Time-Space Lower Bounds for Finding Collisions in Merkle-Damgård Hash Functions

Akshima[1(✉)], Siyao Guo[2], and Qipeng Liu[3]

[1] University of Chicago, Chicago, USA
akshima@uchicago.edu
[2] NYU Shanghai, Shanghai, China
[3] Simons Institute for the Theory of Computing, Berkeley, USA

Abstract. We revisit the problem of finding B-block-long collisions in Merkle-Damgård Hash Functions in the auxiliary-input random oracle model, in which an attacker gets a piece of S-bit advice about the random oracle and makes T oracle queries.

Akshima, Cash, Drucker and Wee (CRYPTO 2020), based on the work of Coretti, Dodis, Guo and Steinberger (EUROCRYPT 2018), showed a simple attack for $2 \leq B \leq T$ (with respect to a random salt). The attack achieves advantage $\widetilde{\Omega}(STB/2^n + T^2/2^n)$ where n is the output length of the random oracle. They conjectured that this attack is optimal. However, this so-called STB conjecture was only proved for $B \approx T$ and $B = 2$. Very recently, Ghoshal and Komargodski (CRYPTO 22) confirmed STB conjecture for all constant values of B, and provided an $\widetilde{O}(S^4TB^2/2^n + T^2/2^n)$ bound for all choices of B.

In this work, we prove an $\widetilde{O}((STB/2^n) \cdot \max\{1, ST^2/2^n\} + T^2/2^n)$ bound for every $2 < B < T$. Our bound confirms the STB conjecture for $ST^2 \leq 2^n$, and is optimal up to a factor of S for $ST^2 > 2^n$ (note as T^2 is always at most 2^n, otherwise finding a collision is trivial by the birthday attack). Our result subsumes all previous upper bounds for all ranges of parameters except for $B = \widetilde{O}(1)$ and $ST^2 > 2^n$.

We obtain our results by adopting and refining the technique of Chung, Guo, Liu, and Qian (FOCS 2020). Our approach yields more modular proofs and sheds light on how to bypass the limitations of prior techniques. Along the way, we obtain a considerably simpler and illuminating proof for $B = 2$, recovering the main result of Akshima, Cash, Drucker and Wee.

1 Introduction

Merkle-Damgård paradigm [Mer89, Dam89] is a domain extension technique for extending a compression function $H : [N] \times [M] \to [N]$ (where $N := 2^n$ and $M > N$) with fixed input length into a full-fledged hash function to handle arbitrary long inputs. Specifically, a B-block message $\mathbf{m} = (m_1, \cdots, m_B)$ with

Y. Dodis and T. Shrimpton (Eds.): CRYPTO 2022, LNCS 13509, pp. 192–221, 2022.
https://doi.org/10.1007/978-3-031-15982-4_7

$m_i \in [M]$ is hashed into $\mathsf{MD}_H(a, \mathbf{m})$ as follows: $\mathsf{MD}_H^1(a, m_1) = H(a, m_1)$ and

$$\mathsf{MD}_H^\ell(a, (m_1, \cdots, m_\ell)) = H(\mathsf{MD}_H^{\ell-1}(a, (m_1, \cdots, m_{\ell-1})), m_\ell), \text{ for } \ell > 1,$$

where $a \in [N]$ is some random given salt. We say $\mathbf{m} \neq \mathbf{m}'$ is a pair of B-block collision with respect to a salt a if they both have at most B blocks and $\mathsf{MD}_H(a, \mathbf{m}) = \mathsf{MD}_H(a, \mathbf{m}')$.

Merkle-Damgård paradigm is widely used in practice for hash functions, including MD5 and SHA family. The primary requirement of a hash function is collision resistance. In this work, we are interested in the collision resistance property of Merkle-Damgård hash functions against preprocessing attackers, which can have an arbitrary (but bounded) precomputed advice about H to help. The power of preprocessing attacks was first demonstrated by Hellman [Hel80] for inverting functions. Recently, several works [DGK17, CDG18, ACDW20, GK22] set out to understand the power of such attacks for finding collisions. All of them studied this question in the auxiliary-input random oracle model (AI-ROM) proposed by Unruh [Unr07], for dealing with non-uniform and preprocessing attackers. In this ideal model, H is treated as a random function, and an adversary $\mathcal{A}$ consists of a pair of algorithms $(\mathcal{A}_1, \mathcal{A}_2)$. (Computationally unbounded) $\mathcal{A}_1$ precomputes S bits of advice about H in an offline stage, then $\mathcal{A}_2$ takes this advice and makes T oracle queries to H during the attack.

Dodis, Guo, and Katz [DGK17] studied the collision resistance of a salted random function (which also corresponds to the $B = 1$ case for Merkle-Damgård). They proved an $\widetilde{O}(S/N + T^2/N)$ security upper bound (with respect to a random salt) where the notation $\widetilde{O}(\cdot)$ hides lower-order factors that are polynomial in $\log N$. This bound shows the optimality of the naive attack, which precomputes collisions for S distinct salts as the advice (the T^2/N term is tight due to the birthday attack).

Since most practical hash functions are based on the Merkle-Damgård paradigm, Coretti, Dodis, Guo and Steinberger [CDGS18] studied finding collisions for salted Merkle-Damgård hash functions (corresponds to the unbounded B case). Interestingly, unlike the $B = 1$ case, they showed an attack achieving advantage $\widetilde{\Omega}(ST^2/N)$, improving the birthday attack by a factor of S. They also proved that this attack is optimal.

Akshima, Cash, Drucker and Wee [ACDW20] observed that the collision produced by the attack of [CDGS18] is very long, which is not appealing for practical relevance. They, therefore, studied the question of finding short collisions, and put forth the following intriguing conjecture.

> STB conjecture [ACDW20]: The best attack with time T and space S for finding collisions of length B in salted MD hash functions built from hash functions with n-bit outputs achieves success probability $\Theta((STB + T^2)/2^n)$.

[ACDW20] showed that, a straightforward modification of the attack of [CDGS18] finds B-block collisions with advantage $\Omega((STB + T^2)/N)$. Unfortunately, they

also showed that the lower bound techniques of [CDGS18] can not rule out attacks with success probability $\Omega(ST^2/N)$, even for $B = 2$. They presented new approaches to prove the STB conjecture for $B = 2$ in AI-ROM. Combining with known results for $B = 1$ and $B = T$, this demonstrates qualitative jumps in the optimal attacks for finding length 1, length 2, and unbounded-length collisions. Very recently, Ghoshal and Komargodski [GK22] confirmed STB conjecture for all constant B. However, for other choices of B, there is still a significant gap between the best-known attack [ACDW20] and known security upper bound $\widetilde{O}(S^4TB^2/N + T^2/N)$ by [GK22] or $\widetilde{O}(ST^2/N)$ by [CDGS18]. That motivates us to study the following question in this paper:

Can we further bridge the gap between the security upper and lower bounds, and prove STB conjecture for more choices of parameters?

Since prior techniques are limited or laborious even for $B = 2$, we start by asking:

Can we prove STB conjecture for $B = 2$ in a simpler way?

Looking ahead, we answer both questions affirmatively.

1.1 Our Results

Our main contribution is the following theorem.

Theorem 1 (Informal). *For any $2 < B < T$, the advantage of the best adversary with S-bit advice and T queries for finding B-block collisions in Merkle-Damgård hash functions in the auxiliary-input random oracle model, is*

$$\widetilde{O}\left((STB/N) \cdot \max\{1, ST^2/N\} + T^2/N\right).$$

Our bound confirms the STB conjecture for any $2 < B < T$ for the range of S, T such that $ST^2 \leq N$. For the other range of S, T, as $T^2 \leq N$ (otherwise, finding a collision is trivial by the birthday attack), Our bound is at most $\widetilde{O}(S^2TB/N + T^2/N)$, which is optimal up to a factor of S.

Comparing to the $\widetilde{O}(STB^2(\log^2 S)^{B-2}/N + T^2/N)$ bound by [GK22], our bound works for any $2 < B < T$, while their bound becomes vacuous when $B > \log N$. However, for $B \leq \log N$, unlike our bound, their bound could be tight even when $ST^2 > N$. In particular, their bound confirms STB conjecture for $B = O(1)$.

Our bound strictly improves the $\widetilde{O}(S^4TB^2/N + T^2/N)$ bound by [GK22], and the $\widetilde{O}(S^2T/N)$ bound by [CDGS18] for any $2 < B < T$ and non-trivial choices of S, T (specifically, when STB attack succeeds with at most a constant probability, i.e., $STB = O(N)$). The two bounds by [GK22] only beat [CDGS18] for $B \ll \sqrt{T}$.

As an additional contribution, we give a considerably simpler proof for proving the tight bound for $B = 2$, recovering the main result of [ACDW20].

Theorem 2 (Informal). *The advantage of the best adversary with S-bit advice and T queries for finding 2-block collisions in Merkle-Damgård hash functions in the auxiliary-input random oracle model, is $\widetilde{O}\left(ST/N + T^2/N\right)$.*

A comparison of our results with the prior works is summarized in Table 1. Overall, our results subsume all previous upper bounds except for the range of S, T, B such that $B \leq \log N$ and $ST^2 > N$.

Table 1. Asymptotic security bounds on the security of finding B-block-long collisions in Merkle-Damgård Hash Functions constructed from a random function $H : [N] \times [M] \mapsto [N]$ against (S, T)-algorithms. For simplicity, logarithmic terms and constant factors are omitted.

	Best attacks	Security bounds	Ref.	Proof techniques
$B = 1$	$\frac{S}{N} + \frac{T^2}{N}$	$\frac{S}{N} + \frac{T^2}{N}$	[DGK17]	Compression
$B = 2$	$\frac{ST}{N} + \frac{T^2}{N}$	$\frac{ST}{N} + \frac{T^2}{N}$	[ACDW20]	Multi-instance problems
$B = 2$	$\frac{ST}{N} + \frac{T^2}{N}$	$\frac{ST}{N} + \frac{T^2}{N}$	Theorem 2	Multi-instance games
$2 < B < T$	$\frac{STB}{N} + \frac{T^2}{N}$	$\frac{STB^2(\log^2 S)^{B-2}}{N} + \frac{T^2}{N}$	[GK22]	Multi-instance problems
$2 < B < T$	$\frac{STB}{N} + \frac{T^2}{N}$	$\frac{S^4TB^2}{N} + \frac{T^2}{N}$	[GK22]	Multi-instance problems
$2 < B < T$	$\frac{STB}{N} + \frac{T^2}{N}$	$\frac{STB}{N} \cdot \max\{1, \frac{ST^2}{N}\} + \frac{T^2}{N}$	Theorem 1	Multi-instance games
Unbounded	$\frac{ST^2}{N}$	$\frac{ST^2}{N}$	[CDGS18]	Presampling

1.2 Our Techniques

In this section, we describe our techniques, how to use them to prove our main results, and what makes our techniques different from prior approaches used in [CDGS18, ACDW20, GK22].

