Hybrid-query bounds with partial input control – framework and application to tight M-eTCR

Andreas Hülsing[1,2], Mikhail Kudinov[1], and Christian Majenz[3]

[1] Eindhoven University of Technology, Eindhoven, Netherlands
[2] SandboxAQ, Paolo Alto, USA
[3] Technical University of Denmark, Kongens Lyngby, Denmark
andreas@huelsing.net, mishel.kudinov@gmail.com, chmaj@dtu.dk

Abstract. In this paper, we present an improved framework for proving query bounds in the Quantum Random Oracle Model (QROM) for algorithms with both quantum and classical query interfaces, where the classical input is partially controlled by the adversary. By extending existing techniques, we develop a method to bound the progress an adversary can make with such partial-control classical queries. While this framework is applicable to different hash function properties, we decided to demonstrate the impact of the new techniques by giving an analysis of the multi-target extended target collision resistance property (m-eTCR). This new approach allows us to achieve an improved bound that significantly reduces the required function key size. Our proof is tight in terms of query complexity and has significant implications for cryptographic applications, especially for signature schemes in the hash & sign paradigm, enabling more efficient instantiations with reduced salt sizes and smaller signature lengths. For an example of multiple signatures aggregation, we achieve a signature size of 30 kB smaller.

Keywords: QROM · Hybrid QROM · TCR · Hash & Sign.

1 Introduction

Hash functions are one of the most widely used primitives in modern cryptography. One of the first steps to analyze a given security property for a hash function is to analyze the property for a random function. To do so, we model the hash function as a random oracle [2, 12] (Random Oracle Model, ROM). The ROM is the standard tool for characterizing generic attacks against hash functions: attacks that only depend on the input-output behavior of the hash function rather than the details of the algorithm. In the ROM, the number of queries required to break a certain security property is used as a proxy for the real (time-) complexity of an attack. On the other hand, such analysis gives a bound on the maximum possible level of security for real-world hash functions for a given property.

The desire for security of quantum computing attacks requires revisiting many cryptographic techniques, including the ROM. A quantum adversary possesses a large-scale quantum computer and may perform local quantum computations, while the honest users remain classical. Such quantum adversary can implement any publicly available primitive as a quantum circuit and run it on a superposition of inputs. As hash functions are public primitives, we need to model them as being quantumly accessible. This also applies to the ROM, resulting in the quantum-accessible random oracle model (QROM) [3].

As mentioned, for post-quantum security, we consider a quantum adversary, but the honest parties remain classical. Hence, any interaction with an honest user must be classical. If an adversary has

A.H. and M.K. were supported by an NWO VIDI grant (Project No. VI.Vidi.193.066). Date: April 6, 2025

access to a keyed functionality which, in turn, queries a random oracle, such access thus remains classical. A typical example would be a pseudorandomness notion for a keyed hash function. An honest user generates a secret key. The adversary may query the hash function instantiated with the secret key and needs to determine whether the responses are generated by actual evaluation of the hash function or if they are just random strings. In this case, the adversary needs to perform classical queries to the honest user while still being able to do quantum queries (containing both the key and the message) to the hash function.

Recently, a plethora of techniques has been developed to prove security in the QROM. Among those, there are reprogramming techniques [14] that allow us to update the outputs of the random oracle dependent on adversaries' queries. While this is a very powerful tool, sometimes we do not actually need to change the outputs of the hash function. In certain scenarios this tool was used to deal with classical challenge queries (as in the PRF example). The reprogramming techniques allow sampling responses for the challenge queries at the very beginning.This way, all the classical queries can be made before any quantum computations, allowing us to analyze the simple setting of a quantum algorithm for a known task. Here, the important part was to learn the outputs in advance, rather then changing the output of a hash function. For the analysis of the quantum part one could use a recently developed framework [7] by Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang and Tai-Ning Liao. This framework helps to analyze quantum query progress using only classical reasoning. In their work the authors view the QRO as a database, which is updated with each query. Such an approach is possible because of the compressed random oracle (CRO) – a technique introduced by Zhandry [27] that allows the investigation of queries made to the oracle. The CRO resembles classical lazy sampling. Unfortunately, these techniques do not give us a way to distinguish records in the database. We can not say which record is a result of a challenge query and which is an attempt to find a solution.

Motivated by the limited resources of a near-term quantum computing scenario, Hamoudi, Liu, and Sinha developed a technique to give a fine-grained analysis of hybrid algorithms [17]. Such algorithms use a mix of quantum and classical queries. The authors' aim was to show lower bounds on the complexities of certain problems. For example, the authors showed that the optimal success probability of an algorithm making q quantum and c classical queries for solving the Collision Finding problem is $\Theta((c^2+cq^2+q^3)/N)$. The crucial part of their framework is that it allows the distinction between classical and quantum queries.

In [17] classical and quantum queries model different implementations of a public primitive. It was thus not necessary to restrict the adversary in their choice of input to the classical query interface. For settings where the classical query interface results from modeling adversarial access to a keyed cryptographic functionality, the adversary might only get partial control over the query input (as in the PRF example). An important open question is thus:

Can we upgrade the hybrid compressed oracle technique to allow tight query bounds for partial-input-control query access?

Our contribution. In this work, we develop a framework for proving QROM query bounds for algorithms with standard quantum access and an additional classical query interface with partially-random input.

In our work, we generalize the techniques used by the authors of [17]. Instead of using the hybrid CRO for analyzing the overall complexity of an algorithm, we use it to model the classical interactions of a quantum adversary. A key insight is that the separate databases for quantum- and classical queries in the framework of [17] allow analyzing problems where input-output pairs obtained from the classical query interface are treated differently than other pairs.

Concretely, we develop a framework for proving query bounds in settings where an adversary can provide part of the input to the classical query interface of a QRO, and the remaining part is sampled

at random. Starting from the work of [17], our technique bounds the "progress", as measured by a database predicate projector, that the adversary can achieve with such partial-control classical queries.

To illustrate the utility of our technique, we analyze the multi-target extended target collision resistance property (M-ETCR) in the QROM. Loosely speaking, the adversary must find a collision for a message of their choice, but the key for that message is chosen uniformly at random: $\mathcal{A}$ chooses M, and then given a uniformly random K, must find $(K', M' \neq M)$, s.t. $F(K', M') = F(K, M)$. A typical use-case for this property can be keyed-hash message authentication codes [20] or signature schemes in the hash & sign paradigm [1, 8, 23].The security of this property has been analyzed in the QROM in [14]. We give it another look. Using the techniques from [17] and the ones we develop in this paper, we were able to improve the state-of-the-art bound for QROM security of M-ETCR. The proof requires an analysis of the progress a quantum and a classical query can make. This is done with the use of our extension and careful inspection of possible amplitude growth.

The new bound allows us to remove more than half of the key size, reducing them from 192 to 72 bits. Moreover, our proof is tight in the number of queries. We provide an attack with a matching number of queries for each term in the bound, essentially closing the question of the security analysis of M-ETCR for random functions.

We also discuss the implications of our new proof. We take a closer look at one of the typical use cases – randomized hashing in the hash & sign paradigm. It is often the case that we want to hash the message before signing it. Adding randomness to the hash function input allows us to decrease the security requirement from collision resistance to M-ETCR. This usually allows us to use a smaller digest length but forces us to append the used randomness to the signature. Hence, using smaller salt gives us smaller signature sizes. This becomes especially important in the case of multiple signature aggregation. By applying our bound to a recent proposal [1], we achieve a significant reduction in the size of the aggregated signature. In case the aggregated signature is formed from 2000 signatures, our analysis enables a size reduction from 165 kB to 136 kB.

Organization. We introduce necessary definitions and discuss CRO in Section 2. Section 3 is devoted to the description of the hybrid CRO framework. In Section 4, we improve the framework by presenting a bound on the progress for a new type of query. The security proof for M-ETCR that uses the results obtained in the previous section is given in Section 5. Section 6 discuss the use of M-ETCR in signature schemes.

2 Preliminaries: Compressed Random Oracle

In this section, we revisit the quantum-accessible random oracle model (QROM) and Zhandry's Compressed Oracle (CRO) [27].

The random oracle methodology has been effective in designing efficient cryptographic protocols and proving their security in a rigorous, albeit idealized, way. This approach treats a cryptographic hash function as an external oracle that an adversary must query to learn the output for a given input. This work will mostly focus on keyed hash functions: $F : [K] \times [M] \to [N]$. While keyed hash functions have two inputs, they can still be modeled as an idealized object with a single input $x \in [M'] = [K] \times [M]$. The random oracle responds to these queries by simulating the behavior of a uniformly random function. Although it is acknowledged that this methodology can potentially fail [5, 19], experience suggests that this rarely occurs for naturally designed protocols.

When dealing with a quantum adversary who has access to a quantum computer, the random oracle must be redefined to be able to handle queries in superposition and accurately reflect the attacker's capabilities in a real-world setting. This is known as the quantum-accessible random oracle model (QROM). The key difference between the classical ROM and the QROM is that in the ROM, security reductions can examine the adversary's queries to the random oracle. However, in

the quantum setting, queries exist as superposition states, making it hard to inspect them without significantly disrupting their or the adversary's state.

To deal with this, Zhandry introduced the Compressed Oracle framework (CRO) [27]. The CRO provides an approach that is useful for establishing lower bounds against quantum algorithms with black-box access to a uniformly random function F, which maps $[M]$ to $[N]$ (in our case $[K] \times [M]$ to $[N]$). The CRO enables the storage of a compressed representation of the random function, conditional on the knowledge obtained from previous queries. Conceptually, the technique resembles the classical "lazy sampling" method. Technically, it considers a quantum purification of the random function F and then analyzes the internal state of the random oracle in the Fourier domain.

We will now closely follow the description of QROM and CRO from [7]. In our model, it is sufficient to work with three different registers $|x, y, z\rangle$, where x will contain a concatenation of elements from $[K]$ and $[M]$, y is the output register, and z is the work register. We will omit the work register in most cases. The standard approach for an ordinary QROM defined with unitary StO works the following way:

$$\mathsf{StO} \sum_{x,y} \alpha_{x,y} |x, y\rangle \to \sum_{x,y} \alpha_{x,y} |x, y + F(x)\rangle$$

To switch to the CRO, we first need to consider a superposition $\sum_F |F\rangle$ of all possible functions F in the defined domain and range. So, the initial state will be $|\Pi_0\rangle = \sum_F |F\rangle$

Now, we want to look at it in a Fourier basis, which we will denote with a hat symbol "$\hat{}$".

$$|\Pi_0\rangle = \sum_F |F\rangle = \bigotimes_x (\sum_y |y\rangle) = \bigotimes_x |\hat{0}\rangle$$

The idea is that we can compress these $|\hat{0}\rangle$ states in a new special state $|\perp\rangle$. This will imply some error for decompressing, but we can make it small enough for our use cases. So, we will define compression in the following way:

$$\mathsf{Comp}_x |\Pi_0\rangle = |\perp\rangle \langle\hat{0}| + \sum_{\hat{w} \neq 0} |\hat{w}\rangle \langle\hat{w}|, \text{ i.e.} \qquad |\hat{y}\rangle \to \begin{cases} |\perp\rangle & \text{if } \hat{y} = \hat{0} \\ |\hat{y}\rangle & \text{if } \hat{y} \neq \hat{0} \end{cases}$$

Now, we can apply this isometry to every register $x \in \mathcal{X}$ and obtain the compression operator $\mathsf{Comp} = \otimes_x \mathsf{Comp}_x$. If we apply Comp to the $|\Pi_0\rangle$, we will get all the $|\perp\rangle$ states.

$$\mathsf{Comp} |\Pi_0\rangle = (\bigotimes_x \mathsf{Comp}_x)(\bigotimes_x |\hat{0}\rangle) = \bigotimes_x \mathsf{Comp}_x |\hat{0}\rangle = \bigotimes_x |\perp\rangle$$

This can be viewed as a trivial database that maps everything to $|\perp\rangle$. This compression will work mostly the same as just working with the Fourier basis, i.e. $\mathsf{Comp} |\hat{F}\rangle = |\hat{D}\rangle$, where $\hat{D}$ is such that $\hat{D}(x) = \hat{F}(x)$ whenever $\hat{F}(x) \neq 0$ and $\hat{D}(x) = \perp$ whenever $\hat{F}(x) = 0$. As a result, after q queries, we will have the internal state of the Compressed Oracle consisting of several state vectors, where $D(x) = \perp$ for all but (at most) q choices of x. The last step is to efficiently store it in terms of the number of qubits. We omit this technical detail. For more information, we refer to [27]. We now use an updated definition of StO to accommodate the superposition of functions:

$$\mathsf{StO} \sum_{x,y,F} \alpha_{x,y,F} |x, y\rangle |F\rangle \to \sum_{x,y,F} \alpha_{x,y,F} |x, y + F(x)\rangle |F\rangle$$

As a result, we have the following compressed random oracle:

$$\mathsf{cO} := \mathsf{Comp} \circ \mathsf{StO} \circ \mathsf{Comp}^\dagger$$

Here, we implicitly refer to F representation as database D. Intuitively, a database D represents a classical lazy sampling technique. The original state is D_0 where for any input the output is the $\perp$ symbol. After a query, we check if the input is in the database or not. In the first case, we respond with the value assigned to that input in D; otherwise, we sample a uniformly random value for the output and update D, i.e., $D_i = D_{i-1}[x_i \to y_i]$. So $D_i(x_i) = y_i$. In the case of superposition queries, we have a superposition of such databases.

According to [10], we bound the difference in working with QROM rather than CRO by the following corollary.

Corollary 1 ([10, Corollary 2.8]). *Let $R \subseteq \mathcal{X}^\ell \times \mathcal{Y}^\ell \times \mathcal{Z}$ be a relation, where $|\mathcal{Y}| = N$. Let $\mathcal{A}$ be an oracle quantum algorithm that outputs $\mathbf{x} = (x_1, \ldots, x_\ell) \in \mathcal{X}^\ell$ and $z \in \mathcal{Z}$. Let $\tilde{\mathcal{A}}$ be the oracle quantum algorithm that runs $\mathcal{A}$, makes ℓ classical queries on the outputs x_i to obtain $\mathbf{y} = (y_1 = F(x_1), \ldots, y_l = F(x_l)) = F(\mathbf{x})$, and then outputs $(\mathbf{x}, \mathbf{y}, z)$. Let*

$$p_\circ(\mathcal{A}) := \Pr[(\mathbf{x}, F(\mathbf{x}), z) \in R]$$

be the considered probability when $\mathcal{A}$ has interacted with the RO. Furthermore, let $p(\tilde{\mathcal{A}})$ be the probability that $\mathbf{y} = F(\mathbf{x})$ and $(\mathbf{x}, \mathbf{y}, z) \in R$ when $\tilde{\mathcal{A}}$ has interacted with the standard random oracle, initialized with a uniformly random function F, and $p'(\tilde{\mathcal{A}})$ be the probability that $\mathbf{y} = D(\mathbf{x})$ and $(\mathbf{x}, \mathbf{y}, z) \in R$ when $\tilde{\mathcal{A}}$ has interacted with the compressed oracle instead and D is obtained by measuring its internal state (in the computational basis). Then

$$p_\circ(\mathcal{A}) = p(\tilde{\mathcal{A}}) \leq p'(\tilde{\mathcal{A}}) + \frac{2\ell}{N}.$$

In our work, we actually need extra features. We need to be able to distinguish classical queries from quantum. This is implemented as a hybrid random oracle [17], which we present in Section 3.

