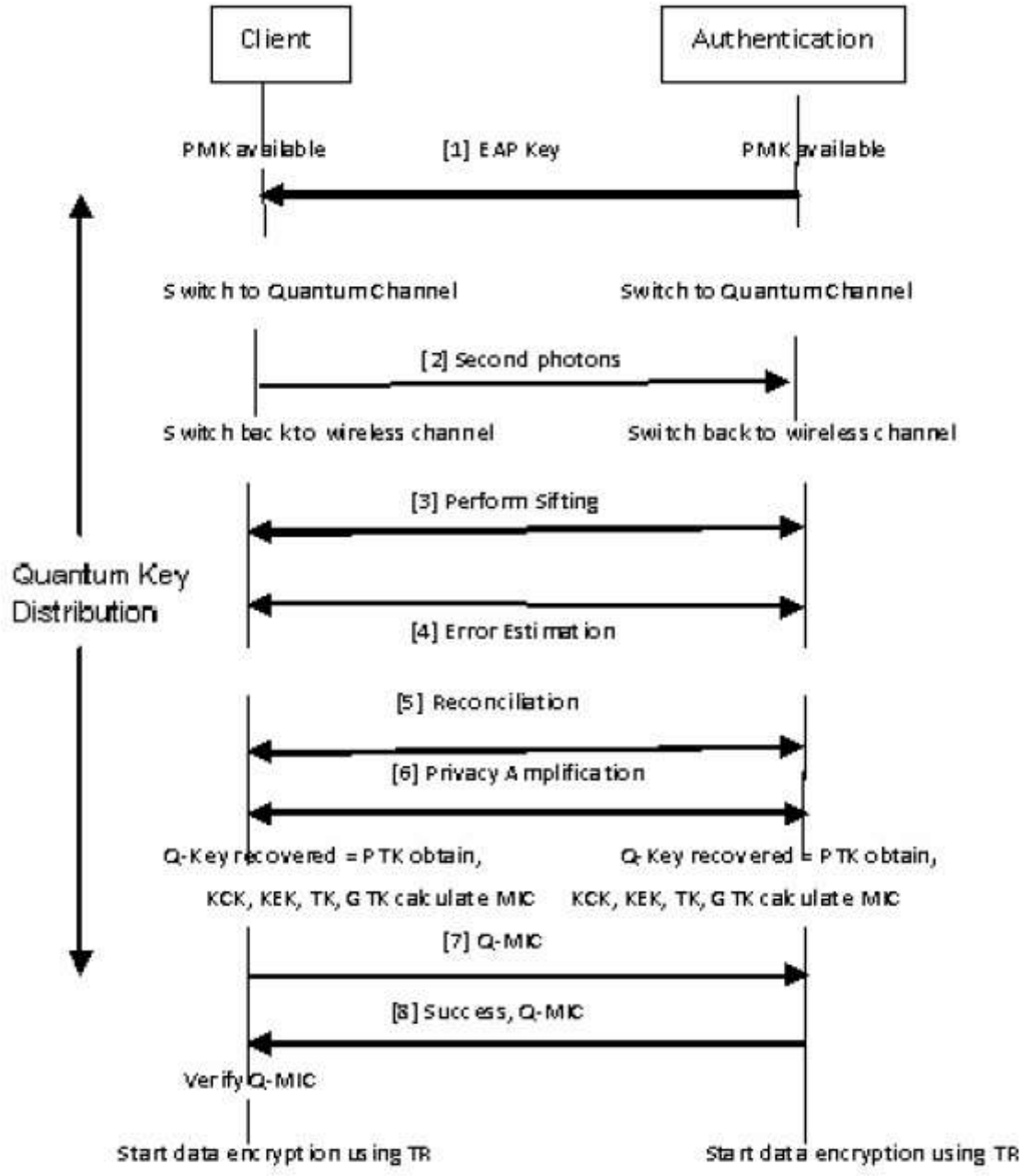

Figure 4. The Proposed 4-Phase Handshake Protocol.

