CHAPTER 4

A HYBRID APPROACH FOR SECURE QKDP IN 802.11i
## CHAPTER 4

Chapter 4: A HYBRID APPROACH FOR SECURE QKDP IN 802.11i

<table>
<thead>
<tr>
<th>S.NO.</th>
<th>Name of the Sub-Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>MOTIVATING SCENARIO</td>
<td>81</td>
</tr>
<tr>
<td>4.2</td>
<td>PRELIMINARIES</td>
<td>82</td>
</tr>
<tr>
<td>4.2.1</td>
<td>TRADITIONAL CRYPTOGRAPHY</td>
<td>82</td>
</tr>
<tr>
<td>4.2.2</td>
<td>QUANTUM CRYPTOGRAPHY</td>
<td>82</td>
</tr>
<tr>
<td>4.2.3</td>
<td>MOTIVATION TO HYBRID APPROACH</td>
<td>83</td>
</tr>
<tr>
<td>4.2.4</td>
<td>USB ASSUMPTION</td>
<td>83</td>
</tr>
<tr>
<td>4.2.5</td>
<td>KEY MANAGEMENT AND AUTHENTICATION IN 802.11i</td>
<td>83</td>
</tr>
<tr>
<td>4.3</td>
<td>INTEGRATING HQKDP WITH 802.11i</td>
<td>85</td>
</tr>
<tr>
<td>4.4</td>
<td>HQKDP MODEL BY USING QUANTUM SUPERPOSITION STATES</td>
<td>87</td>
</tr>
<tr>
<td>4.4.1</td>
<td>QUANTUM SUPERPOSITION STATES</td>
<td>87</td>
</tr>
<tr>
<td>4.4.2</td>
<td>OVERVIEW OF ARCHITECTURAL HQKDP</td>
<td>89</td>
</tr>
<tr>
<td>4.4.2.1</td>
<td>CONFIDENTIALITY PHASE</td>
<td>89</td>
</tr>
<tr>
<td>4.4.2.2</td>
<td>KEY DISTRIBUTION PROTOCOL</td>
<td>90</td>
</tr>
<tr>
<td>4.4.3</td>
<td>CONFIDENTIALITY PHASE EXPLICITLY</td>
<td>92</td>
</tr>
<tr>
<td>4.4.3.1</td>
<td>HQKDP SECURITY PROOF</td>
<td>93</td>
</tr>
<tr>
<td>4.4.4</td>
<td>SECURITY IN HQKDP</td>
<td>94</td>
</tr>
<tr>
<td>4.4.5</td>
<td>HAQKDP SECURITY PROOF</td>
<td>95</td>
</tr>
<tr>
<td>4.5</td>
<td>QKDP PROTOCOLS COMPARISION WITH OTHER PROTOCOLS</td>
<td>96</td>
</tr>
</tbody>
</table>
CHAPTER 4

A HYBRID APPROACH FOR SECURE QKDP IN 802.11i

The advantage of wireless networks is low cost. Wireless networks are widely deployed. Due to use of air as medium for transmission, it is easy for the unauthorized persons to access the wireless network. Hence, wireless networks should be protected from security threats. To protect these networks, there are many solutions. The two based on cryptography are:

- Classical Cryptography
- Quantum Cryptography

In turn, classical cryptography is again classified into two models. They are:

- Symmetric model
- Asymmetric model

The confidentiality of key in classical cryptography mainly depends on computational complexity. With enough computing resources particularly with quantum computing, it is possible to find the key though it is difficult and computationally intensive activity. Hence, researchers are experimenting with quantum cryptography for key distribution in secure way. But, there is need to integrate the good features of both the cryptography methods. This chapter proposes a scheme for the integration.

4.1 MOTIVATING SCENARIO

In this section we present the typical wireless network scenario. A typical WLAN is presented in Figure 4.1. The network has devices such as access point (AP) (1), mobile phone (2), WLAN USB Adapter (3), Wi-Fi Printer (4), WLAN PCMCIA Card (5), Personal Digital Assistant (PDA) (6), Notebook (7), personal computer (PC) (8).etc. All the devices are connected to the WLAN through AP.
With the help of radio waves these devices are going to communicate over the channel.

4.2 PRELIMINARIES

The proposed security architecture can be understood with QKDP.

4.2.1 TRADITIONAL CRYPTOGRAPHY

Traditional Cryptography has mainly two types, namely public key cryptography or asymmetric cryptography, private key cryptography or symmetric key cryptography. In public key cryptography, there is computational complexity and in private key cryptography there is sharing problem. Another type of key distribution is called three party key distribution Bat, traditional cryptography cannot solve some problems. Hence Quantum Cryptography came into existence.

