A Complete Characterization of One-More Assumptions In the Algebraic Group Model

Jake Januzelli

Oregon State University, januzelj@oregonstate.edu

Jiayu Xu

Oregon State University, xujiay@oregonstate.edu

Abstract

One-more problems like One-More Discrete Logarithm (OMDL) and One-More Diffie–Hellman (OMDH) have found wide use in cryptography, due to their ability to naturally model security definitions for interactive primitives like blind signatures and oblivious PRF. Furthermore, a generalization of OMDH called Threshold OMDH (TOMDH) has proven useful for building threshold versions of interactive protocols. However, due to their complexity it is often unclear how hard such problems actually are, leading cryptographers to analyze them in idealized models like the Generic Group Model (GGM) and Algebraic Group Model (AGM). In this work we give a complete characterization of widely used group-based one-more problems in the AGM, using the Q-DL hierarchy of assumptions defined in the work of Bauer, Fuchsbauer and Loss [BFL20].

1. Regarding (T)OMDH, we show (T)OMDH is part of the Q-DL hierarchy in the AGM; in particular, Q-OMDH is equivalent to Q-DL. Along the way we find and repair a flaw in the original GGM hardness proof of TOMDH [JKKX17, Theorem 7], thereby giving the first correct proof that TOMDH is hard in the GGM.
2. Regarding OMDL, we show the Q-OMDL problems constitute an infinite hierarchy of problems in the AGM incomparable to the Q-DL hierarchy; that is, Q-OMDL is strictly harder than Q'-OMDL if $Q < Q'$, and Q-OMDL is incomparable to Q'-DL (i.e., there are no reductions either way) unless $Q = Q' = 0$.

Contents

1 Introduction

One-more type problems — first introduced by Bellare et al. [BNPS02] — require an adversary to solve at least $Q+1$ instances of a problem, given only Q accesses to a solver: prominent examples include *One-More RSA*, *One-More Discrete Logarithm (OMDL)*, and *One-More (computational) Diffie–Hellman (OMDH)*. Such problems have found wide use in cryptography due to naturally modelling security requirements of interactive protocols; for example, in a secure blind signature scheme, an adversary cannot produce $Q+1$ valid signatures after Q interactions with the signer, and in a secure Oblivious PRF (OPRF) scheme, an adversary cannot evaluate $Q+1$ PRF values after Q interactions with the sender who holds the PRF key.

The *Threshold One-More Diffie–Hellman (TOMDH)* assumption is a generalization of OMDH in the threshold setting, with the secret exponent (t, n)-Shamir secret-shared and the adversary may (statically) corrupt t' shares of its choice. It has been used in several works to implement efficient threshold versions of cryptographic primitives, such as threshold OPRF[1] and distributed Verifiable Random Function (distributed VRF). Certain special cases of the assumption were shown to be equivalent to classical OMDH [JKKX17, Theorems 1 and 2], but due to the complexity of the problem it is unclear how it relates in general.[2]

See Table 1 for an incomplete list of applications of one-more assumptions.

assumption	application	reference
OMDL	blind signatures	[FPS20]
	multisignatures	[BN06, NRS21]
	identification schemes	[BP02, BNN04]
OMDH	oblivious PRF	[JKK14, JKKX16]
	blind signatures	[Bol03]
TOMDH	threshold oblivious PRF	[JKKX17]
	password-based threshold authentication	[AMMM18]
	distributed VRF	[KMMM23]
	threshold blind signatures	[LNÖ25]

Table 1: Applications of one-more type assumptions

Proofs in the GGM and the AGM. Since one-more assumptions are interactive and more complex than their "standard" counterparts, researchers often investigate them in the setting of an idealized computational model. For assumptions in prime-order groups, the models of choice are the generic and algebraic group models respectively. In the *Generic Group Model (GGM)* [Sho97], group elements are not given to the adversary directly, but only via random handles. The adversary is also given an oracle to multiply group elements, thus modelling the situation where the adversary interacts with the group "generically". The appeal of the GGM is that one can show unconditional lower bounds on solving problems; indeed, it was introduced for the purpose of proving the lower bound of Discrete Logarithm (DL). However, proofs in the GGM can be inordinately complex, involving careful bookkeeping and probabilistic arguments, with several instances of published proofs

[1] The original work realizing threshold OPRF under TOMDH [JKKX17] was later found to be flawed, but can be fixed by adding another round or making more group operations; the former still relies on TOMDH, whereas the latter only relies on OMDH and DDH. See the end of [GJK+24, Section 1.1] for a discussion.

[2] The TOMDH assumption is so complex that even describing its definition clearly is a delicate task. In Sect. 3.2 we provide an explanation of the intuition behind TOMDH, which we believe is more approachable than existing works.

later found to be incorrect [HLY09, BFP21].

The *Algebraic Group Model (AGM)* [FKL18] lies between the GGM and the standard model, and provides an attractive alternative to the GGM in the analyses of both security properties of schemes and hardness of computational assumptions. In the AGM, the adversary is given group elements directly, and the only requirement is to "explain" how any output Y is computed from their inputs $X_1, \ldots, X_n$ via outputting the *algebraic coefficients* $\lambda_1, \ldots, \lambda_n$ such that

$$Y = X_1^{\lambda_1} \cdots X_n^{\lambda_n}.$$

Unlike the GGM, one cannot prove unconditional lower bounds in the AGM, and security proofs in the AGM still assume hardness of a basic computational problem such as DL. However, one can give a reduction in the AGM to a problem already known to be hard in the GGM, and hardness should intuitively follow from the fact that a generic algorithm is also algebraic. This "lifting result" was first argued in [FKL18, Lemma 2.2] but formalization proved to be a challenge due to definitional ambiguities in the AGM [Zha22, ZZK22]; a complete and correct formalization was only provided in the very recent work of [JM24]. Such hardness proofs in the AGM enjoy two benefits: first, they are strictly stronger than the corresponding GGM proofs; second, the proofs become simpler and more modular. Indeed, as pointed out in [JM24, Section 1], a standard GGM proof usually contains the following steps:

1. Replace "honest" group elements with handles represented by polynomials of the secret exponent x;
2. Show that the assumption reduces to a "bad event" that the adversary defines two different polynomials which evaluate to the same value on input x;
3. Argue that the "bad event" occurs with negligible probability.

Steps 1 and 3 are somewhat "boilerplate" which can be filtered out in an AGM proof, and we can then focus on the more essential step 2. In sum, hardness reductions in the AGM are interesting on their own, but can also be viewed as a methodology that provides a cleaner and less error-prone way to write hardness proofs in the GGM. (This advantage of AGM proofs is also briefly discussed in [FKL18, Section 1.2].)

1.1 Our Contributions

To date, the most comprehensive study of group-based assumptions in the AGM is the work of Bauer, Fuchsbauer and Loss [BFL20], which analyzes a wide class of so-called *Uber assumptions*. Uber assumptions are defined as follows: fix a generator g of group G and polynomials $R_1, \ldots, R_r, F$ in variables $Z_1, \ldots, Z_n$ that are linearly independent; the adversary is given $(g^{R_1(\vec{z})}, \ldots g^{R_r(\vec{z})})$ (for $\vec{z} \leftarrow \mathbb{F}_p^n$), and succeeds if it returns $g^{F(\vec{z})}$. [BFL20] reduces this problem to Q-DL, where $Q = \max \deg(R_i) - 1$. (The Q-DL problem is: given $(g, g^x, g^{x^2}, \ldots, g^{x^{Q+1}})$ for $x \leftarrow \mathbb{F}_p$, compute x.[3]) This covers a vast array of hardness assumptions, including Computational Diffie–Hellman, Square Diffie–Hellman, Strong Diffie–Hellman [BB08], and LRSW [LRSW99], as special cases. Furthermore, [BFL20, Section 9] proves that Q-DL and $(Q+1)$-DL are separate in the AGM, providing a complete hierarchy for the relative hardness of all Uber assumptions.

However, [BFL20] only provides one result on one-more type assumptions: there is no reduction

[3] Our definition of Q-DL is what's commonly called $(Q+1)$-DL in the literature; in particular, the (regular) discrete log problem is 0-DL. We make this change for the sake of consistency with Q-OMDL where the adversary is given $Q+1$ challenge group elements (in addition to the generator g).

from Q-DL to Q'-OMDL in the AGM for any $Q, Q' \geq 1$, unless Q-DL is easy [BFL20, Section 10]. This is shown via the general paradigm of *meta-reductions*: given an efficient reduction $\mathcal{R}$ that solves problem P_1 given access to a solver of P_2, we construct an efficient meta-reduction $\mathcal{M}$ that uses $\mathcal{R}$ to *unconditionally* solve P_1 by simulating $\mathcal{R}$'s access to an adversary $\mathcal{A}$ for P_2.

Regarding the GGM, the Q-DL, OMDL and (T)OMDH problems have been established to be hard generically ([BFL20, Appendix A], [BFP21, Theorem 1, Corollary 1] and [JKKX17, Theorem 7], respectively), but these results imply nothing about where OMDL and (T)OMDH stand in relation to the Q-DL hierarchy in the AGM, leaving the relative hardness of one-more type assumptions in the AGM understudied. Furthermore, in Sect. 5.2 we will show that the proof of [JKKX17, Theorem 7] is incorrect in the static corruption case (i.e., the number of corrupted shares $t' > 0$), so the hardness of TOMDH in the GGM remains unsettled.

In this work, we present a comprehensive study of one-more type assumptions in the AGM. Concretely, our contributions are as follows:

1. We first show in Sect. 4 that Q-OMDH is tightly equivalent to Q-DL in the AGM. In fact, we show a stronger result that covers a much wider class of one-more assumptions, generalizing Q-OMDH to what we call *One-More Uber (OMU) assumptions*: these are the same as Uber assumptions except that the adversary can query an x-th power oracle Q times and needs to compute the x-th power of $Q + 1$ challenge group elements.

2. As our main result, we show in Sect. 5 that TOMDH falls into the Q-DL hierarchy in the AGM, giving a reduction from $(Q(n-t)-1)$-DL to (t', t, n, Q)-TOMDH in the AGM; on the way we point out and repair the flaw in the GGM argument of [JKKX17, Theorem 7].

[BFL20] show that their reductions work in many additional settings (over groups with a billinear pairing, replacing polynomials with rational functions, gap problems, letting the adversary choose the polynomials adaptively, etc.). Since our reductions use the same template, these generalizations also apply to our results: in particular, we cover the Gap TOMDH assumption as originally considered in [JKKX17], and the Bilinear TOMDH assumption as considered in [KMMM23, LNÖ25].

We have mentioned that the "lifting result" of [JM24, Theorem 2] can "translate" an AGM proof into a GGM hardness proof. It is easy to see that our reductions from Thms. 4.2, 4.3 and 5.1 satisfy the syntactical requirements for [JM24, Theorem 2] to apply. Therefore we recover the hardness of (T)OMDH in the GGM, and give the *first correct proof* of TOMDH hardness in the static corruption case.

In addition, we also give two separation results, which close the gaps in the literature remaining from [BFL20]:

3. We show in Sect. 6.1 that there is no reduction from Q-OMDL to $(Q+1)$-OMDL in the AGM unless Q-OMDL is easy.

4. We show in Sect. 6.2 that there is no reduction from Q-OMDL to 1-DL in the AGM for $Q \geq 0$, unless Q-OMDL is easy. Since Q'-DL is easier than 1-DL for $Q' \geq 1$, this separates Q-OMDL and Q'-DL for all $Q, Q' \geq 0$ (unless $Q = Q' = 0$). Both results are shown via the same meta-reduction approach as in [BFL20].

Together, these results establish a *complete characterization* of the hardness of *all widely used group-based one-more problems in the literature* in the AGM, summarized in Figure 1 below. At a high level, we have two hierarchies, one for Q-OMDL and one for Q'-OMDH (which is equivalent to Q'-DL); they are equivalent if $Q = Q' = 0$, but are separate in all other cases. This matches the intuition that Q-OMDL and Q'-OMDH are incomparable: in OMDH the answer is easier to find,

but the oracle is also weaker.

$$\begin{array}{ccccc} \vdots & & \vdots & & \vdots \\ \text{2-OMDL} & & \text{2-DL} & = & \text{2-OMDH} \\ \wedge & & \wedge & & \wedge \\ \text{1-OMDL} & & \text{1-DL} & = & \text{1-OMDH} \\ \wedge & & \wedge & & \wedge \\ \text{0-OMDL} & = & \text{0-DL} & = & \text{0-OMDH} \end{array}$$

Figure 1: Hierarchies of Q-DL, Q-OMDL, and Q-OMDH in the AGM. TOMDH also falls into the Q-DL hierarchy but is not directly equivalent to Q'-DL for some Q'.

Besides being theoretically interesting, these results have implications to the concrete security of existing protocols. As one example, [JKKX17, Theorem 3] shows the security of a threshold OPRF protocol under (t', t, n, Q)-TOMDH, where

- Q is the maximum number of OPRF evaluations initiated by a user;
- n is the number of servers that hold secret shares of the OPRF key;
- t' is the number of servers an adversary can corrupt; and
- $t + 1$ servers need to collaborate to evaluate the OPRF.

Although there are many non-generic attacks on group-based hardness assumptions [Adl79, Che06], to our knowledge there are no *non-algebraic* attacks. (For a more detailed discussion on this, see [FKL18, Section 1].) Therefore our AGM analysis indicates the above TOPRF protocol should be as secure as the $Q(n-t)$-DL problem. The best attack on Q'-DL in a group of prime order p is due to Cheon [Che06] with concrete complexity $\approx \log p(\sqrt{p/Q'} + Q')$ provided $(p-1)/2$ is also prime. Since Q corresponds to evaluation messages sent by users, one should conservatively assume Q can be very large, or even (adaptively) adversarially controlled. In this case the best attack occurs when $Q' \approx p^{1/3}$, with complexity $\approx \log p \cdot p^{1/3}$. As noted in [TCR+22, Theorem 2] this is the best level of security one could expect, as it matches existing non-threshold OPRF protocols like 2HashDH [JKK14].

1.2 Relation to Existing Work

Algebraic reductions vs. reductions in the AGM. Bresson, Monnerat and Vergnaud [BMV08, Theorem 11] prove that there is no algebraic reduction from Q-OMDL to $(Q+1)$-OMDL, unless Q-OMDL is easy. While one might think their result renders our AGM separation for OMDL (Thm. 6.1) superfluous, in fact our result is stronger: [BMV08, Theorem 11] only rules out algebraic reductions that work for *arbitrary* adversaries, whereas our Thm. 6.1 rules out a larger class of algebraic reductions, namely those that only work for *algebraic* adversaries. The work of the latter class of reductions is potentially much easier, as they can additionally exploit the algebraic representations output by the adversary.