3 Hybrid Compressed Oracle

In this section, we briefly recall the construction of the Hybrid Compressed Random Oracle (HCRO) and its properties from [17].

As presented in [17], a hybrid compressed random oracle is a framework that allows us to analyze the success probability of hybrid algorithms that perform a mix of quantum and classical queries. The HCRO framework is built upon the CROM. While in [17] this framework was used to develop general lower-bound techniques that characterize the tradeoffs between the number of quantum queries and classical queries, we use it to address specific properties of hash functions. We focus on the properties that specify the challenges through the oracle interaction.

The hybrid oracle is built by substituting the original classical and quantum query operators with two novel "recording query operators", which ensure a consistent record of classical-quantum queries throughout the algorithm's execution. Notably, when all queries are either exclusively classical or quantum, the framework reduces to the classical lazy sampling method and the quantum compressed oracle technique, respectively, in these extreme cases.

The main addition of the Hybrid Compressed Oracle framework is a history register. The history register $\mathcal{H}$ is dedicated to recording all the classical queries $(x, D(x))$. The contents of the recorded query stay unchanged through the whole run of the algorithm. To incorporate it with the CROM, new compression and uncompression operations are defined. The new ones are conditioned on the content of the history register. If an input x is recorded in history, then it is never compressed or uncompressed for the database again.

3.1 Hybrid CRO overview

Below, we formally present the Hybrid CRO framework [17].

Memory. The memory of an algorithm accessing an oracle $D : [M'] \to [N]$ is made of three quantum registers defined as follows:

- Index register $\mathcal{X}$ holding $x \in [M']$. We will sometimes partition register $\mathcal{X}$ into two: register $\mathcal{K}$ and register $\mathcal{M}$ representing the division of the space $[M']$ into two subspaces. $\mathcal{K}$ holds $k \in [K]$ and $\mathcal{M}$ holds $m \in [M]$.
- Phase register $\mathcal{Y}$ holding $y \in [N]$.
- Workspace register $\mathcal{Z}$ holding $z \in \{0,1\}^*$ (the register size may increase during the computation as we allow appending new qubits to it).

We use $\mathcal{A} = \mathcal{X}\mathcal{Y}\mathcal{Z}$ as a shorthand for the registers on which the algorithm operates. The initial state of the memory is the all-zero basis state $|0,0,0\rangle_{\mathcal{A}}$. In this paper, we consider the phase oracle, which returns the value $D(x)$ in the phase, but it is equivalent to the standard oracle that maps $|x,y,z\rangle_{\mathcal{A}}$ to $|x,y \oplus D(x),z\rangle_{\mathcal{A}}$ up to a unitary transformation.

Quantum Phase Oracle. We define the quantum oracle $\mathcal{O}_0^D$ as the unitary operator acting on the memory of the algorithm as follows.

$$\mathcal{O}_0^D : |x,y,z\rangle_{\mathcal{A}} \mapsto \omega_N^{yD(x)} |x,y,z\rangle_{\mathcal{A}} \quad \text{where} \quad \omega_N = e^{\frac{2i\pi}{N}}.$$

Classical Oracle. A classical oracle query can be viewed as a query to the standard oracle that maps $|x,y,z\rangle_{\mathcal{A}}$ to $\omega_N^{yD(x)} |x,y,z\rangle_{\mathcal{A}}$ followed by a measurement on the index register $\mathcal{X}$ and phase register $\mathcal{Y}$.

To represent the measurement, we add the *history* register $\mathcal{H}$. In our work we consider that the number of classical queries is limited (for example at most t), so the history register can be represented as $\mathcal{H} = \mathcal{H}_1 \cdots \mathcal{H}_t$ where the c-th subregister $\mathcal{H}_c$ is used to purify the c-th classical query and stores a value in $([M'] \times [N]) \cup \{\star\}$, where $\star$ is a new state that represents that the query has not happened yet. The initial state of that register is $|\star,\ldots,\star\rangle_{\mathcal{H}}$. The classical oracle $\mathcal{O}_1^D$ is defined as the unitary operator acting as follows

$$\begin{aligned} \mathcal{O}_1^D \ : \quad & |x,y,z\rangle_{\mathcal{A}} |(x_1,y_1),\ldots,(x_c,y_c),\star,\ldots,\star\rangle_{\mathcal{H}} \\ \mapsto\ & \omega_N^{yD(x)} |x,y,z\rangle_{\mathcal{A}} |(x_1,y_1),\ldots,(x_c,y_c),(x,D(x)),\star,\ldots,\star\rangle_{\mathcal{H}}. \end{aligned}$$

We will denote the list of history records $((x_1,y_1),\ldots,(x_c,y_c),\star,\ldots,\star)$ by H and we say $x \in H$ if and only if there exists $1 \le i \le c$ such that $x_i = x$.

Definition 2 ($H_{x \leftarrow y}$ **[17]).** *Given a history $H = ((x_1,y_1),\ldots,(x_c,y_c),\star,\ldots,\star)$ with at least one star entry, we define the appendment of a new pair (x,y) to H as*

$$H_{x \leftarrow y} = ((x_1,y_1),\ldots,(x_c,y_c),(x,y),\star,\ldots,\star)$$

where the leftmost star has been replaced with (x,y).

Sometimes, we will identify the above list with a function $H : [M'] \to [N] \cup \{\star\}$ if there are no ambiguous pairs, i.e., no pairs of the form (x,y) and (x,y') where $y \neq y'$. We also let $\mathscr{H}$ denote the set of all possible histories H.

Hybrid Oracle. We extend the above definitions by allowing for probabilistic choices between the two oracles. This is represented by a channel that applies the quantum oracle $\mathcal{O}_0^D$ with probability $1-b$, for some $b \in [0,1]$, and applies the classical oracle $\mathcal{O}_1^D$ otherwise. Additionally, we assume that the algorithm is provided with a query type bit (or "flag") indicating which oracle has been applied. We represent this operation by an isometry$\mathcal{O}_b^D$ acting as

$$\mathcal{O}_b^D : |x,y,z\rangle_{\mathcal{A}} |H\rangle_{\mathcal{H}} \mapsto \omega_N^{yD(x)} |x,y\rangle_{\mathcal{XY}} \left(\sqrt{1-b}\cdot|z0\rangle_{\mathcal{Z}} |H\rangle_{\mathcal{H}} + \sqrt{b}\cdot|z1\rangle_{\mathcal{H}} |H_{x\leftarrow D(x)}\rangle_{\mathcal{H}}\right)$$

where the bit appended to the workspace z indicates the nature of the oracle. We recover the quantum and classical oracles when $b=0$ and $b=1$ respectively (ignoring the query type bit). We will not use $b \notin \{0,1\}$ in the analysis, but sometimes it is more convenient to use this representation.

Hybrid Algorithm. An algorithm with t queries is defined as a sequence $U_0,\ldots,U_t$ of unitary transformations acting on the memory register $\mathcal{A}$ and a list of real numbers $b(1),\ldots,b(t) \in \{0,1\}$ that specifies which type of query is used. The state $|\psi_t^D\rangle$ of the algorithm after t queries is

$$|\psi_t^D\rangle = U_t\, \mathcal{O}_{b(t)}^D\, U_{t-1} \cdots U_1\, \mathcal{O}_{b(1)}^D\, U_0\ |0\rangle_{\mathcal{A}} |\star,\ldots,\star\rangle_{\mathcal{H}}\,.$$

The function D is chosen uniformly at random from the set $\{\, D : [M'] \to [N] \,\}$. We model that by adding another purification register (the *database*) $\mathcal{D} = \mathcal{D}_0 \ldots \mathcal{D}_{M-1}$ where each subregister $\mathcal{D}_x$ for $x \in [M']$ holds a value $D(x) \in [N]$ and we define the following joint state,

$$|\psi_t\rangle = \frac{1}{N^{M'/2}} \sum_{D\in[N]^{[M']}} |\psi_t^D\rangle_{\mathcal{AH}} \otimes |D\rangle_{\mathcal{D}} = U_t\, \mathcal{O}_{b(t)}\, U_{t-1} \cdots U_1\, \mathcal{O}_{b(1)}\, U_0\ |\psi_0\rangle\,,$$

where $\mathcal{O}_b := \sum_D \mathcal{O}_b^D \otimes |D\rangle\langle D|_{\mathcal{D}}$ and $|\psi_0\rangle := |0\rangle_{\mathcal{A}} \otimes |\star,\cdots,\star\rangle_{\mathcal{H}} \otimes \frac{1}{N^{M/2}} \sum_D |D\rangle_{\mathcal{D}}$.

Output. The output of a hybrid algorithm is obtained by performing a computational basis measurement on the final state $|\psi_t\rangle$ where we measure a designated part of the workspace register $\mathcal{Z}$.

As we mentioned, we only consider the algorithms that make only two types of queries: quantum and classical. One can further distinguish whether the algorithm is static or adaptive. We say that the algorithm is static if the order of quantum and classical queries is fixed. An adaptive algorithm can adaptively choose the query type of each individual query as long as the total number of quantum (classical) queries is unchanged. Theorem 3 shows that without loss of generality, we can always consider the algorithm to be static.

Theorem 3 ([9, 17]). *In the hybrid random oracle model, for any adaptive hybrid quantum algorithm making at most q quantum queries and c classical, there exists a static hybrid algorithm making at most $2q$ quantum queries and $2c$ classical queries such that their outputs are always identical.*

3.2 Construction

Now, we present the actual construction from [17]. While we include it to give a self-contained presentation, the reader might want to skip this subsection, proceed with the Section 3.3, and return to this one later.

To begin, it is necessary to define a compressed encoding for the database that is compatible with the history register. This involves expanding the alphabet used for the database register, allowing $\mathcal{D}_x$ to hold values from the set $\{\,\perp\,\} \cup [N]$, where $D(x)$ represents the value associated with x. We state that $\omega_N^{yD(x)} = 1$ when $D(x) = \perp$. The initial state of the database is set as $|\perp,\ldots,\perp\rangle_{\mathcal{D}}$, implying all entries are initially undefined. The history register's alphabet is also expanded to accommodate

tuples of the form $(x, \perp)$, where x belongs to $[M']$. Denote $x \in H$ if a tuple (x, y) exists in H, with $y \in \perp \cup [N]$. In cases where no ambiguous pairs are present in the list, H can be viewed as a function that maps elements from $[M']$ to the extended set $\perp, \star \cup [N]$.

We define the uncompression operator S. Let $|\hat{y}\rangle_{\mathcal{D}_x} = \frac{1}{\sqrt{N}} \sum_{p \in [N]} \omega_N^{yp} |p\rangle_{\mathcal{D}_x}$ for $y = 0, \ldots, N-1$, denote the Fourier basis states and let S_x be the unitary operator acting on $\mathcal{D}_x$ such that

$$S_x : \begin{cases} |\perp\rangle_{\mathcal{D}_x} \longmapsto |\hat{0}\rangle_{\mathcal{D}_x} \\ |\hat{0}\rangle_{\mathcal{D}_x} \longmapsto |\perp\rangle_{\mathcal{D}_x} \\ |\hat{y}\rangle_{\mathcal{D}_x} \longmapsto |\hat{y}\rangle_{\mathcal{D}_x} \quad \text{for } y = 1, \ldots, N-1. \end{cases}$$

S_x is unitary and Hermitian. A controlled unitary $S_{x,H}$ acting on $\mathcal{D}_x$ is defined as:

$$S_{x,H} = \begin{cases} \mathbb{I} & \text{if } x \in H \\ S_x & \text{otherwise.} \end{cases}$$

Define the Hermitian unitary operator S acting on $\mathcal{AHD}$ such that:

$$S = \sum_{x \in [M'], H \in \mathscr{H}} |x\rangle\langle x|_{\mathcal{X}} \otimes \mathbb{I}_{\mathcal{YZ}} \otimes |H\rangle\langle H|_{\mathcal{H}} \otimes (\mathbb{I}_{\mathcal{D}_0 \ldots \mathcal{D}_{x-1}} \otimes S_{x,H} \otimes \mathbb{I}_{\mathcal{D}_{x+1} \ldots \mathcal{D}_{M-1}}).$$

The hybrid compressed oracle $\mathcal{R}_b$ is defined as follows,

$$\mathcal{R}_b = S\mathcal{O}_b S \quad \text{where} \quad \mathcal{O}_b = \sum_{D \in (\{\perp\} \cup [N])^M} \mathcal{O}_b^D \otimes |x\rangle\langle x|_{\mathcal{D}}, \text{ for } b \in [0, 1].$$

We acquired an oracle that, for any basis state $|x, y, z\rangle_{\mathcal{A}} |H, D\rangle_{\mathcal{HD}}$ satisfies the following:

- If the queried input x is contained in the history register: $x \in H$, it means that x has been queried classically before; then we stop (un)compressing $\mathcal{D}_x$, and it behaves like a regular phase oracle on input x.
- If $x \notin H$, then $\mathcal{D}_x$ is simulated as a compressed oracle.

The quantum query $\mathcal{R}_0$ only acts on the register $\mathcal{H}$ as control and does not change its records. The joint state $|\phi_t\rangle$ of the algorithm and the oracle after t queries in the hybrid compressed oracle model is defined as

$$|\phi_t\rangle = U_t \mathcal{R}_{b(t)} U_{t-1} \cdots U_1 \mathcal{R}_{b(1)} U_0 (|0\rangle_{\mathcal{A}} |\star, \ldots, \star\rangle_{\mathcal{H}} |\perp, \ldots, \perp\rangle_{\mathcal{D}}).$$

The initial state is defined as $|\phi_0\rangle = |0\rangle_{\mathcal{A}} \otimes |\star, \cdots, \star\rangle_{\mathcal{H}} \otimes |\perp, \ldots, \perp\rangle_{\mathcal{D}}$.

3.3 Basic results regarding HCRO

Below, we will present the main properties and results for HCRO obtained in [17]. We start with a definition of the History-Database Consistent state. Each query can increase the size of the quantum database by no more than one input. And a history database query also increases the number of records maximally by one. The history database must be unambiguous: there should not be two pairs (x, y), (x, y'), where $y \neq y'$. We also want the quantum part to coincide with the classical part: $H(x) = D(x)$, where $H(x) \neq \star$.