In the proposed protocol, the existing 4-way handshake of IEEE 802.11i has been replaced with the QKD based 4-phase handshake as shown in Figure 4. The last message of IEEE 802.1X authentication is the delivery of EAP key as in flow (1). Both parties are in possession of Pair-wise Master Key (PMK) at the end of this message. At this point the proposed QKD based Wi-Fi protocol begins. The communication switches to quantum channel and the photon transmission takes place (flow (2)) from the STA towards the AP. Once the quantum transmission finishes, the communication channel switches back to the wireless channel. Afterwards the final key recovery process begins by executing the 4 phases as shown in flows (3) to (6).

In the first phase (sifting) AP announces the bases it used to interpret the bits received from STA. With this information, they only keep the bits that are recorded against the matching bases. In the next phase, error estimation as shown in flow (3), the STA sends a random sample of its key to AP. AP then compares these bits with its copy of the key can calculate the error level. In the next phase (flow (5)), they extract the final secured key by implementing a reconciliation protocol such as Cascade, Winnow, parity check etc. At the end of this phase both parties hold identical keys, but may not completely secure. In the last phase, privacy amplification (flow (6)), they apply a selected hash function to extract the final key which has been proven to be unconditionally secure. We take this key as the PTK. Knowing the PTK, rest of the 802.11i key hierarchy consisting of KCK (Key Confirmation key), KEK (Key Encryption key), TK (Temporal Key) and GTK (Group Temporal Key) can be retrieved. The TK is used to encrypt data for the subsequent data communication.

A. Sifting

The authenticator starts the public discussion step by announcing the N bases that it used to receive the N photons. The first message of the public discussion is sent to the supplicant over the radio link and appended with a MIC calculated based on the content of the message and the KCK just established. This MIC assures the integrity and the authenticity of the message.

Upon receiving the bases announcement of the authenticator, the supplicant compares the bases used for the sent photon and those used for the received photons. Assuming that there are M ($M < N$) photons which are sent and received with the same basis. In the second message of the public discussion, the supplicant tells the authenticator the M bases which were correct.

B. Error Estimation

The supplicant and the authenticator keep only the M bits corresponding to the M correct bases. These bits can be the shared secret information if there is no eavesdrop and the quantum channel is perfect without noise. However, eavesdrop is always possible and the quantum channel is usually noisy. The supplicant and the authenticator will detect the probably happened eavesdrop based on an error rate estimation. For this task, the authenticator randomly selects P testing bits ($P < M$) among the remaining M bits. P can be one third of M following the BB84 protocol. In the third message of the public discussion, the authenticator reveals the values of the P testing bits to the supplicant.

In theory the photons sent and received with the same basis should yield the same information value. The authenticator and the supplicant should have the same values for the P testing bits. In practice, the photon's polarization can be changed during the transmission over the quantum channel by the presence of eavesdropping or the noise of the quantum channel, leading to the disagreement on the values of the testing bits. A bit 1 encoded by a photon which is sent and received with the same basis can be decoded into a bit 0. Upon receiving of the values of the P testing bits, the supplicant compares them with the values of their original values. In the fourth message of the public discussion, the supplicant confirms the values of the P testing bits with the authenticator. The error rate is calculated as follows.

$$Er = \frac{\text{Number of disagree testing bits}}{P}$$

If the error rate Er is smaller than a threshold Emax, we can conclude that there was no eavesdrop and the error bits are caused by the imperfection of the quantum channel. Otherwise, the quantum transmission was eavesdropped and the photon measurement of the eavesdropper caused an unusual high error rate to quantum transmission. The value of Emax depends on the quantum transmission quality of specific QKD systems. If the quantum transmission is concluded "no eavesdropping" after the estimation of the error rate Er, the P testing bits are removed from the M bits. The remaining M-P bits are used as the sifted keys Kr shared between the supplicant and the authenticator, finishing the BB84 procedure. If eavesdropping is detected, the transmitted photon cannot be used. The Quantum handshake is terminated without establishing necessary keys.

C. Reconciliation

The procedure described for Alice and Bob to reconcile their bits takes place over a public channel. Since Eve presumably listens to all public transmissions, Alice and Bob must reveal as little information as possible while still ensuring that they end up with identical keys. The can do this by agreeing upon a random permutation of the bits in their strings (to randomize the locations of errors) and then splitting the resulting string into blocks of size b. The constant b is chosen so that each block is unlikely to contain more than one error. In the BBBSS implementation,

was chosen by experiment rather than theory. Alice and Bob then compare the parity of each block. If they find a pair of blocks with mismatched parities, they continually bisect the block into smaller and smaller blocks, comparing parities each time, until the error is found. To ensure that Eve learns nothing from this process, Alice and Bob discard the last bit of each block whose parity they disclose.