Other existing works. The works of Schage [Sch24, Corollary 6] and Zhang et al. [ZZC+14] give general separations for many one-more problems, including Q-OMDH/Q-OMDL for different values of Q, without assuming the reduction is algebraic. These separations are complementary

but incomparable to ours: (1) their proofs are far more technically complex than ours, sometimes producing weaker results like meta-reductions running in *expected* polynomial-time, and (2) like [BMV08, Theorem 11], they give no separation in the setting when a reduction is allowed to assume its solver is also algebraic, and so cannot apply to the AGM.

Concerning the analysis of one-more problems in the AGM, Tyagi et al. [TCR+22, Theorem 2] propose a novel one-more Diffie–Hellman-type assumption and shows that it reduces to Q-DL in the AGM. However, their assumption is tailored to their specific use case, and their proof gives little indication as to the hardness of more standard one-more type assumptions like (T)OMDH.

2 Technical Overview

2.1 Notation

Let G be a group of prime order p, with the security parameter $\lambda = \lceil \log p \rceil$, i.e., $2^{\lambda-1} < p < 2^\lambda$. We assume that G and p are public parameters known to any algorithm, and omit them in the descriptions of security games below. Let $\mathbb{F}_p$ be the finite field with p elements, and $[n] = \{1, 2, \ldots, n\}$: $[n]$ may be a subset of $\mathbb{Z}$ or $\mathbb{F}_p$, which one will be clear from context. If S is a finite set, let $|S|$ denote the size of S and $x \leftarrow S$ denote sampling an element x uniformly at random from S. Let $S_{\geq a}$ be the subset of S consisting of elements at least a.

Runtime calculation. While considering the runtime of an algorithm, we only count group operations (so computing an exponentiation costs time up to 2λ). Our reductions will often make use of finite field arithmetic as well (matrix multiplication, row reduction, factorization of polynomials) but it suffices to observe that all of these operations can be done in PPT (e.g., [KU11]).

Basic linear algebra. Notationally, we use uppercase bolded letters to denote matrices: e.x. $\mathbf{A}$. We use arrows to denote vectors: e.x. $\vec{v}$. Given a matrix $\mathbf{A}$, define $\ker(\mathbf{A}) = \{\vec{v} \mid \mathbf{A}\vec{v} = \vec{0}\}$ and $\mathrm{im}(\mathbf{A}) = \{\vec{w} \mid \exists \vec{v} : \mathbf{A}\vec{v} = \vec{w}\}$. Given a vector space V, let $\dim(V)$ be its dimension. Standard uppercase letters will usually denote group elements, formal variables and polynomials in said formal variables. Lowercase letters and Greek letters will usually denote elements of $\mathbb{Z}$ or $\mathbb{F}_p$. We adopt a convenient notational convention whereby given an expression v in lowercase quantities a_i, we let V denote the polynomial given by replacing each occurrence of a_i in v with a new formal variable A_i.

For vectors $\vec{a} = (a_1, \ldots, a_n), \vec{b} = (b_1, \ldots, b_n)$, let $\vec{a}^i = (a_1^i, \ldots, a_n^i)$. Define the Hadamard product of $\vec{a}, \vec{b}$ to be $\vec{a} \odot \vec{b} = (a_1 \cdot b_1, \ldots, a_n \cdot b_n)$. Note that

1. $\vec{a} \odot \left(\sum_{i=1}^m \vec{b}_i\right) = \left(\sum_{i=1}^m \vec{a} \odot \vec{b}_i\right)$.
2. $\vec{a} \odot (x\vec{b}) = x(\vec{a} \odot \vec{b})$ where x is a scalar.
3. If $\vec{a}$ has no zero entries and $\vec{a} \odot \vec{b} = \vec{0}$, then $\vec{b} = \vec{0}$.
4. By Items 1 and 3, if $\vec{a}$ has no zero entries and $\sum_{i=1}^m \vec{a} \odot \vec{b}_i = \vec{0}$, then $\sum_{i=1}^m \vec{b}_i = \vec{0}$.

2.2 Our Techniques

Our reductions will use techniques from [BFL20] and [Rot22], so we first give a high-level review of these works.

Review of [BFL20]. In [BFL20] the authors reduce the $(R_1, \ldots, R_r, F)$-Uber problem to Q-DL in the AGM, where $Q = \max \deg(R_i) - 1$. The idea of the reduction is that if the Uber adversary is algebraic, it returns an algebraic representation $F(\vec{z}) = \sum_{j=1}^{r} \alpha_j R_j(\vec{z})$ to the challenger. By the definition of the Uber problem, $R_1, \ldots, R_r, F$ are linearly independent, so $R'(\vec{Z}) = F(\vec{Z}) - \sum_{j=1}^{r} \alpha_j R_j(\vec{Z})$ is a non-zero polynomial that evaluates to zero on the secret $\vec{z}$. Therefore the adversary has implicitly found a polynomial that "encodes" the secret $\vec{z}$. The reduction then embeds a randomized version of the Q-DL secret x into each of the coordinates of $\vec{z}$ by setting $z_i = y_i x + w_i$ for random $y_i, w_i \leftarrow \mathbb{F}_p$, and solves for x by finding the roots of the polynomial $R'(y_1 X + w_1, \ldots, y_n X + w_n)$. Since $Q = \max \deg(R_i) - 1$, the reduction can simulate the Uber challenges to the adversary by computing the elements $g^{R_j(\vec{z})}$ from the Q-DL challenge.

Review of [Rot22]. In [Rot22, Section 5.2], the author notes that although the reduction of [BFL20] is tight (losing only an additive factor in advantage), it requires a Q-DL challenge dependent on the maximum *total* degree of the polynomials R_j. [Rot22, Theorem 5.4] provides an alternative reduction that reduces from $(d'-1)$-DL, where d' is the maximum degree of the R_j in any *single* variable. In this reduction the $(d'-1)$-DL secret x is only inserted into a single variable Z_i chosen randomly from $Z_1, \ldots, Z_n$, and the other variables are set to known random elements of $\mathbb{F}_p$. As before, the reduction can simulate the Uber challenges to the adversary by computing the elements $g^{R_j(\vec{z})}$ from the $(d'-1)$-DL challenge; however, the reduction loses a factor of n in advantage as the price for reducing Q, since extracting the secret is no longer as simple as finding the roots of a univariate polynomial.

Our techniques for (T)OMDH. We first note why the results of [BFL20, Rot22] do not straightforwardly extend to one-more type assumptions.

The closest problem to the one-more setting is their flexible GeGenUber problem [BFL20, Section 8] (for the sake of exposition we describe a simplified version of the problem). The security game is parameterized by a fixed polynomial R. The game chooses $\vec{z} \leftarrow \mathbb{F}_p^n$, and the adversary has access to an oracle $\mathcal{O}$ that, on input $P_i \in \mathbb{F}_p[Z_1, \ldots, Z_n]$, returns $g^{P_i(\vec{z})}$. The adversary wins the game if it returns $g^{R(\vec{z})}$ and $R \notin \mathrm{Span}(\{P_i\})$. Compare this to the Q-OMDH problem, where the adversary is given challenge elements $g^{z_i}, 1 \le i \le Q+1$, as well as up to Q queries to a z_0-th power oracle, and must return $g^{z_0 z_i}$ for $1 \le i \le Q+1$. Although the problems appear similar, Q-OMDH is *not* a special case of GeGenUber: in GeGenUber there is a single target $g^{R(\vec{z})}$, whereas in Q-OMDH there are $Q+1$ targets, none of which is *fixed*, i.e., it is possible that any single target $g^{z_0 z_i}$ was the result of an oracle query. As such, there is no valid choice for R to make OMDH a special case of GeGenUber. Indeed, to prevent the adversary from trivially succeeding, in GeGenUber one adaptively restricts the allowed *targets* of oracle queries, whereas in Q-OMDH one only restricts the *number* of oracle queries.

Our reduction will produce a collection of polynomials $\{P_j\}$ with $P_j(\vec{z}) = 0$ such that *at least one of them* is guaranteed to be nonzero. By the Schwartz–Zippel lemma the nonzero one can be found efficiently, and the reduction can proceed as before. In general, the primary tasks of a reduction from Q-DL will be to (a) simulate the security game to the challenger using the Q-DL challenge, and (b) use properties of the security game to extract a nontrivial polynomial equation over the Q-DL secret. As evidenced by [BFL20, Rot22] there is a tradeoff between (a) and (b): inserting the secret into more variables increases the probability that the secret can be extracted, but makes it more difficult to simulate the security game.

In the case of OMU (Sect. 4) we closely follow the strategy in [BFL20] and either insert the challenge into all variables at once, or all but one, as the total degree of the polynomials R_j is close

to the maximum degree in a single variable for OMU. However, the case of TOMDH (Sect. 5) is more complex. Concerning (a), there is no a priori bound on the maximum total degree of polynomials the adversary can produce (see Sect. 5 for details), so we must use the strategy of [Rot22] and choose a single variable (either a challenge element or a Shamir secret share) to insert the Q-DL secret into; even then the reduction may fail to always simulate the security game correctly, leading to a looser reduction. Concerning (b), the proof of the hardness of TOMDH in the GGM [JKKX17, Theorem 7] should implicitly contain the extraction of a nontrivial polynomial equation, corresponding to the intuition that in TOMDH, the only thing the adversary can do with shares of the secret is combine them via Lagrange interpolation. However, as noted previously their argument is incorrect in the static corruption case. In Sect. 5.2 we isolate the missing technical piece of the proof, and prove it in Sect. 5.4 using linear-algebraic properties of Vandermonde matrices.

Our techniques for OMDL separation. First note that we cannot trivially adapt the meta-reduction in [BMV08, Theorem 11] — which shows separation between Q-OMDL and $(Q+1)$-OMDL assuming *non-algebraic adversaries* — to the AGM. The discrete-log oracle queries for their simulated adversary depend on the view of the meta-reduction, which is problematic when analyzing its advantage in the AGM (see [BFL20, Section 10] for further discussion).

To circumvent this issue, our meta-reduction $\mathcal{M}$ queries the discrete-log oracle at uniformly random elements and uses linear algebra to extract the solution. $\mathcal{M}$ has a Q-OMDL challenge $(A_{-1}, A_0, \ldots, A_Q)$, which it feeds to a reduction $\mathcal{R}$. $\mathcal{R}$ will then submit a $(Q+1)$-OMDL challenge $(B_{-1}, B_0, \ldots, B_{Q+1})$ to the adversary $\mathcal{A}$, which $\mathcal{M}$ must simulate.[4] Let $B_{-1}^{e_i} = A_i, B_{-1}^{d_j} = B_j$, for $i = -1, \ldots, Q$ and $j = 0, \ldots, Q+1$, so $\mathcal{M}$ needs to compute $(d_0, \ldots, d_{Q+1})$. Our strategy uses that $\vec{d} = (1, d_0, \ldots, d_{Q+1})$ is defined in terms of $\vec{e} = (e_{-1}, \ldots, e_Q)$ via $\mathbf{Z}\vec{e} = \vec{d}$, where $\mathbf{Z}$ is a $(Q+3) \times (Q+2)$ matrix given by the algebraic representations of B_i in terms of A_i. Since the first entry of $\vec{d}$ is known, $\vec{e}$ must satisfy a known nontrivial linear equation specified by the first row of $\mathbf{Z}$, so $\vec{e}$ (and thus $\mathbf{Z}\vec{e}$) has only $Q+1$ degrees of freedom. This is what allows $\mathcal{M}$ to leverage its DL queries: intuitively, $\mathcal{M}$ can now recover $\mathbf{Z}\vec{e}$ by using DL to obtain $Q+1$ additional linear equations over it and solving them. $\mathcal{M}$ queries DL on $Q+1$ random group elements, whose algebraic representations (in terms of B_j) form a $(Q+1) \times (Q+2)$ matrix $\mathbf{U}$. $\mathcal{M}$ adds $(1, 0, \ldots, 0)$ as $\mathbf{U}$'s first row and first column (since $\mathcal{M}$ wants to also use the known equation defined by the first row of $\mathbf{Z}$), making $\mathbf{U}$ a $(Q+2) \times (Q+3)$ matrix. The results of the DL queries are the entries of $\mathbf{U}\vec{d}$. $\mathcal{M}$ can now use row reduction to find some solution $\vec{v}$ to the equation $\mathbf{U}\mathbf{Z}\vec{v} = \mathbf{U}\vec{d}$. Since $\mathbf{Z}$ could have low rank, there is no guarantee that $\vec{v} = \vec{e}$. However, since $\mathbf{U}$ was chosen randomly $\mathbf{Z}\vec{v} = \vec{d}$ holds with high probability, so $\mathcal{M}$ can recover $\vec{d}$.