Definition 4 (History-Database Consistent State [17]). *Given an integer t, we say that (H, D) is a history-database t-consistent pair if it has the following properties:*

1. *(Database SIZE) The database satisfies $D(x) \neq \perp$ for at most t different values of x.*
2. *(History SIZE) The history is of the form $H = ((x_1, y_1), \ldots, (x_c, y_c), \star, \ldots, \star)$ where $x_1, \ldots, x_c \in [X]$ and $y_1, \ldots, y_c \in \{\perp\} \cup [N]$ for some $c \leq t$.*
3. *(Uniqueness) We can identify the history with a function $H : [X] \to \{\star, \perp\} \cup [N]$ where $H(x_j) = y_j$ for all $j \in \{1, 2, \cdots c\}$ (meaning no two pairs in the history can differ on the second coordinate only) and $H(x) = \star$ for $x \notin \{x_1, \ldots, x_c\}$.*
4. *(Equality) The database coincides with the history on non-$\star$ values, meaning that $H(x) \neq \star$ implies $D(x) = H(x)$.*

We let $\mathbb{H}_t$ denote the vector space spanned by all basis states $|x, y, z\rangle_{\mathcal{A}} |H, D\rangle_{\mathcal{HD}}$ where (H, D) is history-database t-consistent. We say that a basis state is history-database consistent if it is in $\mathbb{H}_t$ for some integer t.

Proposition 5 (Consistency [17]). *Any state $|\phi_t\rangle$ obtained after t queries in the compressed oracle model satisfies $|\phi_t\rangle \in \mathbb{H}_t$.*

The following lemmas describe what happens after a quantum or a classical query. Note that a quantum query never changes the history part. For a classical query with an input that is not in the history database, but in the quantum database, there is a small chance of resampling, but most probably the database will remain the same.

Lemma 6 (Quantum Query $\mathcal{R}_0$ [17]). *Let $|x, y, z\rangle |H, D\rangle$ be a history-database consistent basis state. Then, $\mathcal{R}_0$ maps this state to $|x, y, z0\rangle |H\rangle |\varphi\rangle$ where the state $|\varphi\rangle$ of the database register is*

- $\omega^{yD(x)} |D\rangle$ (if $H(x) \neq \star$ or $y = 0$)
- $\displaystyle\sum_{p \in [N]} \frac{\omega^{yp}}{\sqrt{N}} |D_{x \leftarrow p}\rangle$ (if $H(x) = \star$, $D(x) = \perp$, $y \neq 0$)
- $\omega^{yD(x)} |D\rangle + \frac{\omega^{yD(x)}}{\sqrt{N}} |D_{x \leftarrow \perp}\rangle + \displaystyle\sum_{p \in [N]} \frac{1 - \omega^{yD(x)} - \omega^{yp}}{N} |D_{x \leftarrow p}\rangle$ (if $H(x) = \star$, $D(x) \neq \perp$, $y \neq 0$)

Lemma 7 (Classical Query $\mathcal{R}_1$ [17]). *Let $|x, y, z\rangle |H, D\rangle$ be a history-database consistent basis state. Then, $\mathcal{R}_1$ maps this state to $|x, y, z1\rangle |\varphi\rangle$ where the state $|\varphi\rangle$ of the history-database registers is*

- $\omega^{yD(x)} |H_{x \leftarrow D(x)}, D\rangle$ (if $H(x) \neq \star$)
- $\displaystyle\sum_{p \in [N]} \frac{\omega^{yp}}{\sqrt{N}} |H_{x \leftarrow p}, D_{x \leftarrow p}\rangle$ (if $H(x) = \star$, $D(x) = \perp$)
- $\omega^{yD(x)} |H_{x \leftarrow D(x)}, D\rangle + \frac{1}{\sqrt{N}} |H_{x \leftarrow \perp}, D_{x \leftarrow \perp}\rangle - \displaystyle\sum_{p \in [N]} \frac{\omega^{yp}}{N} |H_{x \leftarrow p}, D_{x \leftarrow p}\rangle$ (if $H(x) = \star$, $D(x) \neq \perp$)

3.4 Progress measure

In the following, we present the results from [17] that help us measure the amount of progress an algorithm can make towards solving a given task. In Section 4, we add a new result to the already existing framework. To define the task, we use predicates. In this paper, all progress measures will be defined in terms of the norm of the projection onto basis states satisfying certain predicates.

Definition 8 (Basis-State Predicate [17]). *Let* $\mathrm{P} : (x, y, z, H, D) \mapsto \{False, True\}$ *be a predicate function over all basis states* $|x, y, z\rangle_{\mathcal{A}} |H, D\rangle_{\mathcal{HD}}$*. We define the projection*

$$\Pi_{\mathrm{P}} = \sum_{(x,y,z,H,D)\in \mathrm{P}^{-1}(True)} |x, y, z, H, D\rangle \langle x, y, z, H, D|$$

over all basis states satisfying P*. We let* $\overline{\mathrm{P}}$ *denote the negation of* P *and, given two predicates* P_1 *and* P_2*, we let* $\mathrm{P}_1 \cdot \mathrm{P}_2$ *denote their conjunction and* $\mathrm{P}_1 + \mathrm{P}_2$ *denote their disjunction.*

For the basis-state predicates the following fact will be used in our proofs.

Fact 4.9 [17]. *Let* P_1 *and* P_2 *be two basis-state predicates. Then, the projections* Π_{P_1} *and* Π_{P_2} *are commuting operators. We have* $\Pi_{\overline{\mathrm{P}_1}} = \mathbb{I} - \Pi_{\mathrm{P}_1}, \Pi_{\mathrm{P}_1\cdot\mathrm{P}_2} = \Pi_{\mathrm{P}_1}\Pi_{\mathrm{P}_2}$ *and* $\Pi_{\mathrm{P}_1+\mathrm{P}_2} = \Pi_{\mathrm{P}_1} + \Pi_{\mathrm{P}_2} - \Pi_{\mathrm{P}_1}\Pi_{\mathrm{P}_2}$*. Moreover,* $P_1 \Rightarrow P_2$ *if and only if* $\Pi_{\mathrm{P}_1} \preceq \Pi_{\mathrm{P}_2}$*, where* $\preceq$ *is the Loewner order.*

We define the following general notions of progress measure and overlap. Loosely speaking the progress measure gives a bound on the improvement gained after a query, and the progress overlap represents how the query interacts with the part of the state that did not satisfy some property. We will utilize these notions to derive a bound on the increase in success probability of an adversary after performing query.

Definition 9 (Progress Measure and Progress Overlap [17]). *Given a state* $|\phi\rangle$*, a real* $b \in [0, 1]$ *and a projector* Π *over* $\mathcal{AHD}$*, we define progress measure* (Δ_b) *and progress overlap* (Γ_b) *as*

$$\Delta_b(\Pi, |\phi\rangle) = \|\Pi\mathcal{R}_b |\phi\rangle\|^2 - \|\Pi |\phi\rangle\|^2 \quad \text{and} \quad \Gamma_b(\Pi, |\phi\rangle) = \frac{\|\Pi\mathcal{R}_b(\mathbb{I} - \Pi) |\phi\rangle\|^2}{\|(\mathbb{I} - \Pi) |\phi\rangle\|^2},$$

with the convention that $\Gamma_b(\Pi, |\phi\rangle) = 0$ *if* $\|(\mathbb{I} - \Pi) |\phi\rangle\| = 0$*.*

In this paper we work only with History-Database predicates. This is an analogue of the notion "database property" introduced in [6, 7]. We want the predicate to be satisfied only on the pairs (H, D) that can be obtained through oracle interaction. According to Proposition 5, every such pair is history-database consistent. Next, we want that the order of the history database inputs does not affect the predicate mapping. Lastly, we want that adding new values to the quantum database (making queries on inputs that were not in the quantum database) should not turn a satisfied predicate into unsatisfied.

Definition 10 (History-Database Predicate [17]). *Let* $\mathrm{P} : (H, D) \mapsto \{False, True\}$ *be a predicate function over all history-database pairs. We say that it is a history-database predicate if for every true-pair* $(H, D) \in \mathrm{P}^{-1}(True)$*,*

- *(Consistent) The pair* (H, D) *is history-database consistent (see Definition 4).*
- *(History Invariant) For every list* H' *such that* (H', D) *is history-database consistent and* $H(x') = H'(x')$ *for all* $x' \in [X]$*, we have* $(H', D) \in \mathrm{P}^{-1}(True)$*.*
- *(Database Monotone) For every database* D' *that is obtained by replacing a* $\perp$ *in* D *with another value (i.e.* $D = D'_{x'\leftarrow\perp}$ *for some* $x' \in [X]$ *), we have* $(H, D') \in \mathrm{P}^{-1}(True)$*.*

By extension, we say that $\mathrm{P} : (x, y, z, H, D) \mapsto \{False, True\}$ *is a history-database predicate if it does not depend on* (x, y, w) *and its restriction to* (H, D) *satisfies the above properties.*

The next lemmas are used to bound the progress overlap that an algorithm can gain after a query. To bound this, we want to use the probability that the database will turn into one that satisfies

the predicate. While the analysis of the quantum query case is similar to the analysis in [7, 27], the classical query analysis was introduced in [17]. The restriction for the classical query bound is that the database can not turn into a satisfying one by adding an existing input from the quantum database into the history register. This restriction is reasonable when there is no logical distinction between the two databases. For example, in [17], a collision finding problem was analyzed. The hybrid compressed random oracle model allowed a fine-grained analysis of an algorithm that uses both quantum and classical queries. In our work, the classical queries serve the purpose of challenge definition. Hence, having an input in the quantum database is not the same as having an input in the history register. In Section 4, we extend the framework by giving a bound on progress overlap that is more suitable for such scenarios.

Lemma 11 (Progress Overlap, Quantum Query [17]). *Let* P *be a history-database predicate,* t *be an integer, and* $\gamma \in [0,1]$ *be a real parameter. Suppose that, for every false-state* $(H,D) \in \mathrm{P}^{-1}(\mathrm{False}) \cap \mathbb{H}_t$ *where* $D(x) = \perp$*, the probability to make the predicate true by replacing* $D(x)$ *with a random value* p *is at most*

$$\Pr_{p \leftarrow [N]}\left[(H, D_{x \leftarrow p}) \in \mathrm{P}^{-1}(\mathit{True})\right] \leq \gamma.$$

Then, the quantum progress overlap is at most $\Gamma_0(\Pi_P, |\phi\rangle) \leq 10\gamma$ *for all* $|\phi\rangle \in \mathbb{H}_t$*.*

Lemma 12 (Progress Overlap, Classical Query [17]). *Let* P *be a history-database predicate,* t *be an integer, and* $\gamma \in [0,1]$ *be a real parameter. Suppose that, for every false-state* $(H,D) \in \mathrm{P}^{-1}(\mathrm{False}) \cap \mathbb{H}_t$ *where* $D(x) = \perp$*, the probability to make the predicate true by replacing* $H(x)$ *and* $D(x)$ *with the same random value* p *is at most*

$$\Pr_{p \leftarrow [N]}\left[(H_{x \leftarrow p}, D_{x \leftarrow p}) \in \mathrm{P}^{-1}(\mathrm{True})\right] \leq \gamma.$$

Assume further that, for every false-state $(H,D) \in \mathrm{P}^{-1}(\mathit{False})$*, the predicate does not become true when* $(x, D(x))$ *is appended to the history, i.e.*

$$(H,D) \in \mathrm{P}^{-1}(\mathrm{False}) \quad \Rightarrow \quad (H_{x \leftarrow D(x)}, D) \in \mathrm{P}^{-1}(\mathrm{False})$$

Then, the classical progress overlap is at most $\Gamma_1(\Pi_{\mathrm{P}}, |\phi\rangle) \leq 2\gamma$ *for all* $|\phi\rangle \in \mathbb{H}_t$*.*

Note that γ will often depend on the number of queries that have been performed (equivalently, the maximum number t of values contained in the database and in the history). If Lemma 11 and Lemma 12 hold with parameters γ_0 and γ_1 respectively, then the combined progress can be written as $\Gamma_b(\Pi_P, |\phi\rangle) \leq 10(1-b)\gamma_0 + 2b\gamma_1$.

4 Progress overlap with partially random inputs

As we discussed before, we want to analyze properties that use classical queries as challenge setting. An example can be target collision resistance: the adversary makes a classical query with a message m and gets $\{k, F(k,m)\}$ in response. Here the k is chosen uniformly at random for each query. Then, the adversary is asked to find a collision for one of the messages used in the classical queries. This example shows a typical structure: an adversary is given quantum access to the function, but to get information about the challenges, the adversary must make classical queries to an oracle. Note that these classical queries have a part of the input that is not controlled by the adversary and is usually uniformly random. To address these types of properties, we extend the framework. We bound the probability of success to pulling a value from the quantum database into the history register, which allows us to analyze a wider range of queries.

Lemma 13 (Progress Overlap, Classical Query with randomness). *Let* P *be a history-database predicate,* t *be an integer, and* $\gamma \in [0,1]$ *be a real parameter. Suppose that, for every false-state* $(H,D) \in \mathrm{P}^{-1}(\mathrm{False}) \cap \mathbb{H}_t$ *where* $D(x) = \perp$*, the probability to make the predicate true by replacing* $H(x)$ *and* $D(x)$ *with the same random value* p *is at most*

$$\Pr_{p\leftarrow[N]}\left[(H_{x\leftarrow p}, D_{x\leftarrow p} \in \mathrm{P}^{-1}(\mathrm{True})\right] \leq \gamma.$$

Assume $[M'] = [K] \times [M]$ *and we can write* $x = (k,m) \in [K] \times [M]$*. Suppose that, for every false-state* $(H,D) \in \mathrm{P}^{-1}(\mathrm{False}) \cap \mathbb{H}_t$ *the probability to make the predicate true by adding a value* $((k,m), D(k,m))$ *to the history database, where* k *is chosen uniformly at random, is at most*

$$\Pr_{k\leftarrow[K]}\left[(H_{(k,m)\leftarrow D(k,m)}, D) \in \mathrm{P}^{-1}(\mathrm{True})\right] \leq \varepsilon.$$

Then, the classical progress overlap with a partially random input is at most $\Gamma_1(\Pi_\mathrm{P}, |\phi\rangle) \leq 3\gamma + 2\varepsilon$ *for all* $|\phi\rangle \in \mathbb{H}_t$*.*

Proof. Let $\Pi_{\overline{\mathrm{P}}} |\phi\rangle = \sum_{x,y,z,H,D} \alpha_{x,y,z,H,D} |x,y,z\rangle |H,D\rangle \in \mathbb{H}_t \cap \mathrm{supp}(\Pi_{\overline{\mathrm{P}}})$ be any state supported over consistent basis-states evaluating the predicate P to false. We show that, after making a classical query, the probability of satisfying P is at most $\| \Pi_\mathrm{P} \mathcal{R}_1 \Pi_{\overline{\mathrm{P}}} |\phi\rangle \|^2 \leq (3\gamma + 2\varepsilon) \cdot \| \Pi_{\overline{\mathrm{P}}} |\phi\rangle \|^2$. We define three projections Π_1, Π_2, Π_3 such that $\Pi_1 + \Pi_2 + \Pi_3 = \Pi_{\overline{\mathrm{P}}}$.

