Appendix A

Security Analysis of proposed Three-party QKDPs

The following formally analyzes the security of the proposed KHC and KUC.

A. Security Analysis of the QKDP with Honest Center

This following analyzes the security of the QKDP with honest center (KHC). We formally define the model of game-based proof and prove the security of the QKDP in details. The model of game-based proof includes the definitions of the participants, adversary's abilities, the security notion and the primitives.

Protocol Participant

Two legitimate participants and a center are supposed to take part in the KHC. A participant and the center may have many instances correlated in distinct and concurrent executions of KHC. An instance $s \in N$ of a participant is denoted as Π_U^s. Besides, the instance Π_U^s accepts when the user gains sufficient information to compute a session key K. An instance can accept at any time and only accept once. Moreover, the notation $AAC(\Pi_U^s) = TRUE$ means that a session key K is accepted by the instance Π_U^s.

Adversary's Queries

The adversary is not a legitimate participant, but he controls the communications. The queries, Send query, Reveal query and Test query, represent the capabilities of adversary Eve. We describe the queries as follows:

- $(Send, Q, \Pi_U^s)$: This query models that Eve inputs $(Send, Q)$ to an instance Π_U^s and get the response from the instance Π_U^s, where Q denotes qubits. The Eve can control all qubit transmissions in an execution with the Send query. Besides, we denote the number of the Send queries as q_{se}.
- $(\mathrm{Re}veal, \Pi_U^s)$: This query models that Eve inputs Reveal to a client instance Π_U^s. The Reveal query is only available to Eve when $AAC(\Pi_U^s) = TRUE$. If $AAC(\Pi_U^s) = TRUE$, the Reveal query forces the client instance Π_U^s to output the session key K to Eve. Otherwise, Π_U^s outputs $NULL$. The Reveal query models the known key attack, which means that the loss of one session key should not endanger a session key of another session.
- $(Test, \Pi_U^s)$: This query models that Eve inputs Test to a client instance Π_U^s. Then, the client instance Π_U^s will flip an unbiased coin b and output the session key K if $b = 1$. Otherwise, Π_U^s outputs a u-bit random string. The Test query estimates the semantic security of the session key, i.e., the indistinguishability between a real session key and a random string. It should be noted that the Test query is only available when Π_U^s is Fresh. During an execution of KHC, Eve can only ask the Test query once.

Security Notion

This following describes the Freshness and the Security of KHC in details.

Freshness:

In KHC, an instance Π_U^s is called Fresh when $AAC(\Pi_U^s) = TRUE$ and Π_U^s has not been asked a Reveal query. With the fresh instance Π_U^s, we can describe the freshness of session key. That is, Π_A^s and Π_B^s establish a session key K, the session key K is called Fresh if and only if both Π_A^s and Π_B^s are Fresh.

Security of KHC:

The KHC includes the Setup phase, the Query phase and the Challenge phase, which describes the beginning of KHC, the abilities of the adversary and the advantage for the adversary to win the game.

- Setup Phase: The challenger runs the Setup algorithm, and responds to the adversary with the resulting system parameters, such as u, m and n.
- Query Phase: The adversary can ask various queries, such as the Send query and the Reveal

query.

- Challenge Phase: When the query phase is over, the adversary can submit the Test query to the Fresh instance $\prod_U^s$ in an execution of KHC. The challenger will toss an unbiased coin. If the tossing result $b=1$, the challenger returns the u-bit session key K to the adversary; otherwise, returns a u-bit random string. After receiving the string, the adversary responses b'. The adversary wins if $b'=b$. The advantage for the adversary, Eve, to get the correct session key is $Adv_{KHC}^{K}(Eve)$. The proposed QKDP is called session key secure if $Adv_{KHC}^{K}(Eve)$ is negligible.

$$Adv_{KHC}^{K}(Eve)=\left|\Pr[b'=b]-\frac{1}{2}\right|.$$

Definition of Primitive:

The following describes the primitives used in the security proof of KHC, which are the uncertainty of quantum measurement, the assumption of no cloning, the assumption of pseudo random function, and the assumption of one-way hash function.