Our techniques for Q-DL separation. The idea of our meta-reduction $\mathcal{M}$ is to use linear-algebraic techniques to adapt the meta-reduction in [BFL20, Theorem 9.1] (that separates Q-DL and $(Q+1)$-DL) to work with Q-OMDL. Suppose $\mathcal{M}$ receives a Q-OMDL challenge $(A_{-1}, A_0, \ldots, A_Q)$ and feeds it to $\mathcal{R}$. Let $A_i = A_{-1}^{x_i}$ for $i = 0, \ldots, Q$, so $\mathcal{R}$ needs to recover $\vec{x} = (x_0, \ldots, x_Q)$. When $\mathcal{R}$ invokes $\mathcal{A}$ on a 1-DL challenge $(B_{-1}, B_0 = B_{-1}^u, B_1 = B_{-1}^{u^2})$, the algebraic representations of B_i in terms of A_i allow $\mathcal{M}$ to compute polynomials P_j such that $B_j = A_{-1}^{P_j(\vec{x})}$ (for $j = -1, 0, 1$). Here we start with the observation of [BFL20, Theorem 9.1] that $S(\vec{x}) = 0$ for $S = P_1 P_{-1} - P_0^2$, since $u = P_0(\vec{x})/P_{-1}(\vec{x}) = P_1(\vec{x})/P_0(\vec{x})$. If S is the zero polynomial, $\mathcal{M}$ can compute u directly; otherwise $\mathcal{M}$ obtains a nontrivial polynomial equation over $\vec{x}$. In [BFL20, Theorem 9.1] the secret $\vec{x}$ has a

[4] Note that B_{-1} might be different from A_{-1}, i.e., we do not assume that the group generator g is fixed and even rule out reductions that might change the group generator while running the $(Q+1)$-OMDL solver.

single entry, so this suffices to compute $\vec{x}$. In our case we must obtain Q additional equations over $\vec{x}$ to compute it; we do so by utilizing the Q queries to DL that $\mathcal{R}$ can make. Concretely, $\mathcal{M}$ has to answer $\mathcal{R}$'s discrete log oracle queries $V_1, \ldots, V_Q$. Some queries $\mathcal{M}$ can answer without querying its own DL oracle; for example, if $V_3 = V_1 V_2^2 A_{-1}$, $\mathcal{M}$ can answer directly since it already knows $\mathrm{dlog}_{A_{-1}}(V_1), \mathrm{dlog}_{A_{-1}}(V_2), \mathrm{dlog}_{A_{-1}}(A_{-1})$. In this way $\mathcal{M}$ only makes DL queries corresponding to linearly independent polynomials in $X_0, \ldots, X_Q$, and can "save" some DL queries for the future. When $\mathcal{R}$ invokes $\mathcal{A}$ on (B_{-1}, B_0, B_1), $\mathcal{M}$ must recover $u = \mathrm{dlog}_{B_{-1}}(B_0) = \mathrm{dlog}_{B_0}(B_1)$ as $\mathcal{A}$'s output to $\mathcal{R}$. Suppose $\mathcal{M}$ has already made Q' queries to DL (where $Q' \leq Q$). $\mathcal{M}$ computes S: if the equation $S(\vec{x}) = 0$ is trivial (i.e., implied by the Q' existing equations from the DL queries) we show $\mathcal{M}$ can compute u directly as before. Otherwise $\mathcal{M}$ will make $Q - Q'$ additional queries to DL on random group elements. By a linear-algebraic argument, $S(\vec{x}) = 0$ remains nontrivial in the presence of the new equations with high probability; $\mathcal{M}$ now has $Q + 1$ equations over $Q + 1$ variables and can compute $\vec{x}$, from which u is easily computed.

3 Background

3.1 Polynomial Rings

We denote the ring of polynomials in variables $X_1, \ldots, X_m$ over a field $\mathbb{F}$ by $\mathbb{F}[X_1, \ldots, X_m]$, and the field of rational functions as $\mathbb{F}(X_1, \ldots, X_m)$. The degree of $F \in \mathbb{F}[X_1, \ldots, X_m]$, denoted $\deg(F)$, is the maximum total degree of all monomials in F; if $T \subset \{X_1, \ldots, X_m\}$ then $\deg_T(F)$ is the maximum total degree of F as a polynomial in the variables T. As an example, consider

$$F(X_1, X_2) = X_1^3 X_2 + X_2^2 + X_1.$$

We have $\deg(F) = \deg_{X_1, X_2}(F) = 4$ since $X_1^3 X_2$ has total degree 4, $\deg_{X_1}(F) = 3$ since $X_1^3 X_2$ has degree 3 in X_1, and $\deg_{X_2}(F) = 2$ since X_2^2 has degree 2 in X_2. Define

$$\mathcal{I} = \langle F_1, \ldots, F_n \rangle = \left\{ \sum_{i=1}^{n} G_i F_i \mid G_i \in \mathbb{F}[X_1, \ldots, X_m] \right\}$$

to be the ideal of $\mathbb{F}[X_1, \ldots, X_m]$ generated by $F_1, \ldots, F_n$. Denote by $\mathbb{F}[X_1, \ldots, X_m]/\mathcal{I}$ the quotient ring modulo $\mathcal{I}$, whose elements are equivalence classes where F, G are equivalent if $F - G \in \mathcal{I}$: in this case we write $F \equiv G \pmod{\mathcal{I}}$. Define

$$V(\mathcal{I}) = \{\vec{x} \in \mathbb{F}^m \mid F(\vec{x}) = 0 \ \forall \ F \in \mathcal{I}\}$$

to be the vanishing set of the ideal $\mathcal{I}$. Note that for any $\vec{x} \in V(\mathcal{I})$ and $F \in \mathbb{F}[X_1, \ldots, X_m]/\mathcal{I}$, the value $F(\vec{x})$ is well-defined: if $F - G = H \in \mathcal{I}$ then $F(\vec{x}) - G(\vec{x}) = H(\vec{x}) = 0$.

In this paper the polynomials $F_1, \ldots, F_n$ will be of degree 1 and linearly independent, and $n < m$. Recall that row reduction defines a set $T(\mathcal{I}) \subset \{X_1, \ldots, X_m\}$ of *pivotal variables* of size n corresponding to the system $\{F_i = 0\}$, such that all pivotal variables can be eliminated via the equations $\{F_i = 0\}$. Therefore for any $R \in \mathbb{F}[X_1, \ldots, X_m]$ there is a canonical polynomial $[R]$ such that $R \equiv [R] \pmod{\mathcal{I}}$ and $[R]$ contains none of the variables in $T(\mathcal{I})$. We have the properties

1. $R(\vec{x}) = [R](\vec{x})$ for any $\vec{x} \in V(\mathcal{I})$,
2. $[R] = [R']$ if and only if $R \equiv R' \pmod{\mathcal{I}}$.

As a consequence, the degree of $F \in \mathbb{F}[X_1, \ldots, X_m]/\mathcal{I}$ is well-defined: we simply define $\deg(F \pmod{\mathcal{I}}) = \deg([F])$ (note that as regular polynomials, $\deg(F) \geq \deg([F])$). We retain the standard properties of the degree in this setting, such as

1. $\deg(FG) = \deg(F) + \deg(G)$,
2. $\deg(F + G) \leq \max(\deg(F), \deg(G))$

since $[FG] = [F][G]$ and $[F + G] = [F] + [G]$.

3.2 Computational Problems

The One-More Discrete Log (OMDL) problem. Given a group generator $g \in \bar{G} = G \setminus \{1_G\}$, let $\mathsf{DL}(\cdot)$ be an oracle that on the first Q inputs $X_1, \ldots, X_Q \in G$ returns $\mathrm{dlog}_g(X_i)$, and returns $\perp$ on all subsequent inputs; in other words, $\mathsf{DL}(\cdot)$ is a discrete logarithm oracle that can be queried at most Q times. An adversary $\mathcal{A}$'s advantage in the *Q-OMDL problem* is defined as

$$\Pr\left[\begin{array}{c} g \leftarrow \bar{G}; A_0, \ldots, A_Q \leftarrow G \\ (a_0, \ldots, a_Q) \leftarrow \mathcal{A}^{\mathsf{DL}(\cdot)}(g = A_{-1}, A_0, \ldots, A_Q) \end{array} \;:\; g^{a_i} = A_i \text{ for } i = 0, \ldots, Q \right].$$

Note that the adversary $\mathcal{A}$ is given $Q + 1$ uniformly random group elements (in addition to the generator g), and that the (regular) discrete log problem is 0-OMDL.

The Q-Discrete Log (Q-DL) problem. An adversary $\mathcal{A}$'s advantage in the *Q-DL problem* is defined as

$$\Pr\left[\begin{array}{c} x \leftarrow \mathbb{F}_p; g \leftarrow \bar{G} \\ x^* \leftarrow \mathcal{A}(g, g^x, g^{x^2}, \ldots, g^{x^{Q+1}}) \end{array} \;:\; x^* = x \right].$$

The One-More Diffie–Hellman (OMDH) problem. For any $x \in \mathbb{F}_p$, let $\mathsf{Power}(\cdot)$ be an oracle that on the first Q inputs $Y_1, \ldots, Y_Q \in G$ returns Y_i^x, and returns $\perp$ on all subsequent inputs; in other words, $\mathsf{Power}(\cdot)$ is an x-th power oracle that can be queried at most Q times. An adversary $\mathcal{A}$'s advantage in the *Q-OMDH problem* is defined as

$$\Pr\left[\begin{array}{c} x \leftarrow \mathbb{F}_p; g \leftarrow \bar{G}; A_0, \ldots, A_Q \leftarrow G \\ (B_0, \ldots, B_Q) \leftarrow \mathcal{A}^{\mathsf{Power}(\cdot)}(g, X = g^x, A_0, \ldots, A_Q) \end{array} \;:\; B_i = A_i^x \text{ for } i = 0, \ldots, Q \right].$$

Note that the adversary $\mathcal{A}$ is given $Q + 1$ uniformly random group elements (in addition to g and g^x), and that the (regular) computational Diffie–Hellman problem is 0-OMDH.

Since different sources [BFP21, JKKX17] disagree on whether to provide X to the adversary or not, we define the *Q-OMDH2 problem* to be the same as Q-OMDH, except that $X = g^x$ is not given to the adversary.

The One-More Uber (OMU) problem. Let $Q > 0$ be an integer, $\vec{R} = (R_0, \ldots, R_t)$ with $R_i \in \mathbb{F}_p[Z_0, \ldots, Z_{Q+1}]$ and $t \geq Q$. For any $\vec{z} = (z_0, \ldots, z_{Q+1}) \in \mathbb{F}_p^{Q+2}$, let $\mathsf{Power}(\cdot)$ be an oracle that on the first Q inputs $Y_1, \ldots, Y_Q \in G$ returns $Y_i^{z_0}$, and returns $\perp$ on all subsequent inputs. An adversary $\mathcal{A}$'s advantage in the *$(\vec{R}, Q)$-OMU problem* is defined as

$$\Pr\left[\begin{array}{c} \vec{z} \leftarrow \mathbb{F}_p^{Q+2}; g \leftarrow \bar{G}; A_i := g^{R_i(\vec{z})} \\ (B_0, \ldots, B_Q) \leftarrow \mathcal{A}^{\mathsf{Power}(\cdot)}(g, A_0, \ldots, A_t) \end{array} \;:\; B_i = A_i^{z_0} \text{ for } i = 0, \ldots, Q \right].$$

The adversary $\mathcal{A}$ has $Q + 1$ challenges $A_0, \ldots, A_Q$ (of which $\mathcal{A}$ needs to compute the z_0-th power), plus $t - Q$ additional group elements $A_{Q+1}, \ldots, A_t$ that might help $\mathcal{A}$. As special cases, the Q-OMDH problem is the $(\vec{R}, Q)$-OMU problem with $\vec{R} = (Z_1, \ldots, Z_{Q+1}, Z_0)$, and the Q-OMDH2 problem is

the $(\vec{R}, Q)$-OMU problem with $\vec{R} = (Z_1, \ldots, Z_{Q+1})$ (x in the definition of OMDH and OMDH2 above corresponds to z_0 here).

To ensure the problem cannot be trivially solved, throughout the rest of the paper we require $\{1, R_0, \ldots, R_t, Z_0R_0, \ldots, Z_0R_Q\}$ to be linearly independent in $\mathbb{F}_p[Z_0, \ldots, Z_{Q+1}]$.

The Threshold One-More Diffie–Hellman (TOMDH) problem. The TOMDH problem concerns the setting where the exponent x is (t, n)-Shamir secret shared. It is considerably more complicated than OMDH, so we begin with a concrete example. Say $(t, n) = (1, 3)$, i.e., x has 3 shares x_1, x_2, x_3, and knowing any 2 of them is sufficient for recovering x. The adversary $\mathcal{A}$ is given access to 3 oracles $\mathsf{Power}_1(\cdot), \mathsf{Power}_2(\cdot), \mathsf{Power}_3(\cdot)$, which compute the x_1-th, x_2-th, x_3-th powers of the input, respectively. Suppose $\mathcal{A}$ is given sufficiently many challenge group elements, and is allowed to query its oracles 3, 3, 4 times, respectively. How many x-th powers can $\mathcal{A}$ compute?

Let $\mathcal{V} = \{(1, 1, 0), (1, 0, 1), (0, 1, 1)\}$, i.e., $\mathcal{V}$ is the set of binary vectors whose length is 3 and there are 2 ones. Each vector in $\mathcal{V}$ represents a query strategy for $\mathcal{A}$ to compute an x-th power; for example, $(1, 1, 0)$ corresponds to querying $\mathsf{Power}_1(A)$ and $\mathsf{Power}_2(A)$, and using Lagrange interpolation to compute A^x. $\mathcal{A}$'s "query vector" $(3, 3, 4)$ can be expressed as

$$(3, 3, 4) = (1, 1, 0) + 2 \times (1, 0, 1) + 2 \times (0, 1, 1);$$

using this strategy, $\mathcal{A}$ can compute the x-th powers of 1+2+2=5 challenge group elements. A one-more assumption should say that this is the best $\mathcal{A}$ can do, i.e., $\mathcal{A}$ cannot feasibly compute the x-th powers of 6 challenges.

In general, fix positive integers t, n, Q where $t < n$ and define $W(\vec{v}) = \sum_{i=1}^{n} v_i$ for $\vec{v} = (v_1, \ldots, v_n)$. Let $\mathcal{V}_{t+1} = \{\vec{v} \in \{0, 1\}^n \mid W(\vec{v}) = t + 1\}$. For any n-dimensional vector $\vec{q}$, define $C_{t+1}(\vec{q})$ as the maximum integer m for which there are vectors $\vec{v}_1, \ldots, \vec{v}_m \in \mathcal{V}_{t+1}$ such that $\vec{v}_1 + \cdots + \vec{v}_m \leq \vec{q}$; using the example above, when $(t, n) = (1, 3)$, $C_2(3, 3, 4) = 5$.[5]

For any polynomial $P(\cdot)$ with degree t, let $x = P(0), x_1 = P(1), \ldots, x_n = P(n)$. Let $\mathsf{Power}(\cdot, \cdot)$ be an oracle that on input (i, Y) returns Y^{x_i}, subject to the condition that $C_{t+1}(\vec{q}) \leq Q$, where $\vec{q} = (q_1, \ldots, q_n)$ and q_i is the number of $(i, \star)$ queries made so far. An adversary $\mathcal{A}$'s advantage in the (t, n, Q)-*TOMDH problem* is defined as

$$\Pr\left[\begin{array}{c} P \leftarrow \{\mathcal{P} \in \mathbb{F}_p[X] \mid \deg(\mathcal{P}) = t\}; g \leftarrow \bar{G}; A_0, \ldots, A_Q \leftarrow G \\ (B_0, \ldots, B_Q) \leftarrow \mathcal{A}^{\mathsf{Power}(\cdot,\cdot)}(g, A_0, \ldots, A_Q) \end{array} : \; B_i = A_i^x \text{ for } i = 0, \ldots, Q \right].$$

Note that the Q-OMDH2 problem is the $(0, 1, Q)$-TOMDH problem.