- Π_1 : all basis states $|x,y,z,H,D\rangle \in \mathbb{H}_t \cap \mathrm{supp}(\Pi_{\overline{\mathrm{P}}})$ such that $H(x) = \star$ and $D(x) = \perp$.
- Π_2 : all basis states $|x,y,z,H,D\rangle \in \mathbb{H}_t \cap \mathrm{supp}(\Pi_{\overline{\mathrm{P}}})$ such that $H(x) = \star$ and $D(x) \neq \perp$.
- Π_3 : all basis states $|x,y,z,H,D\rangle \in \mathbb{H}_t \cap \mathrm{supp}(\Pi_{\overline{\mathrm{P}}})$ such that $H(x) \neq \star$.

Below, we prove the (in)equalities

- $\| \Pi_\mathrm{P} \mathcal{R}_1 \Pi_1 |\phi\rangle \|^2 \leq \gamma \| \Pi_1 |\phi\rangle \|^2$
- $\| \Pi_\mathrm{P} \mathcal{R}_1 \Pi_2 |\phi\rangle \|^2 \leq 2(\gamma + \varepsilon) \| \Pi_2 |\phi\rangle \|^2$
- $\| \Pi_\mathrm{P} \mathcal{R}_1 \Pi_3 |\phi\rangle \| = 0$

Hence, by the triangle inequality and Cauchy-Schwarz inequality, we conclude that

$$\begin{aligned}\| \Pi_\mathrm{P} \mathcal{R}_1 \Pi_{\overline{\mathrm{P}}} |\phi\rangle \|^2 &\leq (\| \Pi_\mathrm{P} \mathcal{R}_1 \Pi_1 |\phi\rangle \| + \| \Pi_\mathrm{P} \mathcal{R}_1 \Pi_2 |\phi\rangle \| + \| \Pi_\mathrm{P} \mathcal{R}_1 \Pi_3 |\phi\rangle \|)^2 \\ &\leq (3\gamma + 2\varepsilon) \| \Pi_{\overline{\mathrm{P}}} |\phi\rangle \|^2.\end{aligned}$$

Analysis of Π_1. The analysis of this case matches the case for Lemma 12. We present it below.

$$\begin{aligned}&\| \Pi_\mathrm{P} \mathcal{R}_1 \Pi_1 |\phi\rangle \|^2 \\ &= \| \Pi_\mathrm{P} \mathcal{R}_1 \sum_{\substack{x,y,z,H,D: \\ H(x)=\star, D(x)=\perp}} \alpha_{x,y,z,H,D} |x,y,z\rangle |H,D\rangle \|^2 \\ &= \| \Pi_\mathrm{P} \sum_{\substack{x,y,z,H,D: \\ H(x)=\star, D(x)=\perp}} \alpha_{x,y,z,H,D} |x,y,z1\rangle \Big(\sum_{p\in[N]} \frac{\omega^{yp}}{\sqrt{N}} |H_{x\leftarrow p}, D_{x\leftarrow p}\rangle \Big) \|^2 \\ &= \sum_{\substack{x,y,z,H,D: \\ H(x)=\star, D(x)=\perp}} \| \alpha_{x,y,z,H,D} \|^2 \cdot \Pr_{p\leftarrow[N]}\left[(H_{x\leftarrow p}, D_{x\leftarrow p}) \in \mathrm{P}^{-1}(\mathrm{True})\right] \\ &\leq \gamma \| \Pi_1 |\phi\rangle \|^2\end{aligned}$$

The first line is by definition of Π_1. The second line is by Lemma 7. The third line uses the orthogonality of the basis states. Finally, the last line is the statement of Lemma 13.

Analysis of Π_2. This is the main difference compared to Lemma 12. The projection Π_2 corresponds to all basis states $|x, y, z, H, D\rangle \in \mathbb{H}_t \cap \mathrm{supp}(\Pi_{\overline{\mathrm{P}}})$ such that $H(x) = \star$ and $D(x) \neq \perp$. According to Lemma 7, we have the following result of the classical query:

$$\mathcal{R}_1 |x, y, z\rangle |H, D\rangle = |x, y, z1\rangle \left(\omega^{yD(x)} |H_{x\leftarrow D(x)}, D\rangle + \frac{1}{\sqrt{N}} |H_{x\leftarrow\perp}, D_{x\leftarrow\perp}\rangle - \sum_{p\in[N]} \frac{\omega^{yp}}{N} |H_{x\leftarrow p}, D_{x\leftarrow p}\rangle\right)$$

So, we have three terms that correspond to three possible scenarios:

1. $D(x)$ remains unchanged in the database: $|H_{x\leftarrow D(x)}, D\rangle$.
2. $D(x)$ gets removed: $|H_{x\leftarrow\perp}, D_{x\leftarrow\perp}\rangle$.
3. $D(x)$ is resampled to a new value p: $|H_{x\leftarrow p}, D_{x\leftarrow p}\rangle$.

The third option can be analyzed the same way as in Lemma 12. Loosely speaking, this case is the same as adding a new input. The second option can not make the predicate turn from False to True according to the Monotone property from Definition 10. The first option is now possible. This corresponds to the case when we pull a value from a quantum database into the history register. We aim to bound it with ε.

Now, remember, since we are doing a classical query, we know that there is a part of $x = (k, m)$ that is distributed uniformly at random. Hence, we can write the initial state as

$$|\phi\rangle = \sum_{k,m,y,z,H,D} \frac{1}{\sqrt{K}} \alpha_{m,y,z,H,D} |k, m, y, z\rangle |H, D\rangle \ .$$

We proceed by applying Π_2. We need to include only the inputs that exist in the quantum database. Hence, we need $(k, \cdot) \in D$ and $D(k, m) \neq \perp$. Notice that if $D(k, m) \neq \perp$, then k is guaranteed to be in the database. So we can write

$$\Pi_2 |\phi\rangle = \sum_{\substack{k,m,y,z,H,D:\\ H(k,m)=\star, D(k,m)\neq\perp}} \frac{1}{\sqrt{K}} \alpha_{m,y,z,H,D} |k, m, y, z\rangle |H, D\rangle \ .$$

Now, let us look at how the query behaves for $\Pi_2 |\phi\rangle$.

$$\begin{aligned}
&\| \Pi_{\mathrm{P}} \mathcal{R}_1 \Pi_2 |\phi\rangle \|^2 \\
&= \| \Pi_{\mathrm{P}} \mathcal{R}_1 \sum_{\substack{k,m,y,z,H,D:\\ H(k,m)=\star, D(k,m)\neq\perp}} \frac{1}{\sqrt{K}} \alpha_{m,y,z,H,D} |k, m, y, z\rangle |H, D\rangle \|^2 \\
&= \| \Pi_{\mathrm{P}} \sum_{\substack{k,m,y,z,H,D:\\ H(k,m)=\star, D(k,m)\neq\perp}} \frac{1}{\sqrt{K}} \alpha_{m,y,z,H,D} |k, m, y, z1\rangle \cdot \\
&\quad \Big(\omega^{yD(k,m)} |H_{(k,m)\leftarrow D(k,m)}, D\rangle + \frac{1}{\sqrt{N}} |H_{(k,m)\leftarrow\perp}, D_{(k,m)\leftarrow\perp}\rangle - \\
&\quad \sum_{p\in[N]} \frac{\omega^{yp}}{N} |H_{(k,m)\leftarrow p}, D_{(k,m)\leftarrow p}\rangle\Big) \|^2 \\
&= \| \Pi_{\mathrm{P}} \sum_{\substack{k,m,y,z,H,D:\\ H(k,m)=\star, D(k,m)\neq\perp}} \frac{1}{\sqrt{K}} \alpha_{m,y,z,H,D} |k, m, y, z1\rangle \cdot \\
&\quad \Big(\omega^{yD(k,m)} |H_{(k,m)\leftarrow D(k,m)}, D\rangle - \sum_{p\in[N]} \frac{\omega^{yp}}{N} |H_{(k,m)\leftarrow p}, D_{(k,m)\leftarrow p}\rangle\Big) \|^2 \leq
\end{aligned}$$

$$\leq 2\|\Pi_{\mathrm{P}} \sum_{\substack{k,m,y,z,H,D:\\ H(k,m)=\star, D(k,m)\neq\perp}} \frac{1}{\sqrt{K}}\alpha_{m,y,z,H,D}\left((\omega^{yD(k,m)} - \frac{\omega^{yD(k,m)}}{N})\,|x,y,z1\rangle\,|H_{(k,m)\leftarrow D(k,m)}, D\rangle\right)\|^2$$
$$+ 2\|\Pi_{\mathrm{P}} \sum_{\substack{k,m,y,z,H,D:\\ H(k,m)=\star, D(k,m)\neq\perp}} \frac{1}{\sqrt{K}}\alpha_{m,y,z,H,D}\left(\sum_{p\in[N]/D(k,m)} \frac{\omega^{yp}}{N}\,|x,y,z1\rangle\,|H_{(k,m)\leftarrow p}, D_{(k,m)\leftarrow p}\rangle\right)\|^2$$

The first equality is by definition of Π_2. The second equation is by Lemma 7. The third equation is due to the fact that $|H_{x\leftarrow\perp}, D_{x\leftarrow\perp}\rangle$ can not make the predicate turn from False to True. The last inequality is based on the triangle inequality $(a+b)^2 \leq 2a^2 + 2b^2$.

Let us analyze the last two terms separately. We start with the second one since it has already been analyzed for Lemma 12. Following the reasoning from [17, Lemma 4.13] we get

$$\|\Pi_{\mathrm{P}} \sum_{\substack{k,m,y,z,H,D:\\ H(k,m)=\star, D(k,m)\neq\perp}} \frac{1}{\sqrt{K}}\alpha_{m,y,z,H,D}\left(\sum_{p\in[N]/D(k,m)} \frac{\omega^{yp}}{N}\,|x,y,z1\rangle\,|H_{(k,m)\leftarrow p}, D_{(k,m)\leftarrow p}\rangle\right)\|^2$$
$$\leq \gamma\|\Pi_2\,|\phi\rangle\,\|^2$$

Now, we need to prove that

$$\|\Pi_{\mathrm{P}} \sum_{\substack{k,m,y,z,H,D:\\ H(k,m)=\star, D(k,m)\neq\perp}} \frac{1}{\sqrt{K}}\alpha_{m,y,z,H,D}((\omega^{yD(k,m)} - \frac{\omega^{yD(k,m)}}{N})\,|x,y,z1\rangle\,|H_{(k,m)\leftarrow D(k,m)}, D\rangle)\|^2$$
$$\leq \varepsilon\|\Pi_2\,|\phi\rangle\,\|^2$$

To do so, let us first remove the subtraction from the coefficient.

$$\|\Pi_{\mathrm{P}} \sum_{\substack{k,m,y,z,H,D:\\ H(k,m)=\star, D(k,m)\neq\perp}} \frac{1}{\sqrt{K}}\alpha_{m,y,z,H,D}((\omega^{yD(k,m)} - \frac{\omega^{yD(k,m)}}{N})\,|x,y,z1\rangle\,|H_{(k,m)\leftarrow D(k,m)}, D\rangle)\|^2$$
$$\leq \|\Pi_{\mathrm{P}} \sum_{\substack{k,m,y,z,H,D:\\ H(k,m)=\star,\\ D(k,m)\neq\perp}} \alpha_{m,y,z,H,D}(\frac{1}{\sqrt{K}}\omega^{yD(k,m)})\,|x,y,z1\rangle\,|H_{(k,m)\leftarrow D(k,m)}, D\rangle\,\|^2$$

This holds as we can take a factor $\left(1-\frac{1}{N}\right)$ out of the norm expression. Since we started with the state $\sum_{\substack{x,y,z,H,D:\\ H(x)=\star, D(x)\neq\perp}} \alpha_{x,y,z,H,D}\,|x,y,z\rangle\,|H,D\rangle$ where every term represents an orthogonal state,

adding a value to the history databases does not change orthogonality. This is because if we have a history database of the same size, then we are adding different values, and the order of the inputs in the history database matters for orthogonality; hence, we get orthogonal vectors. If the history database has different sizes, we have different vectors z. Hence, every term in the result represents an orthogonal vector. Note that Π_{P} does not spoil the orthogonality since it is diagonal in the computational basis. Hence, we get:

$$\begin{aligned}
&\|\Pi_{\mathrm{P}} \sum_{\substack{k,m,y,z,H,D:\\ H(k,m)=\star, D(k,m)\neq\perp}} \alpha_{m,y,z,H,D}\left(\frac{1}{\sqrt{K}}\omega^{yD(k,m)}\right)|x,y,z1\rangle\,|H_{(k,m)\leftarrow D(k,m)},D\rangle\,\|^2 \\
&\leq \sum_{\substack{k,m,y,z,H,D:\\ H(k,m)=\star, D(k,m)\neq\perp}} \|\alpha_{m,y,z,H,D}\|^2 \cdot \Pr_{k\leftarrow[K]}\left[(H_{(k,m)\leftarrow D(k,m)},D)\in \mathrm{P}^{-1}(\mathrm{True})\right] \\
&\leq \varepsilon\|\Pi_2\,|\phi\rangle\,\|^2
\end{aligned}$$

Analysis of Π_3. By Lemma 7 the operator $\mathcal{R}_1$ maps any state $|x,y,z\rangle\,|H,D\rangle \in \mathrm{supp}(\Pi_3)$ to $\omega^{yD(x)}\,|x,y,z1\rangle\,|H_{x\leftarrow D(x)},D\rangle$ since $H(x)\neq\star$. Moreover, H and $H_{x\leftarrow D(x)}$ have the same function representation (since the initial state is history-database consistent). Thus, by the history invariant property (see Definition 10), we have $\mathrm{P}(H_{x\leftarrow D(x)},D)=False$ and $\|\Pi_{\mathrm{P}}\mathcal{R}_1\Pi_3|H\,|\phi\rangle\,\|=0$.

This concludes the proof. □

5 The security of M-ETCR

In this section, we show how to apply the improved framework. We choose the M-ETCR notion (see Definition 14) as a good example to illustrate the utility of our technique.