- **Uncertainty of Quantum Measurement:** Let $S_i \in \{0,1\}^n$ be the string of polarization direction, $B_i \in \{R,D\}^n$ be the string of polarization basis and QG be the qubit generating algorithm. With the information of (S_i,B_i), the algorithm QG produces qubits $Q_i=QG(S_i,B_i)\in\{|0\rangle,|1\rangle,|+\rangle,|-\rangle\}$. Let n_{diff} be the number of bits, where $B_1\neq B_2$ and $B_1,B_2\in B_i$. Let $M_{B_i}(\,)$ be the qubit measuring algorithm, which measuring result $MR_i=M_{B_i}(Q)$ is a bit string. The probability for the adversary Eve to break the Uncertainty of Quantum Measurement is denoted as ε_{uqm}. The ε_{uqm} is negligible if

$$\varepsilon_{uqm}=\Pr[MR_1=MR_2 \mid MR_1=MR_{B_1}(Q_1),$$
$$MR_2=MR_{B_2}(Q_1)]\le(\frac{1}{2})^{n_{diff}}.$$

- **Assumption of No Cloning:** Let $S_i\in\{0,1\}^n$ be the string of polarization direction, $B_i\in\{R,D\}^n$ be the string of polarization basis and QG be the qubit generating algorithm. With the information of (S_i,B_i), the algorithm QG produces qubits $Q_i=QG(S_i,B_i)\in\{|0\rangle,|1\rangle,|+\rangle,|-\rangle\}$. Let $QC(\,)$ be the qubit cloning algorithm, which produces qubits $Q'_i=QC(Q_i)$. Let $M_B(\,)$ be the qubit measuring algorithm which produces the measuring result $MR_i=M_B(Q_i)$. The probability for the adversary Eve to break the Assumption of No Cloning is denoted as ε_{clone}. The ε_{clone} is negligible if

$$\varepsilon_{clone}=\Pr[MR_1=MR_2 \mid MR_1=M_B(Q_1),$$
$$MR_2=M_B(Q'_1)\le(\frac{3}{4})^n].$$

- **Assumption of Pseudo Random Function:** Let $B\in\{R,D\}^n$ be the string of polarization basis and $Q\in\{|0\rangle,|1\rangle,|+\rangle,|-\rangle\}^n$. Let $M_B(\,)$ be the qubit measuring algorithm, which measuring result $MR=M_B(Q)$ is the n-bit string. Let $\Gamma_{n,n}$ is the set of pseudo random function, which maps $\{0,1\}^n$ to $\{0,1\}^n$. If the adversary Eve could not distinguish the measuring result MR from the output of the pseudo random function MR', we denote the qubit measuring algorithm as one of the Pseudo Random Function. Moreover, we denote the advantage for Eve to distinguish MR from MR' as ε_{prq}. Then, we have

$$|\Pr[A(MR)=1 \mid MR=M_B(Q)]-\Pr[A(MR')=$$
$$1 \mid f\leftarrow\Gamma_{n,n}, MR'\leftarrow f(\,)]|\le\varepsilon_{prq}.$$

- **Assumption of One-way Hash Function:** Let $H(.)$ be the one-way hash function, which maps $\{0,1\}^u$ to $\{0,1\}^m$. Given a checksum h, the probability for the adversary Eve to compute K is denoted as ε_{hf}, where $h=H(K)$. The one-way hash function is computationally secure if ε_{hf} is negligible under the quantum computation.

The model describes the security environment of the KHC, in which an adversary wants to extract the

session key from the execution of KHC. The adversary can be a registered user or malicious outsider. The KHC will achieve the key security against an adversary if the adversary has the negligible advantage to gain the session key in the model.

Security Analysis of KHC

The following game-based proof played between a challenger and the adversary, which environment is defined in the model. A theorem is firstly presented to demonstrate the security of KHC; and then the proof of the theorem is described in details.

