Improved and formal proposal for device-independent quantum private query

Jyotirmoy Basak, Kaushik Chakraborty, Arpita Maitra and Subhamoy Maitra

1 Applied Statistics Unit, Indian Statistical Institute, Kolkata, India
2 School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom
3 TCG Centre for Research and Education in Science and Technology, Kolkata, India

E-mail: bjyotirmoy.93@gmail.com

Received 6 July 2023; revised 21 January 2024
Accepted for publication 30 January 2024
Published 13 February 2024

Abstract

In this paper, we present a novel quantum private query (QPQ) scheme that is fully device-independent. As far as we know, this is the first QPQ scheme that uses EPR pairs and offers full device independence. Our approach takes into account the self-testing of shared EPR pairs and the self-testing of projective measurement operators in a mistrustful scenario where neither the client nor the server trusts each other. To achieve full device independence, we also exploit self-testing of a specific class of POVM operators used by the client in our scheme. Additionally, we evaluate the performance of our proposal formally and derive an upper limit of the maximum cheating probabilities (when the protocol does not terminate) for both the dishonest client and the dishonest server.

Keywords: quantum private query, device independence, self-test, projective measurement, POVM

1. Introduction

From the initial proposal due to Chor et al [2], Private Information Retrieval (PIR) attracts extensive attention in the domain of classical cryptography [3–7]. PIR is a two-party (a Server, and a Client) mistrustful cryptographic scenario. Usually in PIR, a server holds a database and a client wants to know some bits from the database. In a trivial approach, the entire database is
given to the client so that he can extract the information accordingly. Naturally, in this kind of proposal, the database owner (i.e. the server) cannot learn anything about the query. However, the inefficiency lies in communicating the entire database.

Another variant of PIR is Symmetric Private Information Retrieval (SPIR) introduced by Gertner et al [8]. SPIR is more stringent than PIR as it takes into account the security of the database too. In SPIR, the entire database is not provided; only the required elements queried by the client are supplied in such a manner that the database owner cannot learn anything about the queried elements. The functionality of SPIR is the same as 1 out of \( N \) Oblivious Transfer (OT) where the server is assumed to hold a \( N \)-bit database and the client retrieves 1 out of those \( N \) bits in a manner such that both data security and user privacy holds. Designing information-theoretically secure SPIR schemes that preserve both user privacy and database security is a challenging task (in quantum as well as in classical domain [8, 9]), as these goals appear to be in conflict. However, under the assumption that there exist some distributed databases which can share certain randomness, it is possible to design a secure SPIR scheme [10]. Of course, there exist classical SPIR proposals based on some computational hardness assumptions [3, 11] but unfortunately in the quantum domain, those hardness assumptions are proved to be tractable [12]. So, there is a need for designing a weaker version of the SPIR schemes that are information-theoretically secure.

The focus of this paper is on that weaker version of SPIR known as Private Query (PQ) that offers unconditional security in the single-server scenario. In contrast to SPIR or 1-out-of-\( N \) OT, PQ enables the client to gather additional information about the database while ensuring that the knowledge regarding the unintended data bits remains negligible. The client’s privacy is safeguarded in a cheat-sensitive manner, assuming that the server will abstain from cheating if there is even a small chance of being detected. The PQ primitive is not as strong as SPIR but stronger than PIR. However, like PIR, the server must deal with the entire database to answer the client’s query, or else the server will gain knowledge about the indices of the client’s request.

The history of quantum private query (QPQ) protocols began with Giovannetti et al’s proposal [13]. This was followed by [14, 15], but these protocols relied on quantum memories, which are not implementable in practice. Several modifications and advancements have been made towards the proposal of the first implementable QPQ scheme by Jakobi et al [16], which was based on a Quantum key distribution (QKD) protocol [17]. This was followed by a flexible generalization by Gao et al [18] and further efficiency improvements suggested by Rao and Jakobi [19]. Zhang et al [20] proposed a QPQ protocol based on the counterfactual QKD scheme [21] and Yang et al developed a flexible QPQ protocol [22] based on the B92 QKD scheme [23]. The domain continues to develop, as seen in recent publications [24, 25]. Some QPQ protocols use entangled states to create a shared key between the server (Bob) and the client (Alice), while others use a single qubit sent to the client, which is prepared in specific states and measured to retrieve the key bit. All the QPQ protocols, despite differences in the key generation procedure, share common concepts. Their security is based on the following fundamental principles.

- The server (Bob) holds the entire final key used to encrypt the database using one-time pad.
- The client (Alice) is usually aware of only 1 final key bit.
- The server (Bob) is ignorant of the specific location of the final key bit known to the client (Alice).
In QPQ, either party may act as an adversary and attempt to compromise security. The client (Alice) may try to learn more number of final key bits (that implies the extraction of more data bits in a single query) that are used to encrypt the database, while the server (Bob) may try to learn the indices of the final key bits known to Alice. Because of this, QPQ is a two-party distrustful cryptographic primitive where both parties are allowed to violate privacy with a small probability, depending on the security requirements. In reality, the desired primitive is as follows-

- The objective is to minimize the amount of information that the malicious server Bob can learn about the query indices from the client Alice without being detected, while Alice aims to conceal these indices. At the same time, the malicious client Alice seeks to gain additional knowledge about the database beyond Bob’s intention, while Bob endeavors to prevent Alice from obtaining extra information.

Recently, Maitra et al. [26] pointed out that the security of existing QPQ proposals (up until that time) relies on the assumption that the communicating parties trust the devices involved in their scheme, just as it is in the case of initial QKD proposals. In device independent (DI) scenario, these assumptions are removed and security is guaranteed without them. However, unlike QKD, proving DI security in QPQ is challenging due to its distrustful nature.

Despite this challenge, a DI-QPQ scheme has been proposed recently in [26] and its finite sample analysis has been discussed in [27]. This enhances the overall security by removing trustful assumptions over the devices. However, the protocol in [26] only introduced a testing phase at the server side, making it a semi-DI version of the Yang et al. [22] QPQ scheme.

Like [22], in almost all the QKD-based QPQ schemes, the client generates the partial key at her side by distinguishing non-orthogonal states. One of the limitations of the scheme proposed in [22] is that Bob and Alice share non-maximally entangled states. Additionally, Alice randomly measures her qubits using projective measurement on a specified basis to retrieve the raw key bits selected by Bob with certainty.

It is well known that maximally entangled states (i.e. Bell states) are easier to prepare and are more robust in a DI setting than non-maximally entangled states. Additionally, POVM measurements provide the best (conclusive) distinction between non-orthogonal states, according to [28, 29].

Considering these facts, instead of pursuing the full DI proposal of the QPQ scheme [22], we introduce a novel QPQ scheme that uses shared EPR pairs and optimal POVM measurements at the client’s side to distinguish non-orthogonal quantum states and retrieve the maximum number of raw key bits. This proposal provides full device-independent certification through self-testing of shared EPR states, self-testing of (the client’s) POVM measurements, and self-testing of (the server’s) projective measurements. We also discuss the security issues formally and manage to provide upper limits on the maximum cheating probabilities for both the server and the client. A detailed comparison of our scheme with the existing QPQ proposals is shown in table 1.

1.1. Our contribution

In the current report, we study the QPQ primitive where data privacy and user security contradict each other. Moreover, because of the cheat sensitivity, if any of the parties try to violate data privacy or user security, it will be detected by the other party.
Given the distrustful nature of QPQ, proving its device independent (DI) security is a challenging task. With that in mind, our proposed scheme aims to maintain both data privacy and user security, while also detecting any attempts by a party to compromise the system’s security. The main contributions of this paper are outlined below.

(i) Unlike the previous semi-DI version in [26], here we come up with a full DI-QPQ scheme using maximally entangled EPR pairs for better preparation and robustness in DI certification. Our proposed QPQ scheme removes device trustworthiness by performing self-testing of EPR pairs (following the procedure of CHSH test), projective measurements operators (following the procedure mentioned in [30]), and POVM operators (following a new strategy mentioned here without imposing any dimension bound). All these self-testing mechanisms provide full DI security, a first of its kind in QPQ (as far as we know).

(ii) Our proposal utilizes optimal POVM measurement at the client’s side, replacing the traditional projective measurement. This allows the client Alice to conclusively distinguish two non-orthogonal states with maximum probability, improving the efficiency of the scheme. The result is that, on average, Alice is able to secure the maximum number of raw key bits with certainty. That means, our proposal also enables Alice to retrieve the maximum number of data bits in a single query.

(iii) We introduce the security definitions for data privacy and user privacy in terms of the maximum fraction of information known by the dishonest party, and also in terms of the probability with which the dishonest party guesses more than the expected amount of information in a scenario where the protocol does not abort. We formally evaluate the performance of our proposal in terms of these security definitions considering all types of attacks that maintain the accuracy requirement. Taking into account all our assumptions, we perform a detailed DI security analysis of our proposal to certify all the devices.
We also discuss briefly about the practical implementation (considering finite number of samples) of our scheme.

1.2. Notations and definitions

Let us first list a few notations.

- $K$: The total number of states needed, assumed to be large.
- $I_k$: $k$ dimensional identity matrix.
- $\mathcal{A}$ ($\mathcal{A}^*$): Alice with honest or dishonest behavior.
- $\mathcal{B}$ ($\mathcal{B}^*$): Bob with honest or dishonest behavior.
- $\mathcal{A}_i$ ($\mathcal{B}_i$): Subsystem of honest (dishonest) Alice indexed by $i$.
- $\mathcal{B}_i$ ($\mathcal{B}^*_i$): Subsystem of honest (dishonest) Bob indexed by $i$.
- $|\phi\rangle_{\mathcal{A}_i\mathcal{B}_i}$: The $i$-th shared state with the first qubit belonging to Alice ($\mathcal{A}_i$) and the second to Bob ($\mathcal{B}_i$).
- $\rho_{\mathcal{A}_i\mathcal{B}_i}$: The density matrix of the $i$-th state.
- $\rho_{\mathcal{A}_i}$ ($\rho_{\mathcal{B}_i}$): The reduced density matrices for Alice and Bob, respectively, of the $i$-th state.
- $X$: The database held by Bob, with $N$ bits.
- $R$ ($R_{\mathcal{A}}$): The entire raw key at Bob’s and Alice’s sides, respectively, each with $kN$ bits.
- $F$ ($F_{\mathcal{A}}$): The entire final key at Bob’s and Alice’s sides, respectively, each with $N$ bits.
- $R_i$ ($R_{\mathcal{A}}^i$): The $i$-th raw key bit of Bob and Alice, respectively.
- $F_i$ ($F_{\mathcal{A}}^i$): The $i$-th final key bit of Bob and Alice, respectively.
- $k$: Number of raw key bits combined by XOR to produce each bit of the final key.
- $l$: Size of the query set $I_l$ of Alice.
- $\mathcal{I}_l$: The set of query indices made by Alice.
- $a$: the classical bit announced by Bob for $i$-th shared state.
- $A$ ($\mathcal{B}$): Alice and Bob’s measurement outcomes, respectively.
- $|0\rangle' = \cos \theta |0\rangle + \sin \theta |1\rangle$.
- $|1\rangle' = \sin \theta |0\rangle - \cos \theta |1\rangle$.
- $\in_{\mathcal{R}}$: uniform random selection from a set.

1.2.1. Adversarial model. In QPQ, which is a distrustful cryptographic primitive, each party has distinct security goals. The security of the entire protocol is termed as ‘Correctness of the Protocol’, while the security of the server (Bob) is referred to as ‘Privacy of the Database Owner’ and the security of the client (Alice) is called ‘Privacy of the User’. These terms were already formally defined in the preliminary version of this paper in [1]. Here we revisit those definitions (mentioned in [1]) and also introduce two new definitions of data privacy and user privacy.

**Definition 1.** Correctness of the Protocol: If both the client (Alice) and the server (Bob) are honest, then after the protocol execution, it is highly likely that Alice will correctly retrieve the expected number of data bits in a single database query. That means, if we consider that the random variable $T$ denotes the event whether the protocol terminates or not (where $T = 1$ implies that the protocol terminates and $T = 0$ otherwise) then in case of honest implementation, following the key generation phase, Alice’s actual knowledge of $X$ data bits is expected to match the expected knowledge of $Y$ data bits i.e.

$$\Pr(|X - Y| \leq \delta_t \wedge T = 0) \geq P_e$$  \hspace{1cm} (1)
where Bob allows deviation by $\delta_t$, and $P_e$ (which should ideally be high) represents the likelihood that $X$ falls within the range of $[Y - \delta_t, Y + \delta_t]$.

**Definition 2.** Robustness of the Protocol: The likelihood of Alice not obtaining any final key bits (or equivalently any database bits), leading to a protocol restart, is low if both Alice and Bob act honestly during the key generation phase of our proposal i.e.

$$\Pr(T = 1) \leq P_d$$

(2)

where $T = 1$ denotes the event that the protocol terminates in honest scenario and $P_d$ denotes the likelihood that no final key bits are known to Alice and the protocol terminates, and this probability should ideally be low.

**Definition 3.** Privacy of the Database Owner: A QPQ protocol is considered to protect data privacy if, in the long run, it terminates with a high probability or if dishonest Alice can only extract a small fraction of the database bits in a single query. Formally, if $D_A^*$ is the average number of bits extracted by dishonest Alice ($A^*$), then the protocol meets the data privacy requirement if

$$E_R(D_A^*) \leq \tau N$$

(3)

where $\tau$ being a small fraction compared to $N$ and the expectation is calculated over the random coin $R$ utilized in the proposal.

The data privacy against dishonest Alice can also be defined (from the correctness definition) in terms of the success probability in guessing more than the expected number of data bits. In this notion, after the key generation phase, either the scheme terminates with high likelihood (as the limit approaches infinity), or the probability that dishonest Alice ($A^*$) correctly retrieves more data bits than expected and the protocol does not terminate is very low. This means that if $T = 0$ denotes the event that the scheme does not terminate, and the number of data bits known to Alice is represented by $X$, and the expected number is represented by $Y$, then after the key generation phase,

$$\Pr(|X - Y| > \delta_t \land T = 0) \leq P_d$$

(4)

where $\delta_t$ represents the allowed deviation (by Bob) from the expected number of data bits and the probability that the actual number of data bits known to Alice ($X$) lies outside the range of $[Y - \delta_t, Y + \delta_t]$ is denoted by $P_d$, which should ideally be very low.

**Definition 4.** Privacy of the User: A QPQ protocol protects against dishonest Bob ($B^*$) if after Alice’s $l$ queries, either the protocol is highly likely to terminate (as the limit approaches infinity), or dishonest Bob can only correctly guess a very small fraction $\delta$ of indices from the set of Alice’s query indices i.e. $I_l$. This means that if $l_{B^*}$ denotes the number of accurately predicted indices by dishonest Bob, then after $l$ queries from Alice,

$$E_{R'}(l_{B^*}) \leq \delta l$$

(5)

where the expectation is based on the random coin $R'$ utilized in the proposal.

The user privacy against dishonest Bob can also be defined in terms of the success probability in guessing a query index correctly from the set of Alice’s query indices. In this notion, either the proposal terminates with high likelihood (as the limit approaches infinity), or the probability of dishonest Bob ($B^*$) accurately guessing one of Alice’s query indices from set $I_l$ and the protocol not aborting is very low. In other words, if $T = 0$ denotes the event that
the scheme does not terminate, and Bob guesses any index $i$ from the database such that the protocol continues, then the probability of $i$ being in Alice’s query index set $I_l$ is low, i.e.

$$\Pr(i \in I_l \land T = 0) \leq P_u$$ (6)

where $P_u$ represents the probability that $i$ is in $I_l$ and the protocol does not terminate ($P_u$ should ideally be very small).

1.3. Assumptions for our DI-QPQ proposal

In this section, we present the assumptions necessary for the security of our proposed QPQ scheme. These assumptions are already explicitly mentioned in the preliminary version of this draft in [1]. However, we revisit the assumptions again for the readability issue and summarize as follows.

(i) This proposal assumes that the measurement devices and state generation device, involved follow the principles of quantum mechanics and the corresponding outcomes are determined by the Born rule.

(ii) We follow the similar assumptions here like the recent DI Oblivious Transfer proposal in [31]. That means for this proposal, we assume that the measurement and state generation devices are described by independent Hilbert spaces and the devices operate independently and identically in all trials. It is also assumed that the honest party randomly and independently chooses inputs for each round.

Note: To identify the deceitful activity (if any) of the dishonest party, the i.i.d. assumption on the honest party’s inputs is necessary in this distrustful setting. More general scenarios without this assumption could also be considered, but they are not covered in this work.

(iii) The honest party can only query the unknown devices at their end, but the dishonest party can manipulate any devices (at their end or the honest party’s end) before the protocol starts. However, after the scheme starts, the dishonest party can no longer modify any devices and can only query them like the honest party. The dishonest party is also assumed to process his/her data independently and identically for each trial.

(iv) In the DI scenario, it is usually assumed that the labs of Alice and Bob are secure and there’s no communication between them. In QPQ scheme, both parties are considered as distrustful, meaning each one wants to obtain as much information from the other while leaking as little as possible. To detect cheating, one party must behave honestly during each testing phase. In local tests, the honest party randomly chooses input bits and self-tests the devices. So, there’s no communication between labs. Whereas in distributed tests, the honest party chooses the inputs for both parties and the dishonest party announces the outputs. That means, in distributed tests, communication between labs is allowed for input and output bits. The honest party is also assumed to have a way of shielding his devices to prevent information leaks until he decides to reveal anything.

Note: If in a distributed test, the dishonest party does not perform measurements on their qubits according to the honest party’s input choice, then the honest party’s ability to detect this dishonest behavior is explained later in the section entitled ‘Security in Device Independent Scenario’.

(v) The inputs in self-tests are selected randomly and without any correlation to the other systems involved in the protocol, including the input bit generation device of the other party and their laboratory. We also assume the presence of quantum memories [32] on both the parties end which are capable of efficiently storing the qubits corresponding to each of the states shared between them before the start of the protocol.
2. Our DI-QPQ proposal

The QPQ protocols are composed of several phases. Depending on the functionality, we have divided the entire protocol into five phases. The first phase is termed ‘entanglement distribution phase’. In this phase, a third party (need not be a trusted one and may collude with the dishonest party) distributes several copies of entangled states between the server (Bob) and the client (Alice). The next phase is called ‘source device verification phase’. In this phase, the server and the client self-test their shared entangled states using CHSH game. The third phase is termed as ‘Bob’s measurement device verification phase’. In this phase, Bob self-tests his measurement device (in some specific measurement basis that will be used for the QPQ protocol).

In QPQ, before the protocol, the server Bob decides how much information the client Alice can retrieve from the database in a single query. For this reason, Bob chooses a parameter $\theta$ and performs measurements on his qubits (of the shared entangled states) in this $\theta$ rotated basis (during the protocol) to restrict Alice’s information about the database$^4$. As Alice and Bob get the measurement devices from an untrusted third party, (in device-independent setting) they need to check the devices before proceeding with the protocol. Here we assume that dishonest Bob’s aim is not only to know Alice’s query indices but also to leak as little additional information about the database as possible. For this reason, in ‘Bob’s measurement device verification phase’, only Bob will act as a referee and choose input bits for both parties. They first perform some measurements assuming the devices as unknown boxes and then after getting the outcome, they conclude about their functionality. After measurement, if the probability of winning the specified game is equal to some predefined value, then they can conclude that Bob’s measurement devices are noiseless for those specified bases.

The next phase of this protocol is termed ‘Alice’s POVM device verification phase’. In the phase, Alice first performs specific measurements assuming the POVM devices as unknown boxes and then concludes about their functionality based on the outcome i.e. in this phase Alice checks the functionality of her POVM device. If the POVM device works as expected, then Alice and Bob generate key bits in the next phase for the remaining instances which is termed as ‘key generation phase’. After this phase, Bob has a secret key such that Alice knows some of those bits and Bob does not know the indices of the bits known by Alice.