Finally, we extend the TOMDH problem to allow shares to be (statically) corrupted; that is, the adversary $\mathcal{A}$ can decide a subset of all x_i's. Fix non-negative integer $t' \leq t$ and $F = \{f_1, \ldots, f_{t'}\} \subset [n]$: $\mathcal{A}$ can decide the values $P(f_i)$. Now $\mathcal{A}$ only needs to make $t - t' + 1$ queries to $\mathsf{Power}(\cdot, \cdot)$ to compute an x-th power via Lagrange interpolation, so we now require that $C_{t-t'+1}(\vec{q}) \leq Q$ (and can assume without loss of generality that $q_i = 0$ if $\mathcal{A}$ has set the i-th share). $\mathcal{A}$'s advantage in the (t', t, n, Q)-*TOMDH problem* is defined as

$$\Pr\left[\begin{array}{c} F' = \{f'_1, \ldots, f'_{t'}\} \leftarrow \mathcal{A} : F' \subset \mathbb{F}_p; \\ P \leftarrow \{\mathcal{P} \in \mathbb{F}_p[X] \mid \deg(\mathcal{P}) = t \wedge \mathcal{P}(f_i) = f'_i\}; g \leftarrow \bar{G}; A_0, \ldots, A_Q \leftarrow G \\ (B_0, \ldots, B_Q) \leftarrow \mathcal{A}^{\mathsf{Power}(\cdot,\cdot)}(g, A_0, \ldots, A_Q) \end{array} : \; B_i = A_i^x \text{ for } i = 0, \ldots, Q \right].$$

[JKKX17] also lets the number of challenges N vary instead of being fixed at $Q + 1$. However

[5] [JKKX17] uses the same example and says $C_2(3, 3, 4) = 4$, which is incorrect.

by [JKKX17, Theorem 5] the problems are equivalent, so in this work we set $N = Q + 1$ without loss of generality.

For all computational problems above, we say an adversary $\mathcal{A}$ (T, ϵ)-*solves* the problem if $\mathcal{A}$'s runtime is at most T, and its advantage in the game is at least ϵ.

3.3 Lemmas

We now state several technical lemmas that will be utilized in our proofs.

Lemma 3.1 ([BFL20, Lemma 2.1])**.** *Let $P \in \mathbb{F}_p[X_1 \dots X_m]$ be a non-zero polynomial of total degree d. Define $Q(X) \in (\mathbb{F}_p[Y_1, \dots, Y_m, W_1, \dots, W_m])[X]$ as $Q(X) = P(Y_1X + W_1, \dots, Y_mX + W_m)$. Then the coefficient of maximal degree of Q is a polynomial in $\mathbb{F}_p[Y_1, \dots, Y_m]$ of degree d.*

Lemma 3.2 ([DL78])**.** *Let $P \in \mathbb{F}_p[X_1 \dots X_m]$ be a non-zero polynomial of total degree d. Let $r_1, \dots, r_m$ be independently and uniformly sampled from $\mathbb{F}_p^\times$. Then*

$$\Pr[P(r_1, \dots, r_m) = 0] \leq \frac{d}{p-1}.$$

Given $F \in \mathbb{F}_p[X_1, \dots, X_m]$ with $F \neq 0$, define a sequence of polynomials $\mathcal{S}(F) = (H_1, \dots, H_m)$ as follows:

- $H_1 = F$,
- For each $i \in \{2, \dots, m\}$: if $H_{i-1} = 0$ then $H_i = 0$. Otherwise, write H_{i-1} as a polynomial in X_{i-1} with coefficients in $\mathbb{F}_p[X_i, \dots, X_m]$. That is, write

$$H_{i-1} = \sum_{j=0}^{d} G_j(X_i, \dots, X_m) X_{i-1}^j.$$

  Let j^* be the smallest index such that $G_{j^*} \neq 0$ and set $H_i = G_{j^*}$. If no such index exists, set $H_i = 0$. One can easily see $H_i \in \mathbb{F}_p[X_i, \dots, X_m]$.

Lemma 3.3 ([Rot22, Lemma 5.5])**.** *Let $F \in \mathbb{F}_p[X_1, \dots, X_m]$ with $F \neq 0$ and $\mathcal{S}(F) = (H_1, \dots, H_m)$. Then*

1. *$H_i \neq 0$ for $1 \leq i \leq m$,*
2. *For every $\vec{\alpha} \in \mathbb{F}_p^m$ such that $F(\alpha) = 0$, there is some i^* such that the univariate polynomial $V_{i^*}(X_{i^*}) = H_{i^*}(X_{i^*}, \alpha_{i^*+1}, \dots, \alpha_m)$ is not the zero polynomial, and $V_{i^*}(\alpha_{i^*}) = 0$.*

The following is a generalization of [BFL20, Lemma 9.2]:

Lemma 3.4. *Let $F \in \mathbb{F}_p(X_1, \dots, X_m)$ and let $0 \neq P \in \mathbb{F}_p[X_1, \dots, X_m]$ have degree at most 1. If F^2P is a polynomial and has degree at most 1, then F is constant.*

(When $m = 1$, Lem. 3.4 follows from [BFL20, Lemma 9.2].) The proof of Lem. 3.4 proceeds identically to the proof of [BFL20, Lemma 9.2], since $\mathbb{F}_p[X_1, \dots, X_m]$ is a unique factorization domain and the degree function for multivariable polynomials enjoys the same properties as for univariate polynomials (e.g., $\deg(F + G) \leq \max(\deg(F), \deg(G))$.

Given $\vec{w} \in \mathbb{F}_p^t$, let $(\vec{w})_i$ be the i-th coordinate of $\vec{w}$ and $\vec{v} \cdot \vec{w} = \sum_{i=1}^t v_i w_i$ be the inner product of vectors. Define $H(\vec{w}) = \{\vec{x} \in \mathbb{F}_p^t \mid \vec{x} \cdot \vec{w} = 0\}$ to be the hyperplane defined by $\vec{w}$ and $H(\vec{w}_1, \dots, \vec{w}_k) = \bigcap_{i=1}^k H(\vec{w}_i)$.

Lemma 3.5. *If $\vec{w}_1, \ldots, \vec{w}_k \in \mathbb{F}_p^t$ are sampled according to some probability distribution and $k \leq t$, then*

$$\Pr[\dim(H(\vec{w}_1, \ldots, \vec{w}_k)) = t - k] = \Pr[\dim(\vec{w}_1, \ldots, \vec{w}_k) = k] =$$
$$\Pr[\vec{w}_1 \neq 0] \prod_{i=1}^{k-1} \Pr[\vec{w}_{i+1} \notin \mathrm{Span}(\vec{w}_1, \ldots, \vec{w}_i) \mid \dim(\vec{w}_1, \ldots, \vec{w}_i) = i].$$

Proof. The first equality follows from the rank-nullity theorem (and the fact that row rank equals column rank). For the second, we proceed by induction. For $k = 1$ the equality is trivial. Suppose the equality holds for k; we show it for $k+1$. Since $\dim(\vec{w}_1, \ldots, \vec{w}_{k+1}) = k+1$ implies $\dim(\vec{w}_1, \ldots, \vec{w}_k) = k$ we have

$$\begin{aligned}&\Pr[\dim(\vec{w}_1, \ldots, \vec{w}_{k+1}) = k+1] \\ =&\Pr[\dim(\vec{w}_1, \ldots, \vec{w}_{k+1}) = k+1 \mid \dim(\vec{w}_1, \ldots, \vec{w}_k) = k] \Pr[\dim(\vec{w}_1, \ldots, \vec{w}_k) = k].\end{aligned}$$

If $\dim(\vec{w}_1, \ldots, \vec{w}_k) = k$ then $\dim(\vec{w}_1, \ldots, \vec{w}_{k+1}) = k+1$ if and only if $\vec{w}_{k+1} \notin \mathrm{Span}(\vec{w}_1, \ldots, \vec{w}_k)$; therefore

$$\begin{aligned}&\Pr[\dim(\vec{w}_1, \ldots, \vec{w}_{k+1}) = k+1] \\ =&\Pr[\vec{w}_{k+1} \notin \mathrm{Span}(\vec{w}_1, \ldots, \vec{w}_k) \mid \dim(\vec{w}_1, \ldots, \vec{w}_k) = k] \Pr[\dim(\vec{w}_1, \ldots, \vec{w}_k) = k] \\ \overset{\text{hypothesis}}{=\!=}&\Pr[\vec{w}_1 \neq 0] \prod_{i=1}^{k} \Pr[\vec{w}_{i+1} \notin \mathrm{Span}(\vec{w}_1, \ldots, \vec{w}_i) \mid \dim(\vec{w}_1, \ldots, \vec{w}_i) = i].\end{aligned}$$

□

4 Reductions Between $(\vec{R}, Q)$-OMU and Q'-DL

In this section, we establish the following corollary:

Corollary 4.1. *For any $Q > 0$, Q-OMDH is equivalent to Q-DL, and Q-OMDH2 is equivalent to $(Q-1)$-DL.*

To do so, we give several reductions in both directions between $(\vec{R}, Q)$-OMU and Q'-DL. The reductions to Q'-DL are simple and do not require the AGM, while the reductions from Q'-DL are more involved and are tight (in contrast to the reductions in Sect. 5). Our results do not exactly characterize the hardness of all OMU assumptions in general, but they suffice for OMDH and OMDH2 (for details see the end of this section).

Theorem 4.2. *For any $Q > 0$, there is a reduction from $(\vec{R}, Q)$-OMU to $(Q-1)$-DL; if $R_i(\vec{Z}) = Z_0^k$ for some i and $1 \leq k \leq Q+1$, there is a reduction from $(\vec{R}, Q)$-OMU to Q-DL.*

Concretely,

1. *Suppose $\mathcal{A}$ (T, ϵ)-solves $(Q-1)$-DL. Then there is a reduction $\mathcal{R}^{\mathcal{A}}$ that $(T + 2(Q+1)\lambda, \epsilon)$-solves $(\vec{R}, Q)$-OMU.*
2. *Suppose $\mathcal{A}$ (T, ϵ)-solves Q-DL. Then there is a reduction $(\mathcal{R}')^{\mathcal{A}}$ that $(T + 2(Q+1)\lambda, \epsilon)$-solves $(\vec{R}, Q)$-OMU, if $R_i(\vec{Z}) = Z_0^k$ for some i and $1 \leq k \leq Q+1$.*

Proof. For (1), given such an adversary $\mathcal{A}$ for $(Q-1)$-DL, we define a reduction $\mathcal{R}$ that uses $\mathcal{A}$ to solve $(\vec{R}, Q)$-OMU as follows:

Reduction $\mathcal{R}$:

1. On $(\vec{R}, Q)$-OMU challenge $(g, A_0, \ldots, A_t)$, $\mathcal{R}$ queries $X_1 := \mathsf{Power}(g), X_2 := \mathsf{Power}(X)$, $\ldots, X_Q := \mathsf{Power}(X_{Q-1})$.
2. $\mathcal{R}$ runs $\mathcal{A}$ on $(Q-1)$-DL challenge $(g, X_1, X_2, \ldots, X_Q)$.
3. When $\mathcal{A}$ outputs x^*, $\mathcal{R}$ outputs $(A_0^{x^*}, \ldots, A_Q^{x^*})$.

We have $X_1 = g^{z_0}, X_2 = X_1^{z_0} = g^{z_0^2}, \ldots, X_Q = g^{z_0^Q}$, so $\mathcal{R}$ simulates the $(Q-1)$-DL game to $\mathcal{A}$ correctly. If $\mathcal{A}$ succeeds then $x^* = z_0$, so $\mathcal{R}$ also succeeds. Thus, $\mathcal{R}$'s success probability is no less than $\mathcal{A}$'s. $\mathcal{R}$'s runtime consists of running $\mathcal{A}$ and computing $Q+1$ exponentiations, which is up to $T + 2(Q+1)\lambda$. ($\mathcal{R}$ can additionally be made algebraic by outputting x^* as the algebraic coefficient in step 3.)

For (2), if $R_i(\vec{Z}) = Z_0^k$ for some $k \leq Q+1$, we can construct a reduction $\mathcal{R}'$ to Q-DL that is almost identical: $\mathcal{R}'$ obtains $g^{z_0}, \ldots, g^{z_0^{k-1}}$ via $k-1$ queries to Power on g, and then $g^{z_0^{k+1}}, \ldots, g^{z_0^{Q+1}}$ via $Q-k+1$ queries to Power on A_i. $\mathcal{R}'$ then invokes the Q-DL adversary as before and continues in the same manner. □

In the other direction, we have:

Theorem 4.3. *For any $Q > 0$, let*

$$\begin{aligned} d_0 &= \max_{-1 \leq j \leq t} \deg_{Z_0}(R_j(\vec{Z})), \\ d_1 &= \max_{-1 \leq j \leq t} \deg_{Z_1, \ldots, Z_{Q+1}}(R_j(\vec{Z})), \\ Q_2 &= \max(Q + d_0, d_1) - 1. \end{aligned}$$

Then there is a reduction from Q_2-DL to $(\vec{R}, Q)$-OMU.

Concretely, suppose $\mathcal{A}$ is an algebraic adversary that (T, ϵ)-solves $(\vec{R}, Q)$-OMU.

1. *Let $d = \max_i \deg(R_i)$ and $Q_1 = Q + d - 1$. Then there is a reduction $\mathcal{R}^{\mathcal{A}}$ that $\left(T + O(\lambda), \epsilon - \frac{Q_1+1}{p-1}\right)$-solves Q_1-DL.*
2. *Let $d' = \max(d_0, d_1)$. Then there is a reduction $(\mathcal{R}')^{\mathcal{A}}$ that $\left(T + O(\lambda), \frac{\epsilon}{2}\left(1 - \frac{d_1}{p-1}\right)\right)$-solves Q_2-DL.*[6]

(Observe that $Q_2 \leq Q_1$, so item (2) improves the Q parameter).

Before proving Thm. 4.3 we will need some preliminary results. Suppose $\mathcal{R}$ has access to an adversary $\mathcal{A}$ for $(\vec{R}, Q)$-OMU and that $\mathcal{A}$ is run on an $(\vec{R}, Q)$-OMU instance $(g, g^{R_0(\vec{z})}, \ldots, g^{R_t(\vec{z})})$; to ease notation, let $z_{-1} = 1$ and $R_{-1} = 1$. After $\mathcal{A}$ receives group elements $g^{v_1}, \ldots, g^{v_Q}$ from the Power oracle, to succeed $\mathcal{A}$ must return $B_i = g^{z_0 R_i(\vec{z})}$ for $0 \leq i \leq Q$ with algebraic representations

$$z_0 R_i(\vec{z}) = \sum_{j=-1}^{t} b_{i,j} R_j(\vec{z}) + \sum_{j=1}^{Q} b'_{i,j} v_j. \tag{1}$$

Lemma 4.4. *For any $1 \leq j \leq Q$, v_j is a linear combination of the terms $z_0^\ell R_i(\vec{z})$ for all $1 \leq \ell \leq j$ and $-1 \leq i \leq t$, with coefficients known to $\mathcal{R}$.*

[6] Here we assume that Q, t, d_1, d_2 are all constant and use the big-O notation for the runtime, as the concrete runtime is difficult to calculate and complicated to express (it can be found in the proof). We stress that our theorem statement still holds asymptotically even if Q, t, d_1, d_2 are polynomial in λ.