4.2.2 QUANTUM CRYPTOGRAPHY

Quantum Cryptography is based on Quantum mechanics. There exist many QKDPs which make use of quantum physics.
4.2.3 MOTIVATION TO HYBRID APPROACH

Hwang, Lee and Li in 2007 overworked both cryptographic paradigms and Quantum Cryptography for secure exchange of data over the channel to prevent replay attacks, and man-in-the-middle attacks.

4.2.4 USB ASSUMPTION

Waters and Zurek [50] proposed no-cloning theorem. Hwang, Lee and Li proposed USB assumption for identifying the polarization basis of quantum states. It is helpful in authentication proof of QKDP.

4.2.5 KEY MANAGEMENT AND AUTHENTICATION IN 802.11i

802.11i is a security specification for wireless networks. Before integrating the proposed quantum protocol with WLAN, here are the authentication and key management procedures of 802.11i. 802.11i was implemented in the form of WPA2 which came into existence as WEP failed to secure wireless LANs. Both key management and authentication work together in 802.11i standard.

![Fig. 4.2: Illustrates both authentication and key distribution](image-url)
As shown in Figure 4.2, there are three components involved in authentication and key management in 802.11i standard. The mobile terminal and authentication server participate in mutual authentication process and pair wise master key establishment. The access point takes pair wise master key from authentication server and involve in mutual authentication and key establishment between itself and mobile terminal. There is 4-way handshake between mobile terminal and access point.

![Fig. 4.3: Pair wise key hierarchy](image)

As seen in Figure 1.1, 802.11i makes use of multiple keys at different levels. However, the top most key PMK is used to derive other keys. There are two approaches supported by 802.11i standard for key establishment. They include using pre-shared key and using authentication server. The PTK is split into KCK, KEK and TK. EAPOL stands for EAP over LAN as explored in [51]. EAPOL-Key messages are used to carry out handshaking process between access point and mobile terminal. The KCK is meant for computing message integrity code. To encrypt group temporal
key, the KEK is used. In order to encrypt data traffic of unicast user, the TK is used. In fact 802.11i uses different keys for authentication and encryption.

The PTK is constructed by client using PMK, Access Point Nonce (ANonce), STA Nonce (SNonce), MAC address of access point and MAC address of mobile client (STA). Mobile client sends SNonce and message integrity code to access point. Then access point constructs PTK. Then GTK and message integrity codes are sent from access point to mobile client. The GTK is best used for decrypting broadcast and multicast traffic. Finally mobile client sends acknowledgement to access point. This is illustrated in Figure 1.1.

4.3 INTEGRATING HQKDP WITH 802.11i

HQKDP is used for integrating with 802.11i. Quantum Cryptography is mainly useful for confident key transmission but not for encryption. The HQKDP is utilized in quantum mechanics for 4-way-handshake.
As shown in the Figure 4.4, PMK is mainly used for deriving KCK, while Q-PTK is derived from the HQKDP.

The line by line process of changing 4-way-handshake to integration of classical and quantum cryptography is given in Figure 4.5.
This Quantum Handshake was proposed by T.M.T Nguyen, 2006. The two data messages are interchanged from the mobile client to access point. It is carried out according to the steps of HQKDP. In quantum handshake, the last data messages exchanged completes the quantum handshake. This shows that there is secure data interchange between two parties.
4.4 HQKDP MODEL BY USING QUANTUM SUPERPOSITION STATES

4.4.1 QUANTUM SUPERPOSITION STATES

HQKDP protocol mainly uses quantum superposition states. In Figure 4.6 the measuring and encoding polarized photons are shown. Horizontal and vertical polarizations are used. It is shown as logic zero, \( |0\rangle = (1 \ 0)^T \) and one, \( |1\rangle = (0 \ 1)^T \) respectively.

![Figure 4.6: polarizations with horizontal and vertical angle](image)

| \[ \psi \rangle = |0\rangle \] | \[ \psi \rangle = |1\rangle \] |
|---|---|
| (a) | (b) |

If the user denotes data as \( S_n \), then the data has \( n \) traditional bits, this is used for encoding qubits.
\[ S_m = |i_1\rangle \otimes |i_2\rangle \otimes \ldots \otimes |i_j\rangle, \text{ where } \{ i_j = 0 \text{ or } 1, j = 1, 2, \ldots, n \} \]

The tensor product is shown by \( \otimes \). Here the rotation of superposition and the rotation of polarization are used to avoid malicious user from eavesdropping transmitting photons.