In the last phase, i.e. in ‘PQ phase’, Bob encrypts the database using the key generated at his side and sends the encrypted database to Alice. Alice then decrypts the intended data bits using the known key bits at her side.

Although the entire protocol is already mentioned in the preliminary version of this draft in [1], here we outline different steps of our proposal for readability issue. It should be noted that we have not taken into account channel noise and therefore all operations are assumed to be flawless.

(i) **Entangled State Sharing Phase:**
(a) Before the start of different phases of the protocol, a third party first distributes $K$ (where $K$ is assumed to be asymptotically large) number of states, $|\phi\rangle_{AB}$, between Alice and Bob with Alice receiving subsystem $A$ and Bob receiving subsystem $B$ in each pair.

(ii) **Source Device Certification Phase:**

The source device verification phase is composed of two subphases. In the first subphase, Bob acts as a referee, chooses random samples (for testing phase), receives the

$^4$ Once chosen, this value of $\theta$ remains fixed for the entire QPQ protocol.
Algorithm 1. LocalCHSHtest(\mathcal{S}, \mathcal{P}).

- For every \(i \in \mathcal{S}, \mathcal{P}\) does the following:
  
  (a) The device of \(\mathcal{P}\) measures on first qubit of the \(i\)-th state for inputs \(s_i = 0\) and \(s_i = 1\) and outputs \(c_i = 0\) or \(c_i = 1\).
  
  (b) The device of \(\mathcal{P}\) measures on second qubit of the \(i\)-th state for inputs \(r_i = 0\) and \(r_i = 1\) and outputs \(b_i = 0\) or \(b_i = 1\).

- From the inputs \(s_i, r_i\) and their corresponding outputs \(c_i, b_i\), \(\mathcal{P}\) calculates the following quantity,
  \[ C = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} C_i \]
  where \(C_i\) is defined as,
  \[ C_i := \begin{cases} 
    1 & \text{if } s_i r_i = c_i \oplus b_i \\
    0 & \text{otherwise.} 
  \end{cases} \]

- If \(C = \cos^2 \frac{\pi}{8}\), then \(\mathcal{P}\) continues with the protocol, otherwise \(\mathcal{P}\) aborts the protocol. (In the case of honest implementation, this exact desired value can be obtained for asymptotically large number of samples. However, in practice, with finite number of samples, it is nearly always impossible to exactly match with the desired value of the estimated statistic. Hence, a small deviation from the desired value is allowed in practice. A discussion regarding the variation of the deviation range with the sample size is mentioned in appendix A. However, how the existing security definitions will vary with the noise parameter, is out of the scope of this present work and we will try to explore this issue in our future works.)

corresponding qubits from Alice, generates random input bits for those instances and performs a localCHSH test to certify the states. Similarly, in the second subphase, Alice acts as a referee and does the same that Bob does in the previous phase. In each phase, after receiving the inputs, Alice’s and Bob’s device measure the states and return output bits \((c_i, b_i)\). The detail description of different subphases is as follows.

(a) Bob chooses \(\frac{\gamma K}{2}\) instances randomly from these \(K\) shared states (in practice, how Bob and Alice choose the specific value of \(\gamma_1\) from the set \([0, 1]\) is mentioned in appendix A), declares those instances publicly and constructs the set \(\Gamma_{\text{CHSH}}^B\) with these chosen instances.

(b) For all the instances in \(\Gamma_{\text{CHSH}}^B\), Alice sends her qubits to Bob.

(c) For the instances in \(\Gamma_{\text{CHSH}}^B\), Bob plays the role of the referee as well as the two players and plays Local CHSH game.

(d) For every \(i\)-th sample in \(\Gamma_{\text{CHSH}}^B\), Bob randomly generates input bits \(r_i\) and \(s_i\) for his two measurement devices (these devices act as separate parties without any communication), with \(r_i, s_i \in \{0, 1\}\).

(e) Bob performs LocalCHSHtest(\(\Gamma_{\text{CHSH}}^B\), Bob), according to the procedure outlined in algorithm 1 (which is equivalent to the local version of the CHSH game) for the set \(\Gamma_{\text{CHSH}}^B\).

(f) If Bob passes this LocalCHSHtest(\(\Gamma_{\text{CHSH}}^B\), Bob) then both Alice and Bob proceed further, otherwise they abort.

(g) From the rest \(\left( \mathcal{K} - \frac{\gamma K}{2} \right)\) shared states, Alice randomly chooses \(\frac{\gamma K}{2}\) (in practice, how Bob and Alice choose the specific value of \(\gamma_1\) from the set \([0, 1]\) is mentioned in appendix A) instances, declares those instances publicly and constructs the set \(\Gamma_{\text{CHSH}}^A\) with these chosen instances.

(h) For all the instances in \(\Gamma_{\text{CHSH}}^A\), Bob sends his qubits to Alice.
(i) For these instances in $\Gamma_{\text{CHSH}}^A$, Alice plays the role of the referee as well as the two players and plays Local CHSH game.

(j) For every $i$-th sample in $\Gamma_{\text{CHSH}}^A$, Alice randomly generates input bits $r_i$ and $s_i$ for her two measurement devices (these devices act as separate parties without any communication), with $r_i, s_i \in \{0, 1\}$.

(k) Alice performs LocalCHSHtest($\Gamma_{\text{CHSH}}^A$, Alice), according to the procedure outlined in algorithm 1 (which is equivalent to the local version of the CHSH game) for the set $\Gamma_{\text{CHSH}}^A$.

(l) If Alice passes the LocalCHSHtest($\Gamma_{\text{CHSH}}^A$, Alice) test then both Bob and Alice proceed to the next phase where Bob self-tests his measurement device, otherwise they abort.

(iii) Bob’s Measurement Device Verification Phase:

(a) In the previous phase (i.e. in source device certification phase), Bob and Alice selected a total of $|\Gamma_{\text{CHSH}}| - |\Gamma_{\text{CHSH}}|$ samples, with $\Gamma_{\text{CHSH}} = \Gamma_{\text{CHSH}}^A \cup \Gamma_{\text{CHSH}}^B$. For every $i$-th instance from the remaining $(K - |\Gamma_{\text{CHSH}}|)$ samples, Bob performs the following

- Bob generates random bit $r_i \in_R \{0, 1\}$ for every $i$-th state, as the input of his device. (essentially, these randomly generated bits serve as the initial key bits for Bob, meaning $R_i = r_i$).
- If $r_i = 0$, Bob measures his share of the $i$-th state using the operator $\{B_{0i}^1, B_{0i}^0\}$ and produces the output bit $b_i = 0$ and $b_i = 1$ respectively.
- If $r_i = 1$, Bob measures his share of the $i$-th state using the operator $\{B_{1i}^1, B_{1i}^0\}$ and produces the output bit $b_i = 0$ and $b_i = 1$ respectively.
- Bob announces $a_i = 0$ if his device outputs $b_i = 0$, meaning the operator $B_{0i}^1$ or $B_{0i}^0$ was applied for the $i$-th instance.
- Bob announces $a_i = 1$ if his device outputs $b_i = 1$, meaning the operator $B_{1i}^1$ or $B_{1i}^0$ was applied for the $i$-th instance.

(b) Bob chooses $\frac{\gamma_2 (K - |\Gamma_{\text{CHSH}}|)}{2}$ instances randomly from these $(K - |\Gamma_{\text{CHSH}}|)$ shared states (how Bob and Alice choose the specific value of $\gamma_2$ from the set $\{0, 1\}$ is mentioned in appendix A), declares them publicly and constructs a set $\Gamma_{\text{obs}}^B$ with these instances.

(c) Alice then chooses $\frac{\gamma_2 (K - |\Gamma_{\text{CHSH}}|)}{2}$ (how Bob and Alice choose the specific value of $\gamma_2$ from the set $\{0, 1\}$ is mentioned in appendix A) instances randomly from the rest $(K - |\Gamma_{\text{CHSH}}| - \frac{\gamma_2 (K - |\Gamma_{\text{CHSH}}|)}{2})$ shared states, declares the instances publicly and constructs a set $\Gamma_{\text{obs}}^A$ with these instances.

(d) Bob and Alice create a set $\Gamma_{\text{obs}}$ with all their chosen samples i.e. $\Gamma_{\text{obs}} = \Gamma_{\text{obs}}^A \cup \Gamma_{\text{obs}}^B$.

(e) They then perform OBStest($\Gamma_{\text{obs}}$), by following the procedure mentioned in algorithm 2, for the set $\Gamma_{\text{obs}}$.

(iv) Alice’s POVM Device Verification Phase:

(a) After Bob’s measurement device verification phase, Alice and Bob move on to this phase with the remaining $(K - |\Gamma_{\text{CHSH}}| - |\Gamma_{\text{obs}}|)$ shared states, referred to as $\Gamma_{\text{POVM}}$.

(b) Alice selects $\gamma_3 |\Gamma_{\text{POVM}}|$ randomly chosen samples from $\Gamma_{\text{POVM}}$ (how Alice selects $\gamma_3$ is explained in appendix A) and calls this set $\Gamma_{\text{test}}^\text{POVM}$.

(c) Alice then performs KEYgen($\Gamma_{\text{test}}^\text{POVM}$) followed by POVMtest($\Gamma_{\text{test}}^\text{POVM}$) according to the procedures described in algorithms 3 and 4 respectively, for the set $\Gamma_{\text{test}}^\text{POVM}$.
Algorithm 2. OBStest(S).

- For every $i \in S$, Bob and Alice do the following-
  (a) Bob randomly generates a bit $s_i$ (either 0 or 1) to input into Alice’s device and announces the input publicly.
  (b) Alice measures her share of the $i$-th state for inputs $s_i = 0$ and $s_i = 1$, and obtained outputs $c_i = 0$ or $c_i = 1$.
  (c) Bob has already measured his share of the $i$-th state for inputs $r_i = 0$ and $r_i = 1$, and obtained outputs $b_i = 0$ or $b_i = 1$.
  (d) Alice and Bob announce their inputs $s_i, r_i$ and their corresponding outputs $c_i, b_i$.
- Bob randomly generates a bit $s_i$ (either 0 or 1) to input into Alice’s device and announces the input publicly.
- Bob and Alice estimate the following quantity from their declared outcomes,
  \[ \beta = \frac{1}{4} \sum_{i,r,c,b \in \{0,1\}} (-1)^{d_{srcb}} \alpha^{1/2} \langle \phi_{AB} | A_s^i \otimes B_r^i | \phi_{AB} \rangle \]
  where $\alpha = \frac{(\cos \theta + \sin \theta)}{(\cos \theta - \sin \theta)}$ and $d_{srcb}$ is as follows,
  \[ d_{srcb} := \begin{cases} 0 & \text{if } sr = c \oplus b \\ 1 & \text{otherwise.} \end{cases} \]
- If $\beta = \frac{1}{\sqrt{2}(\cos \theta - \sin \theta)}$, then they continue with the protocol, otherwise they abort the protocol.

Algorithm 3. KEYgen(S).

- For every $i \in S$, Alice performs the following-
  (a) If Bob stated $a_i = 0$, Alice uses measurement device $M_0^i = \{M_0^0, M_1^0, M_2^0\}$ to measure her qubit in the shared state indexed by $i$.
  (b) If Bob stated $a_i = 1$, Alice uses measurement device $M_1^i = \{M_0^1, M_1^1, M_2^1\}$ to measure her qubit in the shared state indexed by $i$.

(v) Key Generation Phase:
  (a) After Alice’s POVM device verification phase, Alice continues with the remaining shared states ($|\Gamma_{\text{POVM}} - \gamma_3| \Gamma_{\text{POVM}}|)$, which she denotes as $\Gamma_{\text{Key}}$.
  (b) Alice performs the KEYgen ($\Gamma_{\text{Key}}$) for these shared states.
  (c) After KEYgen($\Gamma_{\text{Key}}$), Alice determines the original raw key bits based on her measurement results-
    - For each shared state with $a_i = 0$, if Alice gets $M_0^0(M_1^0)$, she concludes the $i$-th raw key bit as 0(1). If she receives $M_2^0$, she ignores it.
    - Similarly, for each shared state with $a_i = 1$, if Alice obtains $M_0^1(M_1^1)$, she concludes the $i$-th raw key bit as 0(1). If she receives $M_2^1$, she ignores it.
  (d) Bob and Alice then proceed to the PQ phase with the shared states in $\Gamma_{\text{Key}}$. This set contains $kN$ many states, where $k > 1$ and $k$ is exponentially smaller than $N$, the number of bits in the database.
Algorithm 4. POVMtest($S$).

- In this step, Alice first separates instances where Bob declared $a_i = 0$ into a set $S^0$, and the rest (where Bob declared $a_i = 1$) into $S^1$.
- Alice assumes that for each set $S^y$ (where $y = a_i$, the values declared by Bob), the states at her side are either $\rho_x^y$ or $\rho_x^y \oplus \frac{1}{2}$ (where $x = r_i$, the raw key bit values randomly chosen by Bob).
- For each set, Alice calculates the parameter $\Omega^y$ as
  \[
  \Omega^y = \sum_{b \in \{0,1\}} (-1)^{b \oplus x} \text{tr} \left[ M_y^b \rho_x^y \right]
  \]
  where $M_y^b$ is Alice’s measurement outcome in KEYgen().
- If for every $S^y$ ($y \in \{0,1\}$),
  \[
  \Omega^y = \frac{2\sin^2 \theta}{1 + \cos \theta}
  \]
  then Alice continues with the scheme, otherwise Alice aborts the scheme.

(e) Alice and Bob conduct the next phase using the $kN$ raw key bits obtained from the shared states.

(vi) Private Query Phase:
(a) Bob and Alice possess a raw key of length $kN$ bits with Bob aware of all its values and Alice aware of some of them (without Bob knowing which bits Alice knows).
(b) Bob rearranges the order of the $kN$-bit string by randomly announcing a permutation, and both parties then apply that permutation to their raw key bits.
(c) Bob divides the raw key into $N$ partitions, each with $k$ bits, and informs Alice of each bit’s position. Alice and Bob then XORed the bits of each substring to form the final key, which is $N$ bits long. If Alice is not aware of any bits of the final key, the protocol must be repeated.
(d) Alice, who knows only the $i$-th bit of Bob’s final key $F$, requests the $j$th bit of the database $m_j$ by announcing a permutation $P_A$. This permutation moves the $i$-th bit of the final key to the $j$th position. Bob applies the permutation $P_A$ on the final key $F$ and uses it to encrypt the database with a one-time pad. Alice can recover $m_j$ as it is encrypted by $F_i$ after receiving the encrypted database.
(e) Alice must announce the permutation $l$ times if she wants to retrieve $l$ bits of the database with only one known final key bit.
(f) If Alice knows more than one final key bit then she announces a permutation that links her known final key bits to the database bits she wants to know. This way she can retrieve multiple intended database bits in a single query.

Our QPQ Proposal (In Case of Honest Implementation)

- Alice and Bob share $K$ EPR pairs, with Alice having the first qubit and Bob having the second qubit.
- For each shared states, they generate raw key bits by following the procedure mentioned below.
  - Bob randomly generates a value of either 0 or 1 for the $i$-th raw key bit $r_i$.
  - If $r_i = 0$, Bob performs measurement on his share for the $i$-th state in $\{|0\}, |1\rangle\}$ basis, otherwise (i.e. for $r_i = 1$) he performs measurement in $\{|0\}', |1\rangle\}'$ basis where
\[ |0\rangle = (\cos \theta |0\rangle + \sin \theta |1\rangle) \text{ and } |1\rangle = (\sin \theta |0\rangle - \cos \theta |1\rangle) \] (the value of \( \theta \) is chosen as per the relation specified in equation (13)).

- Bob announces \( a_i = 0 \) (\( a_i = 1 \)) if the outcome at his side corresponding to the \( i \)-th shared state is either \( |0\rangle \langle 1| \) or \( |0\rangle \langle 1'| \).
- When Bob declares \( a_i = 0 \), Alice performs measurement on her share of the \( i \)-th state using the POVM \( \mathcal{M}^0 = \{ M^0_0, M^0_1, M^0_2 \} \) where
  \[
  M^0_0 = \frac{(\sin \theta |0\rangle - \cos \theta |1\rangle)(\sin \theta \langle 0| - \cos \theta \langle 1|)}{1 + \cos \theta} \\
  M^0_1 = \frac{1}{1 + \cos \theta} |1\rangle \langle 1| \\
  M^0_2 = I - M^0_0 - M^0_1
  \]

- Similarly, for \( a_i = 1 \), Alice performs measurement on her share of the \( i \)-th state using the POVM \( \mathcal{M}^1 = \{ M^1_0, M^1_1, M^1_2 \} \) where
  \[
  M^1_0 = \frac{(\cos \theta |0\rangle + \sin \theta |1\rangle)(\cos \theta \langle 0| + \sin \theta \langle 1|)}{1 + \cos \theta} \\
  M^1_1 = \frac{1}{1 + \cos \theta} |0\rangle \langle 0| \\
  M^1_2 = I - M^1_0 - M^1_1
  \]

- When Bob declares \( a_i = 0 \), if Alice gets \( M^0_0(M^1_0) \), she concludes the \( i \)-th raw key bit as 0 (1). For measurement outcome \( M^0_2 \), Alice remains uncertain.
- When Bob declares \( a_i = 1 \), if Alice gets \( M^1_0(M^1_1) \) for \( a_i = 1 \), she concludes the \( i \)-th raw key bit as 0 (1). For measurement outcome \( M^1_2 \), Alice remains uncertain.

- At first, Bob decides the value of \( \theta \) and the number of raw key bits needed to generate each bit of the final key based on equation (13). Bob and Alice generate a final key by processing their raw key bits through permutation and XOR such that the final key and the database are of the equal size and Bob knows all the bits but Alice knows only some bits of the final key.
- Bob encrypts the whole database using onetime pad with his final key and sends it to Alice.
- Alice recovers the desired bits from the encrypted database using her partial knowledge of the final key. The pictorial representation of our proposed DI-QPQ scheme is depicted in figure 1.

### 3. Analysis of the protocol

In this section, we cover the workings of our proposed scheme. We start by examining the accuracy of the protocol in subsection A, followed by estimating the security parameters involved (subsection B). Finally, we delve into the security aspects of our scheme in subsection C. It’s worth noting that all our analyses are based on asymptotic scenarios, and the actual values of parameters may vary in practice based on the sample size selected.
3.1. Correctness of the protocol

We begin by demonstrating the accuracy of the protocol.

**Theorem 1.** In case of honest implementation of our proposal, on average, Alice can correctly retrieve around \((1 - \cos \theta)kN\) bits of the raw key \(R\) at the end of key generation phase.

**Proof.** Bob and Alice have \(kN\) raw key bits after key generation phase. These raw key bits were generated from \(kN\) copies of maximally entangled states of the form

\[
\frac{1}{\sqrt{2}} \left( |0\rangle_A |0\rangle_B + |1\rangle_A |1\rangle_B \right)
\]

where, \(|0\rangle = (\cos \theta |0\rangle + \sin \theta |1\rangle\) and \(|1\rangle = (\sin \theta |0\rangle - \cos \theta |1\rangle\). Here \(\theta\) may vary from 0 to \(\frac{\pi}{2}\).

The generation of such \(kN\) raw key bits can be redefined as follows.