Existing Reduction to Sequential Multi-instance Games. Our initial inspiration is the recent framework of Chung, Guo, Liu, Qian [CGLQ20] for establishing tight time-space tradeoffs in the quantum random oracle model. Generally speaking, they reduce proving the security of a problem with S-bit advice to proving the security of multiple random instances of the problem, presented one at a time, *without* advice. Specifically, they observe that[1], if any adversary (with no advice) can solve S instances of the problem "sequentially" with success probability at

[1] The framework of Chung, Guo, Liu, Qian [CGLQ20] reduces to analyzing sequential multi-instance security for $S + \log N + 1$ instances instead of S-instances. We slightly improve their parameters and obtain a considerably cleaner version in Theorem 3.

most δ^S, then any adversary with S-bit advice can solve one instance of the problem with success probability at most 2δ.

This idea of reducing the security of a problem with advice to the security of a multi-instance problem without advice was first introduced by Impagliazzo and Kabanets in [IK10]. The idea was also used by later works [ACDW20, GK22]. The difference between [IK10] and the later works, including this work, is that we reduce to a "sequential" multi-instance game as opposed to a "parallel" multi-instance problem. More concretely, in the parallel multi-instance problem, the adversary is presented with all the randomly chosen instances of the challenge problems to solve once at the start. Whereas in the multi-instance game, the adversary gets a new randomly chosen instance of challenge problem one at a time and only after solving all the previous challenges.

Chung et al. [CGLQ20] recently demonstrated a separation between "sequential" multi-instance games and "parallel" multi-instance problems in the context of function inversion in the quantum setting[2]. Guo, Li, Liu and Zhang [GLLZ21] pointed out a connection between "sequential" multi-instance game and the presampling technique (first introduced by Unruh [Unr07], and further optimized by Coretti et al. [CDGS18])—the main technique used by Coretti et al. [CDGS18] for proving the $O(ST^2/N)$ bound. Roughly speaking, all results relying on presampling technique can be reproved using "sequential" multi-instance games. That suggested that "sequential" multi-instance games have the potential to prove stronger results. Therefore we are motivated to adapt and take full advantage of "sequential" multi-instance games in the context of collision finding.

To better illustrate the connection between "sequential" multi-instance games and the presampling technique, we show how to recover the $O(ST^2/N)$ bound by Coretti et al. [CDGS18]. Recall that presampling technique by Coretti et al. [CDGS18] generically reduces security proofs of unpredictability applications (including collision finding) in the AI-ROM to a much simpler P-bit-fixing random-oracle model (BF-ROM), where the attacker can arbitrarily fix the values of the random oracle on some $P := O(ST)$ coordinates, but then the remaining coordinates are chosen at random. Coretti et al. [CDGS18] showed that the security of finding collisions in Merkle-Damågard Hash Functions in the BF-ROM is $O(ST/N)$.

Using "sequential" multi-instance games, it suffices to bound the advantage of any adversary (with no advice) winning a new game, conditioning on winning all previous (up to at most S) ones, by $O(ST^2/N)$. The adversary wins all games with advantage $O(ST^2/N)^S$, which implies the desired security against S-bit advice. The key point is that the adversary (with no advice) made at most ST queries in previous games. Therefore, conditioning on any possible events of earlier games, from the view of the adversary, the random oracle is essentially a

[2] In particular, they showed that "sequentially" inverting S random images (with T quantum queries per round to a given random function $f : [N] \to [N]$) admits security $O(ST/N + T^2/N)^S$, and the corresponding "parallel" multi-instance problems admits an attack with advantage $\Omega(ST^2/N)^S$.

(convex combination of) bit-fixing random oracles (BF-ROM) [CDGS18], where at most ST-positions are known, and the rest remains independent and random. Hence, it suffices to prove the security of a single game in BF-ROM by $O(ST^2/N)$, which has been shown by Coretti et al. [CDGS18] as a necessary step to use the presampling technique.

Barriers of the Above Idea. Akshima et al. [ACDW20] pointed out a barrier to using the vanilla presampling technique towards proving $B = 2$. In particular, one can only hope to achieve $\Omega(ST^2/N)$ in the BF-ROM even for $B = 2$. Recall that, to prove the sequential multi-instance security, it is sufficient to bound the advantage of any adversary that finds a 2-block collision for a fresh salt a, conditioned on it finds 2-block collisions for all the previous random challenge salts $a_1, \cdots, a_S$.

We will call these ST queries made during the first S rounds as offline queries. Among the T queries made for a, we will call the queries that were not made during the first S rounds as online queries. Throughout the discussion, we will focus on the case that the new salt a has never been queried before in offline queries, because the other case happens with probability at most ST/N (so won't affect our conclusion). As a result, all queries starting with the challenge salt a have to be online queries.

It is clear that the adversary learns about the function not only using the online queries but also from the offline queries. The information this algorithm can take advantage of from the offline queries varies by a lot. The followings are two extreme cases:

1. The offline queries consist of exactly one single query for each of ST distinct salts.
2. The offline queries consist of one collision for each of $ST/2$ distinct salts

For the first case, the offline queries can barely help[3]. Whereas, in the second case, as long as an adversary can find a pre-image (starting with the challenge salt a) of any of these $ST/2$ salts, it finds a 2-block collision (Fig. 1). Since there are T online queries, the algorithm achieves advantage at least $ST^2/(2N)$ in the second case.

The vanilla presampling approach works for worst-case offline queries. Given the above example, the best security bound one can hope to achieve in the BF-ROM for $B = 2$ is $\Omega(ST^2/N)$.

Our Main Technical Novelty. Our main insight is that, unlike the presampling technique in which offline queries can be arbitrary, the worst offline queries are not typical and can be tolerated by refining the technique. In the above example, the chance that offline queries form $ST/2$ pairs of collisions is quite unlikely. We define the following "high knowledge gaining" event $\mathbf{E}_1$:

[3] We do not prove it rigorously here. Instead, we focus on the more interesting case – offline queries do provide advantages.

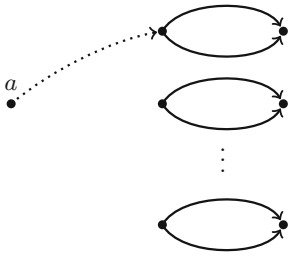


Fig. 1. Nodes indicate salts in $[N]$. An arrow connected two salts means there is a query on the starting salt and a message in $[M]$ such that the output is the other salt. An online query hits an existing collision. Solid lines denote offline queries. The dotted line denotes the online query that forms a 2-block collision.

$\mathbf{E}_1$: By making ST queries, there are more than S distinct salts with 1-block collision.

The name "high knowledge gaining" suggests that whenever this event happens, the online algorithm can behave significantly better than average (following the attack in Fig. 1). If this event $\mathbf{E}_1$ does not happen, the probability that an online algorithm finds a query hitting an existing offline collision is bounded by $O((S/N)\cdot T)$; it is much better compared to the worst case – which is $O(ST^2/N)$. Remember that we have not shown how to prove that $\mathbf{E}_1$ happens with a tiny probability. We will not do that in this section since this is not our main technical novelty.

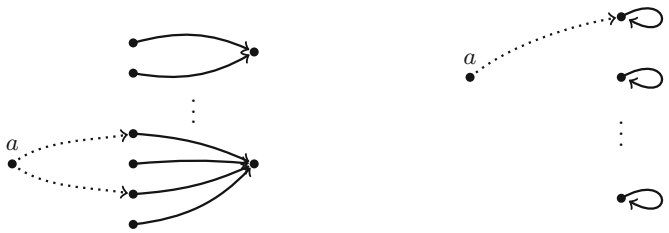
We then show two more "high knowledge gaining" events, which are all the events we consider. Conditioned on none of them happens, no online algorithms can find 2-block collisions with advantage better than $O(ST/N + T^2/N)$. The second event $\mathbf{E}_2$ is defined as:

$\mathbf{E}_2$: By making ST queries, there are more than S^2 pairs of queries forming collisions.

In Fig. 2a, we denote a multi-collision by a claw. $\mathbf{E}_2$ says that many pair-wise collisions are found among all the offline queries. $\mathbf{E}_1$ only cares about collisions starting with the same salt, whereas $\mathbf{E}_2$ counts every pair of collisions (even starting with distinct salts). If there are many pairs of collisions, as long as an online adversary can hit two queries that form a collision, it finds a 2-block collision. The probability that an online algorithm having two queries hitting one particular existing collision is at most $O(T^2/N^2)$; if $\mathbf{E}_2$ does not happen, by union bound, the advantage of this type of attack is bounded by $O(S^2\cdot(T^2/N^2))$, again smaller than $O(ST/N)$.

The final event $\mathbf{E}_3$ is very similar to $\mathbf{E}_1$:

$\mathbf{E}_3$: By making ST queries, there are more than S distinct salts with self-loops.



(a) $\mathbf{E}_2$ (b) $\mathbf{E}_3$

Fig. 2. Other two "high knowledge gaining" events and their corresponding attacks.

If an online algorithm hits an offline self-loop, it forms a 2-block collision. Following the same reasoning as $\mathbf{E}_1$, if $\mathbf{E}_3$ does not happen, the probability that an online algorithm finds a query hitting an existing self-loop is bounded by $O((S/N) \cdot T)$.

By identifying the "high knowledge gaining" events and managing to show that they are all unlikely (which is intuitive but non-trivial to prove), we obtain a considerably simpler proof for the $B = 2$ result from [ACDW20] using our approach in Sect. 3 for illustration. More precisely, with all these "high knowledge gaining" events, we show that[4]: (1) these events happen with probability at most $O(N^{-S})$, even conditioned on the adversary winning all the previous rounds; (2) when none of them happens, an online algorithm making T queries can find a 2-block collision with advantage $O(ST/N + T^2/N)$: such a 2-block collision will consist of either hybrid queries (both online and offline queries) or solely online queries; but for both cases, the probability is small.

It is an upside of our technique that it modularises and separates the bad events, making the overall proof more straightforward and intuitive. Following the same structure, we then extend our proof to larger B by identifying a few events, and obtaining our main result.