Theorem 1. T*he proposed KHC is secure based on the Uncertainty of Quantum Measurement and the Assumption of No Cloning. The advantage for an adversary, Eve, to get the correct session key is:*

$$Adv^{K}_{KHC}(Eve) = \left|\Pr[b' = b] - \frac{1}{2}\right| \le \lambda,$$

where the probability λ *is negligible.*

Proof. In this proof, a sequence of games are incrementally defined starting from the real game G_0 to the simulated game G_2. Besides, the Difference Lemma [11] is used within the sequence of games. That is, with the events A, B, F $(A \wedge \neg F \leftrightarrow B \wedge \neg F)$, we have $|\Pr[A] - \Pr[B]| \le \Pr[F]$. The following describes the games in details.

Game G_0**:** We describe that the game is an interactive computation between Eve and the challenger Ch. The original attack game G_0 is the same as the game described in the security notion. In the setup phase, the challenger responds to Eve with the system parameters. In the query phase, Eve can make a number of the Send queries and the Reveal queries. In the challenge phase, Eve makes a Test query and guesses the received string is a random string $(b = 0)$ or the session key $(b = 1)$. The advantage for the adversary to guess $b' = b$ is:

$$Adv^{K}_{KHC}(Eve) = |2 \cdot \Pr[G_0] - 1|.$$

Game G_1**:** In Step 2, Eve can intercept the qubits Q_2 sent from Alice to Bob and produces new qubits Q'_2 for Bob. If Eve has the qubit cloning ability, the qubits Q'_2 will be the same as Q_2. In this game G_1, the challenger simulates the bases B_1 sent from Alice to center with the random string B'_1 in Step 5. From this simulation, we easily see that the game is indistinguishable from the real attack, unless Eve has the ability of qubit cloning. If Eve has the ability of qubit cloning, Eve will find the difference between G_0 and G_1. We denote the failure event as F, where $F = F_1 \cup F_2 \cup F_3$.

Failure Event F_1**:** With the information of B_1', Eve performs the unitary operation U_j on intercepted qubits Q_2. Then, Eve measures the qubits with basis R and gains the measuring result $K' \| h'$. Eve compares h' with $H(K')$, he would find the difference $h' \neq H(K')$ with the probability $\Pr[F_1]$. Because the condition $h' = H(K')$ may happen, the probability $\Pr[h' = H(K')]$ should be removed from the probability $\Pr[F_1]$. The probability for Eve to get $\Pr[h' = H(K')]$ is $(1/2)^m$. Therefore, the probability of the failure event $\Pr[F_1]$ is

$$\varepsilon_{clone}(1 - (\frac{1}{2})^m) \cdot$$

Failure Event F_2**:** If Eve has the qubit cloning ability, he can correctly output b' by distinguishing the session key from the random string. However, Eve will return $b' \neq b$ with the probability $\Pr[F_2] = \Pr[f_1] \cup \Pr[f_2]$ in the following situations.

- Situation f_1: The challenger returns the session key to Eve with $d = 1$, but Eve judges the string as the random number and outputs $d' = 0$. The probability $\Pr[f_1]$ is

$$\varepsilon_{clone} \cdot (\frac{1}{2})^m - \varepsilon_{clone} \cdot (\frac{1}{2})^m \cdot \varepsilon_{uqm}.$$

- Situation f_2: The challenger returns the random string to Eve with $d = 0$, but Eve's measuring result is the same with the random string and

outputs $d'=1$. The probability $\Pr[f_2]$ is

$$\varepsilon_{clone}\cdot(\frac{1}{2})^m-\varepsilon_{clone}\cdot(\frac{1}{2})^m\cdot(\frac{1}{2})^u.$$

Failure Event F_3: When Eve has the qubit cloning ability, the *m* checksum qubits can be cloned by Eve. Then, Eve performs the unitary operation U_j with B_1', and measures the qubits with basis R on the intercepted *m* qubits to get h'. Eve can compute the input value K' from h' with the probability ε_{hf}. When the challenger returns the session key K to Eve, Eve finds the difference by judging the string as the random number $K'\neq K$. The probability $\Pr[F_3]$ is

$$(\frac{3}{4})^m\cdot\varepsilon_{hf}\cdot(1-(\frac{1}{2})).$$

From the above simulation, we see that the games G_0 and G_1 are indistinguishable unless the failure event occurs. Based on the Difference Lemma, $|\Pr[G_0]-\Pr[G_1]|\leq\Pr[F]$ and the number of the send query is q_s. Besides, Eve can also intercept the qubits Q_3 sent from Bob to center and produces new qubits Q'_3 for the center in Step 3. If Eve has the qubit cloning ability, the qubits Q'_3 will be the same as Q_3. The challenger simulates the bases B_2 sent from Bob to center with the random string B'_2 in Step 5. The probability of the failure event is similar to the above description. Therefore, we have