Bob prepares a random bit stream \(R = r_1 \ldots r_{kN}\) of length \(kN\). If \(r_i = 0\), Bob measures his qubits in \(\{ |0\rangle, |1\rangle \}\) basis. Whereas, if \(r_i = 1\), Bob measures his qubit in \(\{ |0\rangle', |1\rangle' \}\) basis. After each measurement Bob announces a bit \(a_i \in \{ 0, 1 \}\). If he gets \(|0\rangle\) or \(|0\rangle'\), he announces \(a_i = 0\). If he gets \(|1\rangle\) or \(|1\rangle'\), he announces \(a_i = 1\). Now, Alice’s job is to guess the value of each \(r_i\).

Thus, whenever Bob declares \(a_i = 0\), Alice can understand that Bob gets either \(|0\rangle\) or \(|0\rangle'\) and the shared qubit of her side also collapses to \(|0\rangle\) or \(|0\rangle'\) respectively. However, to obtain the value of the raw key bit, Alice has to distinguish these two states with certainty. As, \(|0\rangle\) and \(|0\rangle'\) are non-orthogonal states (when \(\theta \neq \frac{\pi}{2}\), Alice cannot distinguish these two states with certainty for all the instances.

According to the strategy mentioned in the protocol, whenever Bob declares \(a_i = 0\), Alice chooses the POVM \(\{ M_{00}, M_{01}, M_{02} \}\). After measurement, if Alice receives the outcome \(M_{00}\), she concludes that Bob’s measurement outcome was \(|0\rangle\). In such case, Alice concludes that \(r_i = 0\). If Alice receives the outcome \(M_{10}\), she concludes that Bob’s measurement outcome
Similarly, for the state $\langle a | 0 \rangle$, Alice can guess (on average) around
\[ \frac{\cos \theta}{\cos \theta} = 1 \]
So, according to the proposed scheme, the overall success probability of Alice in guessing a
state $\langle a | 0 \rangle$ is
\[ \frac{\cos \theta}{\cos \theta} = 1 \]
We now calculate the corresponding success probabilities of getting different results for the
states $\langle a | 0 \rangle$ and $\langle a | 0' \rangle$.
For $\langle a | 0 \rangle$, the success probabilities will be
\[ \Pr (M_0^a || | 0 \rangle) = \langle 0 | M_0^a | 0 \rangle = (1 - \cos \theta) \]
\[ \Pr (M_1^a || | 0 \rangle) = \langle 0 | M_1^a | 0 \rangle = 0 \]
\[ \Pr (M_2^a || | 0 \rangle) = \langle 0 | M_2^a | 0 \rangle = \cos \theta. \]
Similarly, for the state $\langle a | 0' \rangle$, the success probabilities will be
\[ \Pr (M_0^a || | 0' \rangle) = \langle 0' | M_0^a | 0' \rangle = 0 \]
\[ \Pr (M_1^a || | 0' \rangle) = \langle 0' | M_1^a | 0' \rangle = (1 - \cos \theta) \]
\[ \Pr (M_2^a || | 0' \rangle) = \langle 0' | M_2^a | 0' \rangle = \cos \theta. \]
Similarly, whenever Bob declares $a_i = 1$, Alice chooses the POVM \{ $M_0^a$, $M_1^a$, $M_2^a$ \}. In a similar
way, we can calculate the success probability here. We formalize all the conditional probabilities
in the following table (i.e. in table 2).

| Cond. Probability of Alice | $A = M_0^a / M_0^a$ | $A = M_1^a / M_1^a$ | $A = M_2^a / M_2^a$ |
|----------------------------|-----------------------|-----------------------|-----------------------|
| 0  $B = | 0 \rangle$   | 1 - $\cos \theta$     | 0                     | $\cos \theta$         |
| 0  $B = | 0' \rangle$   | 0                     | 1 - $\cos \theta$     | $\cos \theta$         |
| 1  $B = | 1 \rangle$   | 1 - $\cos \theta$     | 0                     | $\cos \theta$         |
| 1  $B = | 1' \rangle$   | 0                     | 1 - $\cos \theta$     | $\cos \theta$         |

Table 2: Probabilities of different POVM outcomes.

was $|0'\rangle$. In such a case, Alice concludes that $r_i = 1$. However, if the measurement outcome is
$M_0^a$, then Alice remains uncertain about the value of the raw key bit. Alice follows the similar
methodology for $a_i = 1$.

Now, we calculate the success probability of Alice to guess each $r_i$ correctly. Let us assume
that $\Pr (M_j^a || | a \rangle)$ denotes the corresponding success probability of getting the result $M_j^a$
when the given state is $|a \rangle$ i.e.
\[ \Pr (M_j^a || | a \rangle) = \langle a | M_j^a | a \rangle. \]
We now calculate the corresponding success probabilities of getting different results for the
states $|0 \rangle$ and $|0'\rangle$.
For $|0 \rangle$, the success probabilities will be
\[ \Pr (M_0^a || | 0 \rangle) = \langle 0 | M_0^a | 0 \rangle = (1 - \cos \theta) \]
\[ \Pr (M_1^a || | 0 \rangle) = \langle 0 | M_1^a | 0 \rangle = 0 \]
\[ \Pr (M_2^a || | 0 \rangle) = \langle 0 | M_2^a | 0 \rangle = \cos \theta. \]
Similarly, for the state $|0'\rangle$, the success probabilities will be
\[ \Pr (M_0^a || | 0' \rangle) = \langle 0' | M_0^a | 0' \rangle = 0 \]
\[ \Pr (M_1^a || | 0' \rangle) = \langle 0' | M_1^a | 0' \rangle = (1 - \cos \theta) \]
\[ \Pr (M_2^a || | 0' \rangle) = \langle 0' | M_2^a | 0' \rangle = \cos \theta. \]
Similarly, whenever Bob declares $a_i = 1$, Alice chooses the POVM \{ $M_0^a$, $M_1^a$, $M_2^a$ \}. In a similar
way, we can calculate the success probability here. We formalize all the conditional probabilities
in the following table (i.e. in table 2).

According to the protocol, if $a_i = 0$ and Alice gets $M_0^a(M_0^a)$, she outputs $r_{A_i} = 0(1)$. When
$a_i = 1$ and she gets $M_0^a(M_1^a)$, she outputs $r_{A_i} = 0(1)$. Thus, the success probability of Alice to
guess the $i$-th raw key bit $r_i$ of Bob can be written as
\[ \Pr (r_{A_i} = r_i) = \Pr (r_{A_i} = 0, r_i = 0) + \Pr (r_{A_i} = 1, r_i = 1) \]
\[ = (1 - \cos \theta). \]
So, according to the proposed scheme, the overall success probability of Alice in guessing a
raw key bit is equal to $(1 - \cos \theta)$. This implies that at the end of the key establishment phase,
Alice can guess (on average) around $(1 - \cos \theta)kN$ many raw key bits with certainty. ☐
3.2. Estimation of parameters for PQ phase

In this subsection, the different parameter values are calculated to ensure that both user and data privacy are preserved. After key generation phase, Bob and Alice share \( kN \) raw key bits, with Bob having full knowledge of them and Alice having partial knowledge. In private query phase, both Alice and Bob cut their raw keys in some particular positions to prepare \( N \) sub strings of length \( k \) such that \( k = \frac{|\text{Key}|}{N} \) where \(|\text{Key}|\) denotes the total number of raw key bits at the private query phase and \( N \) denotes the number of database bits. Alice and Bob then perform bit wise XOR among the bits of each sub string to get the \( N \) bit final key \( F \). Here, \( r_i (1 \leq i \leq kN) \) denotes the \( i \)-th raw key of Bob and \( f_i (1 \leq i \leq N) \) denotes the \( i \)-th final key of Bob. Based on the procedure mentioned in private query phase for generating final key bits, the relation between \( r_i \) and \( f_i \) can be written as,

\[
f_i = \bigoplus_{j=(i-1)k+1}^{i} r_j \quad (1 \leq i \leq N)
\]

where \( \oplus \) denotes addition modulo 2.

It will be clearer by a toy example. Consider \( N = 10 \) and \( k = 2 \). Let us assume that the raw key at Bob’s side is,

\[
\begin{array}{cccccccccc}
0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1
\end{array}
\]

and after the key generation phase, the raw key at Alice’s side is,

\[
\begin{array}{cccccccccccc}
? & 1 & ? & 0 & ? & ? & ? & 1 & 0 & 1 & ? & 0 & ? & 0 & ? & 1
\end{array}
\]

e.g. Alice knows the values of 2nd, 5th, 11th, 12th, 14th, 17th and 20th key bits of the original raw key (? stands for inconclusive key bit i.e. the positions where Alice cannot guess the key bits with certainty).

Now, after the modulo operation on the raw key, Bob’s final key will be,

\[
\begin{array}{cccccccccc}
1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0
\end{array}
\]

and Alice’s final key will be,

\[
\begin{array}{cccccccc}
? & ? & ? & ? & ? & 1 & ? & ? & ?
\end{array}
\]

Thus, the number of known key bits by Alice is reduced from 7 to 1. The significance of such modulo operation is to enhance the security of the protocol. This is similar to the privacy amplification in a QKD protocol.

Estimation of the security parameter \( \theta \):

In the proposed scheme, Alice can expect to know approximately \((1 - \cos \theta)kN\) of the \( kN \) shared raw key bits. The expected value of \( n_r \), the number of raw key bits known to the client Alice after the key generation phase of our scheme, can be expressed as,

\[
E[n_r] = (1 - \cos \theta)kN. \quad (7)
\]

Bob and Alice combine \( k \) raw key bits using XOR to produce each bit of the final key. So, for Alice to correctly guess a final key bit, she must correctly guess all \( k \) corresponding raw key bits, which has a probability of \((1 - \cos \theta)^k\). Let \( n_f \) denote the total final key bits known to Alice. It follows that \( n_f \) is a binomial random variable with \( N \) trials and probability \( P_f = (1 - \cos \theta)^k \). Hence, the expected number of final key bits known by Alice after the key generation phase is,

\[
E[n_f] = P_fN \approx (1 - \cos \theta)^k N. \quad (8)
\]
Our DI proposal requires that dishonest Alice correctly measure (using the designated POVM) in order to pass the DI testing stage. It is recognized that the maximum probability of distinguishing between two non-orthogonal states is \((1 - \cos \theta)\) [28]. This means that in the non-abort scenario, dishonest Alice’s guess for the \(i\)-th raw key bit, \(R_i\), is limited to at most 
\[
(1 - \cos \theta) \quad \text{(9)}
\]
where \(A_i^*\) denotes dishonest Alice’s subsystem corresponding to the \(i\)-th shared state.

As after Bob’s measurement, Alice’s states are independent and the measurement devices at dishonest Alice’s side are also independent and memoryless, the maximum probability that dishonest Alice can guess the \(i\)-th final key bit \(F_i\) will be 
\[
(1 - \cos \theta) \quad \text{e}.
\]
\[
\Pr[F_{A^*} = F_i] = P_f \leq (1 - \cos \theta)^k \quad \text{(10)}
\]

Based on the results in equations (8) and (10), it can be concluded that in non-abort scenario, the expected maximum number of final key bits guessed correctly by dishonest Alice will be limited by,
\[
E[F_{A^*}] \leq (1 - \cos \theta)^k N. \quad \text{(11)}
\]

Our proposal involves encrypting the database, which is the same size as the final key, by bitwise XORing it with the final key. Thus, a correct guess of a final key bit also implies a correct guess of the corresponding database bit. Hence, in a single query, if the scheme does not terminate, dishonest Alice’s expected number of correctly guessed database bits is also upper bounded by \((1 - \cos \theta)^k N\) i.e.
\[
E[D_{A^*}] \leq (1 - \cos \theta)^k N. \quad \text{(12)}
\]

In our scheme, for the protocol to continue, Alice must know at least one final key bit, while Bob wants Alice to know less than two final key bits. Thus, the following condition must be met in the non-abort scenario.
\[
1 \leq E[n_f] < 2,
\]

This implies that,
\[
1 \leq (1 - \cos \theta)^k N < 2 \quad \Rightarrow \quad \frac{1}{N} \leq (1 - \cos \theta)^k < \frac{2}{N}. \quad \text{(13)}
\]

All these results boil down to the following conclusion,

**Corollary 1.** To ensure that Alice knows at least one final key bit but no more than one, Bob needs to select \(k\) and \(\theta\) such that,
\[
\frac{1}{N} \leq (1 - \cos \theta)^k < \frac{2}{N}.
\]
Now, for our proposal, we derive the limits on the values of $P_a$ (from definition 2) and $P_c$ (from definition 1) set by the correctness condition. These limits are already derived in the preliminary version of this work in [1]. However, for readability issues, we revisit those derivations here.

**Estimation of the security parameters $P_a$ and $P_c$:**

Initially, we evaluate the likelihood that the protocol will not end during the honest scenario. Then, using the obtained upper bound on $(1 - \cos \theta)^k$ from equation (13), we can calculate a lower bound on $P_c$ with the Chernoff–Hoeffding inequality [33] (since we consider a scenario where dishonest Alice measures i.i.d).

Our scheme calculates the probability of Alice correctly guessing a final key bit as $(1 - \cos \theta)^k$. So, the probability of Alice not guessing a final key bit is $1 - (1 - \cos \theta)^k$. Therefore, the likelihood of Alice not knowing any of the $N$ final key bits is

$$
1 - (1 - \cos \theta)^k \approx e^{-(1 - \cos \theta)^k N},
$$

i.e. for our proposal, we obtain the following bound on the value of $P_a$.

$$
P_a \leq e^{-(1 - \cos \theta)^k N}. \tag{15}
$$

If Bob sets $\theta$ such that $(1 - \cos \theta)^k = \frac{1}{N}$, then equation (15) gives us the following result according to the relation in equation (13).

$$
P_a \leq e^{-1}. \tag{16}
$$

That means our proposed scheme results in a small value of $P_a$. If $T = 0$ denotes the event that the scheme does not terminate then the likelihood of our proposal not terminating in an honest scenario i.e. Alice knows at least one final key bit, is calculated as

$$
\Pr(T = 0) \geq [1 - e^{-1}] \tag{17}.
$$

Therefore, our proposed scheme has a high likelihood of not aborting, as demonstrated by the above calculation. We now mention the Chernoff–Hoeffding inequality [33].

**Proposition 1 (Chernoff–Hoeffding inequality).** Let $X = \frac{1}{m} \sum_{1 \leq i \leq m} X_i$ be the average of $m$ independent random variables $X_1, X_2, \ldots, X_m$ with values $(0, 1)$, and let $E[X] = \frac{1}{m} \sum_{1 \leq i \leq m} E[X_i]$ be the expected value of $X$. Then for any $\delta_{CH} > 0$, we have

$$
\Pr(|X - E[X]| \geq \delta_{CH}) \leq \exp(-2\delta_{CH}^2 m).
$$

Our scheme defines $X_i = 1$ if the $i$-th final key bit is known to Alice (i.e. she gets conclusive outcomes from the POVMs), and $X_i = 0$ otherwise. Total $N$ final key bits result in the random variable $X$ as the sum of these $X_i$ values. If the scheme does not terminate, the expected number of final key bits that Alice should know is $Y = (1 - \cos \theta)^k N$.

To ensure the value of $X$ lies within the error margin of $\delta_{CH} = \epsilon (1 - \cos \theta)^k N$ from the expected value, we use the Chernoff–Hoeffding inequality. This is because Alice’s final key bits are independent, as the collapsed states and measurement devices are also independent and memoryless. The value of $X$ and $Y$ are calculated under the assumption that the scheme does not terminate. So, if we assume that $T = 0$ denotes the event that the scheme does not
terminate then using the expression for the Chernoff–Hoeffding bound from proposition 1, we can write that,

\[
\Pr[(X - Y) < \delta_{CH} \land T = 0] \\
\geq 1 - \exp\left(-2\delta_{CH}^2 m\right).
\]

(18)

The key generation phase results in N final key bits shared by Alice and Bob. Among those N bits, we aim to have the number of final key bits known to Alice fall within \( [p - \epsilon, p + \epsilon] \), where \( p = (1 - \cos \theta)^k N \) and the allowed deviation is \( \delta_{CH} = \epsilon (1 - \cos \theta)^k N \). If we assume that \( T = 0 \) denotes the event that the scheme does not terminate then using the expression in (18) with \( \delta_{CH} \) and \( m \) values, we get,

\[
\Pr[(X - Y) < \delta_{CH} \land T = 0] \\
\geq 1 - \exp\left(-2\delta_{CH}^2 N\right)
\]

where \( \delta_{CH} = \epsilon (1 - \cos \theta)^k N \).

(19)

We have already established the bound \( \frac{1}{N} \leq (1 - \cos \theta)^k < \frac{\epsilon}{N} \) for \( (1 - \cos \theta)^k \) from equation (13) for our proposed scheme. If we let Bob choose \( \theta \) and \( k \) such that \( (1 - \cos \theta)^k = \frac{1}{N} \), then substituting this value into equation (19) will yield,

\[
\Pr[(X - Y) < \epsilon \land T = 0] \\
\geq 1 - \exp\left(-2\epsilon^2 N\right)
\]

(20)

In our proposal, a correct guess of a final key bit means a correct guess of the related data bit. So, as per definition 1, we can say that when Alice and Bob are both honest, the lower bound of the parameter \( P_c \) in our proposal is determined by,

\[
P_c \geq 1 - \exp\left(-2\epsilon^2 N\right)
\]

(21)

That means the likelihood of Alice knowing the expected number of final key bits and the scheme not terminating is high in the honest scenario of our proposed scheme, as the value of \( N \) is large in practice.

Bob chooses \( \theta \) and \( k \) so that Alice knows at least one and fewer than two final key bits. So, the deviation, \( \delta_{CH} \), has the following bound according to equation (13).

\[
\epsilon \leq \delta_{CH} < 2\epsilon.
\]

(22)

That means the upper bound of \( \epsilon \) can be derived from \( 2\epsilon \leq 1 \), yielding \( \epsilon \leq \frac{1}{2} \).

To evaluate performance, we consider the variant 1-out-of-2 probabilistic oblivious transfer (for \( N = 2 \) and \( k = 1 \)). If Bob chooses \( \theta \) such that \( (1 - \cos \theta) = \frac{1}{2} \) (minimum value for \( N = 2, k = 1 \)), the expected number of final key bits (or data bits) Alice can retrieve in a single query is \( \left(\frac{1}{2} \times 2\right) = 1 \). Like before, if we assume that \( T = 0 \) denotes the event that the scheme does not terminate then from equation (17), we can say that in honest scenario, for this 1 out of 2 variant,

\[
\Pr(T = 0) \geq (1 - e^{-1}) \approx 0.632.
\]

(23)
The equation (21) implies that if the variant 1-out-of-2 probabilistic oblivious transfer is considered with $\epsilon = \frac{1}{2}$, then the likelihood of Alice receiving the expected number of final key bits and the protocol not aborting is lower bounded by,

$$P_c \geq (1 - e^{-1}) \approx 0.632.$$  

(24)

3.3. Security of the protocol

Here, we point towards the security issues of our proposal. The security results achieved here are also mentioned in the preliminary version of this paper in [1]. However, here we have mentioned the detailed proofs of all our results and showed how these results certify device-independent security, data security, and user security for our proposed scheme.

Based on the results in corollary 2 and theorems 2–4, we conclude about the DI security of our proposed scheme. All these results guarantee that either the proposal terminates with high probability (as the limit approaches infinity) or the devices involved in our proposal attain the desired values of the parameters $C$, $\beta$, $\Omega^0$ and $\Omega^1$. Later on, we move towards deriving upper bounds on the information gained by dishonest Alice and dishonest Bob. In lemma 1, we show that dishonest Alice cannot guess (on average) more than $(1 - \cos \theta)$ fraction from the entire raw key. Lemma 2 together with corollary 5 shows that dishonest Bob can guess only $\frac{1}{\pi}$ fraction from the query index set of Alice.

3.3.1. Security in device independent scenario. The proposed scheme undergoes device independent testing in three phases. The first two are in the source device verification phase and Bob’s measurement device verification phase. The third takes place in Alice’s POVM device verification phase.