After completing this process once, there will still be mismatches in those blocks which happened to contain an even number of errors. So Alice and Bob repeat the process several more times with increasing block sizes until they believe the total number of errors to be low. At this point, the above strategy becomes inefficient because Alice and Bob must discard a bit for each block they compare, and the probability of finding an error in each block is low. So Alice and Bob switch to a new strategy, which they again perform multiple times. Each time, they choose a random subset of the bit positions in their complete strings, and compare parities. The probability of disagreement if the subset strings are not identical is exactly 1/2. If a disagreement occurs, a bisective search for the error is performed, this time using random subsets rather than blocks. The last bit of each subset is discarded. Eventually, all the errors will have been removed, and Alice and Bob will go through enough parity checks without discovering any errors that they may assume their strings are identical.

D. Privacy Amplification

After Reconciliation Alice and Bob posses identical strings, but those strings are not completely private. Eve may have gained some information about them either by beam splitting or through intercept/resend. Although this second strategy may cause some errors in Bob's string, if Eve uses it on only a small number of bits, the induced errors will be lost among the errors caused by noise in the detectors and other physical problems. During the reconciliation phase, Eve did not gain any information, since the last bit of each parity check set was discarded. However, some of her original information about specific bits may have been converted to information about parity bits. For instance, if she knew the value of a bit x in string y, and Alice and Bob revealed the parity of y and discarded x, Eve would then know the parity of the remaining bits of y. If we say that Eve knows a parity bit about a string if she knows the parity of a non-empty subset of that string, then if Eve started out knowing at most k physical bits of the key, she will know at most k parity bits of the key after reconciliation.

In any case, Alice and Bob share an n-bit string S, and we will suppose that Eve knows at most k deterministic (i.e. parity or physical) bits of S. Alice and Bob wish to compute an r-bit key K, where r <n, such that Eve's expected information about K is below some specified bound. To do so, they will choose a compression function g: $\{0,1\}^n \rightarrow \{0,1\}^r$ and compute K = g(S).

Definition: A class G of functions A → B is universal if for any distinct x1 and x2 in A, the probability that g(x1) = g(x2) is at most 1/|B| when g is chosen at random from G according to the uniform distribution.

An universal class is the set of permutations of A onto itself, since for any g in the set, the probability that g(x1) = g(x2) is zero, which is less than 1/|A|. If Eve knows k deterministic bits of S, and Alice and Bob choose their compression function g at random from a universal class of hash functions $\{0; 1\}^n \rightarrow \{0; 1\}^r$ where r = n-k-s for some safety parameter 0 < s < n-k, then Eve's expected information about K = g(S) is less than or equal to 2-s/ln2 bits. One such hash function to generate K is for Alice and Bob to compute an additional r random subset parities of S, this time keeping the results secret. The r results of the parities will be the final r-bit key.

Given this result, one might ask how Alice and Bob are to determine the value of k, i.e. how much information has been leaked to Eve. As a conservative estimate, they can simply assume that all transmission errors were caused by eavesdropping (although most likely some came from detection errors). Eavesdropping errors could come from either intercept/resend or beam splitting. Alice and Bob can use the beam intensity m and the bit error rate to calculate the expected fraction of S that Eve has learned. If they are conservative in their assumptions and add several standard deviations to their results, they will have a safe upper bound on the number of bits leaked to Eve.

The above discussion assumes that Eve knows only deterministic bits, so another issue is whether it might be more useful to her to obtain probabilistic information about S instead. In other words, rather than measuring photons in the same bases as Alice and Bob, she could pick a basis halfway in between them. This will give her a result that matches Alice's with probability approximately 85%, regardless of which basis Alice uses. She will not gain any information when Bob reveals his measurement choices, so with this strategy all of her information is probabilistic rather than deterministic. Conceivably, this probabilistic information could be more resistant to privacy amplification than deterministic information. However, it turns out that this is not the case, so if Eve wishes to optimize her expected information on the final key, she should use the same bases as Alice and Bob, obtaining only deterministic bits.