We start with a definition of the keyed hash function family. In the following let $N\in\mathbb{N}$ be the security parameter, and $\mathcal{F}_N=\{F_k:[M]\to[N]\}_{k\in[K]}$ be a keyed family of hash functions. We will also write $F(k,m)$ to denote $F_k(m)$. We define the Multi-target extended target collision resistance (M-ETCR) property for a keyed hash function family.

Definition 14 (Multi-target extended target collision resistance [18]). *Let $N\in\mathbb{N}$, and $\mathcal{F}_N$ be a hash function family, with key space $[K]$, message space $[M]$, and output space $[N]$. Let $\mathcal{A}$ be a (stateful) algorithm, and $c\in poly(N)$. Consider the following experiment M-ETCR$_{\mathcal{F}_N,c}(\mathcal{A})$:*

1. *Set a list of queries to the challenge oracle to empty $Q=\varnothing$.*
2. *Run $\mathcal{A}(\mathcal{F}_N)$ with (classical) access to an oracle $\mathcal{O}(\cdot)$ that takes a message $m\in[M]$ and works as follows:*
   - *If $|Q|\geq c$ return $\perp$.*
   - *Sample $k\leftarrow_{\$}[K]$.*
   - *Compute $F(k,m)=y$.*
   - *Set $Q=Q\cup\{k,m\}$.*
   - *Output (k,m,y).*
3. *Obtain from $\mathcal{A}$ an output $(k_j,m_j),(k^*,m^*)$ with $j\in[|Q|],k^*\in[K],m^*\in[M]$, and where the jth entry in Q is (k_j,m_j).*
4. *Output 1 if $F(k^*,m^*)=F(k_j,m_j)$ and $m^*\neq m_j$. Otherwise, output 0.*

For any such algorithm $\mathcal{A}$, we define the following success probability:

$$\mathrm{Succ}^{\text{M-ETCR}}_{\mathcal{H}_n,c}(\mathcal{A})=\Pr\left[1\leftarrow_{\$}\mathcal{A}^{\mathcal{O}(\cdot)}(\mathcal{F}_N)\right]$$

For M-ETCR, the adversary is allowed to obtain up to c challenges for a given hash function family. The challenges are generated for a random key. The best-known bound for the M-ETCR in the QROM was obtained in [14]:

$$\mathrm{Succ}^{\text{M-ETCR}}_{\mathcal{F}_N,c}(\mathcal{A}) \leq \frac{8c(c+q+2)^2}{N} + \frac{3c}{2}\sqrt{\frac{q+c+1}{K}},$$

where q is the number of (quantum) queries to $F(\cdot,\cdot)$.

The second term in this inequality comes from the reprogramming the random oracle. The main idea is that previously, it was hard to deal with a mix of classical and quantum queries. One of the possible approaches was to learn the outputs of all the challenge queries before any quantum query happens. This was done by choosing all the responses uniformly at random and then reprogramming them into the hash function as a challenge query happens. Intuitively, the reprogramming was not needed to alter the response but rather to move the challenge queries to the very beginning. We can overcome this difficulty and get a tight bound with the new technique. We get the following theorem:

Theorem 15. *The success probability of an algorithm $\mathcal{A}$, that makes at most q quantum queries to the function family $\mathcal{F}_N$ and c classical challenge queries and outputs a solution of M-ETCR is at most:*

$$\mathrm{Succ}^{\text{M-ETCR}}_{\mathcal{F}_N,c}(\mathcal{A}) \leq 44\left(\frac{cq+c^2}{N} + q\sqrt{\frac{c}{N}} + \left(\frac{cq^{3/2}+c^{3/2}}{KN^{1/2}}\right) + \frac{c}{K}\frac{q^3}{N}\right) + \frac{4}{N}$$

$$= O\left(\frac{c^2}{N} + \frac{cq^2}{N} + \frac{q^3\cdot c}{K\cdot N}\right),$$

Before we proceed to the proof, let us discuss this result. First, we highlight that the bound is tight on the number of queries up to a constant factor as demonstrated by matching attacks, detailed below.

The first term $\frac{cq+c^2}{N}$ comes from a probability that we get either a collision in the challenge queries or we make a challenge query, and it collides with an input that is already in the quantum database. Next, $q\sqrt{\frac{c}{N}}$ is obtained by setting c classical targets and searching for a solution using quantum queries (similar to Grover search [15]). The last terms come from the following attack strategy: First, find a collision for some message m and different keys. Then do challenge queries with this message and hope that the sampled key will match one of your prepared collisions. The terms $\sqrt{\frac{q^3}{N}}$ and $\frac{q^3}{N}$ give a bound on number of queries for finding collisions, and there is a chance of $\frac{c}{K}$ that one of the challenge queries will hit the needed key.

We have the following implications from the new bound compared to the previous one:

1. The number of allowed challenge queries now influence N as c^2 instead of c^3 compared to the previous bound. This is important because we may allow the number of challenges as big as 2^{64} in some applications. For example, in the NIST post-quantum standardization process, the security bound for signature schemes was required to assume that the attacker has access to signatures for up to 2^{64} chosen messages [24, Section 4].
2. Previously, K, the size of the randomness space, had to match the bound $K \geq c^2\cdot(q+c)$. So, for example, for $q = c = 2^{64}$ (which matches the requirements of category one in the NIST call [24, Section 4]), K would have to be greater than 2^{192}. Now we bound K as $K \geq 44(\sqrt{cq}+c+q)$ (assuming $N = q^2\cdot c$). This will us to use $K \geq 2^{72}$ for the same security level. Hence, we can use keys almost 2/3 smaller than with the previous bound.

Before proving Theorem 15, we need to define different predicates on basis states and different types of collisions.

Definition 16 (Collision Type). *Given a history-database consistent pair (H, D), we say that it contains a collision if there exist two values $x_1 \neq x_2$ such that $D(x_1) = D(x_2) \neq \perp$. Additionally, if $x_1, x_2 \notin H$, the collision is said to be quantum, if $x_1, x_2 \in H$, it is said to be classical, and if $x_1 \notin H, x_2 \in H$ (or $x_2 \notin H, x_1 \in H$), it is said to be hybrid.*

Given the three types of collisions, we define the corresponding predicates. We also define a k-collision predicate for the quantum database. The kQ predicate represents our ability to track how many times a query formed a new collision. Each new query that formed a collision can either collide with an existing collision or create a new one. In the first case the number of inputs in the database for which there exists a collision increases by one, and in the second case by two.

Definition 17. *The following predicates evaluate a basis state $|x, y, w, H, D\rangle$ to True if and only if it is history-database consistent (see Definition 4) and satisfies the next conditions:*

- Q, H, C*: there is respectively at least one quantum, one hybrid, or one classical collision contained in (H, D).*
- kQ *- There are $k \leq l \leq 2k$ distinct inputs $x_1, \ldots, x_l$, such that there exists a quantum collision for each of them: $x_i \neq x_j$, $i \neq j$ $i, j \in [1, l]$; $\forall x_i \ \exists x_j : D(x_i) = D(x_j) \neq \perp$, x_i, $x_j \notin H$, $i, j \in [1, l]$.*

The proof of Theorem 15 relies on the results of Lemma 18 and Lemma 20. These lemmas limit the progress that can be achieved with quantum and classical queries. Using these results we bound the success probability of the whole algorithm.

Proof (of Theorem 15). Before analyzing M-ETCR, we need to reflect that working with the CRO introduces a slight disturbance. According to Corollary 1, working with the CRO introduces an error that is dependent on the number of the output values of the algorithm. The output of the algorithm for M-ETCR consists of 2 values. Hence, we can limit the error with an additive term of $\frac{4}{N}$.

First, let us observe that a solution for M-ETCR results in a database that either contains a classical collision or a hybrid one. Let $|\phi_t\rangle = |x, y, w, H, D\rangle$ denote a state that is obtained after t queries, where q queries are quantum and c classical ($t = q + c$). Hence, we need to bound $\|\Pi_{\mathrm{H+C}} |\phi_t\rangle\|^2$. It will also be convenient to keep track of the following progress measure.

$$\Phi_t = \|\Pi_{\mathrm{C}} |\phi_t\rangle\|^2 + \|\Pi_{\mathrm{H}\overline{\mathrm{C}}} |\phi_t\rangle\|^2$$

Observe that $\|\Pi_{\mathrm{H+C}} |\phi_t\rangle\|^2 = \Phi_t$. We claim the following recurrence holds for the potential Φ_t if the t-th query is made to the oracle $\mathcal{R}_b$, $b \in \{0, 1\}$.

$$\Phi_t = \Phi_{t-1} + (1-b)(\Delta_b(\Pi_{\mathrm{C}}, |\phi_{t-1}\rangle) + \Delta_b(\Pi_{\mathrm{H}\overline{\mathrm{C}}}, |\phi_{t-1}\rangle)) + b(\Delta_b(\Pi_{\mathrm{H+C}}, |\phi_{t-1}\rangle))$$

From the initial condition $\Phi_0 = 0$ and Lemma 18, Lemma 20 we get:

$$\Phi_t \leq \frac{10cq}{N} + 3q\sqrt{\frac{10c}{N}} + \frac{2(q+c)c}{N} + 12\frac{c}{K}\sqrt{\frac{q^3}{N}} + 44\frac{c}{K}\frac{q^3}{N} + 8(\frac{c\sqrt{q+c}}{K\sqrt{N}} + \frac{c(q+c)}{KN^{3/2}})$$

$$\leq \frac{20cq + 10c^2}{N} + +3q\sqrt{\frac{10c}{N}} + 12\frac{c}{K}\sqrt{\frac{q^3}{N}} + 44\frac{c}{K}\frac{q^3}{N} + 8\frac{c\sqrt{q+c}}{K\sqrt{N}}$$

$$\leq 20\frac{cq + c^2}{N} + +12q\sqrt{\frac{c}{N}} + 20\frac{cq^{3/2} + c^{3/2}}{KN^{1/2}} + 44\frac{cq^3}{KN}$$

□

Lemma 18 (Progress Measure, Quantum Query). *Given an integer $t = q + c$, where q is the number of quantum queries and c is the number of classical queries, and a state $|\phi\rangle \in \mathbb{H}_t$ with the norm at most 1, the progress made by one quantum query of ϕ satisfies:*

$$\Delta_0(\Pi_{\mathrm{C}}, |\phi\rangle) = 0$$
$$\Delta_0(\Pi_{\mathrm{H}\overline{\mathrm{C}}}, |\phi\rangle) \leq \frac{10c}{N} + 2\sqrt{\frac{10c}{N}}$$

Proof. $\Delta_0(\Pi_{\mathrm{C}}, |\phi\rangle) = 0$ comes from a simple observation that quantum query does not affect the History part of the database (see Lemma 6). Hence, we are left to prove $\Delta_0(\Pi_{\mathrm{H}\overline{\mathrm{C}}}, |\phi\rangle) \leq \frac{10c}{N} + 2\sqrt{\frac{10c}{N}}$. To do so first lets expand $\Delta_0(\Pi_{\mathrm{H}\overline{\mathrm{C}}}, |\phi\rangle)$:

$$\begin{aligned}
\Delta_0(\Pi_{\mathrm{H}\overline{\mathrm{C}}}, |\phi\rangle) &= \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 |\phi\rangle \|^2 - \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} |\phi\rangle \|^2 \\
&= \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 (\Pi_{\mathrm{H}\overline{\mathrm{C}}} + \Pi_{\overline{\mathrm{H}}+\mathrm{C}}) |\phi\rangle \|^2 - \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} |\phi\rangle \|^2 \\
&\leq (\| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 \Pi_{\mathrm{H}\overline{\mathrm{C}}} |\phi\rangle \| + \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 \Pi_{\overline{\mathrm{H}}+\mathrm{C}} |\phi\rangle \|)^2 - \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} |\phi\rangle \|^2 \\
&= \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 \Pi_{\mathrm{H}\overline{\mathrm{C}}} |\phi\rangle \|^2 + 2 \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 \Pi_{\mathrm{H}\overline{\mathrm{C}}} |\phi\rangle \| \cdot \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 \Pi_{\overline{\mathrm{H}}+\mathrm{C}} |\phi\rangle \| \\
&\qquad + \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 \Pi_{\overline{\mathrm{H}}+\mathrm{C}} |\phi\rangle \|^2 - \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} |\phi\rangle \|^2 \\
&\leq 2 \cdot \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 \Pi_{\overline{\mathrm{H}}+\mathrm{C}} |\phi\rangle \| + \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 \Pi_{\overline{\mathrm{H}}+\mathrm{C}} |\phi\rangle \|^2
\end{aligned}$$

The first equation comes from the definition of Δ_0. Next we use that $\mathbb{I} = (\Pi_{\mathrm{H}\overline{\mathrm{C}}} + \Pi_{\overline{\mathrm{H}}+\mathrm{C}})$. In the third inequality, we use triangle inequality.The equality in line 4 is obtained by opening the brackets. For the last inequality we use $\| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 \Pi_{\mathrm{H}\overline{\mathrm{C}}} |\phi\rangle \|^2 \leq \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} |\phi\rangle \|^2$.

The last step is to bound $\| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 \Pi_{\overline{\mathrm{H}}+\mathrm{C}} |\phi\rangle \|$ we look at $\Gamma_0(\Pi_{\mathrm{H}\overline{\mathrm{C}}}, |\phi\rangle)$. According to Lemma 11 we have:

$$\begin{aligned}
&\Gamma_0(\Pi_{\mathrm{H}\overline{\mathrm{C}}}, |\phi\rangle) = \frac{\| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 (\mathbb{I} - \Pi_{\mathrm{H}\overline{\mathrm{C}}}) |\phi\rangle \|^2}{\| (\mathbb{I} - \Pi_{\mathrm{H}\overline{\mathrm{C}}}) |\phi\rangle \|^2} \leq 10\gamma \\
&\| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 (\mathbb{I} - \Pi_{\mathrm{H}\overline{\mathrm{C}}}) |\phi\rangle \|^2 = \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 (\Pi_{\overline{\mathrm{H}}+\mathrm{C}}) |\phi\rangle \|^2 \\
&= \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 (\Pi_{\overline{\mathrm{H}\overline{\mathrm{C}}}}) |\phi\rangle \|^2 \leq 10\gamma \cdot \| \Pi_{\overline{\mathrm{H}}+\mathrm{C}} |\phi\rangle \|^2 \leq 10\gamma = \frac{10c}{N},
\end{aligned}$$

where

$$\gamma = \Pr_{y \leftarrow [N]} \left[(H, D_{x \leftarrow y}) \in \mathrm{H}\overline{\mathrm{C}} \right] \leq \frac{c}{N},$$

for a $(D, H) \in \overline{\mathrm{HC}}$.