Fig. 4.7: polarization and rotation angle \( \theta \)

After an \( n \) bit data is enciphered with \( n \) photons, polarization of every photon can be rotated by an angle \( \theta_j \), which can be chosen randomly for every qubit. The rotation operation can be shown in Jones matrix as below.

\[
R(\theta_j) = \begin{bmatrix}
\cos \theta_j & \sin \theta_j \\
-sin \theta_j & \cos \theta_j
\end{bmatrix}
\]

The data qubit \( |\psi_0\rangle \) with \( \theta_s \) is enciphered with the sender where \( \theta_s \) is chosen randomly and known to sender and receiver. The enciphering process is carried out using the secret key \( K \) and data \( S_m \) is as follows.
\[ E_{\theta}[S_m] = R(\theta) \cdot |0\rangle = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \]

\[ = \begin{bmatrix} \cos\theta \\ -\sin\theta \end{bmatrix} = \cos\theta \cdot |0\rangle - \sin\theta \cdot |1\rangle = |\psi_1\rangle \]

When superposition states are received from the sender, then the rotation of a photon is going to take place before we measure it. If the rotation is in opposite \(-\theta\), then the deciphering is as follows.

\[ R(-\theta) \cdot |\psi_1\rangle = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) \\ -\sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} \cos\theta \\ -\sin\theta \end{bmatrix} \]

\[ = \begin{bmatrix} \cos^2\theta + \sin^2\theta \\ \sin\theta\cos\theta - \cos\theta\sin\theta \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle \]

The sequence of rotation angles can form \( K \) for a given \( n \) bit message (secret key) which is represented as follows.

\[ K = \{ \theta_j : 0 \leq \theta_j < \pi, j=1, 2, 3... n \} \]

The message in which the process of enciphering is applied to angle \( \theta_j \) is shown by the subscript.

### 4.4.2 OVERVIEW OF ARCHITECTURE OF HQKDP

#### 4.4.2.1 CONFIDENTIALITY PHASE

Trusted Third Party is involved in the confidential process. Once it gets notification it has to do the following process.

**Step1:** \( X \) and \( Y \) numbers are generated randomly with TC

\[ X = h(K_{1T}, r_1) \oplus (U_1||U_2) \]

\[ Y = h(K_{2T}, r_2) \oplus (U_2||U_1) \]
where $r_1$ and $r_2$ are random numbers and X and Y are computed values used further. In the same fashion, the mechanism is applied to $r_2 \parallel r_1 \parallel Y$ in order to obtain the result $K_{2,T}$ which is sent to Bob over the secure quantum channel.

Then the pre-shared key is used to polarize and encrypt $r_1 \parallel r_2 \parallel X$. The result of the operation is $K_{1,T}$ which is sent to Alice through a secure quantum channel.

**Step2:** The deciphered qubits are then measured by Alice. Alice will utilize $K_{1,T}$ and $r_1$ to get a hash value of $U_1||U_2$. Afterwards, these values $U_1$ and $U_2$ are checked by Alice.

**Step3:** Bob will decipher and measure the qubits. Then hash value is calculated by $K_{2,T}$ and $r_2$ to get values for $U_2||U_1$. Afterwards $U_1$ and $U_2$ are checked by Bob.

Then the overall session is completed successfully with the Trusted Center.

### 4.4.2.2 KEY DISTRIBUTION PROTOCOL

The key distribution is mainly useful for a confident key ‘k’ among the sender and the receiver. For the key establishment Shamir’s three-pass protocol is used in quantum superposition states.

Without losing the originality, the following can be assumed as X and can be shown as $S_m = |0\rangle$ (i.e., $n=1$ and $i_1=0$) in the single photon encoded format. Then key distribution is initiated by Alice. The procedure is as follows.

**Step1:** The session keys $K_1 = \theta_1$ and $K_2 = \theta_2$ are developed between Alice and Bob.

**Step2:** The data $S_m$ is enciphered by Alice using his session key $K_i$. The result is as follows.

$$ E_{K_i}[S_m]: R(\theta_i) . |0\rangle = \begin{bmatrix} \cos \theta_i & \sin \theta_i \\ -\sin \theta_i & \cos \theta_i \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \cos \theta_i |0\rangle - \sin \theta_i |1\rangle = |\psi_i\rangle $$
where $E_{K_1}$ represents an encryption which is performed using $K_1$. This state is called as the superposition state. It was denoted by $|\psi_1\rangle$ which is sent by Alice to Bob.