$$|\Pr[G_0]-\Pr[G_1]|\leq 2\cdot q_s\cdot\varepsilon_{clone}(1+(\frac{1}{2})^m)$$
$$\cdot(1-\varepsilon_{uqm}-(\frac{1}{2})^u)+q_s\cdot(\frac{3}{4})^m\cdot\varepsilon_{hf}.$$

Game G_2: In Step 6, the challenger replaces C' sent from center to Bob with a random string RS produced by the pseudo random function. The game G_2 is indistinguishable from the Game G_1, unless Eve has the ability of distinguishing the value RS from C'. If Eve has the ability to break the assumption of pseudo random function, Eve will find the difference between G_1 and G_2 with the probability ε_{prq} described in Lemma 1. Moreover, Eve can make the send query q_s times. Hence, we have

$$|\Pr[G_1]-\Pr[G_2]|\leq q_s\cdot\varepsilon_{prq}.$$

Besides, the probability $\Pr[G_2]$ is $1/2$ because all the information sent from the challenger to Eve is random information. According to the games G_0, G_1 and G_2, we can deduce that the advantage for Eve to break the key security of the proposed KHC is

$$Adv^K_{KHC}(Eve)=|\Pr[G_0]-\Pr[G_2]|$$
$$=\left|\Pr[G_0]-\frac{1}{2}\right|\leq q_s\cdot((2\cdot\varepsilon_{clone}(1+(\frac{1}{2})^m$$
$$\cdot(1-\varepsilon_{uqm}-(\frac{1}{2})^u))+(\frac{3}{4})^m\cdot\varepsilon_{hf}+\varepsilon_{prq}).$$

□

The advantage for Eve to distinguish the difference between the random string and the pseudo random quantum measurement is described as follows:

Lemma 1 (Pseudo Random Quantum Measurement). *The qubit measuring algorithm is one of the Pseudo Random Function and the quantum measuring result MR is indistinguishable from the output of the pseudo random function MR'. The advantage for an adversary, Eve, to distinguish MR from MR' is:*

$$|\Pr[A(MR)=1\,|\,MR=M_B(Q)]-\Pr[A(MR')=1\,|$$
$$f\leftarrow\Gamma_{n,n},MR'\leftarrow f(\,)]|\leq\varepsilon_{prq},$$

where $A(.)$ is a distinguishing algorithm, $M_B(.)$ is the qubit measuring algorithm, and $G_{n,n}$ is the set of pseudo random function (map from $\{0,1\}^n$ to $\{0,1\}^n$).

Proof. A sequence of games are used in the proof of Lemma 1, which start from the game G_0 to the simulated game G_2. Moreover, the Difference Lemma [11] is used within the sequence of games. That is, with the events A, B, F $(A\wedge\neg F\leftrightarrow B\wedge\neg F)$, we have $|\Pr[A]-\Pr[B]|\leq\Pr[F]$. The following describes the games in details.

Game G_0: Let $B\in\{R,D\}^n$ be the string of polarization basis and $Q\in\{|0\rangle,|1\rangle,|+\rangle,|-\rangle\}^n$. The measuring result $MR=M_B(Q)$ is the n-bit string. The pseudo random function f is chosen from the set of pseudo random functions $\Gamma_{n,n}$. In this game, Eve can query q times to

the qubit measuring algorithm $M_B(.)$, which takes qubits Q and polarization bases B as the inputs and returns the measuring result MR as the output. When Eve send the query to $M_B(.)$, the challenger decides qubits Q_i and polarization bases B_i, where $i = 1...q$. Then, the challenger simulates the sender and the receiver, and transmits the qubits from the sender to the receiver. During the qubit transmission, Eve may intercept and produce new qubits Q'_i for the receiver. After receiving qubits, the challenger checks whether $Q'_i = Q_i$. If the equation holds, the challenger will randomly generate a string B'_i with the pseudo random function, where $B'_i \neq B_i$; otherwise, the game is terminated. Moreover, the challenger returns the string B'_i and the measuring result MR_i to Eve. With the values B'_i and MR_i, Eve does not know which function, the quantum measuring function or the pseudo random function, is used by the challenger. Eve wins the game G_0 with the advantage $\Pr[G_0]$ if he correctly judges the values came from the quantum measuring function or the pseudo random function.