Applying Our New Techniques to Larger B. As for $B = 2$, we present results for the sequential multi-instance model and use the reduction to prove results in the auxiliary input model. We simplify the sequential multi-instance model into the offline phase and online phase as in the $B = 2$ result and again use our insight that worst offline queries are unlikely and better bounds than $O(ST^2/N)$ can be achieved using a more refined analysis. However, unlike for $B = 2$ analysis, our larger B analysis is not as straightforward and requires some creative case analysis in terms of collision types.

We call offline queries that share an image under H with other offline query/queries as marked queries. We define the following "high knowledge gaining" event:

[4] This is not a formal argument but captures the intuition behind our technique. For the formal proofs, please refer to Sect. 3.

Fig. 3. Dotted lines denote online queries. Solid lines denote offline queries. Dash-dotted lines can be either offline or online queries. Red lines denote 'colliding' queries. (Color figure online)

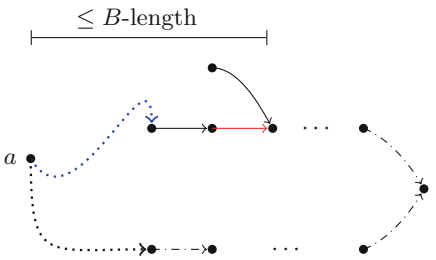


Fig. 4. The B-length collision uses some marked query. The solid red line denotes the *first* marked query along the B-length collisions. The dotted blue line denote the *closest* online query to the red line along the B-length collisions. (Color figure online)

E: By making ST queries, there are more than κ marked queries where $\kappa = S \cdot \max\{1, ST^2/N\}$.

We can show that this event happens with probability at most $O(N^{-S})$, even conditioned on the adversary finding B-length collisions in all the previous rounds. When event E does not happen, there are two possibilities: 1) The B-length collisions found 'use' at least one of these (at most) κ marked queries 2) The B-length collisions found 'use' none of those κ marked queries. For case (1), we will show that some online query should hit one of (at most) $\kappa \cdot B$ offline queries en route to one of κ queries within B steps to succeed, and this happens with probability at most $O(\kappa TB/N)$. For case (2), note that it implies at least one of the two 'colliding' queries among the B-length collisions is a 'new' online query. Then, using this fact along with the structural knowledge of the type of B-length collision, we can show that probability of finding any of these types of B-length collisions is bounded by $O(STB/N + T^2/N)$.

Here, we focus on one type of B-length collisions to reiterate our strategy with more details. Refer to Sect. 4 for the complete proof. Consider the type of B-length collision depicted in Fig. 3a on input salt a.

First, as we have discussed at the beginning of the section, note that the probability that the input salt a has been queried in the offline queries is at most ST/N (as a is randomly and independently sampled). So, it suffices to focus on the case that a has not being queried during offline queries depicted in Fig. 3b. For this case, there should exist some queries (including the queries

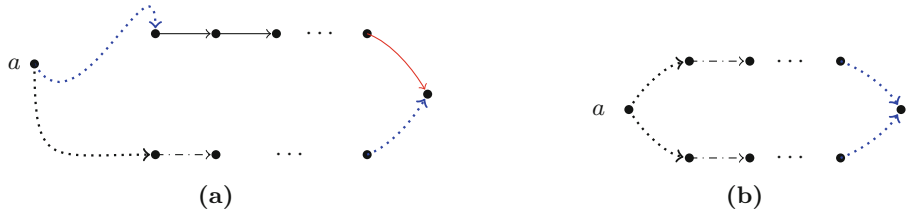


Fig. 5. The B-length collision uses no marked queries. The solid red line (if any) denotes the colliding query made in the offline phase. The dotted blue lines denote the two closest online queries to the colliding queries along the B-length collisions (they can also be colliding queries themselves). (Color figure online)

on a) along with the outputted B-length collisions that are online queries (i.e., made for the first time during the online phase).

In addition, we can also condition on event E not happening as we can show that the probability of event E is at most $O(N^{-S})$, even conditioned on the adversary winning all the previous rounds. Now observe that the queries in any found this type of B-length collisions would satisfy one of the two following possibilities:

1. The B-length collision uses some marked query.
2. None of the offline queries used by B-length collision is a marked query.

We first analyze B-length collisions with queries satisfying (1) above. Refer to Fig. 4 for a pictorial depiction of such collisions. Conditioned on event E not happening, there will be at most κ marked queries. Consider the first such query along the B-length collisions. There is a unique 'chain' consisting of at most B offline queries connecting some online query to this marked query. Thus, the probability of finding B-length collisions satisfying (1) conditioned on event $\overline{E}$ is at most the probability of some online query whose output is one of (the salts of) these κB offline queries, which is at most $O(\kappa TB/N)$.

Note that when queries in the B-length collision satisfy (2) above, it implies at least one of the 'colliding queries' (two queries denoted by red arrows in Fig. 3b) is made for the first time in the online phase.

The probability of both the colliding queries happening for the first time in the online phase (see Fig. 5b) is bounded by $O(T^2/N)$.

In the case exactly one of the colliding queries happens in the offline phase, there are at most ST possibilities for this offline colliding query. There is a unique 'chain' of at most B offline queries from some online query to this query and the output of another online query should be the output of this query (see Fig. 5a). Thus, the probability of finding such B-length collisions is bounded by $O(STB \cdot T/N \cdot T/N) = O(STB/N + T^2/N)$.

For other types of B-length collisions, we can analyze each type in a similar way. Instead of analyzing each type of B-length collisions, we further abstract out

5 conditions such that any type of B-length collisions must satisfy one of them. By considering one more "high knowledge gaining" event, and upper bounding the probability for every condition, we show that the probability of finding B-length collisions is bounded by $O(\kappa TB/N + T^2/N)$. Please see Sect. 4 for the details. It is worth noting that the S^2T^2/N term in κ cannot be further improved, because it is expected to have $\Omega(S^2T^2/N)$ marked queries among ST random oracle queries. Thus, it seems unlikely to obtain a better bound by just improving event E and its analysis.

A Detailed Comparison with Prior Techniques. The similarity between [ACDW20, GK22] and us is that we all adopt the idea of reducing the problem of interest to a multi-instance variant, in which an adversary has to solve multiple copies of the given problem.

Both [ACDW20] and [GK22] directly analyze the probability of solving all instances using the compression paradigm, which typically requires a non-trivial case analysis of the more complicated *multi-instance* problem. These case analyses may be quite laborious and detached from the single-instance problem (thus may not give many insights for the single-instance problem).

Our approach differs significantly from [ACDW20] and [GK22] in two places. First, we focus on analyzing a simple variant of the *single-instance* problem (corresponding to a single round of the sequential multi-instance game conditioning on winning previous games), which is sufficient to establish desired results in multi-instance security. This variant is more similar to the original problem, and may be easier to analyze than the multi-instance problems. The first step (reducing to a variant of the single-instance problem) is somewhat used and captured in the presampling technique (via a different route [CDGS18]). We do think this step is more modular than [ACDW20] and [GK22], but don't consider this as our main technical novelty.

The second place, also our main technical novelty, is that we further introduce "knowledge gaining events" for analyzing the variant of the single-instance problem. These events can be isolated and analyzed on their own, and precisely highlight the correlation in finding collisions given "typical" presampled random oracles. Before this work, all the presampling techniques for time-space tradeoffs considered worst-case presampled random oracles. The worst-case presampling may make the existing analyses sub-optimal. Our approach analyzes the "average-case" presampling random oracles and shows that those "worst-case" ones can never happen except with a tiny probability. To our best knowledge, this is the first work that takes advantage of "average-case" presampling and achieves tight bounds.

Overall, we consider our proofs more modular, because we utilize sequential games to focus on variants of the single-instance game (rather than directly compressing multi-instance games used by [ACDW20] and [GK22]). We further introduce "knowledge gaining events" to take advantage of "average-case" presampling (rather than working with worst-case ones used by [CDGS18]).

1.3 Discussions and Open Problems

A Better Attack or Security Bound for $ST^2 > N$? Our main result suggests that the attack by [ACDW20] is optimal when $ST^2 \leq N$, and is potentially sub-optimal when $ST^2 > N$. This attack shares many similarities with the Hellman's attack for inverting random functions. Interestingly, Hellman's attack is also known to be optimal when $ST^2 \leq N$, and is potentially sub-optimal when $ST^2 > N$. A better attack for $ST^2 > N$ will be exciting and may give insights for improving Hellman's attack. We think that our framework has the potential to prove a better security bound or even the STB-conjecture, by identifying the right set of "high knowledge gaining" events.

Tight Quantum Time-Space Tradeoffs for Finding Collisions in MD? Motivated by analyzing post-quantum non-uniform security, several recent works [CGLQ20, GLLZ21] studied the same question in the quantum setting, in which the adversary is given S-(qu)bit of advice and T quantum oracle queries. However, unlike the classical setting, no matching bounds are known, even for $B = 2$ and $B = T$. The $\Omega(ST^3/N)$ security bound by [GLLZ21], suggests that the optimal attack may speed up the trivial quantum collision finding by a factor of S. However, the best-known attack achieves $O(ST^2/N + T^3/N)$ for every $2 \leq B \leq T$. Is there a security jump for finding 2-block collisions and unbounded collisions in the quantum setting? Can we leverage our new proof for $B = 2$ to prove a tight security bound in the quantum setting?

Other Related Works. We mention that time-space lower bounds of attacks (or non-uniform security) against other fundamental cryptographic primitives, such as one-way functions, pseudorandom random generators, discrete log, have been investigated in various idealized models [DTT10, CHM20, CGK18, CGK19, GGKL21, DGK17, CDG18, CDGS18].

2 Preliminaries

Notation. For non-negative integers N, k, we write $[N]$ for $\{1, 2, \cdots, N\}$ and $\binom{[N]}{k}$ for the collection of all size-k subsets of $[N]$. For a finite set X, we write X^+ for the set of tuples of 1 or more elements of X. Random variables will be written in bold, and we write $\mathbf{x} \leftarrow_\$ X$ to indicate that $\mathbf{x}$ is a uniform random variable in X.

Chernoff Bound. Suppose $\mathbf{X}_1, \cdots, \mathbf{X}_t$ are independent binary random variables. Let $\mathbf{X}$ denote their sum and $\mu = \mathbb{E}[\mathbf{X}]$. For any $\delta \geq 0$,

$$\Pr[\mathbf{X} \geq (1+\delta)\mu] \leq \exp\left(-\frac{\delta^2 \mu}{2+\delta}\right).$$

Random Oracle [BR93]. In random oracle model, we model a hash function as a random function H that is sampled uniformly at random from all functions at the beginning. H is publicly accessible to every entity.