Proof. Let $\mathcal{A}$'s queries to the Power oracle be $g^{v'_1}, \ldots, g^{v'_Q}$, so $v_j = z_0 v'_j$. The proof is by induction on j. For $j = 1$, the only group elements $\mathcal{A}$ has seen when it makes the first Power query are $g^{R_i(\vec{z})}$, so

$$v'_1 = \sum_{i=-1}^{t} \alpha_i R_i(\vec{z}) \Rightarrow$$
$$v_1 = \sum_{i=-1}^{t} \alpha_i z_0 R_i(\vec{z})$$

for some $\alpha_i \in \mathbb{F}_p$ known to $\mathcal{R}$, so the lemma holds.

Assume the lemma for $1, \ldots, j$; we show it for $j+1$. When making the $(j+1)$-th Power query, $\mathcal{A}$ has seen $g^{R_i(\vec{z})}$ and $g^{v_1}, \ldots, g^{v_j}$, so

$$v'_{j+1} = \sum_{i=-1}^{t} \alpha_i R_i(\vec{z}) + \sum_{i=1}^{j} \alpha'_i v_i \Rightarrow$$
$$v_{j+1} = \sum_{i=-1}^{t} \alpha_i z_0 R_i(\vec{z}) + \sum_{i=1}^{j} \alpha'_i z_0 v_i$$

for some $\alpha_i, \alpha'_i \in \mathbb{F}_p$ known to $\mathcal{R}$. Since $v_1, \ldots, v_j$ are linear combinations of $z_0^\ell R_i(\vec{z})$ for $\ell \leq j$ (with coefficients known to $\mathcal{R}$), v_{j+1} is a linear combination of $z_0^\ell R_i(\vec{z})$ for $\ell \leq j+1$ (also with coefficients known to $\mathcal{R}$). □

We now consider the expressions in Equation (1) as polynomials in *formal variables* $Z_0, \ldots, Z_{Q+1}$, i.e., as elements of $\mathbb{F}_p[Z_0, \ldots, Z_{Q+1}]$ (again to ease notation let $Z_{-1} = 1$). We use uppercase lettering to denote a polynomial in these variables; for example, $V_j \in \mathbb{F}_p[Z_0, \ldots, Z_{Q+1}]$ is the polynomial given by replacing each occurrence of z_i in v_j with Z_i.

Lemma 4.5. *There is some $0 \leq i \leq Q$ such that*

$$S^i(\vec{Z}) \stackrel{\text{def}}{=} -Z_0 R_i(\vec{Z}) + \sum_{j=-1}^{t} b_{i,j} R_j(\vec{Z}) + \sum_{j=1}^{Q} b'_{i,j} V_j \neq 0.$$

(Note that if $\mathcal{A}$ succeeds then $S^i(\vec{z}) = 0$ for all i; in other words, there is an i such that $S^i(\vec{Z})$ is a non-zero polynomial, but it evaluates to 0 on $\vec{z}$.)

Proof. Suppose for the sake of contradiction

$$Z_0 R_i(\vec{Z}) = \sum_{j=-1}^{t} b_{i,j} R_j(\vec{Z}) + \sum_{j=1}^{Q} b'_{i,j} V_j \tag{2}$$

for $0 \leq i \leq Q$. Let $W = \{1, R_0(\vec{Z}), \ldots, R_t(\vec{Z}), V_1, \ldots, V_Q\}$. Since $|W| = t + Q + 2$, $\mathrm{Span}(W)$ has dimension at most $t + Q + 2$ as an $\mathbb{F}_p$-vector space. By Equation (2), $Z_0 R_i(\vec{Z}) \in \mathrm{Span}(W)$ for $0 \leq i \leq Q$, so $W' = \{1, R_0(\vec{Z}), \ldots, R_t(\vec{Z}), Z_0 R_0(\vec{Z}), \ldots, Z_0 R_Q(\vec{Z})\} \subset \mathrm{Span}(W)$. But we have assumed that the OMU instance is non-trivial, i.e., W' is a set of $t + Q + 3$ independent elements and thus cannot be a subset of $\mathrm{Span}(W)$. This forms a contradiction. □

Let i^* be the smallest i satisfying the condition of Lem. 4.5, and let $S = S^{i^*}$. Write

$$S(\vec{Z}) = \sum_{j=0}^{d_S} P_j(Z_1, \ldots, Z_{Q+1}) Z_0^j \tag{3}$$

where d_S is the degree of Z_0 in S. Since $S(\vec{Z}) \neq 0$, let j^* be the smallest index so $P_{j^*}(Z_1, \ldots, Z_{Q+1}) \neq 0$. Let $V(Z_0) = S(Z_0, z_1, \ldots, z_{Q+1})$. We can now prove Thm. 4.3.

Proof of Thm. 4.3.

Proof of (1). Given such an adversary $\mathcal{A}$ for $(\vec{R}, Q)$-OMU, we define a reduction $\mathcal{R}$ that uses $\mathcal{A}$ to solve Q_1-DL as follows:

Reduction $\mathcal{R}$:

1. On Q_1-DL challenge $(g, g^x, g^{x^2}, \ldots, g^{x^{Q_1+1}})$, $\mathcal{R}$ samples $y_i \leftarrow \mathbb{F}_p^\times, w_i \leftarrow \mathbb{F}_p$ for $0 \leq i \leq Q+1$. Since $Q_1 + 1 = Q + d > d$, $\mathcal{R}$ knows $g, g^x, \ldots, g^{x^d}$ and thus can compute $g^{R_0(\vec{z})}, \ldots, g^{R_t(\vec{z})}$, where $z_i = y_i x + w_i$. $\mathcal{R}$ runs $\mathcal{A}$ on $(g, g^{R_0(\vec{z})}, \ldots, g^{R_t(\vec{z})})$.

2. When $\mathcal{A}$ queries Power, by Lem. 4.4 $\mathcal{R}$ can answer if it knows $z_0^\ell R_i(\vec{z})$ for all $1 \leq \ell \leq Q$ and $-1 \leq i \leq t$. Note that $\deg(z_0^\ell R_i(\vec{z})) \leq Q + d$, so $\mathcal{R}$ indeed can compute all of them using its Q_1-DL challenges.

3. When $\mathcal{A}$ outputs $B_0, \ldots, B_Q$ together with algebraic coefficients $b_{i,j}, b'_{i,j}$ (see Equation (1)), $\mathcal{R}$ defines $S^i(\vec{Z})$ using $b_{i,j}, b'_{i,j}$ (see Lem. 4.5), finds $S(\vec{Z}) = S^{i^*}(\vec{Z})$, and rewrites $S(\vec{Z})$ as a polynomial of X with $Z_i = y_i X + w_i$:

$$S^*(X) = S(y_0 X + w_0, y_1 X + w_1, \ldots, y_{Q+1} X + w_{Q+1}).$$

   If $S^*(X) = 0$ then $\mathcal{R}$ outputs $\perp$ and aborts. Otherwise, it factors $S^*(X)$ and computes all roots. If for some root x^* we have $g^{x^*} = g^x$ then $\mathcal{R}$ returns x^*; otherwise $\mathcal{R}$ returns $\perp$.

Analysis of $\mathcal{R}$. The analysis follows the same template as in the proof of [BFL20, Theorem 3.5].

Note that $\mathcal{R}$ simulates the $(\vec{R}, Q)$-OMU game to $\mathcal{A}$ correctly. If $\mathcal{A}$ succeeds then $S^*(x) = S(\vec{z}) = 0$, so x is a root of S^*. Therefore as long as $S^*(X) \neq 0$, $\mathcal{R}$ returns x. It remains to upper-bound the probability that $S^*(X) = 0$.

Interpreting S^* as an element of $(\mathbb{F}_p[Y_1, \ldots, Y_m, W_1, \ldots, W_m])[X]$, by Lem. 3.1 the maximal coefficient of S^* is an element S^*_{max} of $\mathbb{F}_p[Y_1, \ldots, Y_m]$ with total degree equal to the maximal total degree of S. Note that the behavior of $\mathcal{A}$ is independent of the values $y_i x$ since they are masked by random w_i, i.e., $\mathcal{A}$'s view only contains some functions of $z_i = y_i x + w_i$ which is independent of $y_i x$. Therefore the values $b_{i,j}, b'_{i,j}, v_j$ are independent of $\vec{y}$, thus S, S^*, S^*_{max} are also independent of $\vec{y}$. The probability that $S^* = 0$ is then upper-bounded by the probability that $S^*_{max}(\vec{y}) = 0$ for a random point $\vec{y}$. By Lem. 3.2 this latter probability is at most $\frac{\deg(S^*)}{p-1}$. Since $\deg(S^*)$ is at most the total degree of S^{i^*}, which is at most $Q_1 + 1$, $S^*(X) = 0$ with probability at most $\frac{Q_1+1}{p-1}$. Thus, $\mathcal{R}$'s success probability is at least $\epsilon - \frac{Q_1+1}{p-1}$.

$\mathcal{R}$'s runtime depends on the polynomials $\vec{R}$ (i.e., the concrete instance of the problem), but in the worst case, computing each $g^{R_i(x)}$ involves $d+1$ exponentiations and d multiplications, which take time $2(d+1)\lambda + d$; therefore, step 1 takes time $t[2(d+1)\lambda + d]$. In step 2, answering each Power query involves $Q+d+1$ exponentiations and $Q+d$ multiplications, which take time $2(Q+d+1)\lambda + Q + d$; adding the

time of running $\mathcal{A}$, step 2 takes time $T+Q[2(Q+d+1)\lambda+Q+d]$. Step 3 involves no group operations and is thus considered "free". Overall, $\mathcal{R}$'s runtime is $T+t[2(d+1)\lambda+d]+Q[2(Q+d+1)\lambda+Q+d] = T+O(\lambda)$.

Proof of (2). Given such an adversary $\mathcal{A}$ for $(\vec{R},Q)$-OMU, we define a reduction $\mathcal{R}'$ that uses $\mathcal{A}$ to solve Q_2-DL as follows:

Reduction $\mathcal{R}'$:

1. On Q_2-DL challenge $(g, g^x, g^{x^2}, \ldots, g^{x^{Q_2+1}})$, $\mathcal{R}'$ samples a bit $b \leftarrow \{0,1\}$ and

   - If $b=0$: $\mathcal{R}'$ samples $w_i \leftarrow \mathbb{F}_p$ for $1 \le i \le Q+1$. Since $Q_2+1 \ge Q+d_0 > d_0$, $\mathcal{R}'$ knows $g, g^x, \ldots, g^{x^{d_0}}$ and thus can compute $g^{R_0(\vec{z})}, \ldots, g^{R_t(\vec{z})}$, where $z_0 = x$ and $z_i = w_i$.
   - If $b=1$: $\mathcal{R}'$ samples $w_0 \leftarrow \mathbb{F}_p$, $w_i \leftarrow \mathbb{F}_p$, $y_j \leftarrow \mathbb{F}_p^\times$ for $1 \le i \le Q+1$. Since $Q_2+1 \ge d_1$, $\mathcal{R}'$ knows $g, g^x, \ldots, g^{x^{d_1}}$ and thus can compute $g^{R_0(\vec{z})}, \ldots, g^{R_t(\vec{z})}$, where $z_0 = w_0$ and $z_i = y_i x + w_i$.

   $\mathcal{R}'$ runs $\mathcal{A}$ on $(g, g^{R_0(\vec{z})}, \ldots, g^{R_t(\vec{z})})$.

2. When $\mathcal{A}$ queries Power, by Lem. 4.4 $\mathcal{R}'$ can answer if it knows $z_0^\ell R_i(\vec{z})$ for all $1 \le \ell \le Q$ and $-1 \le i \le t$. Note that $\deg_{z_0}(z_0^\ell R_i(\vec{z})) \le Q+d_0$ (for $b=0$) and $\deg_{z_1,\ldots,z_{Q+1}}(z_0^\ell R_i(\vec{z})) \le d_1$ (for $b=1$), so $\mathcal{R}'$ indeed can compute all of them using its Q_2-DL challenges.

3. When $\mathcal{A}$ outputs $B_0, \ldots, B_Q$ together with algebraic coefficients $b_{i,j}, b'_{i,j}$ (see Equation (1)), $\mathcal{R}'$ defines $S^i(\vec{Z})$ using $b_{i,j}, b'_{i,j}$ (see Lem. 4.5) and finds $S(\vec{Z}) = S^{i^*}(\vec{Z})$. Then:

   - If $b=0$: $\mathcal{R}'$ computes $V(Z_0)$ using z_i. If $V(X)=0$ then $\mathcal{R}'$ outputs $\perp$ and aborts. Otherwise, it factors $V(X)$ and computes all roots.
   - If $b=1$: $\mathcal{R}'$ computes $P^*(X) = P_{j^*}(y_1X+w_1, \ldots, y_{Q+1}X+w_{Q+1})$ using w_i and y_i. If $P^*(X)=0$ then $\mathcal{R}'$ outputs $\perp$ and aborts. Otherwise, it factors $P^*(X)$ and computes all roots.

   Either way, if for some root x^* we have $g^{x^*} = g^x$ then $\mathcal{R}'$ returns x^*; otherwise $\mathcal{R}'$ returns $\perp$.

Analysis of $\mathcal{R}'$. Note that $\mathcal{R}$ simulates the $(\vec{R},Q)$-OMU game to $\mathcal{A}$ correctly. Suppose $\mathcal{A}$ succeeds. Then:

- If $b=0$ and $V(Z_0)$ is a non-zero polynomial: by Lem. 4.5 it has $z_0 = x$ as a root. Therefore, $\mathcal{R}'$ returns x.
- If $b=1$ and $V(Z_0)=0$ as a polynomial: by Equation (3) we must have $P_{j^*}(z_1, \ldots, z_{Q+1}) = 0$ and $P_{j^*}(Z_1, \ldots, Z_{Q+1}) \neq 0$. But when $b=1$ $\mathcal{R}'$ computes x in almost the same manner as $\mathcal{R}$; the only difference is that $\mathcal{R}$ uses polynomial S and $\mathcal{R}'$ uses polynomial P_{j^*}. Using the same analysis, $\mathcal{R}'$ returns x with probability at least $1 - \frac{d_1}{p-1}$.