Game G_1: The game G_1 is the bridge step of the game G_0, which is based on the indistinguishability. Instead of returning MR_i to Eve, the challenger randomly generates the string B'_i and the measuring result MR'_i. Besides, the challenger saves B_i, MR_i, B'_i and MR'_i into a table. When Eve makes a query, the challenger checks the table to see whether the current query has the corresponding measuring result MR_i. If the challenger finds the same value in the table, he will retrieve the corresponding value MR'_i from the table and returns it to Eve; otherwise, he generates new value for Eve. The advantage for Eve to win this game G_1 is the same as the game G_0. That is, $\Pr[G_1] = \Pr[G_0]$.

Game G_2: The game G_1 is transited to the game G_2 based on the failure event. The difference between G_1 and G_2 is that the challenger will not retrieve MR'_i from a table when $MR_i = MR_j$ in the game G_2. The challenger randomly generates the strings B'_j and MR'_j for Eve without checking the corresponding values in the table. The failure event happens when $MR_i = MR_j (i \neq j)$ but the challenger returns different values $MR'_i \neq MR'_j$ to Eve. The advantage for Eve to detect the failure event is

$$\Pr[F] = C_2^q \cdot (\frac{1}{2})^{n_{diff}} \approx \frac{q^2}{2} \cdot (\frac{1}{2})^{n_{diff}},$$

where q is the number of queries made by Eve and n_{diff} is the upper bound between the number of different qubits $Q_i \neq Q_j$ and the number of different bits $B_i \neq B_j$.

Besides, the execution of our protocol continues when the challenger verifies the correctness of receiving qubits; otherwise, the execution is stopped. The probability for the Q'_i to pass the verification is $(3/4)^p$, where p is the number of qubits attacked by Eve. Therefore,

$$\Pr[F] = (\frac{3}{4})^p \cdot (\frac{q^2}{2}) \cdot (\frac{1}{2})^{n_{diff}}.$$

Based on the Difference Lemma, we have

$$|\Pr[G_1 - G_2]| \leq \Pr[F] = (\frac{3}{4})^p \cdot (\frac{q^2}{2}) \cdot (\frac{1}{2})^{n_{diff}}.$$

Based on three games G_0, G_1 and G_2, we have

$$\begin{aligned} &|\Pr[G_0] - \Pr[G_2]| = |\Pr[A(MR) = 1 | MR = M_B(Q)] \\ &- \Pr[A(MR') = 1 | f \leftarrow \Gamma_{n,n}, MR' \leftarrow f(\,)]| \leq \varepsilon_{prq} \\ &= (\frac{3}{4})^p \cdot (\frac{q^2}{2}) \cdot (\frac{1}{2})^{n_{diff}}, \end{aligned}$$

where the probability ε_{prq} is negligible when the value n_{diff} is big enough.

□

B. Security Analysis of the QKDP with Untrusted Center

This following analyzes the security of the QKDP with untrusted center (KUC) in details.

Adversary's Queries

The adversary's queries represent the capabilities of

adversary Eve, where the adversary isnot a legitimate participant and controls the communications. In KUC, the adversary has the same queries as the KHC, i.e., the Send query, Reveal query and Test query. Moreover, in KUC, the untrusted center can be Eve, tries to eavesdrop the session key, and decrypts the ciphertext transmitted between legitimate users later. To describe the ability, the additional query of Eve has been added and described as follows:

- $(Send, C', \Pi_B^s)$: This query models that Eve is the untrusted center, inputs $(Send, C')$ to an instance Π_B^s, and gets the response from the instance Π_B^s, where $C' = shuffled\_(K \| h)$.

Security Notion

This following describes the Freshness and the Security of KUC in details, which are similar to the KHC.