A useful property about random oracle model is that, instead of sampling H uniformly at random, one can assume H is initialized as a function that always outputs $\perp$; which indicates the response has not been sampled. Whenever an input x is queried and $H(x)$ has not been sampled (i.e. $H(x) = \perp$), the random oracle samples y uniformly from the range and $H(x) := y$.

Definition 1 (Lazy Sampling and Databases). *We refer to the table of sampled queries (for those $H(x) \neq \perp$) on H and their responses as the database or the partially sampled random oracle.*

*The set of **offline queries** is the set of distinct queries made in the offline stage. The set of **online queries** is the set of distinct queries made in the online stage and had not been made in the offline stage.*

While dealing with algorithms with both offline and online stages, the table of only the offline queries on H and their responses is referred to as the offline database.

Note that the outputs of the offline and online queries are independent and uniformly distributed.

2.1 Merkle-Damågard Hash Functions (MD)

A hash function usually is required to function over inputs with different lengths. Many practical hash functions are based on the Merkle-Damågard construction (MD). It takes a hash function with fixed length input to a new hash function with arbitrary input lengths.

We treat the underlying hash function as a random oracle $H : [N] \times [M] \rightarrow [N]$. We call a message $\mathbf{m}$ is a B-block message if $\mathbf{m}$ can be written as $\mathbf{m} =$

$(m_1, \cdots, m_B)$ where each $m_i \in [M]$. The function $\mathsf{MD}_H(a, \mathbf{m})$ evaluates on a salt $a \in [N]$ and a message $\mathbf{m}$ as the follows:

$$\mathsf{MD}_H(a, \mathbf{m}) = \mathsf{MD}_H^{\ell}(a, (m_1, \cdots, m_\ell)) = \begin{cases} H(\mathsf{MD}_H^{\ell-1}(a, (m_1, \cdots, m_{\ell-1})), m_\ell) & \ell > 1 \\ H(a, m_1) & \ell = 1 \end{cases}$$

It applies the fixed-length hash function H on the salt a and the first block m_1 to get a new salt a_2; it then applies H again on a_2 and m_2 until finally it outputs a single string in $[N]$.

2.2 Collision-Resistance Against Auxiliary Input (AI)

We start by defining the security game of collision-resistance against auxiliary input adversaries. The adversary is unbounded in the preprocessing stage and leave nothing but a piece of bounded-length advice for the online stage.

Definition 2 ((S, T)-AI algorithm). *A pair of algorithms $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is an $(S, T) - AI$adversary for* MD *if*

- $\mathcal{A}_1^H$ *is unbounded (making unbounded number of oracle queries to H) and outputs S bits of advice σ;*
- $\mathcal{A}_2^H$ *takes σ and a salt $a \in [N]$, issues T queries to H and outputs $\mathbf{m}_1, \mathbf{m}_2$.*

We are ready to define the security game of collision-resistance against an (S, T)-AI adversary.

Definition 3 (Auxiliary-Input Collision-Resistance). *We define the following game B-*AICR *for a fixed random oracle H and a salt $a \in [N]$ in Fig. 6, where B is a function of N (the range size of the random oracle). The game outputs* 1 *(indicating that the adversary wins) if and only if $\mathcal{A}$ outputs a pair of* MD *collision with at most $B(N)$ blocks.*

Game B-$\mathsf{AICR}_{H,a}(\mathcal{A})$
$\sigma \leftarrow \mathcal{A}_1^H$
$\mathbf{m}_1, \mathbf{m}_2 \leftarrow \mathcal{A}_2^H(\sigma, a)$
If $\mathbf{m}_1$ or $\mathbf{m}_2$ consists of more than $B(N)$ blocks
Then Return 0
If $\mathbf{m}_1 \neq \mathbf{m}_2$ and $\mathsf{MD}_H(a, \mathbf{m}_1) = \mathsf{MD}_H(a, \mathbf{m}_2)$
Then Return 1
Else Return 0

Fig. 6. B-$\mathsf{AICR}_{H,a}(\mathcal{A})$

For an (S, T)-AI adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we define the advantage of $\mathcal{A}$ as its winning probability in the B-$\mathsf{AICR}_{H,a}$ with uniformly random $H \leftarrow \{f :$

Game 2-$\mathsf{AICR}_{H,a}(\mathcal{A})$
$\sigma \leftarrow \mathcal{A}_1^H$
$\mathbf{m}_1, \mathbf{m}_2 \leftarrow \mathcal{A}_2^H(\sigma, a)$
If $\mathbf{m}_1$ or $\mathbf{m}_2$ consists of more than 2 blocks
Then Return 0
If $\mathbf{m}_1 \neq \mathbf{m}_2$ and $\mathsf{MD}_H(a, \mathbf{m}_1) = \mathsf{MD}_H(a, \mathbf{m}_2)$
Then Return 1
Else Return 0

Fig. 7. 2-$\mathsf{AICR}_{H,a}(\mathcal{A})$

$[N] \times [M] \to [N]\}$ *and random* $a \leftarrow [N]$. *We define the* (S, T, B)*-auxiliary-input collision-resistance of Merkle-Damgård, denoted by* $\mathsf{Adv}^{\text{AI-CR}}_{\text{B-MD}}(S, T)$, *as the maximum of advantage taken over all* (S, T)*-AI adversaries* $\mathcal{A}$.

For convenience, we similarly define $\mathsf{Adv}^{\text{AI-CR}}_{\text{2-MD}}(S, T)$ *as the maximum of advantage of winning the game* 2-AICR *(see Fig. 7) taken over all* (S, T)*-AI adversaries* $\mathcal{A}$.

Multi-Instance Collision-Resistance (MI). We then define the sequential multi-instance collision-resistance of Merkle-Damgård. As shown by [CGLQ20], the AI-security is closely related to the (sequential) MI-security. Note that in the MI security, an adversary does not take any advice but tries to solve independent instances sequentially.

Definition 4 (Multi-Instance Collision-Resistance). *Fixing functions* B *and* S, *and a random oracle* H, *we define the following game* B-MICR^S *in Fig. 8. In this game,* $\mathcal{A}$ *will receive* S *freshly independent and uniform salts and it needs to find a* MD *collision with respect to each salt* a_i *of at most* B *blocks,* in a sequential order. *In other words,* $\mathcal{A}$ *will never see the next challenge salt until it solves the current one.*

Game B-$\mathsf{MICR}^S_{H,a}(\mathcal{A})$
For $i \in \{1, 2, \cdots, S\}$:
Sample $a_i \leftarrow [N]$
$\mathbf{m}_1, \mathbf{m}_2 \leftarrow \mathcal{A}^H(a_i)$
If $\mathbf{m}_1$ or $\mathbf{m}_2$ consists of more than B blocks,
or $\mathsf{MD}_H(a_i, \mathbf{m}_1) \neq \mathsf{MD}_H(a_i, \mathbf{m}_2)$
Return 0
Return 1

Fig. 8. Games B-$\mathsf{MICR}^S_{H,a}(\mathcal{A})$.

In this security game, $\mathcal{A}$ is a stateful algorithm that maintains its internal state between each stage. We usually consider an (S,T)-MI adversary $\mathcal{A}$ which makes at most T queries in each of these S stages. We similarly define 2-MICR *by setting $B = 2$ in B-MICR.*

For an (S,T)-MI adversary $\mathcal{A}$, we define the advantage of $\mathcal{A}$ as its winning probability in the B-$\mathsf{MICR}^S_{H,a}$ with uniformly random H and $a \leftarrow [N]$.

We define the (S,T,B)-multi-instance collision-resistance of Merkle-Damgård, denoted by $\mathsf{Adv}^{\text{MI-CR}}_{\text{B-MD}}(S,T)$, as the maximum of advantage taken over all (S,T)-MI adversaries $\mathcal{A}$.

For convenience, we similarly define $\mathsf{Adv}^{\text{MI-CR}}_{\text{2-MD}}(S,T)$ as the maximum of advantage of winning the game 2-$\mathsf{MICR}^S_{H,a}$ (for random H,a) taken over all (S,T)-MI adversaries $\mathcal{A}$.

The following theorem will be useful for proving the AI collision-resistance of Merkle-Damgård. It says a lower bound for the MI collision-resistance implies a lower bound for the AI security. Therefore, in the rest of the paper, we will focus on the MI collision-resistance of Merkle-Damgård with different lengths B. The theorem is based on the idea of Theorem 4.1 in [CGLQ20], which implies that if $\mathsf{Adv}^{\text{MI-CR}}_{\text{B-MD}}(S + \log N + 1, T) \le \delta^{S+\log N+1}$, then $\mathsf{Adv}^{\text{AI-CR}}_{\text{B-MD}}(S,T) \le 4\delta$. We slightly improve their parameter, and obtain a considerably cleaner statement.

Theorem 3. *For any S, T, B and $0 \le \delta \le 1$, if $\mathsf{Adv}^{\text{MI-CR}}_{\text{B-MD}}(S,T) \le \delta^S$, then $\mathsf{Adv}^{\text{AI-CR}}_{\text{B-MD}}(S,T) \le 2\delta$.*

Proof of Theorem 3. We prove by contradiction. Assume there is an (S,T)-AI adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ such that

$$\Pr_{H,a}\left[B\text{-}\mathsf{AICR}_{H,a}(\mathcal{A}) = 1\right] > 2\delta,$$

Consider the following (S,T)-MI adversary $\mathcal{B}$:

1. $\mathcal{B}$ samples a uniformly random σ of S bits.
2. For each stage $i \in [S]$:
   - $\mathcal{B}$ receives a_i from the challenger.
   - $\mathcal{B}$ runs $\mathcal{A}_2^H(\sigma, a_i)$ to obtain and output $\mathbf{m}_1, \mathbf{m}_2$.