In source device verification phase, at first Local CHSH game has been performed by each of Alice and Bob independently (as mentioned in LocalCHSHtest) at their end for some randomly chosen samples. In this phase, both Alice and Bob test individually whether the states provided by the third party are EPR pairs. Bob and Alice choose the samples randomly for which they want to perform LocalCHSHtest and share this information publicly to get the corresponding qubits from the other party and also to identify all the samples for which they perform LocalCHSHtest.

As QPQ is a distrustful scheme, both the parties may not behave honestly in every phase of the protocol. For this reason, here we assume that the party who acts honestly for a particular phase, will take the responsibilities of the referee as well as the two parties in the CHSH game to ensure the random and independent choice of inputs for the devices involved in the LocalCHSHtest at his end. This guarantees that in LocalCHSHtest, the inputs to the devices are random and independent.

The results from the rigidity of CHSH game in [34, lemma 4.2] lead us to the following conclusion.

Corollary 2 (verification of shared states). The LocalCHSHtest of source device certification phase either detects if Alice’s and Bob’s devices achieve $C = \cos^2 \frac{\pi}{8}$, meaning they were given EPR pairs (or the unitary equivalent of the actual states) by the third party, or the scheme is likely to abort in the long run.

In the next phase, Bob verifies his measurement device. Here, Bob is assumed to act honestly in Bob’s measurement device verification phase, as it is clear from lemma 2 that a dishonest Bob who wants to guess Alice’s query indices more accurately must let Alice know more
data bits in a single query, violating assumption 4 that neither party reveals more information to get more information from the other.

At first Bob starts by randomly choosing inputs for his device and measuring the particles. Then Bob and Alice independently pick samples and discuss publicly. Bob generates random input bits for Alice and announces them publicly, so Alice can measure her particles based on the bits. After measurements, they both publicly share inputs and outputs and calculate $\beta$ as in the OBSTest. From this result, one can conclude the following.

**Theorem 2 (Bob’s measurement device verification).** In OBSTest, either Bob’s measurement devices achieve the value of the parameter $\beta = \frac{1}{\sqrt{2\sin^2 \theta}}$ (i.e. his devices correctly measure in $\{0,1\}$ and $\{0',1'\}$ basis where $|0'\rangle = (\cos \theta|0\rangle + \sin \theta|1\rangle)$, $|1'\rangle = (\sin \theta|0\rangle - \cos \theta|1\rangle)$, or the protocol terminates with a high likelihood of failure (as the limit approaches infinity).

A detailed proof of this theorem is provided in appendix B, using the same method outlined in [30] for certifying non-maximally incompatible observables.

This implies that the LocalCHSHTest certifies the states provided by the third party and OBSTest certifies the projective measurement device (for the specific measurement bases used in OBSTest) of Bob. As Bob declares $a_i$ values for all the shared instances before OBSTest and Alice randomly chooses some of those instances for OBSTest, the successful completion of OBSTest also implies that for all the remaining instances (i.e. for the instances which are not chosen for OBSTest), Alice’s state must be either $|0\rangle\langle 0|$ or $|0'\rangle\langle 0'|$ whenever Bob declares $a_i = 0$ and must be either $|1\rangle\langle 1|$ or $|1'\rangle\langle 1'|$ whenever Bob declares $a_i = 1$.

The third DI test is performed in Alice’s POM device verification phase. The protocol moves to this phase once both Alice and Bob have passed the first two DI tests. So, Alice and Bob are in this phase implies that both Bob’s projective measurement device and their shared states are noiseless. Now, this testing phase basically guarantees the functionality of Alice’s POM device. Note that in this phase, Bob need not test his measurement device again. During OBSTest, his devices are tested already. However, Alice has to shift to a new measurement device for better conclusiveness. Device independent security demands that Alice’s new device should be tested further for certification. In this phase, Alice measures the selected instances with either device $M^0 = M^0_0, M^0_1, M^0_2$ or $M^1 = M^1_0, M^1_1, M^1_2$ based on the declared $a_i$ values. She calculates $\Omega^0$ and $\Omega^1$ from the measurement outcomes and verifies if they equal $\frac{2\sin^2 \theta}{1 + \cos \theta}$. Theorem 3 (Theorem 4) shows that, for the instances where $a_i = 0 (a_i = 1)$, if Alice observes that $\Omega^0 = \frac{2\sin^2 \theta}{1 + \cos \theta}$ ($\Omega^1 = \frac{2\sin^2 \theta}{1 + \cos \theta}$) then it guarantees that the measurement devices are the desired POM $\{D^0_0, D^0_1, D^0_2\}$ i.e. $M^0 = D^0_0 (\{D^0_1, D^1_1\}$ i.e. $M^1 = D^1_0$).

**Theorem 3 (Verification of Alice’s measurement device $M_0$).** POMTest either results in a high probability of termination of this proposed scheme (as the limit approaches infinity), or it guarantees that for the instances where Bob declares $a_i = 0$, Alice’s measurement devices attain $\Omega^0 = \frac{2\sin^2 \theta}{1 + \cos \theta}$, meaning they are of this specified form (up to a local unitary),

\[
M^0_0 = \frac{1}{1 + \cos \theta} |1'\rangle\langle 1'| \quad (25)
\]

\[
M^0_1 = \frac{1}{1 + \cos \theta} |1\rangle\langle 1| \quad (26)
\]

\[
M^0_2 = I - M^0_0 - M^0_1, \quad (27)
\]

where $|1'\rangle = \sin \theta|0\rangle - \cos \theta|1\rangle$. 


Theorem 4 (verification of Alice’s measurement device $M_A$). POVMtest either results in a high probability of termination of this proposed scheme (as the limit approaches infinity), or it guarantees that for the instances where Bob declares $a_i = 1$, Alice’s measurement devices attain $\Omega^1 = \frac{2 \sin^2 \theta}{1 + \cos \theta}$, meaning they are of this specified form (up to a local unitary),

$$M_0^1 = \frac{1}{1 + \cos \theta} \langle 0' \rangle \langle 0' |$$

$$M_1^1 = \frac{1}{1 + \cos \theta} \langle 0 \rangle \langle 0 |$$

$$M_2^1 = I - M_0^1 - M_1^1,$$

where $|0' \rangle = \cos \theta |0 \rangle + \sin \theta |1 \rangle$.

The proofs of these two theorems are deferred to the subsection entitled Verification of Alice’s POVM elements of the appendix C. In the proof, we restate the functionality of the POVM devices in the form of a two party game (namely POVMgame), consider a general form for the single qubit three outcome POVM $\{M_0^1, M_1^1, M_2^1\}$ and show that if the input states are chosen randomly between $|0 \rangle |0 \rangle |1 \rangle |1 \rangle$ and $|0' \rangle |0' \rangle |1' \rangle |1' \rangle$ and if $\Omega^0 = \frac{2 \sin^2 \theta}{1 + \cos \theta}$ ($\Omega^1 = \frac{2 \sin^2 \theta}{1 + \cos \theta}$) then $M_0^0 = D_0^1$, $M_1^0 = D_0^1$, $M_2^0 = D_0^1$, $M_3^0 = D_1^0$, $M_4^0 = D_1^0$, $M_5^0 = D_1^0$.

Note: Here, we claim that if Alice and Bob successfully pass the LocalCHSH test, the OBStest and the POVMtest mentioned in our DI proposal, then in the actual QPQ scheme, none of Alice and Bob can retrieve any additional information in the noiseless scenario. Now, suppose that our claim is wrong i.e. Alice and Bob can pass all the tests mentioned in our scheme and later Alice can retrieve more data bits (than what she intends to know) in a single query or Bob can guess Alice’s query indices with a more certain probability (than his intended probability).

We now discuss this issue in the context of a particular form of non-i.i.d. attack, where a specific number of states are independently corrupted (more general attacks are also possible but these are outside the scope of this work). In this context, we will show that if some of the corrupted states are included during the testing phases, then there is some probability of being caught in the asymptotic limit.

At the beginning of our scheme, the untrusted third party shares all the states with Alice and Bob. As in the source device certification phase, both the parties choose the states randomly from the shared instances for the local tests at their end, the dishonest party can not guess beforehand the shared instances that the honest party will choose at his end for the local test. According to our assumption, the dishonest party can not manipulate the honest party’s device once the protocol starts. So, to successfully pass the LocalCHSH test at the honest party’s end, the shared states must be EPR pairs as specified in our scheme. This implies that the source device certification phase certifies all the states provided by the untrusted third party.

We now explain these things more formally. Let us suppose that initially, the untrusted third party colludes with either the dishonest Alice or the dishonest Bob and shares either $\mathcal{K}_A$ corrupted states in favour of Alice (let us denote this type of states as $A$-type) or $\mathcal{K}_B$ corrupted states in favour of Bob (let us denote this type of states as $B$-type) among $\mathcal{K}$ shared states. So, while choosing randomly for the LocalCHSH test at honest Bob’s end, the probability that a chosen state is of $A$-type is $\frac{\mathcal{K}_A}{\mathcal{K}}$. Similarly, for the LocalCHSH test at honest Alice’s end, the probability that a chosen state is of $B$-type is $\frac{\mathcal{K}_B}{\mathcal{K}}$. Let us further assume that for the $A$-type states, the value of the parameter $C$ is $C_A$ (where $C_A = C + \epsilon_A$ such that $\epsilon_A > 0$) and for the $B$-type states, the value of the parameter $C$ is $C_B$ (where $C_B = C + \epsilon_B$ such that $\epsilon_B > 0$).
Now, suppose that only Alice is dishonest and the third party supplies $K_A$ number of corrupted states (in favour of dishonest Alice) along with $(K - K_A)$ actual states. Then, in the localCHSH test at Bob’s end, the probability that a chosen state is not of the $A$-type is $(1 - \frac{K_A}{K})$. One can easily check that this probability is also same for a chosen state in the final QPQ phase. As, dishonest Alice’s aim is to gain as much additional data bits as possible in the final QPQ phase, she needs to choose the value of $K_A$ such that $(K - K_A) = c$ where $c$ is exponentially smaller than $K$ (i.e. she will try to maximize the probability that a state chosen for the final QPQ phase is of the $A$ type). Then, the probability that Bob will choose none of the corrupted states (i.e. the $A$ type states) among his chosen $\frac{2K}{c}$ states for the LocalCHSH test at his end is,

$$\left(1 - \frac{K_A}{K}\right)^{\frac{2K}{c}} = \left(\frac{c}{K}\right)^{\frac{2K}{c}},$$

which is very small compared to $K$. Similarly, whenever Bob is dishonest, the same thing can be shown for the LocalCHSH test at honest Alice’s end. This implies that if the third party colludes with the dishonest party and supplies corrupted states then the probability that none of those corrupted states are chosen for the LocalCHSH test at the honest party’s end is very small.

In our scheme, we consider the ideal scenario where there are no channel noise. So for dishonest Alice, to successfully pass the LocalCHSH test at the honest Bob’s end, the following relation must hold in the noiseless condition.

$$\frac{K_A C_A}{K} + \frac{(K - K_A)C}{K} = C$$

$$K_A C_A + (K - K_A)C = KC$$

$$K_A (C_A - C) = 0.$$ 

Now, replacing the values of $C_A$ from the relation $C_A = C + \epsilon_A$, one can get,

$$K_A \epsilon_A = 0. \quad (31)$$

As the value of $\epsilon_A > 0$, from this relation, one can easily conclude that in the noiseless scenario, the value of $K_A$ must be zero to successfully pass the LocalCHSH test at the honest Bob’s end. Similarly, one can show that whenever Bob is dishonest, the value of $K_B$ must be zero to successfully pass the LocalCHSH test at the honest Alice’s end. In practice, for finite number of samples, one can show that the values of $K_A$ and $K_B$ must be very small to successfully pass the local test at the honest party’s end.

Here, all the states are shared between the two parties before the start of the protocol and the dishonest party can not manipulate the honest party’s device after the start of the protocol. In this study, as the focus is on the i.i.d. scenario, it is easy to conclude that the protocol will either abort with high probability in the long run, or the LocalCHSH test will verify that the states shared in the QPQ scheme have the desired value of $C$.

The next DI testing is done in Bob’s measurement device verification phase where Bob and Alice perform distributed test to certify Bob’s device. Here, one may think that if Bob is dishonest, then for the instances chosen in Bob’s measurement device verification phase and in Alice’s POVM device verification phase, he will measure in the actual measurement basis at his end to detect the fraudulent behaviour of Alice, and later for the instances to be used for the actual QPQ phase, he will measure in some different basis to guess the positions of Alice’s known key bits.
Lemma 2 shows that if dishonest Bob wants to increase his chances of guessing Alice’s query indices, he must let dishonest Alice know more data bits in a single query, but this goes against assumption 4, which prohibits the parties from leaking more information to gain extra information from the other party. Additionally, it was shown in [16] by Jakobi et al. that in the PQ phase of any QPQ proposal, if the dishonest server tries to guess the client’s query indices with more certain probability then it would provide incorrect answers to the honest client which damages the reputation of the server as a database owner. That’s why, for any QPQ proposal, the server is assumed to behave honestly throughout the protocol as there exist a non-zero probability of being caught cheating.

From the discussion in lemma 1, it is also clear that for our scheme, the client Alice performs optimal strategy at her end. That means, dishonest Alice cannot retrieve more data bits in a single query without manipulating the shared states and Bob’s measurement device. Thus, to ensure that dishonest Alice is not getting any additional data bits, Bob must behave honestly in Bob’s measurement device verification phase to certify his device after the successful completion of source device certification phase.

In our scheme, before the Bob’s measurement device verification phase, Bob generates a random bit for each of his qubits and measures his qubits accordingly. In the Bob’s measurement device verification phase, Bob generates random bits for each of the Alice’s qubits chosen for Bob’s measurement device verification phase and declare those bits so that Alice can measure her particles accordingly. As Bob behaves honestly in Bob’s measurement device verification phase (to restrict Alice from knowing additional data bits) and chooses all the inputs randomly for OBStest, there is no possibility that the inputs for OBStest are chosen according to some dishonest distribution. From the analysis of theorem 2, it is clear that if the inputs are chosen randomly then OBStest certifies that Bob’s measurement device measures correctly in \{ |0\rangle, |1\rangle \} and \{ |0'\rangle, |1'\rangle \} basis for our proposed QPQ scheme.

This implies that the successful completion of source device certification phase and Bob’s measurement device verification phase certifies that the shared states are EPR pairs and Bob’s measurement device measures correctly for all the instances. This also implies that for all the remaining instances (that will be used for Alice’s POVM device verification phase and in the actual QPQ phase), Alice has non-orthogonal qubits (i.e. either \{ |0\rangle \} or \{ |0'\rangle \} for \(a_i = 0\) and either \{ |1\rangle \} or \{ |1'\rangle \} for \(a_i = 1\)) at her end.

It is already mentioned that in our scheme, the client Alice performs optimal (POVM) measurement at her end to extract maximal number of data bits in a single query. So, after successful completion of source device certification phase and Bob’s measurement device verification phase, Alice must behave honestly in Alice’s POVM device verification phase to ensure that her measurement device is the optimal one. For this reason, Alice must measure her qubits accordingly as mentioned in KEYgen() and POVMtest() to certify her device. From the analysis of theorems 3 and 4, it is clear that the successful completion of Alice’s POVM device verification phase certifies Alice’s POVM device.

Note that in the proof of theorems 3 and 4 in appendix C (entitled Verification of Alice’s POVM Elements), we have not imposed any dimension bound like the self-testing of POVM in a prepare and measure scenario in [35]. So, the devices that perform a Neumark dilation of this mentioned POVM (i.e. the equivalent larger projective measurement on both the original state and some ancilla system instead of the actual POVM measurement) could still achieve the intended value of \(\Omega\). But both of these operations produce the same output probabilities, which is sufficient for the purposes of this work.

Therefore, from all these discussions, one can conclude the following-
Corollary 3. Our DI proposal either terminates with high likelihood (as the limit approaches infinity), or it confirms that the devices in our QPQ proposal meet the target values of $C$, $\beta$, and $\Omega^p$ (or $\Omega^p$) in the LocalCHSH test, OBStest, and POVMtest respectively.

3.3.2. Security of database against dishonest alice. In this subsection, we calculate the number of raw key bits that an dishonest Alice can determine in our proposed scheme’s key generation phase.

Theorem 5. In our proposal, in the absence of POVMtest, dishonest Alice can retrieve, at most, $\left(\frac{1}{2} + \frac{1}{2} \sin \theta\right)$ fraction of bits from the entire raw key, inconclusively (i.e. the indices of the correctly guessed bits are unknown), during the key generation phase.

Proof. At the end of the key generation phase, dishonest Alice ($A^d$) and honest Bob ($B$) share $kn$ raw key bits obtained from $kn$ EPR pairs. The $i$-th copy of the state is given by $|\phi^+\rangle_{A^d_iB_i} = \frac{1}{\sqrt{2}} |00\rangle_{A^d_iB_i} + \frac{1}{\sqrt{2}} |11\rangle_{A^d_iB_i}$, where $i$-th subsystem of Alice and Bob is denoted by $A^d_i$ and $B_i$ respectively. At Alice’s side the reduced density matrix is of the form

$$\rho_{A^d_i} = \text{tr}_{B_i} \left[ |\phi^+\rangle_{A^d_iB_i} \langle \phi^+ | \right] = \frac{\mathbb{I}_2}{2}.$$

At the beginning, Bob measures each of his part of the state $|\phi^+\rangle_{A^d_iB_i}$ in either $\{ |0\rangle, |1\rangle \}$ basis or in $\{ |0^\prime\rangle, |1^\prime\rangle \}$ basis. The choice of the basis is completely random as this choice depends on the random raw key bit values chosen by Bob. Let $\rho_{A^d_i|a_i}$ denotes the state at Alice’s side after the choice of Bob’s measurement basis. For $r_i = 0$, we have,

$$\rho_{A^d_i|a_i=0} = \text{tr}_{B_i} \left[ |\phi^+\rangle_{A^d_iB_i} \langle \phi^+ | \right] = \frac{1}{2} \left( |00\rangle_{A^d_iB_i} + |11\rangle_{A^d_iB_i} \right) = \frac{\mathbb{I}_2}{2}.$$

Similarly, for $r_i = 1$, we have, $\rho_{A^d_i|a_i=1} = \frac{\mathbb{I}_2}{2} = \rho_{A^d_i}$. This implies that $\rho_{A^d_i|a_i} = \rho_{A^d_i}$. In Bob’s measurement device verification phase, Alice knows the declared $a_i$ values for all the instances. Let $\rho_{A^d_i|a_i}$ denotes the state of Alice given the value of $a_i$. According to the protocol,

$$\rho_{A^d_i|a_i=0} = \frac{1}{2} |0\rangle |0\rangle + \frac{1}{2} |0^\prime\rangle |0^\prime\rangle$$

$$\rho_{A^d_i|a_i=1} = \frac{1}{2} |1\rangle |1\rangle + \frac{1}{2} |1^\prime\rangle |1^\prime\rangle.$$

This implies that for a fixed $a_i = 0$ ($a_i = 1$) if Alice wants to guess the value of $r_i$ then she needs to distinguish the state from the ensemble of states $\{|0\rangle |0\rangle, |0^\prime\rangle |0^\prime\rangle \}$ (or $\{|1\rangle |1\rangle, |1^\prime\rangle |1^\prime\rangle \}$). In other words, whenever Bob measures his qubit and announces the bit $a_i = 0$, Alice knows that Bob gets either $|0\rangle$ or $|0^\prime\rangle$. Similarly, when Bob announces the bit $a_i = 1$, Alice knows that Bob gets either $|1\rangle$ or $|1^\prime\rangle$. So, to retrieve the value of the original raw key bit, Alice needs to distinguish between the states $|0\rangle$ and $|0^\prime\rangle$ or between the states $|1\rangle$ or $|1^\prime\rangle$.