V. EVALUATION OF RESULTS

In the evaluation process SAGR04 is used as the QKD protocol and parity based bisect method as the reconciliation protocol. These test data has been chosen to cover different error rates, enabling to simulate worst case scenarios as well.

The length of PTK for CCMP is 384 bits. Hence the aim is to obtain 384 bit long final key after removing any errors introduced during the quantum transmission. The main contributor to the errors in the quantum transmission is the eavesdroppers. In addition, environmental conditions and poor quality photon apparatus too could contribute to the errors. By using the modified BB84 protocol the time for Quantum Key Distribution in WIFI network can be reduced to half of that used by BB84 protocol.

The four phases of 4-phase handshake is analyzed against different input data. These data consists of key lengths vary between 400 to 800 bits with error rates of 10%, 20%, 30% and 40%.

A. Analysis of Sifting Phase

During the sifting phase, AP and STA agreed on the bits that are recorded against matching bases. This phase consists of two EAPOL communication flows: AP informing STA the bases used, STA then informs the correct bases back to AP. During this simulation, different key lengths with various error rates have been used as the input to sifting model to calculate the time taken to complete the sifting phase.

The analysis shows that irrespective of the key length, the time taken pattern for sifting is very much identical across different error rates. In the worst possible case the longest key length of the key is 800 classical bits. This means there are 800 bases used to be transmitted. By using the Modified BB84 protocol only 400 photons are need to be transmitted for a key length of 800 bits, so the time taken for Shifting Phase can be reduced to 1/4(AP informing STA the bases used, STA then informs the correct bases back to AP). This number can be easily fitted into one EAPOL frame via the key data field of new information element.

B. Analysis of Error Estimation

The main task of error estimation process is to estimate the errors of the mapped key string after quantum transmission. This estimation is done by selecting a random sample of bits from their keys to find out how many of them are in error. This calculated error rate is then compared against the QBER of the quantum transmission. The QKD project work carried out in parallel with this work has achieved QBER levels between 4% - 8.1% in free space.

Like in sifting phase, error estimation comprises of two EAPOL communication flows: one to inform the sample of bits used for comparison, the other to inform the result. Yet again same set of test data has been used to analyze the performance of the error estimation phase. The below table shows the statistics recorded against these input values.

Key Lengths (bits)	**Time taken to complete Error Estimation with various error rates (ms)**			
	10%	**20%**	**30%**	**40%**
400	0.63	0.72	0.80	0.97
500	0.68	0.79	0.90	1.01
600	0.73	0.82	0.98	1.10
700	0.81	0.90	1.01	1.19
800	0.97	1.03	1.11	1.27

The analysis shows that the overall performances of error estimation are similar for different key lengths and error levels. As expected, when the size of the key and the error rate increases, the time taken to complete the error estimation also increases.

C. Analysis of Reconciliation

Reconciliation is the most critical phase of 4-phase handshake protocol where STA and AP remove all the errors present in their respective keys to recover the final identical key. The parity based bisect method chosen in this work consumes more messages flows than other reconciliation protocols. The main reason to choose bisect method is to observe the behavior of 4-phase handshake protocol in worst possible circumstances. The time taken to complete reconciliation process depends on several key factors:

1. Length of the main key.
2. Length of the initial block size of the partition.
3. Number of cycles the parity check will run for.

Fig.5 shows time to complete the reconciliation has been calculated at various error levels for a fixed key length of 500 bits. Block size refers to the size of the partition that the main key is divided to perform parity checks. This block size gets reduced if any block is found to have parity mismatch. In such cases, respective blocks are bisected and parity check continues.