Therefore,

$$\Pr[\mathcal{R}' \text{ succeeds} \mid \mathcal{A} \text{ succeeds} \wedge b=0 \wedge E] = 1,$$
$$\Pr[\mathcal{R}' \text{ succeeds} \mid \mathcal{A} \text{ succeeds} \wedge b=1 \wedge \bar{E}] \ge 1 - \frac{d_1}{p-1},$$

which gives

$$\begin{aligned}\Pr[\mathcal{R}' \text{ succeeds}] &\geq \Pr[\mathcal{R}' \text{ succeeds} \wedge \mathcal{A} \text{ succeeds}] \\ &= \epsilon \cdot \Pr[\mathcal{R}' \text{ succeeds} \mid \mathcal{A} \text{ succeeds}] \\ &= \frac{\epsilon}{2}\left(\Pr[\mathcal{R}' \text{ succeeds} \mid \mathcal{A} \text{ succeeds} \wedge b = 0] + \Pr[\mathcal{R}' \text{ succeeds} \mid \mathcal{A} \text{ succeeds} \wedge b = 1]\right) \\ &\geq \frac{\epsilon}{2}\left[\Pr[E] + \Pr[\bar{E}]\left(1 - \frac{d_1}{p-1}\right)\right] \\ &\geq \frac{\epsilon}{2}\left(1 - \frac{d_1}{p-1}\right).\end{aligned}$$

The runtime analysis of $\mathcal{R}'$ is identical to that of $\mathcal{R}$, except that d is replaced by $d' = \max(d_0, d_1)$. □

Applications to OMDH and OMDH2. We now apply our general results on OMU to OMDH and OMDH2. We have:

- Applying item (1) of Thm. 4.2 to Q-OMDH2 and item (2) to Q-OMDH, we obtain that there is a reduction from Q-OMDH2 to $(Q-1)$-DL and Q-OMDH to Q-DL;
- Applying item (1) of Thm. 4.3 to Q-OMDH and item (2) to Q-OMDH2, we obtain that there is a reduction from Q-DL to Q-OMDH and $(Q-1)$-DL to Q-OMDH2.

From this Cor. 4.1 easily follows. This establishes a separation between Q-OMDH(2) for different values of Q, as [BFL20, Section 9] has shown that Q-DL for different values of Q are separate (assuming the reduction is algebraic). Furthermore, our result also separates Q-OMDH(2) for Q and Q'-OMDL for *any* positive Q and Q', as [BFL20, Section 10] has shown that Q-DL and Q'-OMDL are separate (unless Q-DL is easy).

5 Reductions Between (t', t, n, Q)-TOMDH and Q'-DL

Reduction from (t', t, n, Q)-TOMDH to $(Q-1)$-DL. We first give a reduction from (t', t, n, Q)-TOMDH to $(Q-1)$-DL that does not rely on the AGM (analogous to Thm. 4.2). This reduction is simple so we only provide a sketch. Suppose $\mathcal{A}$ is a $(Q-1)$-DL solver. Reduction $\mathcal{R}$ receives a (t', t, n, Q)-TOMDH challenge $(g, g^{r_1}, \ldots, g^{r_{Q+1}})$ with (statically) corruptible shares $F = \{f_1, \ldots, f_{t'}\} \subset [n]$. $\mathcal{R}$ chooses 0 as the share values, so $P(f_i) = 0$ where P is the secret polynomial in the TOMDH challenge. $\mathcal{R}$ chooses any $U \subset [n] \setminus F$ with $|U| = t - t' + 1$ (such a U must exist since $n \geq t+1$). $\mathcal{R}$ will use the query vector $\vec{q} = (q_1, \ldots, q_n)$, defined by $q_i = Q$ if $i \in U$ and $q_i = 0$ otherwise; obviously $C_{t-t'+1}(\vec{q}) = Q$. $\mathcal{R}$ then uses the $\mathsf{Power}(\cdot,\cdot)$ oracle to simulate a $(Q-1)$-DL challenge for $\mathcal{A}$, as follows. Suppose $\mathcal{R}$ wants to compute a^k for $k = P(0)$. $\mathcal{R}$ knows that $a^{P(f_i)} = 1$ for $f_i \in F$, and can use $\mathsf{Power}(\cdot,\cdot)$ to compute $a^{P(u)}$ for all $u \in U$. $\mathcal{R}$ can then compute a^k from these $t+1$ values by Lagrange interpolation. By repeating this for $a = g, g^k, \ldots, g^{k^{Q-1}}$, $\mathcal{R}$ computes $(g, g^k, \ldots, g^{k^Q})$. $\mathcal{R}$ invokes $\mathcal{A}$ on this challenge, and receives k^* as output. $\mathcal{R}$ then outputs $\{(g^{r_i})^{k^*}\}_{i \in [Q+1]}$. It is easily seen that $\mathcal{R}$'s advantage is the same as $\mathcal{A}$'s. ($\mathcal{R}$ can be made algebraic if $\mathcal{A}$ is.)

The rest of this section is dedicated to giving a reduction in the other direction, namely there is a reduction from (t', t, n, Q)-TOMDH to $(Q(n-t)-1)$-DL. The main theorem is:

Theorem 5.1. *For any $t', t, n, Q > 0$ with $t' \leq t < n$, there is a reduction from $(Q(n-t)-1)$-DL to (t', t, n, Q)-TOMDH.*

Concretely, suppose $\mathcal{A}$ (T,ϵ)-solves (t',t,n,Q)-TOMDH, and let $t_0 = t - t' + 1$. Then there is a reduction $\mathcal{R}^{\mathcal{A}}$ that $(T + O(\lambda), \epsilon')$-solves $(Q(n-t)-1)$-DL, where

$$\epsilon' = \frac{\epsilon}{(Q+t_0+1)^2}\left(Q + \frac{t_0}{\binom{n-t'}{t-t'}} + 1\right).$$

The proof of Thm. 5.1 appears in Sect. 5.3, modulo a technical lemma that is subsequently proven in Sect. 5.4.

5.1 Preliminary Results

Suppose $\mathcal{R}$ has access to an adversary $\mathcal{A}$ for (t',t,n,Q)-TOMDH and that $\mathcal{A}$ is run on an (t',t,n,Q)-TOMDH instance $(g, g^{r_1}, \ldots, g^{r_{Q+1}})$ (to ease notation set $r_0 = 1$) with secret polynomial $P(X) = \sum_{i=0}^{t} a_i X^i$, vector of coefficients $\vec{a} = (a_0, \ldots, a_t)$, and statically corruptible shares $F = \{f_1, \ldots, f_{t'}\} \subset [n]$, with values $F' = \{f'_1, \ldots, f'_{t'}\} \subset \mathbb{F}_p$ supplied by $\mathcal{A}$ such that $P(f_i) = f'_i$. Let $\vec{q} = (q_1, \ldots, q_n)$ be the query vector of $\mathcal{A}$: that is, $\mathcal{A}$ makes q_j queries to $\mathsf{Power}(j,\cdot)$ with $C_{t-t'+1}(\vec{q}) \le Q$ and $q_j = 0$ for $j \in F$. Let g^{v_i} be the result of the i-th query to $\mathsf{Power}(\cdot,\cdot)$, and denote the first argument to the query by k_i (i.e., $\mathcal{A}$ evaluates the $x_{k_i} = P(k_i)$-th power of the second argument). To succeed $\mathcal{A}$ must return elements B_j for indices $1 \le j \le Q+1$ with algebraic representations

$$a_0 r_j = \sum_{u=0}^{Q+1} b_u^{(j)} r_u + \sum_{u=1}^{w} c_u^{(j)} v_u. \tag{4}$$

Lemma 5.2. *Let $\vec{n}_s = (1, s, s^2, \ldots, s^t)$. Then*

$$v_i = \sum_{Z \subset [i], i \in Z} \left[\left(\sum_{j=0}^{Q+1} \beta_{jZ}^{(i)} r_j \right) \prod_{\ell \in Z} (\vec{n}_{k_\ell}^\top \vec{a}) \right]$$

for some choice of coefficients $\beta_{jZ}^{(i)} \in \mathbb{F}_p$ known to $\mathcal{R}$.

(This is analogous to Lem. 4.4.)

Proof. By induction on i. For $i = 1$, the only group elements $\mathcal{A}$ has seen when it makes the first $\mathsf{Power}(k_1,\cdot)$ query are $g^{r_0}, \ldots, g^{r_{Q+1}}$, so the second input is of the form $\sum_{j=0}^{Q+1} \beta_j r_j$, so

$$v_1 = (\vec{n}_{k_1}^\top \vec{a}) \left(\sum_{j=0}^{Q+1} \beta_j r_j \right).$$

Assume the lemma for $1, \ldots, i$; we show it for $i+1$. When making the $(i+1)$-th Power query $\mathcal{A}$ has seen $g^{r_0}, \ldots, g^{r_{Q+1}}$ and $v_1, \ldots, v_i$, so the input is of the form

$$\begin{aligned} &\sum_{j=0}^{Q+1} \beta_j r_j + \sum_{j=1}^{i} \gamma_j v_j \\ &\overset{\text{hypothesis}}{=\!=} \sum_{j=0}^{Q+1} \beta_j r_j + \sum_{j=1}^{i} \gamma_j \left(\sum_{Z \subset [j], j \in Z} \left[\left(\sum_{j'=0}^{Q+1} \beta_{j'Z}^{(j)} r_{j'} \right) \prod_{\ell \in Z} (\vec{n}_{k_\ell}^\top \vec{a}) \right] \right). \end{aligned}$$

We then have

$$v_{i+1} = (\vec{n}_{k_{i+1}}^\top \vec{a}) \sum_{j=0}^{Q+1} \beta_j r_j + (\vec{n}_{k_{i+1}}^\top \vec{a}) \sum_{j=1}^{i} \gamma_j \left(\sum_{Z \subset [j], j \in Z} \left[\left(\sum_{j'=0}^{Q+1} \beta_{j'Z}^{(j)} r_{j'} \right) \prod_{\ell \in Z} (\vec{n}_{k_\ell}^\top \vec{a}) \right] \right).$$

Note that as j ranges from 1 to i, Z ranges over all nonempty subsets of $[i]$. We can then rewrite as

$$v_{i+1} = (\vec{n}_{k_{i+1}}^\top \vec{a}) \sum_{j=0}^{Q+1} \beta_j r_j + \sum_{\substack{Z \subset [i+1], i+1 \in Z \\ |Z| \geq 2}} \left[\left(\sum_{j'=0}^{Q+1} \beta_{j'Z}^{'(j_Z)} r_{j'} \right) \prod_{\ell \in Z} (\vec{n}_{k_\ell}^\top \vec{a}) \right]$$

where $j_Z = \max_{z \in Z \setminus \{i+1\}} z$ and $\beta_{j'Z}^{'(j_Z)} = \gamma_j \beta_{j'Z}^{(j_Z)}$. Since the first term is the contribution of $Z = \{i+1\}$ and the second term is the contribution of all $Z \neq \{i+1\}$ such that $Z \subset [i+1], i+1 \in Z$, combining these two terms yields all $Z \subset [i+1]$ such that $i+1 \in Z$, which is exactly what's covered in the expression of v_{i+1} in the inductive step, so we are done. □

As in Lem. 4.5, we now consider the above quantities in the formal variables $A_0, \ldots, A_t$ and $R_1, \ldots, R_{Q+1}$ ($R_0 = 1$).

Lemma 5.3. *There is some $1 \leq j \leq Q+1$ such that*

$$S^j(\vec{A}, \vec{R}) \stackrel{\text{def}}{=} -A_0 R_j + \sum_{u=0}^{Q+1} b_u^{(j)} R_u + \sum_{u=1}^{w} c_u^{(j)} V_u \neq 0.$$

(Note that if $\mathcal{A}$ succeeds then $S^j(\vec{a}, \vec{r}) = 0$ for all j.)

The rest of this section and Sect. 5.2 are dedicated to the proof of Lem. 5.3. In this section we deal with the $t' = 0$ (i.e., no corrupted shares) case, and then in Sect. 5.2 we extend it to the $t' > 0$ case.

We first recall a lemma in [JKKX17]. Let $\vec{q} = (q_1, \ldots, q_n) \in \mathbb{N}^n, t \in \mathbb{N}$ and suppose $C_{t+1}(\vec{q}) \leq Q, w = W(\vec{q})$. Let $\vec{k} = (k_1, \ldots, k_w) \in \mathbb{Z}_{>0}^w$ be any vector with q_j of its entries equal to j, for each $1 \leq j \leq n$.