We will show that $\Pr_{H,a_1,\ldots,a_S}\left[B\text{-}\mathsf{MICR}^S_H(\mathcal{B}) = 1\right] > \delta^S$. For every fixed choice of H, we define

$$\delta_H := \Pr_a\left[B\text{-}\mathsf{AICR}_{H,a}(\mathcal{A}) = 1\right].$$

Observe that $\mathbb{E}_H[\delta_H] = \Pr_{H,a}\left[B\text{-}\mathsf{AICR}_{H,a}(\mathcal{A}) = 1\right] > 2\delta$. For every fixed choice of H, conditioning on that $\mathcal{B}$ guesses the output of $\mathcal{A}_1^H$ correctly, then $\mathcal{B}$ perfectly simulates $\mathcal{A}$. Therefore,

$$\Pr_{a_1,\ldots,a_S}\left[B\text{-}\mathsf{MICR}_H(\mathcal{B}) = 1\right] \ge \Pr_{a_1,\ldots,a_S}\left[B\text{-}\mathsf{MICR}_H(\mathcal{B}) = 1 \mid \sigma = \mathcal{A}_1^H\right] \cdot \Pr[\sigma = \mathcal{A}_1^H] = \delta_H^S / 2^S.$$

By averaging over the randomness of H,

$$\Pr_{H,a_1,\ldots,a_S}[B\text{-}\mathsf{MICR}_{H,a}(\mathcal{B})=1] \geq \mathbb{E}_H[\delta_H^S]/2^S \geq \mathbb{E}[\delta_H]^S/2^S > \delta^S \ ,$$

where the second inequality is by Jensen's inequality, and the last inequality is by $\mathbb{E}_H[\delta_H] > 2\delta$. □

3 Auxiliary Input Collision Resistance for $B = 2$ Merkle-Damgård

In this section we prove the following theorem, which recovers Theorem 7 in [ACDW20].

Theorem 4. *For any S, T and $N \geq 64$,*

$$\mathsf{Adv}^{\text{AI-CR}}_{\text{2-MD}}(S,T) \leq (200\log^2 N)\cdot\frac{ST+T^2}{N} \ .$$

By Theorem 3, it suffices to prove the following lemma.

Lemma 1. *For any S, T and $N \geq 64$, $\mathsf{Adv}^{\text{MI-CR}}_{\text{2-MD}}(S,T) \leq \frac{100(ST+T^2)\log^2 N}{N}$.*

The purpose of this section is to show the simplicity of our new framework. The proof will also serve as a stepping stone for a better understanding of our proof for larger B cases.

Proof of Lemma 1. Let H be a random oracle in the game 2-MICR^S and $\mathcal{A}$ be an arbitrary (S,T)-MI adversary. We show that its advantage of succeeding in 2-MICR^S is at most $(100(ST+T^2)\log^2 N/N)^S$. In this proof, we will also assume the random oracle H is lazily sampled by the challenger, which is equivalent to being sampled at the very beginning.

Let $\mathbf{X}_i$ be the indicator variable that $\mathcal{A}$ wins the i-th stage on a uniformly random salt a_i. The advantage of $\mathcal{A}$ can be then written as $\Pr[\mathbf{X}_1 \wedge \cdots \wedge \mathbf{X}_S]$. We additionally define the indicator variable $\mathbf{X}_{<i} = \mathbf{X}_1 \wedge \cdots \wedge \mathbf{X}_{i-1}$, meaning whether $\mathcal{A}$ wins the first $(i-1)$ stages of the sequential game. Then

$$\Pr[\mathbf{X}_1 \wedge \ldots \wedge \mathbf{X}_S] = \prod_{i=1}^{S} \Pr[\mathbf{X}_i|\mathbf{X}_{<i}]. \tag{1}$$

We will bound $\Pr[\mathbf{X}_{<i+1}] < (\delta_S)^i$ for each $i \in \{1,\cdots,S\}$ by induction, where $\delta_S = 100\cdot\frac{(ST+T^2)\log^2 N}{N}$.

If $\Pr[\mathbf{X}_{<i}]$ is already bounded by $(\delta_S)^i$, then it trivially holds for $\Pr[\mathbf{X}_{<i+1}]$. Otherwise, we assume $\Pr[\mathbf{X}_{<i}] \geq (\delta_S)^i$.

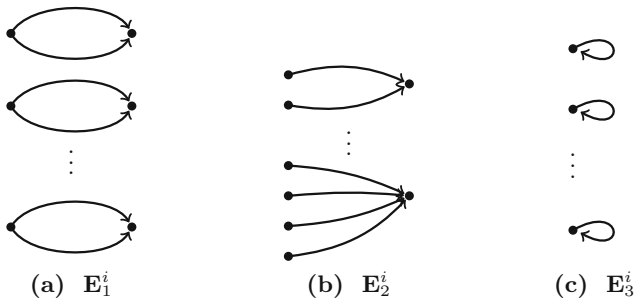
We want to bound $\Pr[\mathbf{X}_i|\mathbf{X}_{<i}] \leq \delta_S$ for any arbitrary $i \in [S]$. In the following proof, we will carefully deal with the conditioning on $\mathbf{X}_{<i}$, since $\mathcal{A}$ learns about the function H not only using the T queries in the i-th stage, but also from

these $(i-1)T$ queries in the early stages. We will call all the queries made in the previous $(i-1)$ stages as "offline" queries and those made in the i-th stage as "online" queries. We also recall the definition for "databases" in Definition 1.

As mention in the introduction, one bad example is that the previous $(i-1)T$ queries consist of $(i-1)T/2$ distinct salts, each has a pair of 1-block collision. An online adversary can use T queries to hit any of these salts and form a 2-block collision with probability roughly iT^2/N. Below, we will show that this event (and other events that give non-trivial advantage to the online adversary) happens with very small probability.

Defining Knowledge-Gaining Events. To bound the knowledge that $\mathcal{A}$ learns in the previous stages, we define the following events: all events are defined for the lazily sampled random oracle right after the first $(i-1)$ stages. We are going to show that these events are the "only events" that $\mathcal{A}$ can learn take advantage of the previous queries but they happen with very small probability.

- Let $\mathbf{E}_1^i$ be the event that 1-block collisions can be found for at least $10i \log N$ distinct salts within $(i-1)T$ queries.
  Formally, in the database, there exist $10i \log N$ salts: for each such salt a, there exists $m \neq m' \in [N]$ satisfying $H(a,m) = H(a,m')$. See Fig. 9a.



(a) $\mathbf{E}_1^i$ **(b)** $\mathbf{E}_2^i$ **(c)** $\mathbf{E}_3^i$

Fig. 9. All events $\mathbf{E}_1^i, \mathbf{E}_2^i, \mathbf{E}_i^3$. Nodes indicate salts in $[N]$. An arrow connected two salts means there is a query on the starting salt and a message in $[M]$, and the output is the other salt.

- Let $\mathbf{E}_2^i$ be the event that at least $10i^2 \log^3 N$ pairs of block collisions can be found within $(i-1)T$ queries.
  Formally, in the database, there exist $10i^2 \log^3 N$ pairs of inputs $(a,m) \neq (a',m')$ satisfying $H(a,m) = H(a',m')$. We emphasize that we do not ask a pair of collision to start with distinct salts. See Fig. 9b.
- Let $\mathbf{E}_3^i$ be the event that self loops can be found for at least $10i \log N$ distinct salts within $(i-1)T$ queries.
  Formally, in the database, there exist $10i \log N$ distinct salts: for each such salt a, there exists some $m \in [N]$ satisfying $H(a,m) = a$. See Fig. 9c.

Then

$$\begin{aligned}\Pr[\mathbf{X}_i|\mathbf{X}_{<i}] &\le \Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_1^i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}] + \Pr[\mathbf{E}_1^i \vee \mathbf{E}_2^i \vee \mathbf{E}_3^i|\mathbf{X}_{<i}] \\ &\le \Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_1^i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}] + \frac{\Pr[\mathbf{E}_1^i]}{\Pr[\mathbf{X}_{<i}]} + \frac{\Pr[\mathbf{E}_2^i]}{\Pr[\mathbf{X}_{<i}]} + \frac{\Pr[\mathbf{E}_3^i]}{\Pr[\mathbf{X}_{<i}]}.\end{aligned}$$

Here we use the fact that $\Pr[\mathbf{A}|\mathbf{B}] \le \Pr[\mathbf{A}]/\Pr[\mathbf{B}]$ for $\Pr[\mathbf{B}] > 0$.

Next, we will show that assuming none of $\mathbf{E}_1^i, \mathbf{E}_2^i, \mathbf{E}_3^i$ happens, an adversary can not take too much advantage of the information from the previous stages. We show that its advantage $\Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_1^i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}]$ is bounded by $98 \cdot (ST + T^2) \log^2 N/N$. Secondly, any of these event happens with very small probability. We can safely "assume" these events never happen. In total, the conditional probability is at most $100 \cdot (ST + T^2) \log^2 N/N = \delta_S$.

Claim 1. *For any $i \in [S]$ and $T^2 \le N/2$, $Pr[\mathbf{E}_1^i] \le N^{-10i}$.*

Claim 2. *For any $i \in [S]$, $iT + T^2 < N/2$ and $N \ge 64$, $Pr[\mathbf{E}_2^i] \le 4N^{-2i}$.*

Claim 3. *For any $i \in [S]$, $N \ge 4$ and $T \le N/2$, $Pr[\mathbf{E}_3^i] \le N^{-4i}$.*

The proofs for these lemma are in the full version of the paper. Readers may skip the proofs for all these claims. The proofs are not necessary for understanding the rest of the proof.

Recall that we assume $\Pr[X_{<i}] \ge (\delta_S)^i$, otherwise $\Pr[\mathbf{X}_1 \wedge \ldots \wedge \mathbf{X}_i] \le (\delta_S)^i$ holds trivially for the first i stages. Therefore,

$$\Pr[\mathbf{X}_i|\mathbf{X}_{<i}] \le \Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_1^i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}] + \frac{\Pr[\mathbf{E}_1^i]}{\Pr[\mathbf{X}_{<i}]} + \frac{\Pr[\mathbf{E}_2^i]}{\Pr[\mathbf{X}_{<i}]} + \frac{\Pr[\mathbf{E}_3^i]}{\Pr[\mathbf{X}_{<i}]} \tag{2}$$

$$\le \Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_1^i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}] + \frac{1}{N}, \tag{3}$$

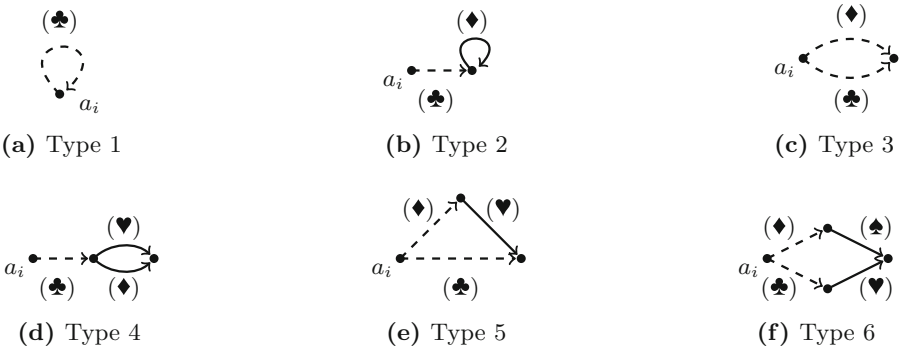
where the last inequality comes from the fact that $1/\Pr[\mathbf{X}_{<i}] \le N^i$ but $(\Pr[\mathbf{E}_1^i] + \Pr[\mathbf{E}_2^i] + \Pr[\mathbf{E}_3^i]) \le 6N^{-2i}$.