Freshness:

In KUC, an instance Π_U^s is called Fresh when $AAC(\Pi_U^s) = TRUE$ and Π_U^s has not been asked a Reveal query. With the fresh instance Π_U^s, we can describe the freshness of session key. That is, Π_A^s and Π_B^s establish a session key K, the session key K is called Fresh if and only if both Π_A^s and Π_B^s are Fresh.

Security of KUC:

The KUC includes the Setup phase, the Query phase and the Challenge phase, which describes the beginning of KUC, the abilities of the adversary and the advantage for the adversary to win the game.

- **Setup Phase:** The challenger runs the Setup algorithm, and responds to the adversary with the resulting system parameters, such as u, m and n.
- **Query Phase:** The adversary can ask various queries, such as the Send query and the Reveal query.
- **Challenge Phase:** When the query phase is over, the adversary can submit the Test query to the Fresh instance Π_U^s in an execution of KUC. The challenger will toss an unbiased coin. If the tossing result $b = 1$, the challenger returns the u-bit session key K to the adversary; otherwise, returns a u-bit random string. After receiving the string, the adversary responses b'. The adversary wins if $b' = b$. The advantage for the adversary, Eve, to get the correct session key is $Adv_{KUC}^K(Eve)$. The proposed QKDP is called session key secure if $Adv_{KUC}^K(Eve)$ is negligible.

$$Adv_{KUC}^K(Eve) = \left| \Pr[b' = b] - \frac{1}{2} \right|$$

Definition of Primitive:

The primitives used in the security proof of KUC are the same as the KHC, which are the uncertainty of quantum measurement, the assumption of no cloning, and the assumption of one-way hash function.

- **Uncertainty of Quantum Measurement:** Let $S_i \in \{0,1\}^n$ be the string of polarization direction, $B_i \in \{R, D\}^n$ be the string of polarization basis and QG be the qubit generating algorithm. With the information of (S_i, B_i), the algorithm QG produces qubits $Q_i = QG(S_i, B_i) \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$. Let n_{diff} be the number of bits, where $B_1 \neq B_2$ and $B_1, B_2 \in B_i$. Let $M_{B_i}()$ be the qubit measuring algorithm, which measuring result $MR_i = M_{B_i}(Q)$ is a bit string. The probability for the adversary Eve to break the Uncertainty of Quantum Measurement is denoted as ε_{uqm}. The ε_{uqm} is negligible if

$$\varepsilon_{uqm} = \Pr[MR_1 = MR_2 \mid MR_1 = M_{B_1}(Q_1), MR_2 = M_{B_2}(Q_1)] \leq (\frac{1}{2})^{n_{diff}}.$$

- **Assumption of No Cloning:** Let $S_i \in \{0,1\}^n$ be the string of polarization direction, $B_i \in \{R, D\}^n$ be the string of polarization basis and QG be the qubit generating algorithm. With the information of (S_i, B_i), the algorithm QG produces qubits

$Q_i = QG(S_i, B_i) \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$. Let $QC(\ )$ be the qubit cloning algorithm, which produces qubits $Q'_i = QC(Q_i)$. Let $M_B(\ )$ be the qubit measuring algorithm which produces the measuring result $MR_i = M_{B_i}(Q)$. The probability for the adversary Eve to break the Assumption of No Cloning is denoted as ε_{clone}. The ε_{clone} is negligible if

$$\varepsilon_{clone} = \Pr[MR_1 = MR_2 \mid MR_1 = M_B(Q_1), MR_2 = M_B(Q'_1)] \le (\frac{3}{4})^n.$$

- **Assumption of One-way Hash Function:** Let $H(.)$ be the one-way hash function, which maps $\{0,1\}^u$ to $\{0,1\}^m$. Given a checksum h, the probability for the adversary Eve to compute K is denoted as ε_{hf}, where $h = H(K)$. The one-way hash function is computationally secure if ε_{hf} is negligible under the quantum computation.

The model describes the security environment of the KUC, in which an adversary wants to extract the session key from the execution of KUC. The adversary can be the center, a registered user or malicious outsider. The KUC will achieve the key security against an adversary if the adversary has the negligible advantage to gain the session key in the model.