Now, in the absence of the POVMtest (i.e. if Alice’s measurement device is not tested), Alice can choose any measurement device at her side to distinguish the non-orthogonal states generated at her side. As it is known that non-orthogonal quantum states cannot be distinguished
perfectly, Alice cannot guess the value of each raw key bit with certainty. This distinguishing probability has a nice relationship with the trace distance between the states in the ensemble \cite{36}. According to this relation we have,

\[
\Pr_{\text{guess}}[r_i|\rho_{A_i^*}|a_i=b] = \frac{1}{2} \left( 1 + \frac{1}{2} ||0\rangle\langle 0| - |0'\rangle\langle 0'| || \right)
\]
\[
\leq \frac{1}{2} \left( 1 + \sqrt{1 - F(|0\rangle\langle 0|, |0'\rangle\langle 0'|)} \right)
\]
\[
= \frac{1}{2} (1 + \sin \theta) = \frac{1}{2} + \frac{1}{2} \sin \theta.
\]

One can check that \(\Pr_{\text{guess}}[r_i|\rho_{A_i^*}|a_i=0] = \Pr_{\text{guess}}[r_i|\rho_{A_i^*}|a_i=1]\). This implies that if Alice is allowed to use any measurement device at her end after Bob’s measurement device verification phase then Alice can successfully retrieve the \(i\)-th raw key bit \(r_i\) with probability at most \(\left( \frac{1}{2} + \frac{1}{2} \sin \theta \right)\). As after Bob’s measurement device verification phase, the qubits at Alice’s side are all independent, dishonest Alice can inconclusively retrieve (on average) atmost \(\left( \frac{1}{2} + \frac{1}{2} \sin \theta \right)\) fraction of bits of the entire raw key.

\[\Box\]

Note: Here, the term ‘inconclusive’ means that Alice cannot determine the positions of the accurately guessed key bits with certainty. For example, whenever Alice tries to guess each of the key bits randomly, she can guess correctly for around half of the instances. However, she cannot tell with certainty what are those instances for which she guesses correctly.

Now let us consider the operator \(E = \{E_0, E_1, E_2\}\) where,

\[E_0 \equiv \frac{1}{\sin \theta} (\sin \theta |0\rangle - \cos \theta |1\rangle) (\sin \theta |0\rangle - \cos \theta |1\rangle)\]
\[E_1 \equiv \frac{1}{\sin \theta} |1\rangle\langle 1|\]
\[E_2 \equiv I - E_0 - E_1.\]

One can easily check that this operator \(E = \{E_0, E_1, E_2\}\) is not a valid POVM as \(E_2\) is not positive semi-definite. Let us consider the operator \(E' = \{E_0', E_1'\}\) where

\[E_0' \equiv E_0 + \frac{E_2}{2}\]
\[E_1' \equiv E_1 + \frac{E_2}{2}.\]

Now, this is a valid POVM to distinguish \(|0\rangle\) and \(|0'\rangle = (\cos \theta |0\rangle + \sin \theta |1\rangle)\). If a party consider the strategy that for the outcome \(E_0'\), he consider the corresponding input qubit as \(|0\rangle\) and \(|0'\rangle\) otherwise, then one can check that this is the POVM corresponding to the optimal success probability (i.e. \(\frac{1}{2} + \frac{\sin \theta}{2}\)) in distinguishing \(|0\rangle\) and \(|0'\rangle\). However, the guessing outcome of this POVM is uncertain as the inconclusive element (the outcome which cannot determine the state with certainty) \(E_2\) is involved in both the elements \(E_0'\) and \(E_1'\) of the POVM \(E'\). So, in the proof of theorem \ref{5}, we refer the optimal guessing probability as inconclusive (i.e. uncertainty about the positions of the known key bits).

In theorem \ref{5}, we show that if Alice is allowed to choose any measurement device at her side then, on average, dishonest Alice can correctly retrieve at most around \(\left( \frac{1}{2} + \frac{\sin \theta}{2}\right)\) fraction from the entire raw key but she remains uncertain about the positions of those known bits.

However, in this DI proposal, dishonest Alice’s (\(A^*\)) main intension is to conclusively (i.e. with certainty about the positions of the correctly guessed key bits) retrieve as many raw
key (as well as final key) bits as possible because otherwise she cannot know which data bits she has retrieved correctly. For this reason, dishonest Alice has to perform the mentioned POVM measurement at her end to retrieve maximum number of raw key bits conclusively. Because of this, one can get a bound on the number of raw key bits that dishonest Alice can guess (on average) in this DI-QPQ proposal.

**Lemma 1.** Either this proposed DI-QPQ scheme terminates with high likelihood in the long run, or dishonest Alice ($A'$) can retrieve (on average) $(1 - \cos \theta)$ fraction of bits from the entire raw key after the key generation phase of our proposal.

**Proof.** According to our proposal, after the Alice’s POVM device verification phase, the client Alice has $kN$ independent non-orthogonal qubits at her end. For each of these instances, dishonest Alice now tries to distinguish between the non-orthogonal states either $|0\rangle$ and $|0'\rangle$ (for $a_i = 0$) or $|1\rangle$ and $|1'\rangle$ (for $a_i = 1$).

In this regard, she chooses the measurement device $\{M^0_i, M^1_i, M^2_i\}$ when Bob announces $a_i = 0$ and measurement device $\{M'_0, M'_1, M'_2\}$ when Bob announces $a_i = 1$.

Whenever the outcome is $M^0_i$ ($M'_i$), Alice concludes that the state is $|0\rangle$ ($|1\rangle$). If it is $M^1_i$ ($M'_i$), she concludes that the state is $|0'\rangle$ ($|1'\rangle$). The guessing remains inconclusive (i.e. cannot guess the outcome with certainty) only when the measurement outcome is $M^2_i$ ($M'_i$).

It is evident from [28] that the maximum probability in successfully distinguishing two non-orthogonal states is $(1 - \cos \theta)$. From theorem 1, we get that in our protocol, the success probability of Alice in guessing a key bit correctly and conclusively is also $(1 - \cos \theta)$. As Alice has to measure each of her qubits independently depending on the declared $a_i$ values, on average she can conclusively retrieve $(1 - \cos \theta)$ fraction from the entire raw key. This concludes the proof.

Dishonest Alice can employ a broader attack strategy by storing all the photons in a quantum memory and deferring the measurements until the initial step of the PQ phase where Bob discloses the qubits that contribute to each final key bit. In this scenario, without performing optimal measurements individually for each of the $k$ qubits, dishonest Alice can perform a joint optimal measurement on all the $k$ qubits associated with a final key bit to extract the key bits. It is well-known that the probability of correctly identifying one of two equally likely quantum states (say $\rho_0$ and $\rho_1$) is upper bounded by $\frac{1}{2} + \frac{1}{2}D(\rho_0,\rho_1)$, where $D(\rho_0,\rho_1)$ represents the trace distance. In the case of a joint Helstrom measurement by dishonest Alice on $k$ qubits (related to a final key bit in our proposal), this probability boils down to $\left(\frac{1}{2} + \frac{\sin^2 \theta}{2}\right)$ as the number of added qubits ($k$) increases. Furthermore, as explained in the note following the proof of theorem 5, this optimal measurement would be inconclusive. In other words, dishonest Alice cannot accurately determine the indices of her correctly guessed final key bits, which is a crucial requirement for the PQ primitive. Therefore, this joint measurement attack is ineffective for our (and any other) PQQ proposal.

Although there is a chance that dishonest Alice can successfully pass all tests and learn more data bits than allowed through statistical fluctuations, the likelihood of this happening is low according to corollary 3. Now from definition 3 and equation (12), we can conclude the following.

**Corollary 4.** In the case of dishonest Alice and honest Bob, either our proposal will terminate (as the limit approaches infinity) or dishonest Alice will, on average, be able to retrieve $\tau$ fraction of bits conclusively from the entire final key, where

$$\tau \leq (1 - \cos \theta)^k.$$  \hspace{1cm} (32)
By using the upper bound from equation (13) in place of \((1 - \cos\theta)^k\), one can obtain the following bound on the value of \(\tau\).

\[
\tau < \frac{2}{N} \tag{33}
\]

This relation signifies that in our DI-QPQ proposal, \(\tau\) is significantly smaller than \(N\).

Now, we validate the probabilistic definition of data privacy for this proposed scheme and show that the probability \(Pr[|X - Y| > \delta \wedge T = 0]\) is negligible where \(X\) and \(Y\) denote the actual and expected number of data bits known to Alice and \(T = 0\) denotes the event that the scheme does not terminate. More specifically, we will calculate the probability with which dishonest Alice can guess more than the expected number of final key bits (with a deviation more than the \(\epsilon\) fraction of the expected number of final key bits).

The negligibility of the probability \(Pr[|X - Y| > \delta \wedge T = 0]\) can be shown using the properties of basic probability theory. Note that the probability \(Pr[|X - Y| > \delta \wedge T = 0]\) is upper bounded by both \(Pr[|X - Y| > \delta]\) and \(Pr[T = 0]\), according to the properties \(Pr[A \wedge B] \leq Pr[A]\) and \(Pr[A \wedge B] \leq Pr[B]\). As in our scheme, we consider the i.i.d. assumption, there will be two different subcases- 1) all the devices attain ideal values in all the testing phases (i.e. in LocalCHSHtest, OBTest and POVMtest) 2) all the devices do not attain ideal values in all the testing phases.

For the first subcase, from the correctness result (i.e. the value of \(P_e\) for our scheme in equation (21)) and the DI security statement in corollary 3, one can easily conclude that \(Pr[|X - Y| > \delta] \leq negl(N)\) where \(negl(N)\) denotes negligible in \(N\). For the second subcase, by an analysis similar to the proof of theorem 2 and from the DI security statement in corollary 3, it can be concluded that \(Pr[T = 0] \leq negl(N)\). This implies that for both of these two subcases, \(Pr[|X - Y| > \delta \wedge T = 0] \leq negl(N)\) (under the i.i.d. assumption).

Although it is easy to derive the negligibility of the expression \(Pr[|X - Y| > \delta \wedge T = 0]\) for both the two subcases, in general for the second subcase, it is hard to derive the exact bound on the probability with which dishonest Alice can guess more than the expected number of final key bits. For our proposed scheme, as Alice performs optimal POVM measurement at her end, it is relatively easier to derive an upper bound on the parameter \(P_d\) for our scheme because it is unlikely that dishonest Alice can retrieve more number of raw key bits (on average) by performing any other measurements at her end.

To derive the exact bound on the parameter \(P_d\) for this proposal, like the discussion in subsection B considering \(X\) and \(Y\) be the actual and expected number of final key bits respectively for Alice, here from the Chernoff–Hoeffding inequality \([33]\) mentioned in proposition 1, one can conclude the following,

\[
Pr[|X - Y| \geq \delta_{CH} \wedge T = 0] \\
\leq \exp\left(-2\delta_{CH}^2N\right). \tag{34}
\]

Here, we aim to calculate the likelihood of the value of \(X\) being outside the error range of \(\delta_{CH} = \epsilon (1 - \cos\theta)^k N\) from its expected value.

From the relation in equation (13), it can be easily derived that whenever Bob selects \(\theta\) so that \((1 - \cos\theta)^k = \frac{1}{N}\), the equation (34) becomes,

\[
Pr[|X - Y| \geq \epsilon \wedge T = 0] \\
\leq \exp\left(-2\epsilon^2N\right). \tag{35}
\]
So, from definition 3, the parameter $P_d$ in our proposed scheme (that corresponds to dishonest Alice and honest Bob) can be upper bounded by,

$$P_d \leq \exp(-2\epsilon^2 N).$$

(36)

That means the probability that dishonest Alice can learn more than the expected amount of final key bits (beyond the $\epsilon$ deviation) while the protocol does not abort is very low in practice because the value of $N$ is very large.

For the purpose of illustration, again we evaluate here the performance of our scheme as a 1 out of 2 probabilistic oblivious transfer, where $N = 2$ and $k = 1$. From expression (36), with an error margin of $\epsilon = \frac{1}{2}$, the probability that dishonest Alice can guess more than the expected number of final key bits (which is 1) is upper bounded by,

$$P_d \leq e^{-1} \approx 0.368.$$  

(37)

The comparison between the highest probability of inconclusive success (i.e. uncertainty in guessing the position of correct bits) and the highest probability of conclusive success (i.e. ability to accurately guess the position of correct bits) is depicted in figure 2. The figure demonstrates that for small values of $\theta$, the highest inconclusive success probability surpasses the highest conclusive success probability.

3.3.3. Security of Alice against dishonest Bob.

In this subsection, we determine the number of indices (I_B*) that dishonest Bob can correctly guess from $\mathcal{I}_d$ (the query index set of Alice). In [16], it was discussed that the dishonest Bob can employ a middle-state attack to gain insight into Alice’s conclusiveness or her known bit values. However, it is impossible (shown in [16]) for dishonest Bob to possess knowledge of both the correct bit values and the conclusiveness information. Engaging in systematic cheating would result in incorrect answers provided to Alice, damaging Bob’s reputation as a database provider. Thus, in the QPQ primitive, Bob is
expected to adhere to the actual protocol, as there exists a non-zero probability of being caught cheating.

Lemma 2. Dishonest Bob can correctly guess a maximum of $\frac{l}{N}$ fraction of the indices from the query set $I_l$ of Alice after $l$ queries to the $N$-bit database, i.e., for a particular index $i$ guessed by Bob,

$$\Pr(i \in I_l) \leq \frac{l}{N}$$

**Proof.** In the key generation phase of our proposal, Alice does not broadcast anything about her measurement outcome. So, dishonest Bob has no information about Alice’s measurement outcomes and her known key bits. Now, Alice queries $l$ many times to the database and retrieves $l$ many data bits. After these $l$ many queries, dishonest Bob will try to guess those query indices of Alice. As Bob has no information about Alice’s known final key bits, he has no other options other than randomly guessing these $l$ many indices (out of the $N$ data bits).

So, for any $i$-th data bit, dishonest Bob can guess whether $i \in I_l$ with probability at most $\frac{l}{N}$. This completes the proof. \(\square\)

This means that when Bob makes a guess about any $i$-th data bit, the chance of it being in Alice’s query index set is roughly $\frac{l}{N}$. Assuming that after $l$ queries, the set $I_l$ of Alice’s query indices has $l$ data bits, and the chosen indices are independent, the expected number of indices ($l_B$) that dishonest Bob correctly guesses from $I_l$ would be,

$$E[l_B] = \Pr(i \in I_l) \cdot l$$

$$\leq \frac{l^2}{N}. \quad (38)$$

However, this guess will be inconclusive i.e. Bob cannot identify his correctly guessed indices with certainty because of the random guess. Now, comparing the expression in definition 4 with equation (38) provides the following upper bound for $\delta$ in our proposal.

**Corollary 5.** Our DI-QPQ proposal either terminates with high likelihood in the long run, or dishonest Bob can guess, on average, $\delta$ fraction of indices in Alice’s query index set $I_l$ where,

$$\delta \leq \left(\frac{l}{N}\right). \quad (39)$$

The typical size of the database is much larger (exponentially so) than the size of Alice’s query index set, i.e., $N = l^n$, where $n$ is a positive integer ($n > 1$). Plugging this into equation (39) gives the following upper bound on the value of $\delta$.

$$\delta \leq \frac{1}{l^{(n-1)}}. \quad (40)$$

This relation shows that for our DI-QPQ proposal, $\delta$ is significantly smaller compared to $l$.

Now, we validate the probabilistic definition of user privacy against dishonest Bob for our full DI proposal and derive the exact bound on the security parameter $P_{gu}$. As shown in lemma 1, dishonest Bob’s chance of guessing if an index $i$ is in the index set $I_l$ (of Alice) is limited to $\frac{l}{N}$. Also, this upper bound is determined incorporating the scenario that the proposal does not terminate (i.e., $T = 0$). This implies that,
\[
\Pr[i \in I \land T = 0] 
\leq \frac{l}{N}.
\]  
(41)

So, from definition 4, the parameter \(P_u\) in our proposed scheme (that corresponds to dishonest Bob and honest Alice) can be upper bounded by,

\[
P_u \leq \frac{l}{N}.
\]  
(42)

In practice, the probability of dishonest Bob correctly guessing a database index in Alice’s query index set is low due to a large difference in size between the database (\(N\)) and query index set (\(l\)).

Here also, we evaluate the performance of our proposal considering it as 1-out-of-2 probabilistic oblivious transfer (i.e. \(N = 2, k = 1\) and \(l = 1\)). From expression (42), we get that the value of \(P_u\) for our scheme is upper bounded by,

\[
P_u \leq \frac{1}{2} \approx 0.5.
\]  
(43)

### 4. Discussion and conclusion

The initial QPQ schemes assumed trust in the devices involved, leading to security issues depending on device functionality. Maitra et al [26] first introduced DI in the QPQ domain by proposing a semi-DI version of the QPQ scheme [22] to address these assumptions. In this present draft, we move one step further and propose a novel fully DI-certified QPQ scheme using maximally entangled states for improved robustness. Our scheme achieves the optimal number of raw key bits for client Alice. We analyze security in a general way against all attacks preserving correctness. We provide upper bounds on the cheating probabilities for both the dishonest client and server. This new QPQ scheme with the incorporation of QKD has the potential to become a crucial near-term application of the quantum internet.

**Data availability statement**

All data that support the findings of this study are included within the article (and any supplementary files).

**Acknowledgments**

The authors would like to thank the anonymous reviewers for their thoughtful and valuable comments that helped in improving the technical as well as the editorial quality of this paper.

**Conflict of interest**

The authors declare no conflict of interest.
Appendix

Here we mention the procedure of choosing the initial sample size such that the two parties can certify the devices with desired accuracy and confidence. We also mention here the proofs of theorem 2–4 which confirms the functionality of the measurement devices involved in our protocol. We further show how ‘up to unitary’ devices preserve the correctness condition of our proposed scheme. In the first subsection, we show how one can choose the initial sample size for the protocol in practice. In the next subsection, we restate theorem 2 and mention the detail proof of the theorem. We further restate the functionality of the POVM devices in the form of a two party game (namely POVMgame) and mention the detail proofs of theorems 3 and 4 in the next subsection. Finally, in the last subsection, we show the correctness of our scheme whenever the devices are ‘up to unitary’ as compared to the original devices.

Appendix A. Choice of initial sample size in practice

In this section, we discuss how Bob and Alice choose the initial sample size required for the proposed DI-QPQ scheme. In practice, Alice and Bob have to allow some deviation (from the actual value of the parameter because of finite number of samples) in each testing phase to certify the devices.

It is well-known that the approximate number of samples required to distinguish two events having probabilities $p$ and $p(1 + \epsilon)$ (for small $\epsilon$) is $O\left(\frac{1}{p^2}\right)$. One may require approximately $\frac{64}{p^2}$ samples to achieve a confidence of more than 99% in distinguishing these two events. A more involved expression of the sample size is recently derived in [27] using Chernoff–Hoeffding [33] bound which is stated in proposition 1.

For the testing phases mentioned in our proposed scheme, we consider $X_i = 1$ whenever Bob and Alice win the $i$-th instance and $X_i = 0$ otherwise. Now if we consider $E[X_i] = p$ and want to estimate the success probability $p$ within an error margin of $\epsilon p$ and confidence $1 - \eta$, then from the result mentioned in [27], we can write that the required sample size $m_{\text{req}}$ will be,

$$m_{\text{req}} \geq \frac{1}{2\epsilon^2p^2} \ln \frac{1}{\eta} \quad (44)$$

From this expression of $m_{\text{req}}$, Bob and Alice can estimate the expected number of samples required for a particular testing phase to certify a device with certain accuracy and confidence.