\textbf{Step3:} From the state $|\psi_1\rangle$, Alice sends a photon that can be received by Bob which can be encrypted using $K_2$.

$$E_{K_2}[E_{K_1}[S_m]]: R(\theta_2) . |\psi_1\rangle = \cos(\theta_2 + \theta_1) |0\rangle - \sin(\theta_2 + \theta_1) |1\rangle = |\psi_2\rangle$$

The resulting state $|\psi_2\rangle$ is still a superposition state. Bob sends it back to Alice.

\textbf{Step4:} On getting it, Alice deciphers it and the resultant superposition state $|\psi_3\rangle$ is sent back. The decryption can be done with angle $\theta_1$.

$$D_{K_1}[E_{K_2}[E_{K_1}[S_m]]] = E_{K_2}[S_m]: R(-\theta_1) . |\psi_2\rangle$$

$$= \cos \theta_2 |0\rangle - \sin \theta_2 |1\rangle = |\psi_3\rangle$$

where $D_{K_1}$ shows the deciphering process using the session key $K_1$.

\textbf{Step5:} Bob deciphers through the act of reading it back with certain angle $\theta_2$.

$$D_{K_2}[E_{K_2}[S_m]]: R(-\theta_2) . |\psi_3\rangle = \begin{bmatrix} \cos(-\theta_2) & \sin(-\theta_2) \\ -\sin(-\theta_2) & \cos(-\theta_2) \end{bmatrix} \begin{bmatrix} \cos \theta_2 \\ -\sin \theta_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$$

Bob has the originality of data denoted as $S_m = |0\rangle$. Confidential and actual KDP are shown in Figure. 4.8.
The steps followed are

- **TC-> Alice**: \( E_{k_{1,T}} [r_1 \ | \ r_2 \ | \ X] \)

Where \( X = h(K_{1,T}, r_1) \oplus (U_1 \ | \ U_2) \).

- **TC-> Bob**: \( E_{k_{2,T}} [r_2 \ | \ r_1 \ | \ Y] \)

Where \( Y = h(K_{2,T}, r_2) \oplus (U_2 \ | \ U_1) \).

- **Alice**: \( D_{k_{1,T}} [E_{k_{1,T}} [r_1 \ | \ r_2 \ | \ X]] \) and verifies \( U_1 \ | \ U_2 \).

- **Bob**: \( D_{k_{2,T}} [E_{k_{2,T}} [r_2 \ | \ r_1 \ | \ Y]] \) and verifies \( U_2 \ | \ U_1 \).

- **Alice**: \( R(\theta_1), |\psi_0\rangle = R(\theta_1) . |0\rangle = |\psi_1\rangle \)

- **Alice->Bob**: \( |\psi_1\rangle = \cos \theta_1 . |0\rangle - \sin \theta_1 . |1\rangle \)

- **Bob**: \( |\psi_2\rangle = R(\theta_2). |\psi_1\rangle \)

- **Bob-> Alice**: \( |\psi_2\rangle = \cos (\theta_2 + \theta_1). |0\rangle - \sin (\theta_2 + \theta_1). |1\rangle \)
Alice: $|\psi_3\rangle = R(-\theta_1). |\psi_2\rangle$

Alice->Bob: $|\psi_3\rangle = \cos\theta_2 |0\rangle - \sin\theta_2 |1\rangle$

Bob: $R(-\theta_2). |\psi_3\rangle = |0\rangle = |\psi_0\rangle$

Alice and Bob have the key ‘k’ when KDP completes.

4.4.3 CONFIDENTIALITY PHASE EXPLICITLY

For mutual authentication, we have two parties in communication.

Step1: With the help of shared key and random number, Alice can encipher and transmit the results to Bob. And Bob has to decipher and get the value $r_1$. The confidentiality is in positive manner if $r_2 = r_1$.

Step2: Bob uses the shared key ‘K’ and random number $r_1$ for enciphering and sends to Alice. Then Alice deciphers to obtain value $r_1$. Hence successful authentication of Bob to Alice is done if $r_1 = r_1$. The explicit secure process is shown in Figure 4.5.