Lemma 5.4 ([JKKX17, Lemma 3])**.** *There are no matrices $\mathbf{A} \in \mathbb{F}_p^{(Q+1) \times w}, \mathbf{B} \in \mathbb{F}_p^{w \times (Q+1)}, \mathbf{K} \in \mathbb{F}_p^{w \times w}$ such that:*

1. *$\mathbf{K}$ is diagonal with entries k_i;*
2. *$\mathbf{AB} = \mathbf{I}$ and $\mathbf{AK}^i\mathbf{B} = \mathbf{0}$ for $1 \leq i \leq t$.*

Proof of Lem. 5.3 when $t' = 0$. This proof follows the same outline as the proof of [JKKX17, Theorem 6]. Assume towards a contradiction that $S^j(\vec{A}, \vec{R}) = 0$ for all j, and let V_i' be V_i but with all terms of degree ≥ 2 in the A_i removed. By Lem. 5.2,

$$V_i' = (\vec{n}_{k_i}^\top \vec{A}) \left(\sum_{j=0}^{Q+1} \beta_j^{(i)} R_j \right)$$

(for notational convenience set $\beta_j^{(i)} = \beta_{j\{i\}}^{(i)}$). Since $A_0 R_j$ has degree ≤ 1 in all A_i, we have

$$A_0 R_j = \sum_{u=0}^{Q+1} b_u^{(j)} R_u + \sum_{u=1}^{w} c_u^{(j)} V_u'$$

for $1 \le j \le Q+1$. These equations can be written in matrix form as

$$\begin{bmatrix} A_0 R_1 \\ \vdots \\ A_0 R_{Q+1} \end{bmatrix} = \mathbf{B}_1 \vec{R} + \vec{b}_0 + \mathbf{C} \begin{bmatrix} V_1' \\ \vdots \\ V_w' \end{bmatrix}, \tag{5}$$

$$\begin{bmatrix} V_1' \\ \vdots \\ V_w' \end{bmatrix} = (\mathbf{B}_2 \vec{R} + \vec{\beta}_0) \odot \begin{bmatrix} \vec{n}_{k_1}^\top \vec{A} \\ \vdots \\ \vec{n}_{k_w}^\top \vec{A} \end{bmatrix},$$

where

$$\mathbf{C} = \begin{bmatrix} c_1^{(1)} & \cdots & c_w^{(1)} \\ \vdots & \ddots & \vdots \\ c_1^{(Q+1)} & \cdots & c_w^{(Q+1)} \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} b_1^{(1)} & \cdots & b_{Q+1}^{(1)} \\ \vdots & \ddots & \vdots \\ b_1^{(Q+1)} & \cdots & b_{Q+1}^{(Q+1)} \end{bmatrix}, \mathbf{B}_2 = \begin{bmatrix} \beta_1^{(1)} & \cdots & \beta_{Q+1}^{(1)} \\ \vdots & \ddots & \vdots \\ \beta_1^{(w)} & \cdots & \beta_{Q+1}^{(w)} \end{bmatrix},$$

$$\vec{R} = \begin{bmatrix} R_1 \\ \vdots \\ R_{Q+1} \end{bmatrix}, \vec{b}_0 = \begin{bmatrix} b_0^{(1)} \\ \vdots \\ b_0^{(Q+1)} \end{bmatrix}, \vec{\beta}_0 = \begin{bmatrix} \beta_0^{(1)} \\ \vdots \\ \beta_0^{(w)} \end{bmatrix}.$$

Let $\vec{\beta}_1 = \mathbf{B}_2 \vec{R} + \vec{\beta}_0, \vec{b}_1 = \mathbf{B}_1 \vec{R} + \vec{b}_0$. We can write

$$\begin{bmatrix} \vec{n}_{k_1}^\top \vec{A} \\ \vdots \\ \vec{n}_{k_w}^\top \vec{A} \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^t A_i k_1^i \\ \vdots \\ \sum_{i=0}^t A_i k_w^i \end{bmatrix} = \sum_{i=0}^t \left(A_i \begin{bmatrix} k_1^i \\ \vdots \\ k_w^i \end{bmatrix} \right),$$

so by properties of the Hadamard product

$$\begin{bmatrix} V_1' \\ \vdots \\ V_w' \end{bmatrix} = \vec{\beta}_1 \odot \begin{bmatrix} \vec{n}_{k_1}^\top \vec{A} \\ \vdots \\ \vec{n}_{k_w}^\top \vec{A} \end{bmatrix} = \sum_{i=0}^t \left(\vec{\beta}_1 \odot A_i \begin{bmatrix} k_1^i \\ \vdots \\ k_w^i \end{bmatrix} \right) = \sum_{i=0}^t A_i \mathbf{K}^i \vec{\beta}_1 \tag{6}$$

where

$$\mathbf{K} = \begin{bmatrix} k_1 & & \\ & \ddots & \\ & & k_w \end{bmatrix}.$$

Substituting Equation (6) into Equation (5) gives

$$\begin{bmatrix} A_0 R_1 \\ \vdots \\ A_0 R_{Q+1} \end{bmatrix} = \vec{b}_1 + \mathbf{C} \left(\sum_{i=0}^t A_i \mathbf{K}^i \vec{\beta}_1 \right) = \vec{b}_1 + \sum_{i=0}^t A_i \mathbf{C} \mathbf{K}^i \vec{\beta}_1. \tag{7}$$

As the above is an equality of polynomials, we have $\mathbf{C}\vec{\beta}_1 = \vec{R}$ and $\mathbf{C}\mathbf{K}^i\vec{\beta}_1 = 0$ for $1 \le i \le t$. Plugging $\vec{\beta}_1$ back in, we get $(\mathbf{C}\mathbf{B}_2 - \mathbf{I})\vec{R} + \mathbf{C}\vec{\beta}_0 = 0$ and $\mathbf{C}\mathbf{K}^i\mathbf{B}_2\vec{R} + \mathbf{C}\mathbf{K}^i\vec{\beta}_0 = 0$ for $1 \le i \le t$. By Lem. 5.4, at least one of

$$\mathbf{C}\mathbf{B}_2 - \mathbf{I}, \mathbf{C}\mathbf{K}\mathbf{B}_2, \mathbf{C}\mathbf{K}^2\mathbf{B}_2, \ldots, \mathbf{C}\mathbf{K}^t\mathbf{B}_2$$

is nonzero, so for some e, i the e-th row of the i-th matrix ($0 \le i \le t$) is nonzero. Denote this row by $\vec{u}^\top$ and let $\vec{c}^\top$ be the e-th row of $\mathbf{C}$. We then have $\vec{u}^\top\vec{R} + \vec{c}^\top\mathbf{K}^i\vec{\beta}_0 = 0$. We have arrived at a contradiction: $\vec{u}^\top$ is nonzero so the left hand-side cannot be the zero polynomial. □

5.2 The Flaw for $t' > 0$

We now prove Lem. 5.3 in the case $t' > 0$: in the process we show the flaw in the proof of [JKKX17, Theorem 7] (hardness of TOMDH in the GGM) and repair it by proving a new technical lemma. Suppose $\mathcal{A}$ has chosen values $\vec{f'} = (f'_1, \ldots, f'_{t'})$ for the corruptible shares $\vec{f} = (f_1, \ldots, f_{t'})$ so that $P(f_i) = f'_i$ (note that f_i are distinct). Written in matrix notation, the equalities $P(f_i) = f'_i$ become

$$[\vec{f}^t, \ldots, \vec{f}, \vec{1}] \begin{bmatrix} A_t \\ A_{t-1} \\ \vdots \\ A_0 \end{bmatrix} = \vec{f'}. \tag{8}$$

By the Vandermonde determinant, $\{\vec{f}^0, \vec{f}, \ldots, \vec{f}^{t'-1}\}$ is linearly independent. Furthermore, since $\vec{f}$ has no zero entries, $\{\vec{f}^u, \vec{f}^{u+1}, \ldots, \vec{f}^{u+t'-1}\}$ is linearly independent for any $u \geq 0$. Therefore bringing Equation (8) to row-reduced echelon form gives

$$\begin{bmatrix} 1 & 0 & \ldots & 0 & \alpha_{t-t'}^{(t)} & \ldots & \alpha_0^{(t)} \\ 0 & 1 & & 0 & \alpha_{t-t'}^{(t-1)} & \ldots & \alpha_0^{(t-1)} \\ \vdots & & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 & \alpha_{t-t'}^{(t-t'+1)} & \ldots & \alpha_0^{(t-t'+1)} \end{bmatrix} \begin{bmatrix} A_t \\ A_{t-1} \\ \vdots \\ A_0 \end{bmatrix} = \begin{bmatrix} \alpha_{-1}^{(t)} \\ \alpha_{-1}^{(t-1)} \\ \vdots \\ \alpha_{-1}^{(t-t'+1)} \end{bmatrix}$$

for coefficients $\{\alpha_j^{(\ell)}\}$. By standard properties of row reduction,

$$A_\ell + \sum_{j=0}^{t-t'} \alpha_j^{(\ell)} A_j = \alpha_{-1}^{(\ell)} \text{ for } t - t' + 1 \leq \ell \leq t, \tag{9}$$

$$\vec{f^j} = \sum_{\ell=t-t'+1}^{t} \alpha_j^{(\ell)} \vec{f^\ell} \text{ for } 0 \leq j \leq t - t'.$$

Returning to the proof of Lem. 5.3 when $t' > 0$, by Equation (9) all A_ℓ are known linear combinations of $A_0, \ldots, A_{t-t'}$, so S^j is a polynomial in $A_0, \ldots, A_{t-t'}, \vec{R}$. Since A_ℓ has degree 1 in $A_0, \ldots, A_{t-t'}$, the previous argument for the $t' = 0$ case goes through up to Equation (7). The proof of [JKKX17, Theorem 7] claims that at this point the equalities $\mathbf{C}\vec{\beta}_1 = \vec{R}$ and $\mathbf{C}\mathbf{K}^i\vec{\beta}_1 = 0$ for $1 \leq i \leq t - t'$ hold. However, this is incorrect: $A_{t-t'+1}, \ldots, A_t$ *have not been set to zero; rather, they are fixed linear combinations of the free variables* $A_0, \ldots, A_{t-t'}$. To give a correct proof, using Equation (9) yields

$$\begin{aligned} \begin{bmatrix} A_0 R_1 \\ \vdots \\ A_0 R_{Q+1} \end{bmatrix} = \vec{b}_1 + \mathbf{C}\left(\sum_{i=0}^{t} A_i \mathbf{K}^i \vec{\beta}_1\right) &= \vec{b}_1 + \sum_{j=0}^{t-t'} A_j \mathbf{C}\mathbf{K}^j \vec{\beta}_1 + \sum_{\ell=t-t'+1}^{t} \left(\alpha_{-1}^{(\ell)} - \sum_{j=0}^{t-t'} \alpha_j^{(\ell)} A_j\right) \mathbf{C}\mathbf{K}^\ell \vec{\beta}_1 \\ &= \vec{b}_1 + \sum_{\ell=t-t'+1}^{t} \alpha_{-1}^{(\ell)} \mathbf{C}\mathbf{K}^\ell \vec{\beta}_1 + \sum_{j=0}^{t-t'} A_j \mathbf{C}\left(\mathbf{K}^j - \sum_{\ell=t-t'+1}^{t} \alpha_j^{(\ell)} \mathbf{K}^\ell\right) \vec{\beta}_1, \end{aligned}$$

Only after this can we equate coefficients of A_j like in Equation (7) for the $t' = 0$ case: doing so

gives $\mathbf{CM}_0\vec{\beta}_1 = \vec{R}, \mathbf{CM}_j\vec{\beta}_1 = \vec{0}, j \in [t-t']$ where

$$\mathbf{M}_j = \mathbf{K}^j - \sum_{\ell=t-t'+1}^{t} \alpha_j^{(\ell)} \mathbf{K}^\ell.$$

Plugging in the definition of β_1 yields

$$(\mathbf{CM}_0\mathbf{B}_2 - \mathbf{I})\vec{R} + \mathbf{CM}_0\vec{\beta}_0 = \vec{0},\ \mathbf{CM}_j\mathbf{B}_2\vec{R} + \mathbf{CM}_j\vec{\beta}_0 = \vec{0} \text{ for } j \in [t-t'].$$

As before, to complete the proof of Lem. 5.3 it suffices to show $\mathbf{CM}_0\mathbf{B}_2 - \mathbf{I}$ or one of $\mathbf{CM}_j\mathbf{B}_2$ is nonzero. Lem. 5.4 is no longer sufficient, so we require the following new lemma.

Lemma 5.5. *Let $\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_{t-t'} \in \mathbb{F}_p^{w \times w}$ be as above. There are no matrices $\mathbf{A} \in \mathbb{F}_p^{(Q+1)\times w}, \mathbf{B} \in \mathbb{F}_p^{w\times(Q+1)}$ such that $\mathbf{AM}_0\mathbf{B} = \mathbf{I}$ and $\mathbf{AM}_j\mathbf{B} = \mathbf{0}$ for $1 \le j \le t-t'$.*

As the proof of Lem. 5.5 is technically complex, we defer it to Sect. 5.4. For now, we present the proof of Thm. 5.1 assuming Lem. 5.5.

5.3 Proof of Thm. 5.1 Assuming Lem. 5.5

Proof. Given such an adversary $\mathcal{A}$ for (t', t, n, Q)-TOMDH, we define a reduction $\mathcal{R}$ that uses $\mathcal{A}$ to solve $(Q(n-t)-1)$-DL as follows:

Reduction $\mathcal{R}$:

1. On $(Q(n-t)-1)$-DL challenge $(g, g^x, \ldots, g^{x^{Q(n-t)}})$, $\mathcal{R}$ first samples an index $i^* \leftarrow [Q+t_0+1]$. (Intuitively i^* is a guess of where the $(Q(n-t)-1)$-DL challenge x should be inserted: if $i^* \le t_0$, $\mathcal{R}$ inserts it into the t_0-th secret share of P; if $i^* > t_0$, $\mathcal{R}$ inserts it into the (i^*-t_0)-th challenge element.) Then $\mathcal{R}$ runs $\mathcal{A}$ and implicitly defines polynomial $P(\cdot)$ as follows: $\mathcal{A}$, given corruptible shares $F = \{f_1, \ldots, f_{t'}\}$ ($\mathcal{R}$ works regardless of which F is used in the TOMDH game), outputs $F' = \{f'_1, \ldots, f'_{t'}\} \subset \mathbb{F}_p$ to $\mathcal{R}$ as the share values. $\mathcal{R}$'s polynomial $P(\cdot)$ must satisfy $P(f_i) = f'_i$ for each i. $\mathcal{R}$ samples a uniform random subset $C = \{c_1, \ldots, c_{t_0}\} \subset [n] \setminus F$, and samples $c'_i \leftarrow \mathbb{F}_p, i \in [t_0]$: if $i^* \le t_0$ $\mathcal{R}$ implicitly defines $P(c_i) = c'_i, P(c_{i^*}) = x$ for $i \in [t_0] \setminus \{i^*\}$, otherwise $\mathcal{R}$ implicitly defines $P(c_i) = c'_i$ for $i \in [t_0]$. We have defined P on $t' + t_0 = t + 1$ points, so P is uniquely determined.

2. Next, if $i^* > t_0$, $\mathcal{R}$ defines $r_i := w_i$ for $i \in [Q+1] \setminus \{i^* - t_0\}$ and $r_{i^*-t_0} := x$; otherwise $\mathcal{R}$ defines $r_i := w_i$ for $i \in [Q+1]$. $\mathcal{R}$ feeds $(g, g^{r_1}, \ldots, g^{r_{Q+1}})$ to $\mathcal{A}$ as the TOMDH challenge.

3. When $\mathcal{A}$ queries $\mathsf{Power}(j, b)$, $\mathcal{R}$ must return $b^{P(j)}$. By Lagrange interpolation $P(j)$ is a linear combination of $\{P(c)\}_{c\in C} \cup F'$ with known coefficients. If $i^* > t_0$, all of these values are known to $\mathcal{R}$, so $\mathcal{R}$ can directly answer the query. Otherwise, all of these values are known to $\mathcal{R}$ except $P(c_{i^*}) = x$. Therefore the query results $\mathcal{R}$ can be computed – by linearity – from (1) the group elements given to $\mathcal{R}$ as input, (2) the algebraic representations of the queries, and (3) the elements in the $(Q(n-t)-1)$-DL challenge according to Lem. 5.2. If the degree of the query output in x is larger than $Q(n-t)$, $\mathcal{R}$ outputs $\mathsf{HighDeg}$ and aborts.