Security Analysis of KUC

The following game-based proof played between a challenger and the adversary, which environment is defined in the model. A theorem is firstly presented to demonstrate the security of KUC; and then the proof of the theorem is described in details.

Theorem 2. *The proposed KUC is secure based on the Uncertainty of Quantum Measurement and the Assumption of No Cloning. The advantage for an adversary, Eve, to get the correct session key is:*

$$Adv^K_{KUC}(Eve) = \left|\Pr[b' = b] - \frac{1}{2}\right| \le \lambda,$$

where the probability l *is negligible.*

Proof. In this proof, a sequence of games are incrementally defined starting from the real game G_0 to the simulated game G_2. The following describes the games in details.

Game G_0: We describe that the game is an interactive computation between Eve and the challenger Ch. The original attack game G_0 is the same as the game described in the security notion. In the setup phase, the challenger responds to Eve with the system parameters. In the query phase, Eve can make a number of the Send queries and the Reveal queries. In the challenge phase, Eve makes a Test query and guesses the received string is a random string $(b = 0)$ or the session key $(b = 1)$. The advantage for the adversary to guess $b' = b$ is:

$$Adv^K_{KUC}(Eve) = |2 \cdot \Pr[G_0] - 1|.$$

Game G_1: The untrusted center can play the role of Eve and try to get the session key without being noticed by users. The difference between G_0 and G_1 is that Eve has been removed from being the untrusted center. In Step 2, the untrusted center (Eve) can intercept the qubits Q_2 sent from Alice to Bob and randomly produces new qubits $Q'_2 \neq Q_2$ for Bob. In this game G_1, the challenger sends the bases B_1 sent from Alice to Bob in Step 4. Eve can perform the U_j operations on the intercepted qubits Q_2, measure the operated qubits with the R basis, and get the correct string $K \| h$. In Step 5, Bob sends the processed qubits $Q'_3 \neq Q_3$ to Eve. In Step 6, Eve (the center) inputs $(Send, C')$ to Bob through the authenticated classical channel, where $C' = shuffled\_(K \| h)$. The response sent from Bob to Alice is always Fail since the probability for Eve to get the correct C' is negligible. With the correct string $K \| h$, Eve knows the number of 1's and 0's. We assume that the number of 1's and 0's in the string $K \| h$ is $n/2$ and $n/2$, respectively. Therefore, Eve can guess $C' = shuffled\_(K \| h)$ with the probability $1/C^n_{n/2}$. If Eve has the ability of gaining the correct C', Eve will find the difference between G_0 and G_1. Therefore, we

have

$$|\Pr[G_0]-\Pr[G_1]|\le \frac{1}{C_{n/2}^{n}}.$$

Game G_2: This game is similar to the game G_1 of the security analysis in KHC. In Step 2, Eve can intercept the qubits Q_2 sent from Alice to Bob and produces new qubits Q'_2 for Bob. If Eve has the qubit cloning ability, the qubits Q'_2 will be the same as Q_2. In this game, the challenger simulates the bases B_1 sent from Alice to Bob with the random string B'_1 in Step 4. From this simulation, we easily see that the game is indistinguishable from the real attack, unless Eve has the ability of qubit cloning. If Eve has the ability of qubit cloning, Eve will find the difference between G_1 and G_2. We denote the failure event as F. Therefore, we have

$$|\Pr[G_1]-\Pr[G_2]|\le q_s\cdot\varepsilon_{clone}(1+(\frac{1}{2})^m)$$
$$\cdot(1-\varepsilon_{uqm}-(\frac{1}{2})^u)+q_s\cdot(\frac{1}{2})\cdot(\frac{3}{4})^m\cdot\varepsilon_{hf}.$$