4.4.3.1 HQKDP SECURITY PROOF

Theorem 1

In HQKDP when security is broken, adversaries get the QKD confident. Assuming that the authentication process does not resolve in the protocol in a given time after making $q_h$ Hash queries, $q_{se}$ Send queries and $q_{ini}$ Initiate queries,

$$\text{adv}_{HQKDP}^{QKD}(A) \leq 2. \text{adv}_{HQKDP}^{QKD}(A) + q_h/2^n$$

The $t \leq t + (q_{ini} + q_h + q_{se}) T_{rc}$;
T_re is shown as replay query.

If it tries to break MA, security of our protocol in time period $t$, after $q_h$ Hash queries, $q_{se}$ Send queries and $q_{ini}$ Initiate queries is

$$\text{adv}^{\text{MA}}_{H^Q\text{KDP}}(A) \leq 2 \cdot \text{adv}^{QKD}_{H^Q\text{KDP}}(\Delta) + q_h / 2^U - 1$$

Where $t \leq t + (q_{ini} + q_{se} + q_h + q_h^1) T_{re}$;

**Proof**

For instance, the attackers launch an attack on QKD and break the QKD security of HQKDP, the attacker will be able to break the MA security of HQKDP.

$A_1$ breaks the security of HQKDP and gets advantage $^{QKD}_{H^Q\text{KDP}}(A_1)$. The advantage gained by attacking $\Delta_1$, is denoted as $^{QKD}_{H^Q\text{KDP}}(\Delta_1)$. The successful attack means it accesses to seek after breaking the security of HQKDP. The adversary $\Delta_1$ executes $A_1$ and provides responses about it. Moreover, randomly generated qubits are also sent from $\Delta_1$ to $A_1$ in order to simulate the authentication process of HQKDP. The $\Delta_1$ can generate a hash query which is padded with 0 prior to returning the result to $A_1$.

If $A_1$ makes a (Test; $\pi_s\Upsilon$) query, $\Delta_1$ will send a (Test; $\pi_s\Upsilon$) query to HQ, then $\Delta_1$ will get HQKDP. Once $S$ is got from (Test; $\pi_s\Upsilon$) the query, a bit 0 is returned by $A_1$ as output. The adversary $\Delta_1$ gives 1 as output when $h(w, 0)$. When $\Delta_1$ helps for breaking the security of the proposed protocol HQKDP and get the session key, the hash table of $A_1$ should have $h(w, 0)$.

Let $\delta$ be the probability that $A_1$ makes the hash query, $h(w, 0)$. We know that, $\delta \geq ^{QKD}_{H^Q\text{KDP}}(A_1)$, since $A_1$ may make the hash query $h(w, 0)$, but does not make the (Test; $\pi_s\Upsilon$) query. It is done purely by chance with security probability $w = sk$

$$\frac{1}{2} \cdot ^{QKD}_{H^Q\text{KDP}}(A_1) - q_h^U \leq \frac{1}{2} \cdot (\delta - q_h^U) = ^{QKD}_{H^Q\text{KDP}}(\Delta_1)$$
The same is shown as follows.

\[ t^i \leq t + (q_{\text{ini}} + q_{\text{se}} + q_{\text{h}}) T_{\text{re}}; \]

4.4.4 SECURITY IN HQKDP

The adversary \( A_2 \) has \( \text{adv}_{\text{HQKDP}}^{M_A}(A_2) \) to break \( M_A \) confident of HQKDP in the the given time period. Likewise, for breaking \( M_A \) confidentiality an adversary \( \Delta_2 \) gain \( \text{adv}_{\text{HQKDP}}^{QKD}(\Delta_2) \) in the given time \( t^i \). The adversary \( \Delta_2 \) runs subroutine in the form of \( A_2 \), besides answering queries of \( A_2 \). Likewise, the adversary gets queries from \( A_2 \) and they will be forwarded the results that are obtained. The results will be sent to \( A_2 \). Sometimes \( \Delta_2 \) sends \((\text{Test}; \pi^U)\) query to get a u-bit from \( \Delta_2 \). Once \( A_2 \) is completed, it verifies the presence of hash query, denoted as \( h'(w, r) \). In the hash table \( h'(w, r) \) the adversary returns 1. Let \( T_{\text{re}} \) is the probability of making hash query \( h'(w, r) \) by \( A_2 \). It is well known that \( \text{adv}_{\text{HQKDP}}^{M_A}(A_2) \) is the gain of \( A_2 \) by making hash query.