Bounding the Last Term. Finally, we are going to bound $\Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_1^i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}]$. In order to do that, we define another event $\mathbf{G}$ as the event that the input salt a_i has been queried among the queries in the previous $(i-1)$ iterations; i.e., for some $m \in [N]$, (a_i, m) is in the lazily sampled hash function. Then it holds that:

$$\begin{aligned}&\Pr\left[\mathbf{X}_i \middle| \mathbf{X}_{<i} \wedge \overline{\mathbf{E}_1^i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}\right] \\ \le &\Pr\left[\mathbf{G} \middle| \mathbf{X}_{<i} \wedge \overline{\mathbf{E}_1^i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}\right] + \Pr\left[\mathbf{X}_i \middle| \mathbf{X}_{<i} \wedge \overline{\mathbf{E}_1^i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i} \wedge \overline{\mathbf{G}}\right] \\ \le &\frac{(i-1)T}{N} + \Pr\left[\mathbf{X}_i \middle| \mathbf{X}_{<i} \wedge \overline{\mathbf{E}_1^i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i} \wedge \overline{\mathbf{G}}\right].\end{aligned}$$

Now all that remains to bound is $\Pr\left[\mathbf{X}_i \middle| \mathbf{X}_{<i} \wedge \overline{\mathbf{E}_1^i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i} \wedge \overline{\mathbf{G}}\right]$, which requires collision type-wise analysis. By enumeration, there are total 6 types of 2-block collisions (Fig. 10).

A dashed line origins from a_i. It indicates that the query should be made online, conditioned on $\overline{\mathbf{G}}$. Other queries can be either made online or offline in the previous iterations. The label ♣, ♦, ♥ and ♠ will be used later for a better presentation of our proof. By enumerating each solid edge being an online query or a offline query, we show that it is sufficient to consider the cases in Claim 4.



(a) Type 1 **(b)** Type 2 **(c)** Type 3

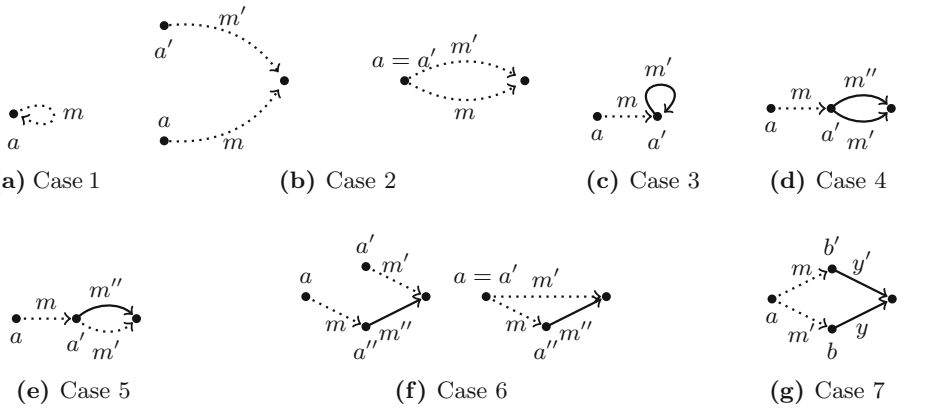
(d) Type 4 **(e)** Type 5 **(f)** Type 6

Fig. 10. All types of 2-block collisions.

Claim 4. *For any $i \in [S]$, to find a 2-block collision on a_i conditioned on $\overline{\mathbf{G}}$, the queries should satisfy at least one of the following conditions:*

1. *There exists an online query (i.e., a query among the T queries in the i-th iteration after receiving the challenge input a_i), denoted (a, m) such that $H(a, m) = a$.*
   In other words, a self loop is found among the online queries. This covers the case when (♣) edge in type 1 collisions and the (♦) edge in type 2 collisions are online queries. See Fig. 11a.
2. *There exists two online queries, denoted (a, m) and (a', m'), such that $(a, m) \neq (a', m')$ and $H(a, m) = H(a', m')$.*
   A collision is found among the online queries. This covers the case when the (♣) and (♦) edges in Type 3 collisions, the (♦) and (♥) edges in Type 4 collisions, the (♣) and (♥) edges in Type 5 collisions, the (♥) and (♠) edges in Type 6 collisions are online queries. See Fig. 11b.
3. *There exists an online query, denoted by (a, m), and one offline query, denoted by (a', m'), such that $a \neq a'$, $H(a, m) = a'$ and $H(a', m') = a'$.*
   This denotes an online query hits an existing self loop. This covers the case when the (♣) edge in type 2 collisions is an online query. See Fig. 11c.

4. *There exists an online query, denoted by* (a, m)*, and two offline queries, denoted by* (a', m') *and* (a', m'')*, such that* $a \neq a'$*,* $H(a, m) = a'$ *and* $H(a', m') = H(a', m'')$*.*
   This denotes an online query hits an existing collision (starting with the same salt a'*). This covers the case when (♣) edge in type 4 collisions is an online query. See Fig. 11d.*
5. *There exists two online queries, denoted by* (a, m) *and* (a', m')*, and an offline query, denoted by* (a', m'') *such that* $a \neq a'$*,* $H(a, m) = a'$ *and* $H(a', m') = H(a', m'')$*.*
   This covers the case when the (♣) and (♦) edges in type 4 collisions are online queries. See Fig. 11e.
6. *There exists two online queries, denoted by* (a, m) *and* (a', m')*, and an offline query, denoted by* (a'', m'') *such that* $H(a, m) = a'$ *and* $H(a', m') = H(a'', m'')$*.*
   This denotes two online queries hit two ends of an existing queries. This covers the case when the (♣) and (♦) edges in type 5 collisions, the (♣) and (♠) edges in type 6 collisions are online queries. See Fig. 11f.
7. *There exists two online queries, denoted by* (a, m) *and* (a, m')*, and two offline queries, denoted by* $(b, y), (b', y')$ *such that* $b \neq b'$*,* $H(a, m) = b, H(a, m') = b'$ *and* $H(b, y) = H(b', y')$*.*
   This covers the case when the (♣) and (♦) edges in type 6 collisions are online queries. See Fig. 11g.



(a) Case 1 **(b)** Case 2 **(c)** Case 3 **(d)** Case 4

(e) Case 5 **(f)** Case 6 **(g)** Case 7

Fig. 11. All possible types of collisions. A dotted line denotes an online query. A solid line denotes a offline query.

Proof for Claim 4. We only prove for type 6 collisions. Other five cases are easier and similar.

When both (♥) and (♠) are offline queries, it is Case 7. If only one of the two edges is offline, it is Case 6. If they are all online queries, we can reduce it to Case 2. □

Finally, we show that for each case in Claim 4, the advantage is bounded by $(98(ST + T^2)\log^2 N)/N$.

Case 1. By making T new queries, each query (a, m) has $1/N$ chance to satisfy $H(a, m) = a$. Therefore, the probability is bounded by T/N.
Case 2. The probability of finding a collision among these T new queries is smaller than T^2/N, by birthday bound.
Case 3. Recall $\overline{\mathbf{E}_3^i}$: there are at most $10i \log N$ salts that has a self loop in the offline queries. By making T new queries, each query (a, m) has $(10i \log N)/N$ chance to hit any of these salts. Therefore, the probability is bounded by $(10iT \log N)/N$.
Case 4. Recall $\overline{\mathbf{E}_1^i}$: there are at most $10i \log N$ salts that has a collision starting from it in the offline queries. By making T new queries, each query (a, m) has $(10i \log N)/N$ chance to hit any of these salts. Therefore, the probability is bounded by $(10iT \log N)/N$.
Case 5. and **Case 6**. The proofs are identical. Fixing any offline query (a'', m''), by making T queries, the chance of hitting both ends is T^2/N^2. This is because we can enumerate which are the first queries that hit the starting salt a'' and the end $H(a'', m'')$. Each case happens w.p. at most $1/N^2$.
Since there are total $(i-1)T$ offline queries, by union bound, the advantage is at most $(i-1)T \cdot T^3/N^2 \leq \frac{iT}{N} \cdot \frac{T^2}{N}$ for both cases.
Case 7. Recall $\overline{\mathbf{E}_2^i}$: there are at most $10i^2 \log^3 N$ pair-wise collisions. For every such collision that start with different salts, the probability of hitting both salts within T queries is T^2/N^2. This is due to the same counting argument in the analysis of Case 5 and Case 6.
By union bound, the advantage is at most $(10i^2T^2 \log^3 N)/N^2$.

We have shown all the cases in Claim 4. Therefore,

$$\Pr[\mathbf{X}_i | \mathbf{X}_{<i} \wedge \overline{\mathbf{E}_1^i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}] \leq \frac{98(iT + T^2)\log^2 N}{N}.$$

Combining with Eq. (1) and Eq. (2), we conclude Lemma 1: $\Pr[\mathbf{X}_1 \wedge \ldots \wedge \mathbf{X}_S] \leq (\delta_S)^S$. □

4 Auxiliary Input Collision Resistance for B Merkle-Damgård

In this section we prove the following theorem.

Theorem 5. *For any functions S, T, B, and $N \geq 64$*

$$\mathsf{Adv}^{\text{AI-CR}}_{\text{B-MD}}(S, T) \leq (34 \log^2 N) \cdot \frac{STB}{N} \cdot \max\left\{1, \frac{ST^2}{N}\right\} + 2 \cdot \frac{T^2}{N}.$$

Lemma 2. *For any functions S, T, B, and $N \geq 64$,*

$$\mathsf{Adv}^{\text{MI-CR}}_{\text{B-MD}}(S,T) \leq \left(\frac{17\kappa TB\log^2 N + T^2}{N}\right)^S$$

where $\kappa = S \cdot \max\{1, ST^2/N\}$.

As for the case of $B = 2$, we prove an upper bound on the advantage of B-block collision finding adversary in the MI-CR model, which implies an upper bound in the AI-CR model via Theorem 3.

Proof of Lemma 2. We prove this lemma in similar fashion as Lemma 1. Let H be a random oracle (which is lazily sampled) in the game B-MICR^S and $\mathcal{A}$ be any (S,T)-MI adversary.