It could be seen that smaller the bisect block size, quicker the completion of reconciliation process. With initial block sizes of 4, 8 and 16 m reconciliation completes within 5 ms time frame which is quite acceptable considering the amount of work involved in removing the errors. Main reason to this result is that when the initial block size is smaller, the errors can locate more quickly. Further, if a parity

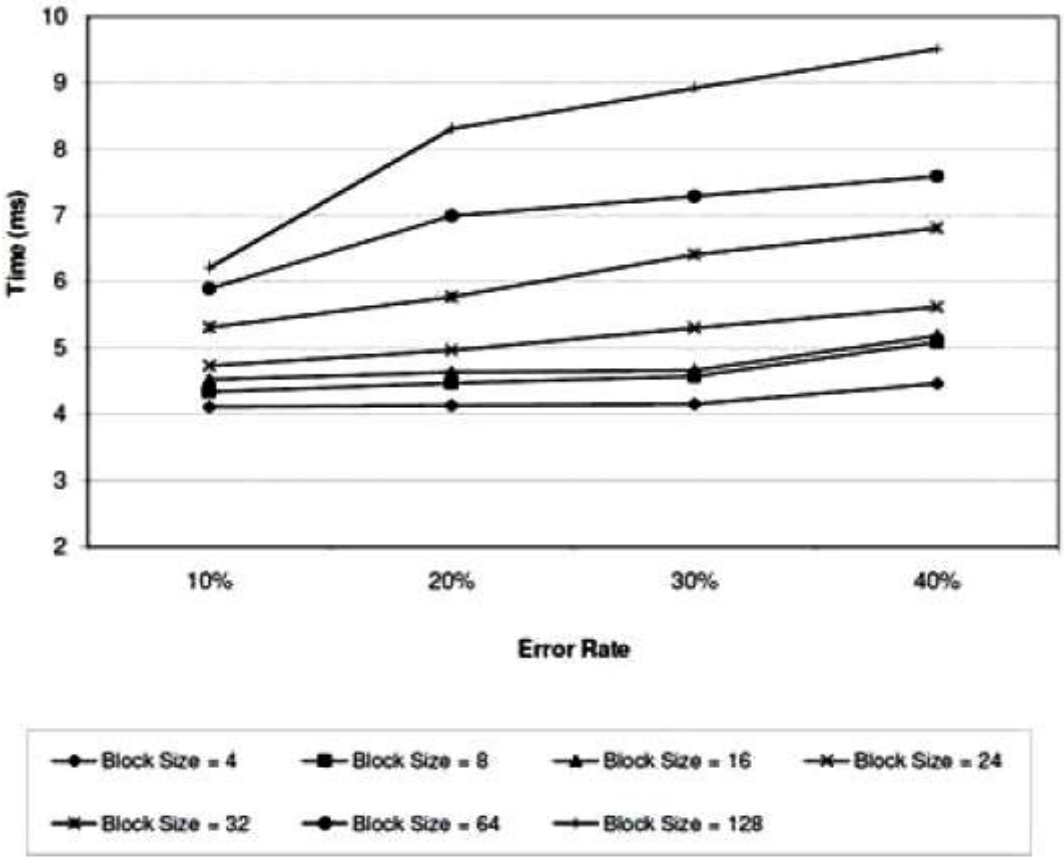

Figure 5. Time Taken to Complete Reconciliation

mismatch is found, then the respective block gets bisected and eventually another round of parity verification is added to the overall communication. With a smaller initial block size, number of sub-blocks required is considerably low. Hence the reconciliation can be completed more quickly. The main reason to this observation is the number of cycles that needed to process the full key length is considerably large. Larger block length will subsequently bisect into more and more sub-blocks hence parity check requires more time to complete.

Other notable observation is when the error level is high (error level of 40%), it takes significant amount of time for reconciliation irrespective of the block size. This is because the entire key is required to process right up to the smallest sub-block level, hence consuming more time for completion.

D. Analysis of Privacy Amplification

During privacy amplification, the key gets further reduced by applying a selected universal hash function to eliminate whatever information that may have leaked to an eavesdropper. The amount of information an eavesdropper might know is based on the sifted bit error rate determined during reconciliation and is entirely applied to potential eavesdropping, rather than distributing that error rate between eavesdropping and system losses.

It can be seen that the performance behavior of privacy amplification remains similar across different key lengths. When the error rate is low, it is evident that privacy amplification gets completed much quicker as the amount of processing on the key is low. In addition to error rate and length of the key, security parameter and the hash function being used also have an impact on the performance. Most of the time consumption during this phase depends on the type of universal hash function being used. The hash function requires some mathematical calculations to compute how the bits are removed to eliminate the information that may have leaked to eavesdroppers.