Here the equality $\| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 (\Pi_{\overline{\mathrm{H}}+\mathrm{C}}) |\phi\rangle \|^2 = \| \Pi_{\mathrm{H}\overline{\mathrm{C}}} \mathcal{R}_0 (\Pi_{\overline{\mathrm{H}\overline{\mathrm{C}}}}) |\phi\rangle \|^2$ comes from the fact that a quantum query can not turn the predicate C from True to False. Hence, we do not need to take into consideration the databases that satisfy the C predicate; after a quantum query, they will still satisfy it. □

Before analyzing the classical case, let us present an intermediate result.

Lemma 19 ($\Pi_{\mathrm{H}+\mathrm{C}}$ classical progress overlap). *Given an integer $t = q + c$, where q is the number of quantum queries and c is the number of classical queries, and a state $|\phi\rangle \in \mathbb{H}_t$ with the norm at most 1, the progress overlap obtained through one classical query on ϕ satisfies:*

$$\Gamma_1(\Pi_{\mathrm{H}+\mathrm{C}}, |\phi\rangle) \leq (3\gamma + 2\varepsilon),$$

where $\gamma \leq (q + c)/N$ and $\varepsilon \leq \frac{6}{K}\sqrt{\frac{q^3}{N}} + \frac{22q^3}{K \cdot N}$.

Proof. According to Lemma 13 we have:

$$\begin{aligned}
&\Gamma_1(\Pi_{\mathrm{H+C}}, |\phi\rangle) = \frac{\|\Pi_{\mathrm{H+C}}\mathcal{R}_1(\mathbb{I}-\Pi_{\mathrm{H+C}})\,|\phi\rangle\,\|^2}{\|(\mathbb{I}-\Pi_{\mathrm{H+C}})\,|\phi\rangle\,\|^2} \leq 3\gamma + 2\varepsilon \\
&\|\Pi_{\mathrm{H+C}}\mathcal{R}_1(\mathbb{I}-\Pi_{\mathrm{H+C}})\,|\phi\rangle\,\|^2 = \|\Pi_{\mathrm{H+C}}\mathcal{R}_1\Pi_{\overline{\mathrm{H}}\cdot\overline{\mathrm{C}}}\,|\phi\rangle\,\|^2 \\
&\leq (3\gamma+2\varepsilon)\cdot\|\Pi_{\overline{\mathrm{H}}\cdot\overline{\mathrm{C}}}\,|\phi\rangle\,\|^2 \leq (3\gamma+2\varepsilon),
\end{aligned}$$

where

$$\gamma = \Pr_{y\leftarrow[N]}\left[(H_{x\leftarrow y}, D_{x\leftarrow y}) \in (\mathrm{H}+\mathrm{C})\right] \leq \frac{q+c}{N}$$

for false-state $(H, D) \in \overline{\mathrm{H}}\cdot\overline{\mathrm{C}} \cap \mathbb{H}_t$ where $D(x) = \perp$; and

$$\varepsilon = \Pr_{k\leftarrow[K]}\left[(H_{(k,m)\leftarrow D(k,m)}, D) \in (\mathrm{H}+\mathrm{C})\right],$$

for false-state $(H, D) \in \overline{\mathrm{H}}\cdot\overline{\mathrm{C}} \cap \mathbb{H}_t$.

Note that ε depends on the state of the quantum part of the database. We need to analyze when adding a value from D turns $\overline{\mathrm{H}}\cdot\overline{\mathrm{C}}$ into $(\mathrm{H}+\mathrm{C})$. In this case, we get either a hybrid collision or a classical one. Assume we move a value from D to H and get a classical collision. This means that before, we had a collision between D and H, which is excluded (we start from $\overline{\mathrm{H}}\cdot\overline{\mathrm{C}}$). Hence, the only possibility is to get a hybrid collision. If we move a value from D to H and get a hybrid collision, this means there was a collision in D before the classical query.

Our classical query contains a chosen input m and a random key k: $x = (k||m)$. Denote with j maximum number of different keys $k_1, \ldots, k_j$ for which there exists a colliding pair of the following type: $[D(k_1, m) = D(k'_1, m'_1)], \ldots, [D(k_j, m) = D(k'_j, m'_j)]$, where $(k_i, m) \neq (k'_i, m'_i)$. Note that we do not have any extra requirement for the k'_i, m'_i. If we can bound the probability that after q queries, we know j colliding inputs, then the maximum number of different keys we can obtain from these collisions is $2j$.

In [21, Section 4.3], the authors give a bound on finding j distinct collisions. However, their argument actually works by bounding the probability that a new query collides with some input that is already in the database. They do not distinguish whether it is a new collision or if we have formed a 3-collision, for example. Hence, we can use their bound and claim that $\|\Pi_{j\mathrm{Q}}\,|\phi_t\rangle\,\|^2 \leq (\frac{e\cdot q^{3/2}}{j\cdot\sqrt{N}})^j$.

Using this result, we deduce the bound on ε :

$$\begin{aligned}
\varepsilon &= \Pr_{k\leftarrow[K]}\left[(H_{(k,m)\leftarrow D(k,m)}, D) \in (\mathrm{H}+\mathrm{C})\right] \\
&\leq \sum_{j=1}^{q-1}\left(\frac{2j}{K}\cdot\|\Pi_{j\mathrm{Q}}\,|\phi_t\rangle\,\|^2\right) \leq \sum_{j=1}^{q-1}\frac{2j}{K}\cdot\left(\frac{e\cdot q^{3/2}}{j\cdot\sqrt{N}}\right)^j \\
&\leq e\cdot\frac{2}{K}\left(\frac{q^3}{N}\right)^{1/2} + \frac{2e\cdot q^{3/2}}{K\sqrt{N}}\sum_{j=2}^{q-1}\left(\frac{e\cdot q^{3/2}}{j\cdot\sqrt{N}}\right)^{j-1}.
\end{aligned}$$

We analyze the last sum separately.

$$\begin{aligned}\sum_{j=2}^{q-1}\left(\frac{e\cdot q^{3/2}}{j\cdot\sqrt{N}}\right)^{j-1} &= \sum_{j=0}^{q-3}\left(\frac{e\cdot q^{3/2}}{(j+2)\cdot\sqrt{N}}\right)^{j+1} \leq \frac{e\cdot q^{3/2}}{\sqrt{N}}\sum_{j=0}^{q-3}\left(\frac{e\cdot q^{3/2}}{(j+2)\cdot\sqrt{N}}\right)^{j}\\ &\leq \frac{e\cdot q^{3/2}}{\sqrt{N}}\sum_{j=0}^{q-3}\left(\frac{e\cdot q^{3/2}}{j\cdot\sqrt{N}}\right)^{j} \leq \frac{e\cdot q^{3/2}}{\sqrt{N}}\sum_{j=0}^{q-3}\left(\frac{e\cdot q^{3/2}}{\sqrt{N}}\right)^{j}(j!)^{-1}\\ &\leq \frac{e\cdot q^{3/2}}{\sqrt{N}}\sum_{j=0}^{\infty}\left(\frac{e\cdot q^{3/2}}{\sqrt{N}}\right)^{j}(j!)^{-1} = \frac{e\cdot q^{3/2}}{\sqrt{N}}\exp\left(\frac{e\cdot q^{3/2}}{\sqrt{N}}\right).\end{aligned}$$

For the first inequality we have used $j+2\geq 1$ for all $j\geq 0$. In the second inequality we have used $j+2\geq j$. In the third inequality we have used $j!\leq j^j$ for all $j\geq 0$ (note that by convention, $0^0=1$). In the 4. inequality we have extended the domain of the sum.

We are striving to bound the a progress overlap which is trivially upper-bonded by 1. If $\frac{e^2\cdot q^{3/2}}{\sqrt{N}}\geq 1$ then this quantity is also a trivial upper bound. Assume now $\frac{e^2\cdot q^{3/2}}{\sqrt{N}}<1$. Then $\exp\left(\frac{e\cdot q^{3/2}}{\sqrt{N}}\right)\leq \exp\left(\frac{1}{e}\right)$. In summary we get

$$\varepsilon \leq \frac{2e}{K}\left(\frac{q^3}{N}\right)^{1/2} + \exp\left(\frac{1}{e}\right)\frac{2e^2\cdot q^3}{K\cdot N} \leq \frac{6}{K}\left(\frac{q^3}{N}\right)^{1/2} + \frac{22\cdot q^3}{K\cdot N}$$

□

Lemma 20 (Progress Measure, Classical Query). *Given an integer $t=q+c$, where q is the number of quantum queries and c is the number of classical queries, and a state $|\phi\rangle\in\mathbb{H}_t$ with the norm at most 1, the progress made by one classical query of ϕ satisfies:*

$$\Delta_1(\Pi_{\mathrm{H+C}},|\phi\rangle) = \frac{3(q+c)}{N} + 2\varepsilon + 8\left(\frac{\sqrt{q+c}}{K\sqrt{N}} + \frac{q+c}{KN^{3/2}}\right),$$

where $\varepsilon\leq \frac{6}{K}\sqrt{\frac{q^3}{N}} + \frac{22\cdot q^3}{K\cdot N}$

Proof. Lets expand $\Delta_1(\Pi_{\mathrm{H+C}},|\phi\rangle)$:

$$\begin{aligned}\Delta_1(\Pi_{\mathrm{H+C}},|\phi\rangle) &= \|\Pi_{\mathrm{H+C}}\mathcal{R}_1|\phi\rangle\|^2 - \|\Pi_{\mathrm{H+C}}|\phi\rangle\|^2\\ &= \|\Pi_{\mathrm{H+C}}\mathcal{R}_1(\Pi_{\mathrm{H+C}}+\Pi_{\overline{\mathrm{H}}\cdot\overline{\mathrm{C}}})|\phi\rangle\|^2 - \|\Pi_{\mathrm{H+C}}|\phi\rangle\|^2\\ &= \|\Pi_{\mathrm{H+C}}\mathcal{R}_1\Pi_{\mathrm{H+C}}|\phi\rangle + \Pi_{\mathrm{H+C}}\mathcal{R}_1\Pi_{\overline{\mathrm{H}}\cdot\overline{\mathrm{C}}}|\phi\rangle\|^2 - \|\Pi_{\mathrm{H+C}}|\phi\rangle\|^2\\ &= \||\phi_1\rangle + |\phi_2\rangle\|^2 - \|\Pi_{\mathrm{H+C}}|\phi\rangle\|^2\\ &= \langle\phi_1|\phi_1\rangle + \langle\phi_1|\phi_2\rangle + \langle\phi_2|\phi_1\rangle + \langle\phi_2|\phi_2\rangle - \|\Pi_{\mathrm{H+C}}|\phi\rangle\|^2\\ &= \|\Pi_{\mathrm{H+C}}\mathcal{R}_1\Pi_{\mathrm{H+C}}|\phi\rangle\|^2 + \|\Pi_{\mathrm{H+C}}\mathcal{R}_1\Pi_{\overline{\mathrm{H}}\cdot\overline{\mathrm{C}}}|\phi\rangle\|^2\\ &+ \langle\phi_1|\phi_2\rangle + \langle\phi_2|\phi_1\rangle - \|\Pi_{\mathrm{H+C}}|\phi\rangle\|^2\\ &\leq \|\Pi_{\mathrm{H+C}}\mathcal{R}_1\Pi_{\overline{\mathrm{H}}\cdot\overline{\mathrm{C}}}|\phi\rangle\|^2 + \langle\phi_1|\phi_2\rangle + \langle\phi_2|\phi_1\rangle,\end{aligned}$$

where $|\phi_1\rangle = \Pi_{\mathrm{H+C}}\mathcal{R}_1\Pi_{\mathrm{H+C}}|\phi\rangle$ and $|\phi_2\rangle = \Pi_{\mathrm{H+C}}\mathcal{R}_1\Pi_{\overline{\mathrm{H}}\cdot\overline{\mathrm{C}}}|\phi\rangle$. The first equality comes from the definition. In the next equality we use that $\mathbb{I} = (\Pi_{\mathrm{H+C}}+\Pi_{\overline{\mathrm{H}}\cdot\overline{\mathrm{C}}})$. The fifth equality comes from the fact that $\|\psi\|^2 = \langle\psi|\psi\rangle$ and distributivity and associativity of the inner product. The last inequality comes from the fact $\|\Pi_{\mathrm{H+C}}\mathcal{R}_1\Pi_{\mathrm{H+C}}|\phi\rangle\|^2 \leq \|\Pi_{\mathrm{H+C}}|\phi\rangle\|^2$.

To bound $\|\Pi_{\mathrm{H+C}}\mathcal{R}_1\Pi_{\overline{\mathrm{H}}\cdot\overline{\mathrm{C}}}\,|\phi\rangle\,\|^2$ we look at $\Gamma_1(\Pi_{\mathrm{H+C}}, |\phi\rangle)$. According to our result from Lemma 19 we can deduce that $\|\Pi_{\mathrm{H+C}}\mathcal{R}_1\Pi_{\overline{\mathrm{H}}\cdot\overline{\mathrm{C}}}\,|\phi\rangle\,\|^2 \leq (3\gamma + 2\varepsilon)$, where $\gamma \leq (q+c)/N$ and $\varepsilon \leq \frac{6}{K}\sqrt{\frac{q^3}{N}} + \frac{22\cdot q^3}{K\cdot N}$.

The last step is to bound $\langle\phi_1|\phi_2\rangle + \langle\phi_2|\phi_1\rangle$. First note that $\langle\phi_1|\phi_2\rangle = \langle\phi_2|\phi_1\rangle^\dagger$. Hence, we can use that $\langle\phi_1|\phi_2\rangle + \langle\phi_2|\phi_1\rangle \leq 2|\,\langle\phi_1|\phi_2\rangle\,|$. Lets call the nonorthogonal parts of $|\phi_1\rangle$ and $|\phi_2\rangle$ as $|\mu_1\rangle$ and $|\mu_2\rangle$. Being more precise, let us define the projector Π_i with support spanned by the computational basis states $|w\rangle$ such that $\langle w|\phi_i\rangle \neq 0$, $i \in \{1,2\}$. Then $|\mu_1\rangle = \Pi_2\,|\phi_1\rangle$ and $|\mu_2\rangle = \Pi_1\,|\phi_2\rangle$. Hence,

$$|\,\langle\phi_1|\phi_2\rangle\,| \leq |\,\langle\mu_1|\mu_2\rangle\,| \leq |\|\mu_1\|\|\mu_2\|e^{i\theta}| = |\sqrt{\langle\mu_1|\mu_1\rangle}\sqrt{\langle\mu_2|\mu_2\rangle}e^{i\theta}|,$$

where θ is the angle between the two states, $|e^{i\theta}| \leq 1$.