Now to ensure that Bob and Alice get the expected number of samples in each phase (to conclude with certain accuracy and confidence), they choose the total initial sample size (i.e. the value of $K$) as follows-:

- Before the start of the protocol, Alice and Bob (based on the protocol description) calculate the minimum number of samples required (according to the expression in inequality (44)) in each testing phase to conclude with chosen accuracy and confidence.
- Then they choose the value of $k$ to calculate the total number of samples required in private query phase.
- At last, they sum up all these number of samples required in each testing phase along with the number of samples required in PQ phase to calculate the total initial sample size.
- After getting the initial sample size, Bob and Alice proceed to each of the testing phases (according to the description of the protocol), select the required number of samples randomly from the shared instances and check whether the value of a predefined parameter lies.
within the interval \([V - \epsilon p, V + \epsilon p]\) where \(V\) is the actual value of the parameter obtained for asymptotically large number of samples. If this is the case, then with accuracy \(\epsilon p\) and chosen confidence \((1 - \eta)\), they conclude that the devices behave accordingly.

As an example, here we demonstrate the method of choosing samples for the first phase namely source device verification phase. Before the start of the protocol, Bob and Alice choose the accuracy and confidence parameter for this phase with which they want to certify the source device and let \(n_1\) be the required number of samples. Now, similar to this source device certification phase, they calculate the required number of samples for the other phases also and from that calculate the required number of total initial samples \(K\).

Bob and Alice then calculate the value of \(\gamma_1\) such that,

\[n_1 = \gamma_1 K.\]

After getting the value of \(\gamma_1\), Bob first chooses \(\frac{\gamma_1 K}{2}\) number of samples randomly from the \(K\) shared states and then from the rest \((K - \frac{\gamma_1 K}{2})\) number of samples, Alice randomly chooses \(\frac{\gamma_1 K}{2}\) number of samples. They then discuss their chosen instances publicly, get the qubits from the other party and perform LocalCHSHtest for their chosen \(\gamma_1 K\) samples. In this similar way, they choose the samples for the remaining testing phases.

Note that this is a particular way of choosing samples that we demonstrate here from the several other possibilities. It is needless to say that one may follow any other strategies for choosing samples in different testing phases.

Appendix B. Statement and proof of theorem 2

**Theorem 2.** In OBStest, either Bob’s measurement devices achieve the value of the parameter \(\beta = \frac{1}{\sqrt{2}(|\cos \theta - \sin \theta|)}\) (i.e. his devices correctly measure in \([[0], |1]\}) and \([[0'], |1']\}) basis where \([0'] = (\cos \theta|0\rangle + \sin \theta|1\rangle), |1') = (\sin \theta|0\rangle - \cos \theta|1\rangle))\), or the protocol terminates with a high likelihood of failure (as the limit approaches infinity).

**Proof.** Suppose, Alice’s measurement operators are \(|A^s_c\rangle, s \in \{0, 1\}\), corresponding to the input \(s\) and output \(c\). Similarly, Bob’s measurement operators are \(|B^r_b\rangle, r \in \{0, 1\}\), corresponding to the input \(r\) and output \(b\). This implies that Alice’s observable, corresponding to the input \(s \in \{0, 1\}\) is,

\[A_s = \sum_{c \in \{0, 1\}} (-1)^c A^s_c.\]  \((45)\)

Similarly, Bob’s observable corresponding to the input \(r \in \{0, 1\}\) is,

\[B_r = \sum_{b \in \{0, 1\}} (-1)^b B^r_b.\]  \((46)\)

Note that, in the OBStest, the fraction \(\beta\) is being computed as follows,

\[\beta = \frac{1}{4} \sum_{s, r, c, b \in \{0, 1\}} (-1)^{d_{scb}} \alpha^{1/2} \langle \phi_{AB}|A^s_c \otimes B^r_b|\phi_{AB}\rangle \]

\[= \frac{1}{4} \langle \phi_{AB}|W|\phi_{AB}\rangle,\]  \((47)\)

\[= \frac{1}{4} \langle \phi_{AB}|W|\phi_{AB}\rangle,\]  \((48)\)
where \( W_\alpha := \left( \sum_{r,c,b \in \{0,1\}} (-1)^{d_{abc}} \alpha A_r^0 \otimes B_b^c \right) \) which is the operator corresponding to the OBTest. We can also rewrite the expression of \( W_\alpha \) in the following way,

\[
W_\alpha = \left( \sum_{r,c,b \in \{0,1\}} (-1)^{d_{abc}} \alpha A_r^0 \otimes B_b^c \right) + \left( \sum_{r,c,b \in \{0,1\}} (-1)^{d_{abc}} A_r^1 \otimes B_b^c \right)
\]

\[
W_\alpha = W_\alpha^0 + W_\alpha^1, \tag{49}
\]

where \( W_\alpha^0 := \left( \sum_{r,c,b \in \{0,1\}} (-1)^{d_{abc}} \alpha A_r^0 \otimes B_b^c \right) \) and \( W_\alpha^1 := \left( \sum_{r,c,b \in \{0,1\}} (-1)^{d_{abc}} A_r^1 \otimes B_b^c \right) \).

Note that, we can simplify further the expression of \( W_\alpha^0 \) in following way,

\[
W_\alpha^0 = \sum_{r,c,b \in \{0,1\}} (-1)^{d_{abc}} \alpha A_r^0 \otimes B_b^c
\]

\[
= \sum_{c \oplus b = 0} \alpha A_r^0 \otimes B_b^c - \sum_{c \oplus b \neq 0} \alpha A_r^0 \otimes B_b^c
\]

\[
= \alpha \left( A_0^0 \otimes B_0^0 + A_0^0 \otimes B_1^1 + A_1^0 \otimes B_0^1 + A_1^0 \otimes B_1^0 + A_0^1 \otimes B_0^1 + A_1^0 \otimes B_1^0 \right)
\]

\[
- \alpha \left( A_0^0 \otimes B_0^0 + A_0^0 \otimes B_1^1 + A_1^0 \otimes B_0^1 + A_1^0 \otimes B_1^0 + A_0^1 \otimes B_0^1 + A_1^0 \otimes B_1^0 \right)
\]

\[
= \alpha \left[ A_0^0 \otimes (B_0^0 - B_0^1) + A_1^0 \otimes (B_0^0 - B_1^0) + A_0^1 \otimes (B_0^0 - B_1^0) + A_1^0 \otimes (B_0^0 - B_1^0) \right]
\]

By substituting the values of \( (A_0^0 - A_1^0), (B_0^0 - B_0^1) \) and \( (B_0^0 - B_1^0) \) from equations (45) and (46) on the right-hand side of the above expression we get,

\[
W_\alpha^0 = \alpha A_0 \otimes (B_0 + B_1). \tag{50}
\]

Using similar approach we get the following simplified version of the expression \( W_\alpha^0 \).

\[
W_\alpha^1 = A_1 \otimes (B_0 - B_1). \tag{51}
\]

By substituting the values of \( W_\alpha^0 \) and \( W_\alpha^1 \) from equations (50) and (51) to equation (49) we get,

\[
W_\alpha = \alpha A_0 \otimes (B_0 + B_1) + A_1 \otimes (B_0 - B_1). \tag{52}
\]

Note that, the right-hand side of this OBTest operator \( W_\alpha \) is exactly same as the tilted CHSH operator, described in [30].

So, the expression of \( W_\alpha^2 \) can be written as

\[
W_\alpha^2 = \alpha^2 A_0^2 \otimes (B_0^2 + B_1^2 + \{B_0, B_1\})
\]

\[
+ A_1^2 \otimes (B_0^2 + B_1^2 + \{B_0, B_1\})
\]

\[
= \alpha^2 A_0^2 + A_1^2 + \alpha \{A_0, A_1\} \otimes B_0^2 + A_1^2 + \alpha \{A_0, A_1\} \otimes B_1^2
\]

\[
+ \alpha^2 A_0^2 + A_1^2 - \alpha \{A_0, A_1\} \otimes B_0^2 + A_1^2 - \alpha \{A_0, A_1\} \otimes B_1^2
\]

\[
+ \alpha^2 A_0^2 - A_1^2 \otimes \{B_0, B_1\} + \alpha \{A_0, A_1\} \otimes \{B_0, B_1\}.
\]
Using the property $A_j^2 \leq I$, we can rewrite this expression as,

$$W_\alpha^2 \leq \left[ (\alpha^2 + 1) I + \alpha \{A_0, A_1\} \right] \otimes B_0^2$$
$$+ \left[ (\alpha^2 + 1) I - \alpha \{A_0, A_1\} \right] \otimes B_1^2$$
$$+ I \otimes (\alpha^2 - 1) \{B_0, B_1\} - \alpha [A_0, A_1] \otimes [B_0, B_1].$$

Since $-2I \leq \{A_0, A_1\} \leq 2I$, we have,

$$\left[ (\alpha^2 + 1) I + \alpha \{A_0, A_1\} \right] \geq 0.$$

We can use the property $B_k^2 \leq I$ and get the following simplified expression

$$W_\alpha^2 \leq 2 (\alpha^2 + 1) I \otimes I + I \otimes (\alpha^2 - 1) \{B_0, B_1\}$$
$$- \alpha [A_0, A_1] \otimes [B_0, B_1].$$

We can further upper bound the commutators by their matrix modulus and use the relation $|[A_0, A_1]| \leq 2I$ to get the following expression

$$W_\alpha^2 \leq 2 (\alpha^2 + 1) I \otimes I + T_\alpha \otimes I$$

where $T_\alpha := (\alpha^2 - 1) \{B_0, B_1\} + 2\alpha [B_0, B_1].$

Now the expression of $T_\alpha$ can also be upper bounded by upper bounding the anti commutators by its matrix modulus. So, the value of $T_\alpha$ will be upper bounded by,

$$T_\alpha \leq (\alpha^2 - 1) |\{B_0, B_1\}| + 2\alpha |[B_0, B_1]|.$$

Again one can easily check that,

$$|\{B_0, B_1\}|^2 + |[B_0, B_1]|^2$$
$$= |B_0B_1 + B_1B_0|^2 + |B_0B_1 - B_1B_0|^2$$
$$= (B_0B_1 + B_1B_0)^\dagger (B_0B_1 + B_1B_0)$$
$$+ (B_0B_1 + B_1B_0)^\dagger (B_0B_1 + B_1B_0)$$
$$= 2 (B_0B_1)^\dagger (B_0B_1) + 2 (B_1B_0)^\dagger (B_1B_0).$$

Let us consider that the measurement operators are projective i.e. $(A_j^\dagger)^2 = A_j^\dagger$ and $(B_k^\dagger)^2 = B_k^\dagger$. Now for the projectors $B_0^0$ and $B_1^0$, $(B_0^0 + B_1^0) = I$. From this relation we can write,

$$(B_0^0 + B_1^0) (B_0^0 + B_1^0)^\dagger = I$$
$$B_0^0 B_0^0 + B_1^0 B_1^0 + B_0^0 B_1^0 + B_1^0 B_0^0 = I$$

This implies,

$$\left( B_0^0 B_1^0 + B_1^0 B_0^0 \right) = 0.$$
Now $B_0 = (B_0^0 - B_0^1)$. From this we can get,

$$
B_0B_0^\dagger = (B_0^0 - B_0^1)(B_0^0 - B_0^1)^\dagger
= B_0^0B_0^\dagger - B_0^0B_0^1\dagger - B_0^1B_0^\dagger + B_0^1B_0^1\dagger
= (B_0^0 + B_0^1) - (B_0^0B_1^\dagger + B_0^1B_0^\dagger)
= \mathbb{I} + 0 = \mathbb{I}.
$$

Similarly, it can be shown that, $B_1B_1^\dagger = B_1^1B_1 = \mathbb{I}$.

So, from equation (54), we can write that for unitary observables $B_0$ and $B_1$,

$$
|\{B_0, B_1\}|^2 + |[B_0, B_1]|^2 = 2(B_0B_1)^\dagger(B_0B_1)
+ 2(B_1B_0)^\dagger(B_1B_0)
= 2\mathbb{I} + 2\mathbb{I} = 4\mathbb{I}.
$$

This implies,

$$
|\{B_0, B_1\}| = \sqrt{4\mathbb{I} - |[B_0, B_1]|^2}.
$$

So, the simplified expression of $T_\alpha$ will be of the form

$$
T_\alpha = (\alpha^2 - 1)\sqrt{4\mathbb{I} - |[B_0, B_1]|^2 + 2\alpha|[B_0, B_1]|}.
$$

This is the maximum value of $T_\alpha$ and here $T_\alpha$ attains this maximum value because of projective observables. Now one can easily check that the value of $|[B_0, B_1]|$ which maximizes the value of $T_\alpha$ is $|[B_0, B_1]| = \frac{4\alpha}{(\alpha^2 + 1)}\mathbb{I}$ and the corresponding value of $T_\alpha$ is $2(\alpha^2 + 1)\mathbb{I}$. This implies that,

$$
T_\alpha = 2(\alpha^2 + 1)\mathbb{I}.
$$

From this value of $T_\alpha$ and from the expression of $W_\alpha$, mentioned in equation (53), we can easily write that the value of $W_\alpha$ is upper bounded by the following quantity.

$$
W_\alpha \leq \sqrt{2(\alpha^2 + 1)\mathbb{I} \otimes \mathbb{I} + T_\alpha \otimes \mathbb{I}} \tag{55}
$$

where $T_\alpha = 2(\alpha^2 + 1)\mathbb{I}$.

Now, the value $\beta$ obtained in OBS test of our algorithm can be written alternatively as $\beta = \frac{\text{tr}(W_\alpha\rho_{AB})}{\text{tr}(\rho_{AB})}$ where $\rho_{AB}$ is the density matrix representation of the shared states $|\phi\rangle_{AB}$ i.e. $\rho_{AB} = |\phi\rangle_{AB}\langle\phi|$. From this expression of $\beta$, one can easily derive that the value of $\beta^2$ is upper bounded by the following quantity,

$$
\beta^2 \leq \frac{\text{tr}(W_\alpha\rho_{AB})}{16}.
$$

Now if we assume $t_\alpha := \frac{1}{2}\text{tr}(T_\alpha\rho_{BG}) - \frac{1}{2}(\alpha^2 - 1)$ (where $\rho_{BG}$ is the reduced state at Bob’s side) then using this value of $t_\alpha$ along with the value of $W_\alpha$ obtained from expression (55) and
the upper bound on the value of $\beta^2$, we can write that the $\beta$ value mentioned in OBStest is upper bounded by the following quantity,

$$\beta \leq \frac{\sqrt{\alpha^2 + T_\alpha}}{2},$$  \hfill (56)

Now here, the observables are projective (i.e. $B_j^2 = \mathbb{I}$) and the anti commutator $\{B_0, B_1\}$ is a positive semi definite operator. Since we have already shown that the value of the anti-hermitian operator $[B_0, B_1]$ is $[B_0, B_1] = \frac{4\alpha_i}{(\alpha^2 + 1)} \mathbb{I}$ for the maximum value of $T_\alpha$, the spectral decomposition of $[B_0, B_1]$ can be written as,

$$[B_0, B_1] = \frac{4\alpha_i}{(\alpha^2 + 1)} (P_+ - P_-)$$

for some orthogonal projectors $P_+$ and $P_-$ such that $(P_+ + P_-) = \mathbb{I}$. As it is well-known that for projective observables, the commutator holds the property $B_0[B_0, B_1]B_0 = -[B_0, B_1]$, we can easily conclude that $B_0 P_+ B_0 = P_+$. Let us consider that $\{ |e_j^d\rangle \}_j$ is an orthonormal basis for the support of $P_+$ and $\{|e_j^d\rangle \}_j$ is an orthonormal basis for the support of $P_-$ where $|e_j^d\rangle = B_0 |e_j^d\rangle$. We define the unitary operator $U_0$ as

$$U_0 |e_j^d\rangle = \frac{1}{\sqrt{2}} \left( |0\rangle + (-1)^d |1\rangle \right) |j\rangle,$$

for $d \in \{0, 1\}$. Then we can easily verify that

$$U_0 [B_0, B_1] U_0^d = \frac{4\alpha_i}{(\alpha^2 + 1)} \sigma_Y \otimes \mathbb{I}.$$

Since $\{I, \sigma_X, \sigma_Y, \sigma_Z\}$ constitute an operator basis for linear operators acting on $\mathbb{C}^2$, without loss of generality we can write

$$U_0 B_0 U_0^d = I \otimes K_0 + \sigma_X \otimes K_x + \sigma_Y \otimes K_y + \sigma_Z \otimes K_z,$$

for some hermitian operator $K_0, K_x, K_y, K_z$. For projective observable $B_0$, one can easily check that $\{B_0, [B_0, B_1]\} = 0$. This relation satisfies only when $K_0 = K_y = 0$. As $B_0^d = I$, $K_x$ and $K_z$ must satisfy the relation

$$K_x^2 + K_z^2 = \mathbb{I} \text{ and } [K_x, K_z] = 0.$$

So, we can easily write $K_x$ and $K_z$ in the following form.

$$K_x = \sum_j \sin 2\gamma_j |j\rangle \langle j|$$

$$K_z = \sum_j \cos 2\gamma_j |j\rangle \langle j|,$$

for some angle $\gamma_j$ and some orthonormal basis $\{|j\rangle \}$. This implies that

$$U_0 B_0 U_0^d = \sigma_X \otimes K_x + \sigma_Z \otimes K_z$$

$$= \sum_j (\sin 2\gamma_j \sigma_X + \cos 2\gamma_j \sigma_Z) \otimes |j\rangle \langle j|.$$
We now consider the following controlled unitary to align the qubit observables.

\[ U_1 = \sum_j \exp(i\gamma_j \sigma_y) \otimes |j\rangle \langle j| \]

Now for this defined unitary operator, one can easily check that

\[ U_1 U_0 B_0 U_0^\dagger U_1^\dagger = \sigma_z \otimes I \]

\[ U_1 U_0 [B_0, B_1] U_0^\dagger U_1^\dagger = \frac{4\alpha i}{\alpha^2 + 1} \sigma_y \otimes I. \]

Like observable \( B_0 \), an analogous reasoning can also be applied for observable \( B_1 \) and from that, without loss of generality we can write

\[ U_1 U_0 B_1 U_0^\dagger U_1^\dagger = \sigma_z \otimes K_z' + \sigma_z \otimes K_x'. \]

Since the commutators are positive semi definite and the observables are projective, we can easily check that

\[ \{B_0, B_1\} = |\{B_0, B_1\}| = \sqrt{4I - |[B_0, B_1]|^2} = \frac{2(\alpha^2 - 1)}{(\alpha^2 + 1)} I. \]

Now we define \( 2\theta := \arcsin \left(\frac{\alpha^2 - 1}{\alpha^2 + 1}\right) \in [0, \frac{\pi}{2}] \). From this relation, imposing consistency on the anti commutator, we get,

\[ K_z' = \sin 2\theta. I. \]

On the other hand, imposing consistency on the commutator, we get,

\[ K_x' = \cos 2\theta. I. \]

Now, from the relation \( 2\theta := \arcsin \left(\frac{\alpha^2 - 1}{\alpha^2 + 1}\right) \), we can get the value of \( \alpha \) which is

\[ \alpha = \frac{(\cos \theta + \sin \theta)}{|(\cos \theta - \sin \theta)|}. \]