Then the adversary breaks the security of HQKDP similar to the probability of hash query \( h'(w, r) \) of \( A_2 \) minus hash query \( h'(sk, r) \) of \( A_2 \) and the same is multiplied as required such that \( w = sk \)

\[ \frac{1}{2} \cdot \text{adv}_{\text{HQKDP}}^{M_A}(A_2) - \frac{1}{2} \cdot \text{adv}_{\text{HQKDP}}^{QKD}(\Delta_2) \leq \frac{1}{2} \cdot \left( \delta - \frac{q_{\text{h}}}{2^u} \right) = \text{adv}_{\text{HQKDP}}^{QKD}(\Delta_2) \]

Consider \( T_{\text{re}} \) is the required time to relay a query. Then it will be observed that \( \Delta_2 \) running time is less than that of \( A_2 \) plus

\[ t^i \leq t + (q_{\text{ini}} + q_{\text{se}} + q_{\text{h}}) T_{\text{re}} \]

4.4.5 HAQKDP SECURITY PROOF

If the opponent succeeds, the USB distinguisher \( \Delta \) is used to fracture the USB statement.

Theorem 2
Let $\text{adv}^{\text{AQKD}}_{\text{HAQKDP}} (A)$ breaks the AQKD of HAQKDP.

Let $\text{adv}^{\text{USB}}_{\phi} (\Delta)$ breaks the USB statement in $\phi$.

If A breaks the AQKD confidentiality of HAQKDP within the time $t$, $q_{\text{ini}}$ Initiate queries, $q_{\text{se}}$ Send queries, and $q_{\text{h}}$ Hash queries within the time $t$, a USB statement will have benefit of $\Phi$ that is:

$$\text{adv}^{\text{AQKD}}_{\text{HAQKDP}} (A) \leq \frac{2 \cdot (q_{\text{ini}} + q_{\text{se}})}{q_{\text{ini}}} \cdot \text{adv}^{\text{USB}}_{\phi} (\Delta).$$

Where $t^1 \leq t + (q_{\text{ini}} q_{\text{se}} T_{\text{rn}})$ $T_{\text{rn}}$ is the time to generate a random number.

**Proof**

This protocol is confident of HAQKDP on the AQKD security of $\phi$. The same is described as follows.

In time $t$, assume that an adversary $A_1$ breaks the security of HQKDP, and gets advantage $\text{adv}^{\text{AQKD}}_{\text{HAQKDP}} (A_1)$. The attacker can gain the advantage $\Delta_1$ with this event that is denoted as $\text{adv}^{\text{USB}}_{\phi} (\Delta_1)$. The adversary $A_1$ executes and gives response to the queries about it. Likewise, $A_1$ is queried from an adversary and sent to AQKD and gets answers and sends them to $A_1$. The test query can be revealed by $A_1$. $\Delta_1$ can generate a hash query which is padded with 0 prior to $A_1$.

$$\text{adv}^{\text{AQKD}}_{\text{HAQKDP}} (A_1) \cdot \frac{\varepsilon_{\text{ini}}}{2(q_{\text{ini}} + q_{\text{se}})} \leq \text{adv}^{\text{AQKD}}_{\text{HAQKDP}} (A_1) \cdot \frac{\varepsilon_{\text{ini}}}{2(q_{\text{ini}} + q_{\text{se}})} = \text{adv}^{\text{USB}}_{\phi} (\Delta_1)$$

Consider the time relay query. The same is shown as follows.

$$t^1 \leq t + q_{\text{ini}} T_{\text{rn}}.$$

**4.5. QKDP PROTOCOLS COMPARISION WITH OTHER PROTOCOLS**
It compares “HQKDP” with other QKDP protocols of different attackers; the results are represented in the Table 4.1
Table 4.1: Comparison of QKD protocols

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cryptographic Mechanism</td>
<td>Quantum + Classical</td>
<td>Quantum + Classical</td>
<td>Classical</td>
<td>Classical</td>
<td>Quantum + Classical</td>
<td>Quantum + Classical</td>
<td>Quantum + Classical</td>
<td></td>
</tr>
<tr>
<td>Pre-shared Secret Key</td>
<td>Lengthy-termed</td>
<td>Lengthy-termed</td>
<td>EPR Pairs</td>
<td>Lengthy-termed</td>
<td>Long-termed</td>
<td>EPR Pairs</td>
<td>Lengthy-termed</td>
<td>Lengthy-termed</td>
</tr>
<tr>
<td>Meeting Round</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>QC</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Clock Synchronization</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>exposed to man-in-the-Middle Attack</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Exposed to Passive Attack</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Exposed to Replay Attack</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Formal Security Proof</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>