We define $|\psi_1\rangle = \Pi_{\mathrm{H+C}}\,|\phi\rangle$ and $|\psi_2\rangle = \Pi_{\overline{\mathrm{H}}\cdot\overline{\mathrm{C}}}\,|\phi\rangle$ (the sates before the oracle queries). Below we will analyze the terms of $|\psi_1\rangle$ and $|\psi_2\rangle$. We want to deduce which terms of $|\psi_1\rangle$ and $|\psi_2\rangle$ will form the terms of nonorthogonal parts of $|\mu_1\rangle$ and $|\mu_2\rangle$. We will write $|\psi_i[k,m,y,z,H,D]\rangle$to relate to the term of $|\psi_i\rangle$ that corresponds to these parameters. When the parameters are known from the context we will write $|\psi_i[\alpha]\rangle$ to mark that we are a talking about a specific term in the state.

We can expand $|\psi_1\rangle$ as

$$|\psi_1\rangle = \sum_{k,m,y,z,H_1,D_1} \frac{1}{\sqrt{K}}\alpha_{m,y,z,H_1,D_1}\,|k,m,y,z\rangle\,|H_1,D_1\rangle$$

and $|\psi_2\rangle$ as

$$|\psi_2\rangle = \sum_{k,m,y,z,H_2,D_2} \frac{1}{\sqrt{K}}\alpha'_{m,y,z,H_2,D_2}\,|k,m,y,z\rangle\,|H_2,D_2\rangle$$

Now lets discuss on the requirements on the terms of ψ_1 and corresponding terms of ψ_2, so that the oracle calls can produce parts of $|\mu_1\rangle$ and $|\mu_2\rangle$. In other words, given a term $|\psi_1[\alpha]\rangle = \frac{1}{\sqrt{K}}\alpha_{m,y,z,H_1,D_1}\,|k,m,y,z\rangle\,|H_1,D_1\rangle$ we want to identify if it can be used to produce the nonorthogonal part of ϕ_1. Further we analyze, given such a term $|\psi_1[\alpha]\rangle$, what are the requirements on the terms of $|\psi_2\rangle$ (that we call $|\psi_2[\alpha']\rangle$) so that $\Pi_{\mathrm{H+C}}\mathcal{R}_1\,|\psi_1[\alpha]\rangle \not\perp \Pi_{\mathrm{H+C}}\mathcal{R}_1\,|\psi_2[\alpha']\rangle$. The list of used requirements is the following:

1. Given history register H_1 of $|\psi_1[\alpha]\rangle$ the history register H_2 of $|\psi_2[\alpha']\rangle$ must be the same. This is because the $\mathcal{R}_1$ query will not affect the existent content of history registers.
2. Due to the first requirement, H_1 can not have classical collisions. Otherwise, H_2 will also have them, and this is excluded by $\Pi_{\overline{\mathrm{H}}\cdot\overline{\mathrm{C}}}$. Hence, we can say that all the terms we are interested in are contained in $\Pi_{\mathrm{H}\cdot\overline{\mathrm{C}}}\,|\phi\rangle$.
3. The history register H_1 after the query: $H_1 \in \mathcal{R}_1\,|\psi_1[\alpha]\rangle$ must match the history register $H_2 \in \mathcal{R}_1\,|\psi_1\rangle\,[\alpha']$. Hence, the query inputs (k,m) must be the same.
4. There can be only a single hybrid collision in $|\psi_1[\alpha]\rangle$. Otherwise any term in $|\psi_2\rangle$ will produce terms orthogonal to $\mathcal{R}_1\,|\psi_1[\alpha]\rangle$. Note that a classical query can add only a single hybrid collision to $|\psi_2\rangle$. This collision will be formed by the input used in the query. The inputs used in $|\psi_1[\alpha]\rangle$ and $|\psi_2[\alpha']\rangle$ must be the same. Hence, we will not be able to obtain the second hybrid collision.
5. Assume a hybrid collision in $|\psi_1[\alpha]\rangle$ is formed by $H_1(k_1,m_1) = D_1(k_1^*,m_1^*)$. Then $D_2(k_1^*,m_1^*) \neq D_1(k_1^*,m_1^*)$. Otherwise $|\psi_2[\alpha']\rangle \in \Pi_{\mathrm{H}\cdot\overline{\mathrm{C}}}\,|\phi\rangle$, which is excluded.
6. Following the reasoning of the fourth and fifth requirements, we conclude that the input index (k,m) in both terms $|\psi_1[\alpha]\rangle$ and $|\psi_2[\alpha']\rangle$ must be the input index in D_1 that forms a hybrid collision. In other words if the hybrid collision is formed by $H_1(k_1,m_1) = D_1(k_1^*,m_1^*)$, then the query index must be (k_1^*,m_1^*). Otherwise, the result can not form matching hybrid and quantum registers.

7. Consider a case, when $D_2(k^*, m^*) \neq \perp \wedge H_2(k^*, m^*) = \star$. If after a query to $\mathcal{R}_1$ with $|\psi_2[\alpha']\rangle$ the output for $H_2(k_1^*, m_1^*)$ is pulled from D_2. Then the corresponding nonorthogonal outcome of a query to $\mathcal{R}_1$ with $|\psi_1[\alpha]\rangle$ can be obtained only by resampling the value of $D_1(k_1^*, m_1^*)$. This is because originally $D_1(k_1^*, m_1^*) \neq D_2(k_1^*, m_1^*)$, but for the outcomes to be nonorthogonal these values must match.

Analysis of $\langle\mu_1|\mu_1\rangle$. Lets denote all the terms of $|\psi_1\rangle$ that fulfill our requirements by $|\widehat{\psi}_1\rangle$. We know, $|\mu_1\rangle$ is a part of $\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_1\rangle$, more precisely that there exists a projector Π_1 in computational basis such that $|\mu_1\rangle = \Pi_1\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_1\rangle$. So let us look at the possible outcomes of a query $\mathcal{R}_1|\widehat{\psi}_1\rangle$. We remember that all the terms in $|\widehat{\psi}_1\rangle$ must contain a single hybrid collision without any classical collision, and the queried index should match the input that forms this hybrid collision in the quantum register. Then the possible outcomes are the following:

(a) The value is pulled from D_1, as a result it forms a classical collision in H_1.
(b) The value is set to $\perp$ both in H_1 and D_1. Hence, we lose the only hybrid collision that existed and do not satisfy H + C anymore.
(c) The value is resampled. Here, there is a chance that the output will form a hybrid collision with one of the inputs in the quantum database or a classical collision with the inputs in the history register.

According to Lemma 7, we can formalize the statements above as:

$$\mathcal{R}_1|\widehat{\psi}_1\rangle = \mathcal{R}_1 \sum_{k_1^*,m_1^*,y,z,H_1,D_1} \frac{1}{\sqrt{K}}\alpha_{m_1^*,y,z,H_1,D_1}|k_1^*, m_1^*, y, z\rangle|H_1, D_1\rangle,$$
$$\text{where } k_1^*,\ m_1^* \text{ and } H_1, D_1 \text{ fulfill our requirements}$$

$$\mathcal{R}_1 \sum_{k_1^*,m_1^*,y,z,H_1,D_1} \frac{1}{\sqrt{K}}\alpha_{m_1^*,y,z,H_1,D_1}|k_1^*, m_1^*, y, z\rangle|H_1, D_1\rangle =$$
$$\sum_{\substack{k_1^*,m_1^*,y,z,\\H_1,D_1}} \frac{1}{\sqrt{K}}\alpha_{m_1^*,y,z,H_1,D_1}|k_1^*, m_1^*, y, z\rangle(\omega^{yD_1(k_1^*,m_1^*)}|H_{1\ (k_1^*,m_1^*)\leftarrow D_1(k_1^*,m_1^*)}, D_1\rangle$$
$$+\frac{1}{\sqrt{N}}|H_{1\ (k_1^*,m_1^*)\leftarrow\perp}, D_{1\ (k_1^*,m_1^*)\leftarrow\perp}\rangle - \sum_{p\in[N]}\frac{\omega^{yp}}{N}|H_{1\ (k_1^*,m_1^*)\leftarrow p}, D_{1\ (k_1^*,m_1^*)\leftarrow p}\rangle)$$

As we discussed, after applying $\Pi_{\mathrm{H+C}}$ we get:

$$\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_1\rangle$$
$$= \sum_{\substack{k_1^*,m_1^*,y,\\z,H_1,D_1}} \frac{1}{\sqrt{K}}\alpha_{m_1^*,y,z,H_1,D_1}|k_1^*, m_1^*, y, z\rangle(\omega^{yD_1(k_1^*,m_1^*)}|H_{1\ k_1^*,m_1^*\leftarrow D_1(k_1^*,m_1^*)}, D_1\rangle$$
$$-\sum_{p\in D_1\cup H_1}\frac{\omega^{yp}}{N}|H_{1\ (k_1^*,m_1^*)\leftarrow p}, D_{1\ (k_1^*,m_1^*)\leftarrow p}\rangle)$$
$$\leq \sum_{\substack{k_1^*,m_1^*,y,\\z,H_1,D_1}} \frac{1}{\sqrt{K}}\alpha_{m_1^*,y,z,H_1,D_1}|k_1^*, m_1^*, y, z\rangle(\omega^{yD_1(k_1^*,m_1^*)}|H_{1\ k_1^*,m_1^*\leftarrow D_1(k_1^*,m_1^*)}, D_1\rangle$$
$$+\sum_{p\in D_1\cup H_1\setminus D_1(k_1^*,m_1^*)}\frac{\omega^{yp}}{N}|H_{1\ (k_1^*,m_1^*)\leftarrow p}, D_{1\ (k_1^*,m_1^*)\leftarrow p}\rangle)$$

To obtain the bound on $\langle\mu_1|\mu_1\rangle$ we observe that there exists a projector Π_1 in computational basis such that $|\mu_1\rangle = \Pi_1\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_1\rangle$. Hence, $\langle\mu_1|\mu_1\rangle \leq (\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_1\rangle)^\dagger\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_1\rangle$.

$$\langle\mu_1|\mu_1\rangle \leq (\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_1\rangle)^\dagger\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_1\rangle \leq \sum_\alpha \frac{1}{K}\alpha\alpha^\dagger(1+\frac{t}{N^2}) \leq (\frac{1}{K}+\frac{t}{KN^2}).$$

Here, we used the fact that the number of possible $p \in D_1 \cup H_1$ is upper bounded by t.

Analysis of $\langle\mu_2|\mu_2\rangle$. Lets denote all the terms of $|\psi_2\rangle$ that fulfill our requirements by $|\widehat{\psi}_2\rangle$. We know, that there exists a projector Π_2 in computational basis such that $|\mu_2\rangle = \Pi_2\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_2\rangle$. So let us look at the possible outcomes of a query $\mathcal{R}_1|\widehat{\psi}_2\rangle$. We will split the terms of $|\widehat{\psi}_2\rangle$ into two parts: where $D_2(k_1^*,m_1^*)=\perp$: $|\widehat{\psi}_{2,\perp}\rangle$, and where $D_2(k_1^*,m_1^*)\neq\perp$: $|\widehat{\psi}_{2,\not\perp}\rangle$.

$$\begin{aligned}\langle\mu_2|\mu_2\rangle &\leq \|\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_2\rangle\|^2 \leq \|\Pi_{\mathrm{H+C}}\mathcal{R}_1(|\widehat{\psi}_{2,\perp}\rangle+|\widehat{\psi}_{2,\not\perp}\rangle)\|^2\\ &\leq 2\|\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_{2,\perp}\rangle\|^2 + 2\|\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_{2,\not\perp}\rangle\|^2\end{aligned}$$

Lets first look at $\|\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_{2,\perp}\rangle\|^2$. After the query, we will obtain the following state:

$$\begin{aligned}&\mathcal{R}_1|\widehat{\psi}_{2,\perp}\rangle =\\ &\sum_{\substack{k_1^*,m_1^*,y,z,\\ H_2,D_2}} \frac{1}{\sqrt{K}}\alpha_{m_1^*,y,z,H_1,D_1}|k_1^*,m_1^*,y,z\rangle\Big(\sum_{p\in[N]}\frac{\omega^{yp}}{\sqrt{N}}|H_{2(k_1^*,m_1^*)\leftarrow p},D_{2(k_1^*,m_1^*)\leftarrow p}\rangle\Big)\\ &\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_{2,\perp}\rangle =\\ &\sum_{\substack{k_1^*,m_1^*,y,\\ z,H_2,D_2}} \frac{1}{\sqrt{K}}\alpha_{m_1^*,y,z,H_1,D_1}|k_1^*,m_1^*,y,z\rangle\Big(\sum_{\substack{p\in\\ D_2\cup H_2}}\frac{\omega^{yp}}{\sqrt{N}}|H_{2(k_1^*,m_1^*)\leftarrow p},D_{2(k_1^*,m_1^*)\leftarrow p}\rangle\Big)\end{aligned}$$

Then $\|\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_{2,\perp}\rangle\|^2 \leq \frac{t}{KN}$.

Now lets look at $\|\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_{2,\not\perp}\rangle\|^2$. After the query, we will obtain the following state:

$$\begin{aligned}&\mathcal{R}_1|\widehat{\psi}_{2,\not\perp}\rangle =\\ &\sum_{\substack{k_1^*,m_1^*,y,z,\\ H_2,D_2}} \frac{1}{\sqrt{K}}\alpha_{m_1^*,y,z,H_2,D_2}|k_1^*,m_1^*,y,z\rangle\,(\omega^{yD_2(k_1^*,m_1^*)}|H_{2\ k_1^*,m_1^*\leftarrow D_2(k_1^*,m_1^*)},D_2\rangle\\ &+\frac{1}{\sqrt{N}}|H_{2\ (k_1^*,m_1^*)\leftarrow\perp},D_{2\ (k_1^*,m_1^*)\leftarrow\perp}\rangle - \sum_{p\in[N]}\frac{\omega^{yp}}{N}|H_{2\ (k_1^*,m_1^*)\leftarrow p},D_{2\ (k_1^*,m_1^*)\leftarrow p}\rangle\Big)\end{aligned}$$

As discussed before, setting the values to $\perp$ won't help. A possible way to get a hybrid or a classical collision is by resampling the output of $D_2(k_1^*,m_1^*)$ to one of the values contained in the quantum or history registers. Also note that $D_2(k_1^*,m_1^*)$ does not match $D_1(k_1^*,m_1^*)$, hence pulling it into the history register will not create a nonorthogonal state to $|\mu_1\rangle$, unless the value in $|\widehat{\psi}_1\rangle$ is resampled. This corresponds to database registers in $|\psi_1\rangle$ of the form

$\frac{\omega^{yD_2(k_1^*,m_1^*)}}{N}|H_{1\ (k_1^*,m_1^*)\leftarrow D_2(k_1^*,m_1^*)}, D_{1\ (k_1^*,m_1^*)\leftarrow D_2(k_1^*,m_1^*)}\rangle$ (see requirement 7).