We analogously define $\mathbf{X}_i$ to be the indicator variable that $\mathcal{A}$ finds at most B-length collisions on uniformly random salt a_i given as input in the i-th stage of the game. We also define $\mathbf{X}_{<i} = \mathbf{X}_1 \wedge \cdots \wedge \mathbf{X}_{i-1}$. So, the advantage of $\mathcal{A}$ is

$$\Pr[\mathbf{X}_1 \wedge \ldots \wedge \mathbf{X}_S] = \prod_{i=1}^{S} \Pr[\mathbf{X}_i | \mathbf{X}_{<i}].$$

As in the proof for $B = 2$ case, we will inductively bound $\Pr[\mathbf{X}_{<i+1}]$ for each $i \in [S]$. Here we will bound $\Pr[\mathbf{X}_{<i+1}]$ to $((17\kappa_i TB\log^2 N + T^2)/N)^i$ where $\kappa_i = i \cdot \max\{1, iT^2/N\}$. Recall that we will analogously assume $\Pr[\mathbf{X}_{<i}] \geq ((17\kappa_i TB\log^2 N + T^2)/N)^i$. Otherwise $\Pr[\mathbf{X}_{<i+1}] \leq ((17\kappa_i TB\log^2 N + T^2)/N)^i$ holds trivially.

In order to prove the lemma, it suffices to upper bound $\Pr[\mathbf{X}_i|\mathbf{X}_{<i}]$ by $17\kappa_i TB\log^2 N/N + T^2/N$ for any arbitrary $i \in [S]$. That is because $\Pr[\mathbf{X}_{<i+1}] = \Pr[\mathbf{X}_i|\mathbf{X}_{<i}] \cdot \Pr[\mathbf{X}_{<i}]$ where $\Pr[\mathbf{X}_{<i}] \leq ((17\kappa_i TB\log^2 N + T^2)/N)^{i-1}$ by the inductive hypothesis. In the proof, we will handle the conditioning on $\mathbf{X}_{<i}$ in a similar fashion to our proof for $B = 2$ case.
First we state some useful definitions.

Definition 5. *A list of elements $(a_1, m_1), \ldots, (a_\ell, m_\ell)$ in $[N] \times [M]$ are said to form a chain for H when for every $j \in [\ell - 1]$, $H(a_j, m_j) = a_{j+1}$.*

A chain $(a_1, m_1), \ldots, (a_\ell, m_\ell)$ for H is called a cycle when $H(a_\ell, m_\ell) = a_1$. The length of a cycle is the number of elements in it, ℓ here.

Definition 6. *Two distinct chains $(a_1, m_1), \ldots, (a_\ell, m_\ell)$ and $(a'_1, m'_1), \ldots, (a'_{\ell'}, m'_{\ell'})$ are called colliding chains for H if $H(a_\ell, m_\ell) = H(a'_{\ell'}, m'_{\ell'})$.*

Definition 7. *For any $a \in [N]$, a set of elements $(a_1, m_1), \ldots, (a_\ell, m_\ell)$ in $[N] \times [M]$ are said to form a claw at a under H if $\ell > 1$, $a_1, \ldots, a_\ell$ are distinct and $H(a_1, m_1) = \ldots = H(a_\ell, m_\ell) = a$. We refer to $a_1, \ldots, a_\ell$ as the pre-images of a.*

Next, we define events to illustrate the bound on 'useful' information gained by $\mathcal{A}$ from the prior iterations in the B-MICR game. Each of these events are defined over responses from the random oracle in the first $(i-1)$ iterations.

- Let Y be the set of salts with more than one pre-image on it in the offline database. Then we define $\mathbf{E}_2^i$ to be the event that $\sum_{a \in Y}$ (# pre-images on a) $\geq 16\kappa_i \log^2 N$ after $(i-1)T$ queries where $\kappa_i = \max\left\{i, \frac{i^2T^2}{N}\right\}$.
- Let $\mathbf{E}_3^i$ be the event that there exists at least $i \log N$ 'special' cycles of length in $[B-1]$ among the $(i-1)T$ offline queries. A cycle $(a_1, m_1), \ldots, (a_\ell, m_\ell)$ is called 'special' if the number of pre-images on a_i is exactly 1 for every $i \in [\ell]$.

Next, we can write

$$\begin{aligned}
\Pr[\mathbf{X}_i|\mathbf{X}_{<i}] &= \Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}] + \Pr[\mathbf{E}_2^i \vee \mathbf{E}_3^i|\mathbf{X}_{<i}] \\
&\leq \Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}] + \frac{\Pr[\mathbf{E}_2^i]}{\Pr[\mathbf{X}_{<i}]} + \frac{\Pr[\mathbf{E}_3^i]}{\Pr[\mathbf{X}_{<i}]} \\
&\leq \Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}] + \frac{1}{N}
\end{aligned}$$

where the last inequality holds via Claim 5, Claim 6 (which are stated next) and our assumption that $\Pr[\mathbf{X}_{<i}] \geq ((17\kappa_i TB \log^2 N + T^2)/N)^i$.

Claim 5. *For any $i \in [S]$, $iT + T^2 < N/2$, $2i \log N + 1 \leq N/2$ and $N \geq 64$, $Pr[\mathbf{E}_2^i] \leq \frac{5}{N^{2i}}$.*

Claim 6. *For any $i \in [S]$, $Pr[\mathbf{E}_3^i] \leq \left(\frac{T}{N}\right)^{i \log N}$.*

As before, we will prove Claim 5 and 6 in the full version of the paper. Readers may safely skip the proofs and assume these "knowledge-gaining events" happen with exponentially small probability.

Next, we want to study $\Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}]$. We define $\mathbf{G}$ to be the event that input salt a_i has been queried among the previous $(i-1)$ iterations or that input salt a_i is the output of some query among the previous $(i-1)$ iterations. So, we can rewrite $\Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}]$ as follows:

$$\begin{aligned}
\Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}] &\leq \Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i} \wedge \overline{\mathbf{G}}] + \Pr\left[\mathbf{G}\middle|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}\right] \\
&\leq \Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i} \wedge \overline{\mathbf{G}}] + \frac{2(i-1)T}{N}.
\end{aligned}$$

Note that a_i is chosen uniformly and independently and as queries in the previous iterations could be made on at most $(i-1)T$ distinct salts and can output at most $(i-1)T$ distinct salts in the previous $(i-1)$ iterations, it is easy to bound

$$\Pr\left[\mathbf{G}\middle|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}\right] \leq \frac{2(i-1)T}{N}.$$

Finally, we analyze $\Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i} \wedge \overline{\mathbf{G}}]$.

Claim 7. *For any $i \in [S]$,*

$$Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i} \wedge \overline{\mathbf{G}}] \leq \frac{16\kappa_i TB\log^2 N + T^2}{N}.$$

Proof of Claim 7 requires different analysis for different types of colliding chains which we show in Subsect. 4.1. Before we move onto that subsection, we first show how we obtain the lemma by putting together all the claims.

$$\begin{aligned}\Pr[\mathbf{X}_i|\mathbf{X}_{<i}] &\leq \Pr[\mathbf{X}_i|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i} \wedge \overline{\mathbf{G}}] + \Pr\left[\mathbf{G}\middle|\mathbf{X}_{<i} \wedge \overline{\mathbf{E}_2^i} \wedge \overline{\mathbf{E}_3^i}\right] + \Pr[\mathbf{E}_2^i \vee \mathbf{E}_3^i|\mathbf{X}_{<i}] \\ &\leq \frac{16\kappa_i TB\log^2 N + T^2}{N} + \frac{2(i-1)T}{N} + \frac{1}{N} \\ &\leq \frac{17\kappa_i TB\log^2 N}{N} + \frac{T^2}{N}\end{aligned}$$

where the last inequality holds from that $\kappa_i = \max\{i, i^2T^2/N\}$ and $N \geq 4$.

4.1 Proof of Claim 7

To this end, we state the following claim.

Claim 8. *For any $i \in [S]$, to find a B-length collision on a_i, the queries in the database should satisfy at least one of the following conditions given there exists no query in the offline database that takes a_i as input or outputs a_i:*

1. *There exists an online query (i.e., a query among at most T queries that were made for the first time in the i-th iteration after receiving the challenge input a_i), denoted (a, m) such that $H(a, m) = a_i$.*
2. *There exists two distinct online queries, denoted (a, m) and (a', m') such that $H(a, m) = H(a', m')$.*
   This includes both of the following possibilities: the online queries are such (1) $a = a'$ (and thus m and m' will be distinct); (2) $a \neq a'$.
3. *There exists an online query, denoted (a, m), a chain (recall Definition 5) of offline queries[5], denoted $(b_1, m_1), \ldots, (b_\ell, m_\ell)$ for some $0 < \ell < B$, and an offline query $(b, m') \neq (b_\ell, m_\ell)$ such that $H(a, m) = b_1$, $H(b, m') = H(b_\ell, m_\ell)$ and the number of pre-images for every salt in $\{b_2, \ldots, b_\ell\}$ in the offline database is exactly 1.*

[5] The set of **Offline queries** is the set of distinct queries made in the previous $(i-1)$ iterations. So there are at most $(i-1)T$ of these queries and their outputs are independent and uniformly distributed. The set of **Online queries** is the set of distinct queries made in the i-th iteration after receiving the challenge input a_i that had not been made in any of the previous $(i-1)$ iterations. Note that the outputs of online queries are also independent and uniformly distributed.