For this value of \( \alpha \), we can easily derive that \( t_\alpha = 1 \). This implies that the simplified expression for \( \beta \) is,

\[ \beta = \frac{\sqrt{1 + \alpha^2}}{2} \quad (57) \]

where \( \alpha = \frac{(\cos \theta + \sin \theta)}{|(\cos \theta - \sin \theta)|} \). Now from this value of \( \alpha \), we can derive the value of \( \sqrt{1 + \alpha^2} \) which is,

\[ \sqrt{1 + \alpha^2} = \frac{\sqrt{2}}{|(\cos \theta - \sin \theta)|}, \quad (58) \]
So, the value of \( \beta \) corresponding to these observables \( B_0 \) and \( B_1 \) will be,

\[
\beta = \frac{1}{\sqrt{2(\cos \theta - \sin \theta)}}.
\]

If we consider \( U_B = U_B^0 U_B^1 \) then the observables \( B_0 \) and \( B_1 \) will be of the form

\[
B_0 = U_B (\sigma_Z \otimes \mathbb{I}) U_B^\dagger
\]
\[
B_1 = U_B (\cos 2\theta \sigma_X + \sin 2\theta \sigma_Z \otimes \mathbb{I}) U_B^\dagger.
\]

This implies that in the OBTest, if \( \beta \) is equal to \( \frac{1}{\sqrt{2(\cos \theta - \sin \theta)}} \), then the corresponding observables of Bob are same as the one described in the OBTest. This concludes the proof. \( \square \)

### Appendix C. Verification of Alice’s POVM elements

In the QPQ protocol, Alice needs to make sure her measurement device works properly, i.e., she should be able to distinguish between \( |0\rangle \langle 1| \) and \( |0\rangle \langle 0| + |1\rangle \langle 1| \) with certainty for (on average) around \( (1 - \cos \theta) \) fraction of instances, where, \( |0\rangle' = \cos \theta|0\rangle + \sin \theta|1\rangle \) \( (|1\rangle' = \sin \theta|0\rangle - \cos \theta|1\rangle) \). Let \( M^0 = \{M_0^0, M_0^1, M_0^2\} \) \( (M^1 = \{M_1^0, M_1^1, M_1^2\}) \) the set of Alice’s POVMs, which distinguishes the states \( \{|0\rangle, |0\rangle' \} \) \( (\{|1\rangle, |1\rangle' \}) \). Here we show that if the input states are of the form \( |0\rangle \langle 0| \) \( \text{or} \) \( |1\rangle \langle 1| \) \( \text{and} \) Alice manages to distinguish the states with certainty for (on average) around \( (1 - \cos \theta) \) fraction of instances then \( M^0_i = D^0_i \) \( (M^1_i = D^1_i) \) for \( i \in \{0, 1, 2\} \). In order to prove this, here we first represent the interactions between Bob and Alice in the proposed DI-QPQ protocol in the form of a game, called POVMgame\( (M^i, y) \) for better understanding, where the agent \( A_1 \) represents Bob and the agent \( A_2 \) represents Alice. The game is as follows,

**Algorithm 5.** POVMgame\( (M^i, y) \).

- \( A_1 \) declares \( y \) whenever the state at his side (and also at \( A_2 \)’s side) is either \( \rho_0^i \) or \( \rho_1^i \), for the randomly chosen \( x \) values (i.e., for \( x \in \{0, 1\} \)), where \( \rho_0^0 = |0\rangle \langle 0| \), \( \rho_1^1 = |0\rangle \langle 0| \) \( \text{or} \) \( |1\rangle \langle 1| \) \( \text{and} \) \( \rho_1^0 = |1\rangle \langle 1| \) \( \text{or} \) \( |0\rangle \langle 0| \).
- \( A_2 \) measures her state (which is either \( \rho_0^i \) or \( \rho_1^i \)) using the POVM \( M^i \) \( (\text{where} \ M^i = \{M_0^i, M_1^i, M_2^i\}) \) and sends the outcome \( b \in \{0, 1, 2\} \) to \( A_1 \).
- \( A_2 \) wins if and only if, \( \Omega^0 = \sum_{b, i \in \{0, 1, 2\}} (-1)^{b+i} \text{Tr}[M_b^i \rho_0^i] = \frac{2 \sin^2 \theta}{1 + \cos \theta} \).

**Theorem 6.** In POVMgame\( (M^i, y) \), if \( A_1 \) chooses \( y = 0 \) and the states at \( A_2 \)'s end are \( \rho_0^0 = |0\rangle \langle 0| \) \( \text{and} \) \( \rho_1^0 = |0\rangle \langle 0| \) \( \text{and} \) if \( A_2 \) manages to win the game, i.e., \( \Omega^0 = \frac{2 \sin^2 \theta}{1 + \cos \theta} \), then this implies, \( A_2 \)’s measurement devices are of the following form (up to a local unitary),

\[
M_0^0 = \frac{1}{(1 + \cos \theta)} |1\rangle \langle 1|  \tag{60}
\]
\[
M_0^1 = \frac{1}{(1 + \cos \theta)} |0\rangle \langle 1|  \tag{61}
\]
\[
M_2^0 = I - M_0^0 - M_0^1,  \tag{62}
\]

where, \( |1\rangle = \sin \theta|0\rangle - \cos \theta|1\rangle \).
**Proof.** In the POVM game \((M^0, y)\), \(A_2\) applies \(M^0\) on a single qubit state \(\rho^0_x\) (where \(x \in \{0, 1\}\)). So, without any loss of generality we can assume that \(M^0_i \in M^0\) has the following form,

\[
M^0_i = \lambda^0_i (\mathbb{I} + \vec{m}^0_i \cdot \vec{\sigma}),
\]

where \(\vec{m}^0_i = [m^0_{i0}, m^0_{i1}, m^0_{i2}]\) and it is the Bloch vector with length at most one, \(\vec{\sigma} = [\sigma_X, \sigma_Y, \sigma_Z]\) are the Pauli matrices and \(\lambda_i \geq 0\). In this case, one may wonder how we can fix the dimension of \(M^0_i\) in the proof in DI scenario? The answer to this question is that here we are able to fix the dimension of \(M^0_i\) and choose this particular general form because of the tests mentioned earlier in the source device verification phase (corresponding result mentioned in corollary 2) and DI testing phase for Bob’s measurement device (corresponding result mentioned in theorem 2) which certifies that the states shared between Alice and Bob are EPR pairs (up to a unitary) and after Bob’s projective measurements, the reduced states at Alice’s side are one qubit states. Now, the condition \(\sum_{i=0}^2 M^0_i = \mathbb{I}\) leads us to the following relations,

\[
\sum_{i=0}^2 \lambda^0_i = 1 \quad (64)
\]

\[
\sum_{i=0}^2 \lambda^0_i \vec{m}^0_i = 0. \quad (65)
\]

In terms of Bloch vector we can rewrite \(\rho^0_0, \rho^0_1\) in following way,

\[
\rho^0_0 = \frac{1}{2} (\mathbb{I} + \sigma_Z)
\]

\[
\rho^0_1 = \frac{1}{2} (\mathbb{I} + \sin 2\theta \sigma_X + \cos 2\theta \sigma_Z). \quad (67)
\]

In the POVM game \((M^0, y)\) if \(A_2\) would like to maximizes her winning probability then she needs to maximize the following expression,

\[
\Omega^0 = \sum_{b, x} (-1)^b \text{tr} [M^0_b \rho^0_x]. \quad (68)
\]

In terms of \(\lambda^0_i, \vec{m}^0_i, \vec{\sigma}\) we have,

\[
\text{tr} [M^0_b \rho^0_0] = \lambda^0_b (1 + m^0_{b2})
\]

\[
\text{tr} [M^0_b \rho^0_1] = \lambda^0_b (1 + m^0_{b0} \sin 2\theta + m^0_{b2} \cos 2\theta)
\]

\[
\text{tr} [M^1_b \rho^0_0] = \lambda^1_b (1 + m^1_{b2})
\]

\[
\text{tr} [M^1_b \rho^0_1] = \lambda^1_b (1 + m^1_{b0} \sin 2\theta + m^1_{b2} \cos 2\theta).
\]

In terms of \(\lambda^0_i, \vec{m}^0_i, \vec{\sigma}\) we can rewrite \(\Omega^0\) as,

\[
\Omega^0 = \lambda^0_0 (1 + m^0_{02}) + \lambda^1_0 (1 + m^1_{00} \sin 2\theta + m^1_{02} \cos 2\theta)
\]

\[- \lambda^0_0 (1 + m^0_{00} \sin 2\theta + m^0_{02} \cos 2\theta) - \lambda^0_1 (1 + m^1_{02}). \quad (69)
\]
As both $\text{tr}[M^0_0 \rho^0_0]$ and $\text{tr}[M^0_1 \rho^0_0]$ are positive quantity, hence

$$
\Omega^0 \leq \lambda^0_0 (1 + m^0_{02}) + \lambda^0_1 (1 + m^0_{10}\sin 2\theta + m^0_{12}\cos 2\theta),
$$

(70)

and this implies,

$$
(1 + m^0_{00}\sin 2\theta + m^0_{02}\cos 2\theta) = 0
$$

(71)

$$
(1 + m^0_{12}) = 0.
$$

(72)

According to the equation (72) we have $m^0_{12} = -1$. As both of $\rho^0_0, \rho^0_1$ lie on the XZ plane and due to the freedom of local unitary without loss of generality we can assume $m^0_{01} = m^0_{11} = m^0_{21} = 0$. Due to the positivity constraint ($M^0_i \geq 0$) we have,

$$
m^0_{00}^2 + m^0_{02}^2 \leq 1
$$

(73)

$$
m^0_{10}^2 + m^0_{12}^2 \leq 1
$$

(74)

$$
m^0_{20}^2 + m^0_{22}^2 \leq 1.
$$

(75)

By combining the constraint equation (72) with the equation (74) we get, $m^0_{10} = 0$. Hence,

$$
\vec{m}^0_1 = [0, 0, -1],
$$

(76)

and by substituting the values of $m^0_{10}, m^0_{12}$ in equation (70) we get the following expression of $\Omega^0$.

$$
\Omega^0 \leq \lambda^0_0 (1 + m^0_{02}) + \lambda^0_1 (1 - \cos 2\theta).
$$

(77)

Note that the expression of $\Omega^0$ maximizes when $\lambda^0_0, m^0_{02}, \lambda^0_1$ maximizes and from the constraint equation (73) we get that $m^0_{00}^2 + m^0_{02}^2 \leq 1$. Hence, without any loss of generality we can assume that for the maximum value of $\Omega^0$, $m^0_{00}^2 + m^0_{02}^2 = 1$. So, we can parameterize $m^0_{00}, m^0_{02}$ as $\sin \alpha, \cos \alpha (0 \leq \alpha \leq 2\pi)$. By substituting $m^0_{00} = \sin \alpha, m^0_{02} = \cos \alpha$ in equation (71) we get,

$$
1 + \sin \alpha \sin 2\theta + \cos \alpha \cos 2\theta = 0.
$$

This implies,

$$
\cos (\alpha - 2\theta) = -1.
$$

As $0 \leq \alpha \leq 2\pi$, so $\cos(\alpha - 2\theta) = -1$ this implies,

$$
\alpha - 2\theta = \pi \quad \text{and},
$$

$$
\alpha = \pi + 2\theta.
$$

(78)

From the equation (78) we get,

$$
\vec{m}^0_0 = [-\sin 2\theta, 0, -\cos 2\theta].
$$

(79)

By substituting the expression of $\vec{m}_0$ in equation (77) we get,

$$
\Omega^0 \leq (\lambda^0_0 + \lambda^0_1) (1 - \cos 2\theta).
$$

(80)
By substituting the values of \( \vec{m}_0 \), \( \vec{m}_1 \) in equation (65) we get,
\[
\begin{align*}
\lambda_0^2 m_{22}^0 - \lambda_0^0 \cos 2\theta &= \lambda_1^0 \\
\lambda_0^0 m_{20}^0 &= \lambda_0^0 \sin 2\theta.
\end{align*}
\tag{81}
\]

Due to the constraint equation (75), similar to \( \vec{m}_0 \), here we parameterize the expression of \( m_{20}^0, m_{22}^0 \) as \( \sin \beta, \cos \beta \) respectively. By substituting \( m_{20}^0 = \sin \beta \) and \( m_{22}^0 = \cos \beta \) in the equations (81) and (82) we get,
\[
\begin{align*}
\lambda_0^2 \cos \beta - \lambda_0^0 \cos 2\theta &= \lambda_1^0 \\
\lambda_0^0 \sin \beta &= \lambda_0^0 \sin 2\theta.
\end{align*}
\tag{82}
\]

By solving equations (83) and (84) together with equation (64) we get,
\[
\begin{align*}
\lambda_0^0 &= \frac{\sin \beta}{\sin \beta + \sin 2\theta + \sin (2\theta - \beta)} \\
\lambda_1^0 &= \frac{\sin (2\theta - \beta)}{\sin \beta + \sin 2\theta + \sin (2\theta - \beta)}.
\end{align*}
\tag{85}
\]

Hence,
\[
\begin{align*}
\lambda_0^0 + \lambda_1^0 &= \frac{\sin \beta + \sin (2\theta - \beta)}{\sin \beta + \sin 2\theta + \sin (2\theta - \beta)} \\
&= \frac{\cos (\theta - \beta)}{\cos \theta + \cos (\theta - \beta)}.
\end{align*}
\tag{86}
\]

According to equation (80), for getting a tight upper bound on \( \Omega^0 \) we need to maximize \( (\lambda_0^0 + \lambda_1^0) \). By equating \( \frac{d(\lambda_0^0 + \lambda_1^0)}{d\beta} = 0 \) in equation (88) we get,
\[
\frac{\sin (\theta - \beta) \cos \theta}{\cos \theta + \cos (\theta - \beta)} = 0.
\tag{89}
\]

This implies,
\[
\beta = \theta.
\tag{90}
\]

It is also easy to check that for \( \theta = \beta \), the expression \( \frac{d(\lambda_0^0 + \lambda_1^0)}{d\beta} < 0 \). Hence, the expression \( \lambda_0^0 + \lambda_1^0 \) maximizes at the point \( \beta = \theta \). Substituting this relation in equations (85) and (86) we get,
\[
\lambda_0^0 = \lambda_1^0 = \frac{1}{2(1 + \cos \theta)}.
\tag{91}
\]

By substituting the values of \( \lambda_0^0 + \lambda_1^0 \) in equation (64) we get,
\[
\lambda_2^0 = \frac{\cos \theta}{1 + \cos \theta}.
\tag{92}
\]
Hence, we get,
\[ \Omega^0 \leq \frac{2\sin^2 \theta}{1 + \cos \theta}, \quad (93) \]
and
\[
\begin{align*}
M^0_0 &= \frac{1}{2(1 + \cos \theta)} (\mathbb{I} - \sin 2\theta \sigma_X - \cos 2\theta \sigma_Z) \\
M^0_1 &= \frac{1}{2(1 + \cos \theta)} (\mathbb{I} - \sigma_Z) \\
M^0_2 &= \frac{\cos \theta}{1 + \cos \theta} (\mathbb{I} + \sin \theta \sigma_X + \cos \theta \sigma_Z).
\end{align*}
\]
We can rewrite the above expressions as follows,
\[
\begin{align*}
M^0_0 &= \frac{1}{2} (1 + \cos \theta) \langle 1' \rangle \langle 1' | \\
M^0_1 &= \frac{1}{2} (1 + \cos \theta) \langle 1 \rangle \langle 1 | \\
M^0_2 &= \mathbb{I} - M^0_0 - M^0_1,
\end{align*}
\]
where \( |1' \rangle = \sin \theta |0 \rangle - \cos \theta |1 \rangle \). This concludes the proof.

Similarly for the input states \(|1\rangle, |1'\rangle\), one can conclude the following.

**Theorem 7.** In POVMgame\((M^y, y)\), if \(A_1\) chooses \(y = 1\) and the states at \(A_2\)'s end are \(\rho^0_1 = |1\rangle \langle 1|\) and \(\rho^1_1 = |1'\rangle \langle 1'|\) and if \(A_2\) manages to win the game, i.e. \(\Omega^1 = \frac{\sin \theta}{1 + \cos \theta}\), then this implies, \(A_2\)'s measurement devices are of the following form (up to a local unitary),
\[
\begin{align*}
M^0_0 &= \frac{1}{(1 + \cos \theta)} \langle 0' \rangle \langle 0'| \\
M^0_1 &= \frac{1}{(1 + \cos \theta)} \langle 0 \rangle \langle 0 | \\
M^0_2 &= \mathbb{I} - M^0_0 - M^0_1,
\end{align*}
\]
where \(|0'\rangle = \cos \theta |0 \rangle + \sin \theta |1 \rangle\).

**Proof.** In the POVMgame\((M^y, y)\), \(A_2\) applies \(M^1\) on a single qubit state \(\rho^1_1\) (where \(x \in \{0, 1\}\)). So, without any loss of generality we can assume that \(M^1_1 \in M^1\) has the following form,
\[
M^1_1 = \lambda^1_1 (\mathbb{I} + \tilde{m}^1 \cdot \tilde{\sigma}),
\]
where \(\tilde{m}^1 = [m^1_0, m^1_1, m^1_2]\) and it is the Bloch vector with length at most one, \(\tilde{\sigma} = [\sigma_X, \sigma_Y, \sigma_Z]\) are the Pauli matrices and \(\lambda^1_1 \geq 0\). The condition \(\sum_{i=0}^{2} M^1_i = \mathbb{I}\) leads us to the following relations,
\[ \sum_{i=0}^{2} \lambda_i^1 = 1 \]  
(101)

\[ \sum_{i=0}^{2} \lambda_i^1 \bar{m}_i^1 = 0. \]  
(102)

In terms of Bloch vector we can rewrite \( \rho_{10}^1, \rho_{11}^1 \) in following way,

\[ \rho_{10}^1 = \frac{1}{2} (I - \sigma_Z) \]  
(103)

\[ \rho_{11}^1 = \frac{1}{2} (I - \sin 2\theta \sigma_X - \cos 2\theta \sigma_Z). \]  
(104)

In the POVM game \((M, y)\) if \( A_2 \) would like to maximizes her winning probability then she needs to maximize the following expression,

\[ \Omega^1 = \sum_{b, x \in \{0, 1\}} (-1)^{b+1} \text{tr} [M_b^1 \rho_x]. \]  
(105)

In terms of \( \lambda_i^1, \bar{m}_i^1, \bar{\sigma} \) we have,

\[ \text{tr} [M_b^0 \rho_0^1] = \lambda_0^1 (1 - m_{02}^1) \]
\[ \text{tr} [M_b^0 \rho_1^1] = \lambda_0^1 (1 - m_{00}^1 \sin 2\theta - m_{02}^1 \cos 2\theta) \]
\[ \text{tr} [M_b^1 \rho_0^1] = \lambda_1^1 (1 - m_{12}^1) \]
\[ \text{tr} [M_b^1 \rho_1^1] = \lambda_1^1 (1 - m_{10}^1 \sin 2\theta - m_{12}^1 \cos 2\theta). \]

In terms of \( \lambda_i^1, \bar{m}_i^1, \bar{\sigma} \) we can rewrite \( \Omega^1 \) as,

\[ \Omega^1 = \lambda_0^1 (1 - m_{02}^1) + \lambda_1^1 (1 - m_{10}^1 \sin 2\theta - m_{12}^1 \cos 2\theta) 
- \lambda_0^1 (1 - m_{00}^1 \sin 2\theta - m_{02}^1 \cos 2\theta) - \lambda_1^1 (1 - m_{12}^1). \]  
(106)

As both \( \text{tr}[M_b^0 \rho_1^1] \) and \( \text{tr}[M_b^1 \rho_0^1] \) are positive quantity, hence

\[ \Omega^1 \leq \lambda_0^1 (1 - m_{02}^1) + \lambda_1^1 (1 - m_{10}^1 \sin 2\theta - m_{12}^1 \cos 2\theta), \]  
(107)

and this implies,

\[ (1 - m_{00}^1 \sin 2\theta - m_{02}^1 \cos 2\theta) = 0 \]  
(108)

\[ (1 - m_{12}^1) = 0. \]  
(109)

According to the equation (109) we have \( m_{12}^1 = 1. \) As both of \( \rho_{00}^1, \rho_{11}^1 \) lie on the XZ plane and due to the freedom of local unitary without loss of generality we can assume \( m_{01}^1 = m_{11}^1 = m_{21}^1 = 0. \) Due to the positivity constraint \((M_b^1 \geq 0)\) we have,

\[ m_{00}^1 + m_{02}^1 \leq 1 \]  
(110)

\[ m_{10}^1 + m_{12}^1 \leq 1 \]  
(111)

\[ m_{20}^1 + m_{22}^1 \leq 1. \]  
(112)
By combining the constraint equation (109) with the equation (111) we get, \( m_{10}^1 = 0 \).