In Step 5, the qubits Q_3 can be measured with the R basis by Eve. However, the bases B_1 sent from Alice to Bob has been replaced with the random string B'_1 in Game G_2. Therefore, the measuring result of the qubits Q_3 is a random string. The probability $\Pr[G_2]$ is $1/2$ because all the information sent from the challenger to Eve is random information. According to the games G_0, G_1 and G_2, we can deduce that the advantage for Eve to break the key security of the proposed KUC is

$$Adv_{KUC}^{K}(Eve)=|\Pr[G_0]-\Pr[G_2]|=\left|\Pr[G_0]-\frac{1}{2}\right|\le$$
$$\frac{1}{C_{\frac{n}{2}}^{n}}+q_s\cdot\varepsilon_{clone}(1+(\frac{1}{2})^m)\cdot(1-\varepsilon_{uqm}-(\frac{1}{2})^u)+q_s\cdot(\frac{1}{2})\cdot(\frac{3}{4})^m\cdot\varepsilon_{hf}.$$

Appendix B

Phoenix et al.'s QKDP

The following describes Phoenix et al.'s QKDP in details (see also Figure 3).

Step 1. The center produces *n* qubits Q_1 in a known polarization state, such as $|0\rangle$, and sends the qubits to Alice through the quantum channel.

Step 2. After receiving Q_1, Alice generates a random string K and performs the unitary operation U_i on a qubit based on the bit K_i. When the bit K_i is 0 (1), the unitary operation U_i is $U_0(U_2)$ as indicated in Subsection 3.1, where U_0 does not affect the qubit and U_2 flips the polarization basis of qubit to $|+\rangle$. Furthermore, Alice sends the qubits $Q_2 = U_i(Q_1)$ to Bob through the quantum channel.

Step 3. After receiving the qubits Q_2, Bob generates the random strings R. Moreover, Bob performs the unitary operations U_i on a qubit based on the bit R_i. When the bit R_i is 0

(1), the unitary operation U_i is $U_0(U_2)$. Bob transforms the polarization states of Q_2 to Q_3 and sends Q_3 to the center through the quantum channel.

Step 4. After receiving the qubits Q_3, the center measures a qubit with its original polarization basis (the R basis), which measuring result can be 0 (identical with the original qubit $|0\rangle$) or 1 (inconsistent with the original qubit $|0\rangle$). Then, the center publishes the measuring results to Alice and Bob.

Step 5. After receiving the string, Alice and Bob can derive the shared key string. The identical

measuring result of a qubit is useless for the users; on the other hand, the inconsistent measuring result helps the receiver to deduce the secret bit sent from the sender. For instance, the center can get the measuring result 0 from the qubit measurement of $|0\rangle$ or $|+\rangle$ with the R basis. On the other hand, the center can only get the measuring result 1 from $|+\rangle$, which means that only one user (either Alice or Bob) performs the U_2 operation on the qubit. When the the measuring result of *ith* qubit is inconsistent and $R_i = 0$ (1), Bob can derive the bit K_i as 1 (0). The probability for the center to get the inconsistent measuring results of all qubits is 25% in average. Moreover, users perform the random sampling public discussion to discuss the correctness of their shared bits, in which half of shared bits are spent. Thus, the QKDP only has 13% qubit efficiency in the end of the execution.

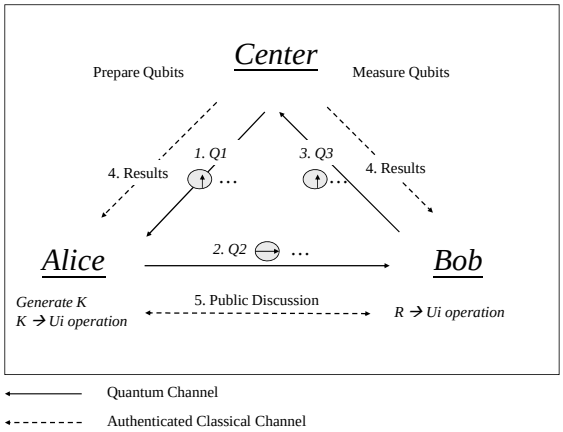

Figure 3: Phoenix et al.'s QKDP.

In this three-party QKDP, legitimate users only equip with unitary operators during the key distribution. Moreover, the QKDP can be executed with the untrusted center. When the measuring result is 1, the center cannot predict that which user (Alice or Bob) has performed the U_2 operation on the qubit. Moreover, if the center publishes incorrect measuring results, it will be detected while users discuss the correctness of their shared bits with the random sampling public discussion. The disadvantage of the QKDP is that it has only 13% qubit efficiency.