E. Analysis of the 4-Phase Handshake Protocol

The first, second and fourth phases, namely, sifting, error estimation and privacy amplification respectively, do not consume heavy resources as they the amount of activities involved are limited to few bit comparisons and restructuring the key etc.
In contrast, the bisect algorithm using parity check involves some considerable amount of processing during the reconciliation phase. During the simulation, lots of different scenarios with several input data have been used. The variations of chosen reconciliation protocol are: the length of the key, initial block size of the partition and error rate. However the first, second and fourth phases do not

have any such variations that would impact the overall process. Different key lengths with error rates 10%, 20% for initial block size 8 and have been used for these analyses.

It could be seen that the 4-phase handshake protocol gets completed well below 9 ms. This is because the key get partitioned only during the reconciliation phase. The length of the key has most of the impact on the reconciliation as longer the key, more partitions are required, hence more parity comparisons needed. However for other phases, key length does not impact in such a magnitude. During those phases, most time consuming operations being the key reconstruction at the end of each of the phase.

VI. CONCLUSION

The 4-way handshake protocol of the existing IEEE 802.11 has been replaced with the QKD based 4-phase handshake protocol. QKD is the most advanced application of Quantum Information Science. It reached already the world of commercial implementation. The main advantage of this proposed protocol is its ability to offer unconditional security to the users and each photons carry two bits. However, the key exchanged in the existing protocol needs to be refreshed at regular intervals or upon requested by STA to maintain security of data encryption. The AP can refresh the PTK either periodically or upon the request from the STA by running another 4-Way Handshake with the same PMK. But in the proposed QKD based Wi-Fi protocol, such key refresh cycles are not needed as the key exchanged provide unconditional security. Hence with this new protocol significant amount of processing time could be saved. Overall, this can compensate to the extra cycles of flows taken during the reconciliation process.

REFERENCES:

[1] Thi Mai Trang Nguyen and Mohamed Ali Sfaxi and Solange Ghernaouti-Helie “802.11i Encryption Key Distribution Using Quantum Cryptography”, 3rd ed. Journal of Networks 2006.

[2] Xu Huang and Shirantha Wijesekera, and Dharmendra Sharma “Novel Protocol and Its Implementation QKD in Wi-Fi Networks”,3rd ed. Eigth IEEE/ACIS International Conference on Computer and Information Science.

[3] C H Bennett and G Brassard “Quantum cryptography: Publickey distribution and coin tossing”, 3rd ed.Proceedings of IEEE International Conference on Computers, Systems and Signal Processing,Bangalore, India, December 1994.

[4] S Wijesekera “Quantum cryptography based key distribution in Wi-Fi Networks-protocol modificationin IEEE 802.11” in 5th international conference on software and data technology, Athens, July 2010.

[5] Brub D “Optimal eavesdropping in quantum cryptography with six states” Phys. Rev. Lett. 81, 3018-3021 (1998).

[6] Bechmann-Pasquinucci and H Peres A “Quantum cryptography with 3-state systems” Phys. Rev. Lett. 85, 3313- 3316 (2000).

[7] Bourennane M and Karlsson A and Bjork G “Quantum key distribution using multilevel encoding” Phys. Rev. A 64, 012306 (2001).

[8] N J Bourennane and M Karlsson and A Gisin “Security of quantum key distribution using d-Level systems”. Phys. Rev. Lett. 88, 127902 (2002).

[9] Groblacher S and Jennewein T and Vaziri A and Weihs and G Zeilinger, “Experimental quantum cryptography with qutrits”. New J. Phys. 8, 1-8 (2006).

[10] Allen L and Beijersbergen M and W Spreeuw “ Orbital angular momentum of light and the transformation of Laguerre Gaussian laser modes” . Phys. Rev. A 45, 8185-8189 (1992).

[11] Bennett, C. H., Wiesner, S. J “Communication via one and two-particle operators on Einstein-Podolsky-Rosen states”, Phys. Rev. Lett. 69, 2881-2884 (1992).

[12] Hanggi E and Renner R and Wolf S “ Quantum cryptography based solely on Bell’s theorem”. Available at arxiv:quant ph/0911.4171 (2009)