Hence,
\[
\vec{m}_1^1 = [0, 0, 1],
\]
(113)

and by substituting the values of \( m_{10}^1, m_{12}^1 \) in equation (107) we get the following expression of \( \Omega^1 \),
\[
\Omega^1 \leq \lambda_0^1 (1 - m_{02}^1) + \lambda_1^1 (1 - \cos 2\theta).
\]
(114)

Note that the expression of \( \Omega^1 \) maximizes when \( \lambda_0^1, \lambda_1^1 \) maximizes and \( m_{02}^1 \) minimizes and from the constraint equation (73) we get that \( m_{00}^2 + m_{02}^2 \preceq 1 \). Hence, without any loss of generality we can assume that for the maximum value of \( \Omega^1 \), \( m_{00}^1, m_{02}^1 \) are \( \sin \alpha, \cos \alpha \) respectively. By substituting \( m_{00}^1 = \sin \alpha, m_{02}^1 = \cos \alpha \) in equation (71) we get,
\[
1 - \sin \alpha \sin 2\theta - \cos \alpha \cos 2\theta = 0.
\]

This implies,
\[
\cos (\alpha - 2\theta) = 1.
\]

As \( 0 \leq \alpha \leq 2\pi \), so \( \cos (\alpha - 2\theta) = 1 \) this implies,
\[
\alpha - 2\theta = 0 \quad \text{or} \quad 2\pi \quad \text{and,}
\]
\[
\alpha = 2\theta \quad \text{or} \quad (2\pi + 2\theta).
\]
(115)

One can easily check that for both these values of \( \alpha \), the value of \( m_{10}^1 \) and \( m_{12}^1 \) are \( \sin 2\theta \) and \( \cos 2\theta \) respectively. From the equation (115) we get,
\[
\vec{m}_0^1 = [\sin 2\theta, 0, \cos 2\theta].
\]
(116)

By substituting the expression of \( \vec{m}_0^1 \) in equation (114) we get,
\[
\Omega^1 \leq \left( \lambda_0^1 + \lambda_1^1 \right) (1 - \cos 2\theta).
\]
(117)

By substituting the values of \( \vec{m}_0^1, \vec{m}_1^1 \) in equation (102) we get,
\[
\lambda_2^1 m_{22}^1 + \lambda_0^1 \cos 2\theta + \lambda_1^1 = 0
\]
(118)
\[
\lambda_2^1 m_{20}^1 + \lambda_0^1 \sin 2\theta = 0.
\]
(119)

Due to the constraint equation (112), similar to \( \vec{m}_0^1 \), here we parameterize the expression of \( m_{20}^1, m_{12}^1 \) as \( \sin \beta, \cos \beta \) respectively. By substituting \( m_{20}^1 = \sin \beta \) and \( m_{12}^1 = \cos \beta \) in the equations (118) and (119) we get,
\[
\lambda_2^1 \cos \beta + \lambda_0^1 \cos 2\theta + \lambda_1^1 = 0
\]
(120)
\[
\lambda_2^1 \sin \beta + \lambda_0^1 \sin 2\theta = 0.
\]
(121)
By solving equations (120) and (121) together with equation (101) we get,

\[ \lambda_0 = \frac{\sin\beta}{\sin\beta + \sin(2\theta - \beta) - \sin2\theta} \]  
\[ \lambda_1 = \frac{\sin(2\theta - \beta)}{\sin\beta + \sin(2\theta - \beta) - \sin2\theta} \]  

Hence,

\[ \lambda_0 + \lambda_1 = \frac{\sin\beta + \sin(2\theta - \beta)}{\sin\beta + \sin(2\theta - \beta) - \sin2\theta} \]  
\[ = \frac{\cos(\theta - \beta)}{\cos(\theta - \beta) - \cos\beta} \]  

According to equation (117), for getting a tight upper bound on \( \Omega^1 \) we need to maximize \( (\lambda_0 + \lambda_1) \). By equating \( \frac{d(\lambda_0 + \lambda_1)}{d\beta} = 0 \) in equation (125) we get,

\[ -\frac{\sin(\theta - \beta) \cos\beta}{\cos\beta + \cos(\theta - \beta)} = 0. \]  

This implies,

either \( \beta = \theta \) or \( (\theta - \beta) = \pi \).  

Now, one can easily check that for \( \theta = \beta \), the eigen value of \( M_1^2 \) becomes negative which is not possible. So, the solution here is \( (\theta - \beta) = \pi \). One can also check that for \( (\theta - \beta) = \pi \), the expression \( \frac{d^2(\lambda_0 + \lambda_1)}{d\beta^2} < 0 \). Hence, the expression \( \lambda_0 + \lambda_1 \) maximizes at the point \( (\theta - \beta) = \pi \). Substituting this relation in equations (122) and (123) we get,

\[ \lambda_0 = \lambda_1 = \frac{1}{2(1 + \cos\theta)}. \]  

By substituting the values of \( \lambda_0 + \lambda_1 \) in equation (101) we get,

\[ \lambda_2 = \frac{\cos\theta}{1 + \cos\theta}. \]  

Hence, we get,

\[ \Omega^1 < \frac{2\sin^2\theta}{1 + \cos\theta}. \]  

The corresponding measurement operators using which \( A_2 \) can achieve \( \Omega^1 = \frac{2\sin^2\theta}{1 + \cos\theta} \) is given by,

\[ M_0^1 = \frac{1}{2(1 + \cos\theta)}(I + \sin 2\theta \sigma_X + \cos 2\theta \sigma_Z) \]  
\[ M_1^1 = \frac{1}{2(1 + \cos\theta)}(I + \sigma_Z) \]  
\[ M_2^1 = \frac{\cos\theta}{1 + \cos\theta}(I - \sin \theta \sigma_X - \cos \theta \sigma_Z). \]
We can rewrite the above expressions as follows,

\[ M_{0}' = \frac{1}{1 + \cos \theta} \left( |0\prime\rangle \langle 0\prime| \right) \]

\[ M_{1}' = \frac{1}{1 + \cos \theta} \left( |0\rangle \langle 0| \right) \]

\[ M_{2}' = I - M_{0}' - M_{1}' , \]

where, \( |0\prime\rangle = \cos \theta |0\rangle + \sin \theta |1\rangle \). This concludes the proof.

From the results of theorems 6 and 7, it is clear that the success probability \( (1 - \cos \theta) \) in distinguishing two non-orthogonal states \{\( |0\rangle, |0\prime\rangle \) (or \{\( |1\rangle, |1\prime\rangle \)\}) can be achieved only when the chosen POVM’s are of the specified form as chosen by Alice for the QPQ scheme. From the results mentioned in [28], one can easily conclude that \( (1 - \cos \theta) \) is the optimal success probability that can be achieved in distinguishing two non-orthogonal states. So from these two results, one can easily conclude that Alice can get optimal number of raw key bits in this QPQ scheme.

Appendix D. Correctness of the scheme considering devices ‘up to a unitary’

In the device independent testing phases of our proposed scheme (i.e. in source device verification phase, Bob’s measurement device verification phase and Alice’s POVM device verification phase), the tests certify that the devices perform exactly same as that is mentioned in the proposed scheme or ‘up to a unitary’ of the actual device. This implies that the source device supplies states that are exactly of the same form or ‘up to a unitary’ (i.e. the states received after applying a unitary operation) of the original state and the measurement devices measure in exactly the same specified basis or ‘up to a unitary’ (i.e. the measurement bases received after applying a unitary operation) of the actual basis.

Thus, because of this ‘up to unitary’ deviation, it is necessary to check whether the protocol preserves its correctness condition whenever the devices are ‘up to unitary’ of the actual devices.

Let us consider that the measurement devices of Alice and Bob perform measurements in the bases which are up to unitary \( U_2 \) such that

\[ U_2 = \begin{pmatrix} a & b \\ -e^{i\phi}b^* & e^{i\phi}a^* \end{pmatrix} \]

where, \( a, b \in \mathbb{C} \) such that \( |a|^2 + |b|^2 = 1 \) and \( \phi \) is the relative angle. Let us also assume that the source device supplies states which are up to unitary \( U_4 \) where

\[ U_4 = U_2 \otimes U_2 . \]

This implies that the states supplied by the source device are of the form

\[ U_4(\phi,AB) = \frac{1}{\sqrt{2}} \left[ |00\rangle + e^{i\phi} (a^*b - ab^*) |01\rangle \right. \]

\[ + \left. e^{i\phi} (a^*b - ab^*) |10\rangle + e^{2i\phi} \left( a^2 + b^2 \right) |11\rangle \right] . \]

Bob’s device measures in the basis \{\( U_2|0\rangle, U_2|1\rangle \} = \{ (a|0\rangle - e^{i\theta}b^*|1\rangle) \}

\( (b|0\rangle + e^{i\phi}a^*|1\rangle) \} \) and \{\( U_2|0\prime\rangle, U_2|1\prime\rangle \} = \{ (a\cos \theta + b\sin \theta)|0\rangle + e^{i\phi}(a^*\sin \theta - b^*\cos \theta)|1\rangle, \}

47
\((a \sin \theta - b \cos \theta)|0\rangle - e^{i\phi} (a^* \cos \theta + b^* \sin \theta)|1\rangle\} \text{ instead of the basis } \{|0\rangle, |1\rangle\} \text{ and } \{|0', 1'\rangle\} \text{ respectively. Alice’s POVM devices are either } D^0 = \{D^0_0, D^0_1, D^0_2\} \text{ or } D^1 = \{D^1_0, D^1_1, D^1_2\} \text{ for } a_i = 0 \text{ and } a_i = 1 \text{ respectively where}

\[
D^0_0 = \frac{1}{1 + \cos \theta} (U_2|1\rangle\langle 1'|U^*_2), \\
D^0_1 = \frac{1}{1 + \cos \theta} (U_2|0\rangle\langle 0'|U^*_2), \\
D^0_2 = I - D^0_0 - D^0_1,
\]

and

\[
D^1_0 = \frac{1}{1 + \cos \theta} (U_2|0\rangle\langle 0'|U^*_2), \\
D^1_1 = \frac{1}{1 + \cos \theta} (U_2|1\rangle\langle 1'|U^*_2), \\
D^1_2 = I - D^1_0 - D^1_1,
\]

One can easily check that whenever Bob measures in \(\{U_2|0\rangle, U_2|1\rangle\} \text{ or } \{U_2|0', 1'\rangle\} \text{ basis randomly on his qubit of the shared state } U_4(\phi, \Lambda_B), \text{ the qubit at Alice’s side will also collapse to } U_2|0\rangle \text{ or } U_2|1\rangle \text{ for the first case and } U_2|0'\rangle \text{ or } U_2|1'\rangle \text{ for the second case.}

Now, if Alice chooses POVM device \(D^0 = \{D^0_0, D^0_1, D^0_2\} \text{ for } a_1 = 0\), the probabilities of getting different outcomes for two different input states are as follows-

\[
\text{Pr}(D^0_0|U_2|0\rangle) = (1 - \cos \theta), \\
\text{Pr}(D^0_1|U_2|0\rangle) = 0, \\
\text{Pr}(D^0_2|U_2|0\rangle) = \cos \theta, \\
\text{Pr}(D^0_0|U_2|0'\rangle) = 0, \\
\text{Pr}(D^0_1|U_2|0'\rangle) = (1 - \cos \theta), \\
\text{Pr}(D^0_2|U_2|0'\rangle) = \cos \theta.
\]

Similarly, if Alice chooses POVM device \(D^1 = \{D^1_0, D^1_1, D^1_2\} \text{ for } a_i = 1\), the probabilities of getting different outcomes for two different input states are as follows-

\[
\text{Pr}(D^1_0|U_2|1\rangle) = (1 - \cos \theta), \\
\text{Pr}(D^1_1|U_2|1\rangle) = 0, \\
\text{Pr}(D^1_2|U_2|1\rangle) = \cos \theta, \\
\text{Pr}(D^1_0|U_2|1'\rangle) = 0, \\
\text{Pr}(D^1_1|U_2|1'\rangle) = (1 - \cos \theta), \\
\text{Pr}(D^1_2|U_2|1'\rangle) = \cos \theta.
\]

According to the protocol, whenever \(a_i = 0\) and Alice gets \(D^0_0(D^1_0)\), she outputs \(r_{A_i} = 0(1)\). Whenever, \(a_i = 1\) and she gets \(D^0_0(D^1_0)\), she outputs \(r_{A_i} = 0(1)\). So, in this case, the success probability of Alice to guess the \(i\)-th raw key bit \(r_i\) of Bob will be,
\[ Pr(\tau_{A_i} = r_i) = Pr(\tau_{A_i} = 0, r_i = 0) + Pr(\tau_{A_i} = 1, r_i = 1) = (1 - \cos \theta). \]

This shows that whenever the devices (both source and measurement devices) involved in this scheme are "up to a unitary" of the original specified device, then also the proposed scheme satisfies the correctness condition.

**ORCID iDs**

Jyotirmoy Basak [https://orcid.org/0000-0003-0303-8365](https://orcid.org/0000-0003-0303-8365)

Arpita Maitra [https://orcid.org/0000-0002-7007-0384](https://orcid.org/0000-0002-7007-0384)

**References**

[1] Basak J, Chakraborty K, Maitra A and Maitra S 2022 A proposal for device independent probabilistic quantum oblivious transfer Indocrypt 2022 (Lecture Notes in Computer Science vol 13774) (Springer)

[2] Chor B, Goldreich O, Kushilevitz E and Sudan M 1995 Private information retrieval Proc. 36th Annual Symp. on Foundations of Computer Science pp 41–50

[3] Kushilevitz E and Ostrovsky R 1997 Replication is not needed: single database, computationally-private information retrieval Proc. 38th Annual Symp. on Foundations of Computer Science pp 364–73

[4] Cachin C, Micali S and Stadler M 1999 Computationally private information retrieval with polynomial communication Advances in Cryptology—EUROCRYPT '99 pp 402–14

[5] Gentry C and Ramzan Z 2005 Single-database private information retrieval with constant communication rate Int. Coll. on Automata, Languages and Programming (Lecture Notes in Computer Science vol 3580) pp 803–15

[6] Ostrovsky R and Skeith W E III 2007 A survey of single-database private information retrieval: techniques and applications Proc. 10th Int. Conf. on Practice and Theory in Public-Key Cryptography pp 393–411

[7] Di Crescenzo G, Malkin T and Ostrovsky R 2000 Single database private information retrieval implies oblivious transfer EUROCRYPT 2000 (Lecture Notes in Computer Science vol 1807) pp 122–38

[8] Gertner Y, Ishai Y, Kushilevitz E and Malkin T 1998 Protecting Data Privacy in Private Information Retrieval Schemes Proc. 30th Annual ACM Symp. on Theory of Computing pp 151–60

[9] Lo H K 1997 Insecurity of quantum secure computations Phys. Rev. A 56 1154

[10] Kon W Y and Lim C C W 2021 Provably-secure symmetric private information retrieval with quantum cryptography Entropy 23 54

[11] Mishra S K and Sarkar P 2000 Symmetrically private information retrieval INDOCRYPT (Lecture Notes in Computer Science vol 1977) pp 225–36

[12] Shor P W 1994 Algorithms for quantum computation: discrete logarithms and factoring Proc. 35th Annual Symp. on Foundations of Computer Science (IEEE Computer Society Press) pp 124–34

[13] Giovannetti V, Lloyd S and Maccone L 2008 Quantum private queries Phys. Rev. Lett. 100 23

[14] Giovannetti V, Lloyd S and Maccone L 2010 Quantum private queries: security analysis IEEE Trans. Inf. Theory 56 3465–77

[15] Olejnik L 2011 Secure quantum private information retrieval using phase-encoded queries Phys. Rev. A 84 022313

[16] Jakobi M, Simon C, Gisin N, Bancal J D, Branciard C, Walenta N and Zbinden H 2011 Practical private database queries based on a quantum-key-distribution protocol Phys. Rev. A 83 022301

[17] Scarani V, Acín A, Ribordy G and Gisin N 2004 Quantum cryptography protocols robust against photon number splitting attacks for weak laser pulse implementations Phys. Rev. Lett. 92 057901

[18] Gao F, Liu B, Wen Q Y and Chen H 2012 Flexible quantum private queries based on quantum key distribution Opt. Express 20 17441–20

[19] Rao MV P and Jakobi M 2013 Towards communication-efficient quantum oblivious key distribution Phys. Rev. A 87 012331
[20] Zhang J L, Guo F Z, Gao F, Liu B and Wen Q Y 2013 Private database queries based on counterfactual quantum key distribution Phys. Rev. A 88 022334
[21] Noh T G 2009 Counterfactual quantum cryptography Phys. Rev. Lett. 103 230501
[22] Yang Y G, Sun S J, Xu P and Tiang J 2014 Flexible protocol for quantum private query based on B92 protocol Quantum Inf. Process. 13 805
[23] Bennett C H 1992 Quantum cryptography using any two nonorthogonal states Phys. Rev. Lett. 68 3121–4
[24] Wei C Y, Gao F, Wen Q Y and Wang T Y 2014 Practical quantum private query of blocks based on unbalanced-state Bennett-Brassard-1984 quantum-key-distribution protocol Sci. Rep. 4 7537
[25] Liu B, Gao F, Huang W and Wen Q Y 2015 QKD-based quantum private query without a failure probability Sci. China Phys. Mech. Astron. 58 100301
[26] Maitra A, Paul G and Roy S 2017 Device-independent quantum private query Phys. Rev. A 95 042344
[27] Basak J and Maitra S 2018 Clauser–Horne–Shimony–Holt versus three-party pseudo-telepathy: on the optimal number of samples in device-independent quantum private query Quantum Inf. Process. 17 77
[28] Ivanovic I D 1987 How to differentiate between non-orthogonal states Phys. Lett. A 123 257–9
[29] Peres A and Terno D R 1998 Optimal distinction between non-orthogonal quantum states J. Phys. A: Math. Gen. 31 7105
[30] Kaniewski J 2017 Self-testing of binary observables based on commutation Phys. Rev. A 95 062323
[31] Broadbent A and Yuen P 2023 Device-independent oblivious transfer from the bounded-quantum-storage-model and computational assumptions New J. Phys. 25 053019
[32] Khodjasteh K, Sastrawan J, Hayes D, Green T J, Biercuk M J and Viola L 2013 Designing a practical high-fidelity long-time quantum memory Nat. Commun. 4 2045
[33] Hoeffding W 1963 Probability inequalities for sums of bounded random variables J. Am. Stat. Assoc. 58 13–30
[34] Reichardt B, Unger F and Vazirani U 2013 Classical command of quantum systems Nature 496 456
[35] Tavakoli A, Smania M, Vértesi T, Brunner N and Bourennane M 2020 Self-testing non-projective quantum measurements in prepare-and-measure experiments Sci. Adv. 6 16
[36] Helstrom C W 1976 Quantum Detection and Estimation Theory (Mathematics in Science and Engineering vol 123) (Academic)