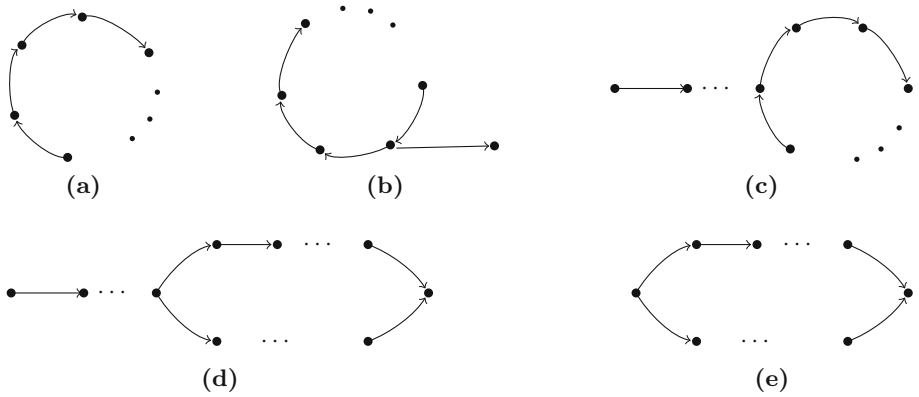


Fig. 12. All types of colliding chains

4. *There exists two online queries, denoted* (a, m) *and* (a', m')*, and a chain of offline queries, denoted* $(b_1, m_1), \ldots, (b_\ell, m_\ell)$ *for some* $\ell < B$*, such that* $H(a, m) = b_1$*,* $H(a', m') = H(b_\ell, m_\ell)$ *and the number of pre-images on every salt in* $\{b_2, \ldots, b_\ell\}$ *in the offline database is exactly* 1*.*
5. *There exists an online query, denoted* (a, m)*, and a cycle in the offline database, denoted* $(b_1, m_1), \ldots, (b_\ell, m_\ell)$ *for some* $\ell < B$*, such that* $H(a, m) = b_1$ *and the number of pre-images on every salt in* $\{b_1, b_2, \ldots, b_\ell\}$ *in the offline database is exactly* 1*.*

Proof for Claim 8. Figure 12 enumerates all the possible types of colliding chains. Depending on where the queries in the chains are first made for each of the types, we show that the list of conditions in the claim is complete. (Refer to Fig. 13 for a visual representation of the conditions in the claim.)

We know that all the queries with output a_i or of the form $(a_i, \cdot)$ in the colliding chains are online queries. This implies if the colliding chains are of the types in Fig. 12a or 12b, the queries in the database will satisfy condition 1.

For the remaining types of colliding chains (ref Fig. 12c, 12d, 12e), one of the following 3 cases can happen:

1. **Both the 'colliding' queries are online.** In this case, the queries in the database will satisfy condition 2.
2. **Both the 'colliding' queries are offline.** In this case, the queries in the database will satisfy condition 3. Note that b_ℓ can be thought of as the earliest query among the chains that has more than one pre-image in the offline database.
3. **One of the 'colliding' queries is offline and online each.** For the colliding chains of types in Fig. 12d and 12e), the queries in the database will satisfy condition 4. For the colliding chains of type in Fig. 12c, there are two possibilities as shown in Fig. 14. For the possibility in Fig. 14a, the queries

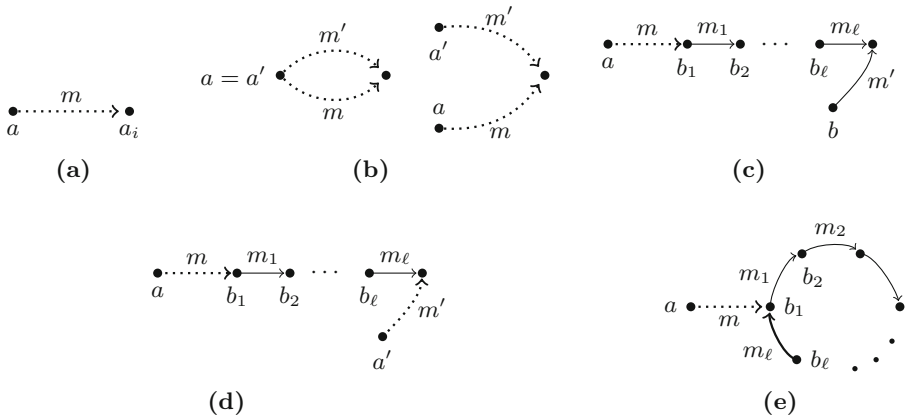


Fig. 13. Pictorial depiction of Conditions 1–5. A dotted line denotes an online query. A solid line denotes an offline query.

Fig. 14. A dotted line denotes an online query. A solid line denotes an offline query.

in the database satisfy condition 4. On the other hand, for the possibility in Fig. 14b, the queries in the database satisfy condition 5.

□

Claim 9. *For $j \in [5]$, let ϵ_j be the advantage in achieving condition j from Claim 8 when $\overline{\mathbf{E}_2^i}$, $\overline{\mathbf{E}_3^i}$ and $\overline{\mathbf{G}}$ hold. Then for any $i \in [S]$, the results summarized in Table 2 on the upper bounds of ϵ_j hold.*

Table 2. Summary of upper bounds on ϵ_j for $j \in [5]$ where $\kappa_i := \max\{i, i^2T^2/N\}$.

Condition j	1	2	3	4	5
ϵ_j	$\frac{T}{N}$	$\frac{T^2}{N}$	$\frac{16\kappa_i TB\log^2 N}{N}$	$\frac{iT}{N} \cdot \frac{T^2}{N}$	$\frac{iTB\log N}{N}$

We prove the bounds stated in Claim 9 next.

Condition 1. Recall that online queries are 'new' queries, as in they are made for the first time among the T queries in the i-th iteration after receiving a_i. Thus, the output of online queries is independent of output from offline queries and has $1/N$ chance to be a_i under H via lazy sampling. By taking a union bound over at most T online queries, we can bound the probability to T/N.

Condition 2. By birthday bound, it holds that the probability of finding 'colliding' queries among T online queries is at most T^2/N.

Condition 3. Given $\overline{\mathbf{E}_2^i}$ implies that there can be at most $16\kappa_i \log^2 N$ queries in the offline database that are part of some claw. As per the definition of condition 4, there will be a unique chain of length $< B$ in the offline database ending in each of these at most $16\kappa_i \log^2 N$ queries, such that an online query hits the start of this chain. The probability of hitting one of these at most $B \cdot 16\kappa_i \log^2 N$ salts within T queries is at most $16\kappa_i TB \log^2 N/N$.

Condition 4. As per the definition of condition 5, there can be at most iT such chains of length $< B$ in the offline database, such that an online query hits the start of this chain and another online hits the end of this chain. The probability of hitting both the salts within at most T queries is bounded by T^2/N^2. By union bound the advantage is at most iT^3/N^2.

Condition 5. Given $\overline{\mathbf{E}_3^i}$ implies there are at most $i \log N$ 'special' cycles in the offline database, each with at most B queries in it. So, there are at most $iB \log N$ queries in these cycles and the probability of hitting one of the starting salts of these queries within T online queries is bounded by $iB \log N \cdot T/N$.

From Claim 9 it holds that the advantage of achieving any of the conditions in Claim 8 given $\overline{\mathbf{E}_2^i}$, $\overline{\mathbf{E}_3^i}$ and $\overline{\mathbf{G}}$ is bounded by $(16\kappa_i TB \log^2 N + T^2)/N$. Note that for $i \leq S$, when $ST^2 < N$ implies $iT^2 < N$. Hence $\kappa_i = i$ if $\kappa_S = S$. □

Acknowledgements. We thank CRYPTO reviewers and Xiaoqi Duan for their constructive comments. We thank Ashrujit Ghoshal and Ilan Komargodski for sharing an early draft of their work. Akshima is supported in part by NSF Grant No. 1925288. Siyao Guo is supported by National Natural Science Foundation of China Grant No.62102260, Shanghai Municipal Education Commission (SMEC) Grant No. 0920000169, NYTP Grant No. 20121201 and NYU Shanghai Boost Fund. Qipeng Liu is supported in part by the Simons Institute for the Theory of Computing, through a Quantum Postdoctoral Fellowship and by the DARPA SIEVE-VESPA grant No. HR00112020023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or DARPA.

References

[ACDW20] Akshima, Cash, D., Drucker, A., Wee, H.: Time-space tradeoffs and short collisions in Merkle-Damgård hash functions. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part I. LNCS, vol. 12170, pp. 157–186. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-56784-2_6

[BR93] Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security, pp. 62–73 (1993)

[CDG18] Coretti, S., Dodis, Y., Guo, S.: Non-uniform bounds in the random-permutation, ideal-cipher, and generic-group models. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part I. LNCS, vol. 10991, pp. 693–721. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96884-1_23

[CDGS18] Coretti, S., Dodis, Y., Guo, S., Steinberger, J.: Random oracles and non-uniformity. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 227–258. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_9

[CGK18] Corrigan-Gibbs, H., Kogan, D.: The discrete-logarithm problem with preprocessing. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 415–447. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78375-8_14

[CGK19] Corrigan-Gibbs, H., Kogan, D.: The function-inversion problem: barriers and opportunities. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019, Part I. LNCS, vol. 11891, pp. 393–421. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36030-6_16

[CGLQ20] Chung, K.-M., Guo, S., Liu, Q., Qian, L.: Tight quantum time-space tradeoffs for function inversion. In: Irani, S. (ed.) 61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, 16–19 November 2020, pp. 673–684. IEEE (2020)

[CHM20] Chawin, D., Haitner, I., Mazor, N.: Lower bounds on the time/memory tradeoff of function inversion. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part III. LNCS, vol. 12552, pp. 305–334. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64381-2_11

[Dam89] Damgård, I.B.: A design principle for hash functions. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 416–427. Springer, New York (1990). https://doi.org/10.1007/0-387-34805-0_39

[DGK17] Dodis, Y., Guo, S., Katz, J.: Fixing cracks in the concrete: random oracles with auxiliary input, revisited. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part II. LNCS, vol. 10211, pp. 473–495. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56614-6_16

[DTT10] De, A., Trevisan, L., Tulsiani, M.: Time space tradeoffs for attacks against one-way functions and PRGs. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 649–665. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_35

[GGKL21] Gravin, N., Guo, S., Kwok, T.C., Lu, P.: Concentration bounds for almost k-wise independence with applications to non-uniform security. In: Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, 10–13 January 2021, pp. 2404–2423 (2021)

[GK22] Ghoshal, A., Komargodski, I.: On time-space tradeoffs for bounded-length collisions in Merkle-Damgård hashing. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, LNCS 13509, pp. xx–yy (2022)

[GLLZ21] Guo, S., Li, Q., Liu, Q., Zhang, J.: Unifying presampling via concentration bounds. In: Nissim, K., Waters, B. (eds.) TCC 2021, Part I. LNCS, vol. 13042, pp. 177–208. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-90459-3_7

[Hel80] Hellman, M.E.: A cryptanalytic time-memory trade-off. IEEE Trans. Inf. Theory **26**(4), 401–406 (1980)

[IK10] Impagliazzo, R., Kabanets, V.: Constructive proofs of concentration bounds. In: Serna, M., Shaltiel, R., Jansen, K., Rolim, J. (eds.) APPROX/RANDOM -2010. LNCS, vol. 6302, pp. 617–631. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15369-3_46

[Mer89] Merkle, R.C.: A certified digital signature. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 218–238. Springer, New York (1990). https://doi.org/10.1007/0-387-34805-0_21

[Unr07] Unruh, D.: Random oracles and auxiliary input. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 205–223. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74143-5_12