$$\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_{2,\not L}\rangle \leq$$
$$\sum_{\substack{k_1^*,m_1^*,y,z,\\ H_2,D_2}} \frac{1}{\sqrt{K}}\alpha_{m_1^*,y,z,H_2,D_2}|k_1^*,m_1^*,y,z\rangle\left(\omega^{yD_2(k_1^*,m_1^*)}|H_{2\ (k_1^*,m_1^*)\leftarrow D_2(k_1^*,m_1^*)}, D_2\rangle\right.$$
$$\left. + \sum_{p\in D_2\cup H_2\setminus D_2(k_1^*,m_1^*)} \frac{\omega^{yp}}{N}|H_{2\ (k_1^*,m_1^*)\leftarrow p}, D_{2\ (k_1^*,m_1^*)\leftarrow p}\rangle\right)$$

Recall that $|\mu_2\rangle$ can be obtained as $|\mu_2\rangle = \Pi_2\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_2\rangle$, where $|\mu_2\rangle$ is the nonorthogonal part to ϕ_1. We can actually split $|\mu_2\rangle$ into two parts: $|\mu_2\rangle = |\mu_2'\rangle + |\mu_2''\rangle$, where

$$|\mu_2'\rangle =$$
$$\sum_{\substack{k_1^*,m_1^*,y,z,\\ H_2,D_2}} \frac{1}{\sqrt{K}}\alpha_{m_1^*,y,z,H_2,D_2}|k_1^*,m_1^*,y,z\rangle\,\omega^{yD_2(k_1^*,m_1^*)}|H_{2\ (k_1^*,m_1^*)\leftarrow D_2(k_1^*,m_1^*)}, D_2\rangle$$

and $|\mu_2''\rangle = |\mu_2\rangle - |\mu_2'\rangle$. Hence, we can write

$$\langle\mu_1|\mu_2\rangle \leq 2\langle\mu_1|\mu_2'\rangle + 2\langle\mu_1|\mu_2''\rangle \leq 2\langle\mu_1|\mu_2'\rangle + 2\sqrt{\langle\mu_1|\mu_1\rangle\langle\mu_2''|\mu_2''\rangle}.$$

Due to the part in $|\psi_1\rangle$:

$$\frac{1}{\sqrt{K}}\alpha'_{m_1^*,y,z,H_1,D_1}\frac{\omega^{yD_2(k_1^*,m_1^*)}}{N}|H_{1\ (k_1^*,m_1^*)\leftarrow D_2(k_1^*,m_1^*)}, D_{1\ (k_1^*,m_1^*)\leftarrow D_2(k_1^*,m_1^*)}\rangle,$$

which will correspond to $\frac{1}{\sqrt{K}}\alpha_{m_1^*,y,z,H_2,D_2}|k_1^*,m_1^*,y,z\rangle\,\omega^{yD_2(k_1^*,m_1^*)}|H_{2\ (k_1^*,m_1^*)\leftarrow D_2(k_1^*,m_1^*)}, D_2\rangle$ we get:

$$\langle\mu_1|\mu_2'\rangle \leq \frac{1}{KN}, \quad \langle\mu_2''|\,\Pi_{\mathrm{H+C}}\mathcal{R}_1|\widehat{\psi}_{2,\not L}\rangle \leq \frac{t}{KN^2}.$$

As a result, we get

$$\langle\mu_2''|\mu_2''\rangle \leq 2\cdot\frac{t}{KN} + 2\cdot\frac{t}{KN^2} \leq 4\frac{t}{KN}.$$
$$\langle\phi_1|\phi_2\rangle \leq 2\sqrt{\langle\mu_1|\mu_1\rangle}\sqrt{\langle\mu_2''|\mu_2''\rangle} + 2\langle\mu_1|\mu_2\rangle \leq 2\sqrt{(\frac{1}{K}+\frac{t}{KN^2})}\sqrt{4\frac{t}{KN}} + \frac{2}{KN}$$
$$\leq \sqrt{16\frac{t}{K^2N} + 16\frac{t^2}{K^2N^3}} + \frac{1}{KN} \leq 4(\frac{\sqrt{t}}{K\sqrt{N}} + \frac{t}{KN^{3/2}})$$

The last inequality comes from the observation that $\sqrt{t} \geq 1$ and $\sqrt{N} \leq N$.

Combining all the results, we obtain the bound from the theorem. □

6 Applications

In this section, we discuss practical implications of our result on the M-ETCR security of a hash function under generic attacks. The main application of M-ETCR is the analysis of the hash & sign transform [8,23]. The hash & sign transform allows to turn a fixed message-length signature scheme

into a variable message-length signature by first computing a message digest and then signing the digest: $\sigma = \text{Sign}(H(m))$. This plain version requires a collision-resistant hash function for security. When SHA1 was broken, collision-resistance was considered an unfavorable requirement that was to be avoided where possible. Such avoidance usually also comes with shorter digest sizes as other properties are not vulnerable to birthday attacks, improving efficiency too.

For hash & sign it was suggested to randomize the digest computation [16] to avoid the need for collision resistance. In this case, the message is hashed with a random salt r, which is then attached to the signature of the original scheme: $\sigma = (r, \text{Sign}(H(r, m)))$. The authors introduced the extended target collision resistance notion (eTCR) to analyze the security of this construction. eTCR matches the M-ETCR notion if we make just one challenge query. By a plug & pray argument, eTCR implies M-ETCR up to the number of challenge queries. While adding randomization to the message hashing allows us to reduce the security requirements from collision resistance to M-ETCR, potentially reducing the digest size, it also increases the signature size, since the salt must be added to the signature. In total, this is usually still beneficial. However, to optimize the scheme's performance, we aim for a M-ETCR bound that allows to use minimal-length salts.

The hash & sign paradigm is often used to allow the signing of arbitrary long messages. An example of such an application can be found in [4, Section 14.1.1]. The authors show how the hashing of the message can improve the efficiency of one-time hash-based digital signatures. They also rely on randomized hashing and eTCR security. Note that for one-time hash-based signatures, the signed digest's size directly affects the scheme's overall efficiency and signature sizes. By requiring only M-ETCR security, we can avoid using long digests, which we would have to use if we did rely on collision resistance. Due to our analysis, we can also use short salts, shrinking the total size of the signature.

The hash & sign paradigm is also widely used in lattice-based signature schemes. For example, Falcon [22] - a lattice-based digital signature scheme, recently chosen for standardization by NIST [26]. The authors suggest randomized hashing of the message (see [22, Section 2.2.2]). Since the security of Falcon is based on the GPV framework [13], it is important that two different signatures are never generated for the same digest. To achieve this, the authors require the size of the salt to be 320 bits. Note that the M-ETCR property covers the required properties for message hashing. If we aim for the highest security level for the NIST parameters, we can estimate the number of classical queries as 2^{64} and the number of quantum queries 2^{128}. The hashing in Falcon is done with SHAKE256 [25]. Hence, we set $N = 2^{256}$. Using these parameters, we get that the 200 bits is enough for the salt space. This is significantly smaller than the sizes used by the Falcon team.

The salt part may not play a big role in the signature sizes, especially for post-quantum schemes. While this is true, salts can play a significant role when we look at signature aggregation. In [1], the authors analyze the aggregation of multiple Falcon signatures. This approach is very useful when sending a large number of signatures over a low bandwidth network – a typical case for large-scale blockchains. Due to a conflict between the random oracle model and viewing a hash function as a circuit (see [19]), the authors decided to include the salts from all the signatures in the final aggregated signature (so the verifier can compute $H(r, m)$ locally). If we require salts of size 200, instead of 320 bits, for the parameters that the authors suggest (see [1, Table 1]), for 2000 signatures, we get a total size reduction from 165 kB to 136 kB, which is an almost 18% decrease in signature size.

Sometimes, it is possible to include the salt in the aggregation. For example, a recent work [11] did this for hash-based multi-signatures. In this case, the effect on the signature size will be minimal. However, a general approach to aggregate signatures involves building a circuit that verifies multiple signatures and then producing a succinct argument for this circuit. Larger salt increases this circuit's complexity, affecting the singing and verification efficiency.

References

1. Aardal, M.A., Aranha, D.F., Boudgoust, K., Kolby, S., Takahashi, A.: Aggregating falcon signatures with LaBRADOR. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology – CRYPTO 2024, Part I. Lecture Notes in Computer Science, vol. 14920, pp. 71–106. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 18–22, 2024). https://doi.org/10.1007/978-3-031-68376-3_3
2. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM CCS 93: 1st Conference on Computer and Communications Security. pp. 62–73. ACM Press, Fairfax, Virginia, USA (Nov 3–5, 1993). https://doi.org/10.1145/168588.168596
3. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology – ASIACRYPT 2011. Lecture Notes in Computer Science, vol. 7073, pp. 41–69. Springer Berlin Heidelberg, Germany, Seoul, South Korea (Dec 4–8, 2011). https://doi.org/10.1007/978-3-642-25385-0_3
4. Boneh, D., Shoup, V.: A Graduate Course in Applied Cryptography (2023), https://toc.cryptobook.us/
5. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited (preliminary version). In: 30th Annual ACM Symposium on Theory of Computing. pp. 209–218. ACM Press, Dallas, TX, USA (May 23–26, 1998). https://doi.org/10.1145/276698.276741
6. Chiesa, A., Manohar, P., Spooner, N.: Succinct arguments in the quantum random oracle model. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019: 17th Theory of Cryptography Conference, Part II. Lecture Notes in Computer Science, vol. 11892, pp. 1–29. Springer, Cham, Switzerland, Nuremberg, Germany (Dec 1–5, 2019). https://doi.org/10.1007/978-3-030-36033-7_1
7. Chung, K.M., Fehr, S., Huang, Y.H., Liao, T.N.: On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In: Canteaut, A., Standaert, F.X. (eds.) Advances in Cryptology – EUROCRYPT 2021, Part II. Lecture Notes in Computer Science, vol. 12697, pp. 598–629. Springer, Cham, Switzerland, Zagreb, Croatia (Oct 17–21, 2021). https://doi.org/10.1007/978-3-030-77886-6_21
8. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory **22**(6), 644–654 (1976). https://doi.org/10.1109/TIT.1976.1055638
9. Don, J., Fehr, S., Huang, Y.H.: Adaptive versus static multi-oracle algorithms, and quantum security of a split-key PRF. In: Kiltz, E., Vaikuntanathan, V. (eds.) TCC 2022: 20th Theory of Cryptography Conference, Part I. Lecture Notes in Computer Science, vol. 13747, pp. 33–51. Springer, Cham, Switzerland, Chicago, IL, USA (Nov 7–10, 2022). https://doi.org/10.1007/978-3-031-22318-1_2
10. Don, J., Fehr, S., Majenz, C., Schaffner, C.: Efficient NIZKs and signatures from commit-and-open protocols in the QROM. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology – CRYPTO 2022, Part II. Lecture Notes in Computer Science, vol. 13508, pp. 729–757. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 15–18, 2022). https://doi.org/10.1007/978-3-031-15979-4_25
11. Drake, J., Khovratovich, D., Kudinov, M., Wagner, B.: Hash-based multi-signatures for post-quantum ethereum. Cryptology ePrint Archive, Paper 2025/055 (2025), https://eprint.iacr.org/2025/055
12. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology – CRYPTO'86. Lecture Notes in Computer Science, vol. 263, pp. 186–194. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 1987). https://doi.org/10.1007/3-540-47721-7_12
13. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th Annual ACM Symposium on Theory of Computing. pp. 197–206. ACM Press, Victoria, BC, Canada (May 17–20, 2008). https://doi.org/10.1145/1374376.1374407
14. Grilo, A.B., Hövelmanns, K., Hülsing, A., Majenz, C.: Tight adaptive reprogramming in the QROM. In: Tibouchi, M., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2021, Part I. Lecture Notes in Computer Science, vol. 13090, pp. 637–667. Springer, Cham, Switzerland, Singapore (Dec 6–10, 2021). https://doi.org/10.1007/978-3-030-92062-3_22
15. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: 28th Annual ACM Symposium on Theory of Computing. pp. 212–219. ACM Press, Philadephia, PA, USA (May 22–24, 1996). https://doi.org/10.1145/237814.237866

16. Halevi, S., Krawczyk, H.: Strengthening digital signatures via randomized hashing. In: Dwork, C. (ed.) Advances in Cryptology – CRYPTO 2006. Lecture Notes in Computer Science, vol. 4117, pp. 41–59. Springer Berlin Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2006). https://doi.org/10.1007/11818175_3
17. Hamoudi, Y., Liu, Q., Sinha, M.: The NISQ complexity of collision finding. In: Joye, M., Leander, G. (eds.) Advances in Cryptology – EUROCRYPT 2024, Part IV. Lecture Notes in Computer Science, vol. 14654, pp. 3–32. Springer, Cham, Switzerland, Zurich, Switzerland (May 26–30, 2024). https://doi.org/10.1007/978-3-031-58737-5_1
18. Hülsing, A., Rijneveld, J., Song, F.: Mitigating multi-target attacks in hash-based signatures. In: Cheng, C.M., Chung, K.M., Persiano, G., Yang, B.Y. (eds.) PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part I. Lecture Notes in Computer Science, vol. 9614, pp. 387–416. Springer Berlin Heidelberg, Germany, Taipei, Taiwan (Mar 6–9, 2016). https://doi.org/10.1007/978-3-662-49384-7_15
19. Khovratovich, D., Rothblum, R.D., Soukhanov, L.: How to prove false statements: Practical attacks on fiat-shamir. Cryptology ePrint Archive, Paper 2025/118 (2025), https://eprint.iacr.org/2025/118
20. Krawczyk, H., Bellare, M., Canetti, R.: HMAC: Keyed-hashing for message authentication. IETF Internet Request for Comments 2104 (Feb 1997)
21. Liu, Q., Zhandry, M.: On finding quantum multi-collisions. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2019, Part III. Lecture Notes in Computer Science, vol. 11478, pp. 189–218. Springer, Cham, Switzerland, Darmstadt, Germany (May 19–23, 2019). https://doi.org/10.1007/978-3-030-17659-4_7
22. Prest, T., Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: FALCON. Tech. rep., National Institute of Standards and Technology (2022), available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022
23. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the Association for Computing Machinery **21**(2), 120–126 (Feb 1978). https://doi.org/10.1145/359340.359342
24. of Standards, N.I., Technology: Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf, accessed: 2025-01-31
25. of Standards, N.I., Technology: FIPS PUB 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. http://dx.doi.org/10.6028/NIST.FIPS.202, accessed: 2025-02-11
26. of Standards, N.I., Technology: Post-Quantum Cryptography. https://csrc.nist.gov/projects/post-quantum-cryptography, accessed: 2023-09-12
27. Zhandry, M.: How to record quantum queries, and applications to quantum indifferentiability. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology – CRYPTO 2019, Part II. Lecture Notes in Computer Science, vol. 11693, pp. 239–268. Springer, Cham, Switzerland, Santa Barbara, CA, USA (Aug 18–22, 2019). https://doi.org/10.1007/978-3-030-26951-7_9