4. When $\mathcal{A}$ outputs $B_0, \ldots, B_Q$, $\mathcal{R}$ defines the nonzero polynomial $S_1(A_0, \ldots, A_{t-t'}, R_1, \ldots, R_{Q+1})$ given by Lem. 5.3, and rewrites it as a polynomial S_2 of $C'_1, \ldots, C'_{t_0}, R_1, \ldots, R_{Q+1}$ with $\sum_{j=0}^{t} A_j c_i^j = C'_i$. By Lagrange interpolation this change of variables is invertible, so S_2 is also nonzero.

After that, $\mathcal{R}$ computes the polynomial V_{i^*} corresponding to S_2 as defined in Lem. 3.3. Note that $\mathcal{R}$ can always compute V_{i^*}, as it is a function of values known to $\mathcal{R}$:

$$V_{i^*} = \begin{cases} V_{i^*}(R_{i^*-t_0}) = H_{i^*}(R_{i^*-t_0}, r_{i^*-t_0+1}, \ldots, r_{Q+1}) & \text{if } i^* > t_0 \\ V_{i^*}(C'_{i^*}) = H_{i^*}(C'_{i^*}, c'_{i^*+1}, \ldots, c'_{t_0}, r_1, \ldots, r_{Q+1}) & \text{otherwise} \end{cases}$$

If $V_{i^*} = 0$ then $\mathcal{R}$ outputs $\perp$ and aborts. Otherwise, it factors V_{i^*} and computes all roots. If for some root x^* we have $g^{x^*} = g^x$ then $\mathcal{R}$ returns x^*; otherwise $\mathcal{R}$ returns $\perp$.

Analysis of $\mathcal{R}$. Note that $\mathcal{R}$ simulates the T-OMDH game to $\mathcal{A}$ correctly, unless it outputs $\mathsf{HighDeg}$ in step 3. $\mathsf{HighDeg}$ occurs only if $i^* \leq t_0$ (which happens with probability $t_0/(Q+t_0+1)$). In this case, if $c \in C \setminus \{c_{i^*}\}$ then $\mathsf{Power}(c, b)$ can always be answered. The queries for the other $(n-t') - (t-t') = n-t$ choices of c return b exponentiated by a linear polynomial in x. Since $C_{t_0}(\vec{q}) \leq Q$, there are $\leq t - t'$ "bad" indices $i \in [n] \setminus F$ such that $q_i > Q$. Since elements of C are chosen uniformly at random from $[n] \setminus F$, with probability at least $1/\binom{n-t'}{t-t'}$ all "bad" indices are in $C \setminus \{c_{i^*}\}$. If this is the case, $\mathcal{R}$ must then answer $n-t$ other queries up to Q times: as the degree in x increases by at most 1 per query, the degree of x in any query is at most $Q(n-t)$, so $\mathsf{HighDeg}$ does not occur. In total, $\mathsf{HighDeg}$ occurs only if $i^* \leq t_0$ *and* $C \setminus \{c_{i^*}\}$ does not contain all "bad" indices, so

$$\Pr[\mathsf{HighDeg}] \leq \frac{t_0}{Q+t_0+1}\left(1 - \frac{1}{\binom{n-t'}{t-t'}}\right) \Rightarrow \Pr[\overline{\mathsf{HighDeg}}] \geq \frac{1}{Q+t_0+1}\left(Q + \frac{t_0}{\binom{n-t'}{t-t'}} + 1\right).$$

Next we assume that $\mathsf{HighDeg}$ does not occur. We have that

$$\Pr[\mathcal{A} \text{ succeeds} \mid \overline{\mathsf{HighDeg}}] = \epsilon.$$

If $\mathcal{A}$ succeeds, by Lem. 3.3 there is some $j^* \in [Q+t_0+1]$ such that the polynomial V_{j^*} corresponding to S_2 is nonzero. Since i^* is chosen uniformly and is independent of $\mathcal{A}$'s view, $i^* = j^*$ happens with probability $1/(Q+t_0+1)$. If $i^* = j^*$, then V_{i^*} is a polynomial in $R_{i^*-t_0}$ if $i^* > t_0$ and C'_{i^*} otherwise; in both cases, by $\mathcal{R}$'s simulation of the TOMDH game and Lem. 3.3, x is a root of V_{i^*}, so $\mathcal{R}$ outputs x and succeeds. Therefore,

$$\Pr[\mathcal{R} \text{ succeeds} \mid \overline{\mathsf{HighDeg}} \wedge \mathcal{A} \text{ succeeds}] \geq \frac{1}{Q+t_0+1}.$$

Putting these all together,

$$\begin{aligned} \Pr[\mathcal{R} \text{ succeeds}] &\geq \Pr[\mathcal{R} \text{ succeeds} \mid \overline{\mathsf{HighDeg}}] \Pr[\overline{\mathsf{HighDeg}}] \\ &\geq \frac{\Pr[\mathcal{R} \text{ succeeds} \mid \overline{\mathsf{HighDeg}}]}{Q+t_0+1}\left(Q + \frac{t_0}{\binom{n-t'}{t-t'}} + 1\right) \\ &= \frac{\epsilon \Pr[\mathcal{R} \text{ succeeds} \mid \overline{\mathsf{HighDeg}} \wedge \mathcal{A} \text{ succeeds}]}{Q+t_0+1}\left(Q + \frac{t_0}{\binom{n-t'}{t-t'}} + 1\right) \\ &\geq \frac{\epsilon}{(Q+t_0+1)^2}\left(Q + \frac{t_0}{\binom{n-t'}{t-t'}} + 1\right). \end{aligned}$$

We will not compute $\mathcal{R}$'s runtime in concrete terms, but it suffices to notice that it is $\mathcal{A}$'s runtime plus $O(\lambda)$. □

5.4 Proof of Lem. 5.5

Henceforth we assume the definitions in Sect. 5.2 of $t', t, n, \vec{q}, w, \vec{k}, \{\alpha_j^{(\ell)}\}$ etc. The following lemmas are proven in Appx. A.

Lemma 5.6. *Let* $Q = C_{t-t'+1}(\vec{q}) + 1$ *and*

$$\vec{m}_j = \vec{k}^j - \sum_{\ell=t-t'+1}^{t} \alpha_j^{(\ell)} \vec{k}^\ell \in \mathbb{F}_p^w.$$

Then for any w*-dimensional vectors* $\vec{b}_1, \ldots, \vec{b}_Q$ *the set*

$$V = \{\vec{m}_j \odot \vec{b}_i\}_{\substack{j \in \{0,\ldots,t-t'\} \\ i \in [Q]}}$$

is linearly dependent over $\mathbb{F}_p$.

Lemma 5.7. *Let* $1 \le j \le \lambda \le t - t'$. *If* $t - t' + 1 \le \ell' \le t - j$ *then*

$$\alpha_{\lambda-j}^{(\ell')} = \alpha_\lambda^{(\ell'+j)} + \sum_{\ell=t-t'+1}^{t-t'+j} \alpha_\lambda^{(\ell)} \alpha_{\ell-j}^{(\ell')},$$

and if $t - j + 1 \le \ell' \le t$ *then*

$$\alpha_{\lambda-j}^{(\ell')} = \sum_{\ell=t-t'+1}^{t-t'+j} \alpha_\lambda^{(\ell)} \alpha_{\ell-j}^{(\ell')}.$$

Lem. 5.6 is almost identical to [JKKX17, Lemma 2], while Lem. 5.7 is a new technical lemma we require to handle the $t' > 0$ case.

Proof of Lem. 5.5: Let $\vec{a}_1^\top, \ldots, \vec{a}_Q^\top$ be the rows of $\mathbf{A}$ and $\vec{b}_1, \ldots, \vec{b}_Q$ be the columns of $\mathbf{B}$ for $Q = C_{t-t'+1}(\vec{q}) + 1$. The conditions $\mathbf{A}\mathbf{M}_0\mathbf{B} = \mathbf{I}$ and $\mathbf{A}\mathbf{M}_j\mathbf{B} = \mathbf{0}$ for all $j \in [t - t']$ are equivalent to

$$\vec{a}_i^\top \vec{b} = \begin{cases} 1, & \text{if } \vec{b} = \vec{m}_0 \odot \vec{b}_i \\ 0, & \text{if } \vec{b} \in V \setminus \{\vec{m}_0 \odot \vec{b}_i\} \end{cases} \tag{10}$$

We shall use Equation (10) to show the following, from which Lem. 5.5 easily follows:

Claim: Let $j \in \{0, \ldots, t - t'\}, i \in [Q]$. Then

$$\vec{m}_j \odot \vec{b}_i \notin \mathrm{Span}(V_{j,i}),$$

where

$$V_{j,i} = \{\vec{m}_\lambda \odot \vec{b}_\gamma \mid j \le \lambda \le t - t', \gamma \in [Q], (\lambda, \gamma) \ne (j, i)\}.$$

Proof of Claim. Equation (10) immediately implies $\vec{m}_0 \odot \vec{b}_i \notin \mathrm{Span}(V \setminus \{\vec{m}_0 \odot \vec{b}_i\})$, the $j = 0$ case of the Claim. For $j > 0$, unwrapping Equation (10) further, the first case gives

$$1 = \vec{a}_i^\top (\vec{m}_0 \odot \vec{b}_i) = \vec{a}_i^\top \vec{b}_i - \sum_{\ell=t-t'+1}^{t} \alpha_0^{(\ell)} \vec{a}_i^\top (\vec{k}^\ell \odot \vec{b}_i). \tag{11}$$

For the second case, if $\vec{b} = \vec{m}_s \odot \vec{b}_u$ for $0 \le s \le t - t', u \in [Q]$ such that $(s,u) \neq (0,i)$, then

$$0 = \vec{a}_i^\top (\vec{m}_s \odot \vec{b}_u) = \vec{a}_i^\top \left(\vec{k}^s \odot \vec{b}_u - \sum_{\ell=t-t'+1}^{t} \alpha_s^{(\ell)} (\vec{k}^\ell \odot \vec{b}_u) \right) \Rightarrow$$

$$\vec{a}_i^\top (\vec{k}^s \odot \vec{b}_u) = \sum_{\ell=t-t'+1}^{t} \alpha_s^{(\ell)} \vec{a}_i^\top (\vec{k}^\ell \odot \vec{b}_u). \tag{12}$$

Suppose for the sake of contradiction there are coefficients $\delta_{\lambda,\gamma}$ such that

$$\vec{m}_j \odot \vec{b}_i = \sum_{\substack{j \le \lambda \le t-t', \gamma \in [Q] \\ (\lambda,\gamma) \neq (j,i)}} \delta_{\lambda,\gamma} (\vec{m}_\lambda \odot \vec{b}_\gamma) \Rightarrow$$

$$\vec{k}^j \odot \vec{b}_i - \sum_{\ell=t-t'+1}^{t} \alpha_j^{(\ell)} (\vec{k}^\ell \odot \vec{b}_i) = \sum_{\substack{j \le \lambda \le t-t', \gamma \in [Q] \\ (\lambda,\gamma) \neq (j,i)}} \delta_{\lambda,\gamma} \left(\vec{k}^\lambda \odot \vec{b}_\gamma - \sum_{\ell=t-t'+1}^{t} \alpha_\lambda^{(\ell)} (\vec{k}^\ell \odot \vec{b}_\gamma) \right).$$

Since $\vec{k}$ has no zero entries, we have

$$\vec{b}_i - \sum_{\ell=t-t'+1}^{t} \alpha_j^{(\ell)} (\vec{k}^{\ell-j} \odot \vec{b}_i) = \sum_{\substack{j \le \lambda \le t-t', \gamma \in [Q] \\ (\lambda,\gamma) \neq (j,i)}} \delta_{\lambda,\gamma} \left(\vec{k}^{\lambda-j} \odot \vec{b}_\gamma - \sum_{\ell=t-t'+1}^{t} \alpha_\lambda^{(\ell)} (\vec{k}^{\ell-j} \odot \vec{b}_\gamma) \right). \tag{13}$$

To derive a contradiction, we show the right-hand side of Equation (13) is orthogonal to $\vec{a}_i$, but the left-hand side is not.

The right-hand side of Equation (13) *is orthogonal to* $\vec{a}_i$*:* We need to show

$$0 = \vec{a}_i^\top \left(\vec{k}^{\lambda-j} \odot \vec{b}_\gamma - \sum_{\ell=t-t'+1}^{t} \alpha_\lambda^{(\ell)} (\vec{k}^{\ell-j} \odot \vec{b}_\gamma) \right) \tag{14}$$

$$= \vec{a}_i^\top (\vec{k}^{\lambda-j} \odot \vec{b}_\gamma) - \sum_{\ell=t-t'+1}^{t-t'+j} \alpha_\lambda^{(\ell)} \vec{a}_i^\top (\vec{k}^{\ell-j} \odot \vec{b}_\gamma) - \sum_{\ell=t-t'+j+1}^{t} \alpha_\lambda^{(\ell)} \vec{a}_i^\top (\vec{k}^{\ell-j} \odot \vec{b}_\gamma). \tag{15}$$

Applying Equation (12) to the summands in the first two terms of (15) yields

$$\sum_{\ell'=t-t'+1}^{t} \alpha_{\lambda-j}^{(\ell')} \vec{a}_i^\top (\vec{k}^{\ell'} \odot \vec{b}_\gamma) - \sum_{\ell=t-t'+1}^{t-t'+j} \alpha_\lambda^{(\ell)} \vec{a}_i^\top \left(\sum_{\ell'=t-t'+1}^{t} \alpha_{\ell-j}^{(\ell')} (\vec{k}^{\ell'} \odot \vec{b}_\gamma) \right) - \sum_{\ell=t-t'+j+1}^{t} \alpha_\lambda^{(\ell)} \vec{a}_i^\top (\vec{k}^{\ell-j} \odot \vec{b}_\gamma)$$

$$= \sum_{\ell'=t-t'+1}^{t} \vec{a}_i^\top \left(\alpha_{\lambda-j}^{(\ell')} - \sum_{\ell=t-t'+1}^{t-t'+j} \alpha_\lambda^{(\ell)} \alpha_{\ell-j}^{(\ell')} \right) (\vec{k}^{\ell'} \odot \vec{b}_\gamma) - \sum_{\ell=t-t'+j+1}^{t} \alpha_\lambda^{(\ell)} \vec{a}_i^\top (\vec{k}^{\ell-j} \odot \vec{b}_\gamma)$$

$$= \sum_{\ell'=t-t'+1}^{t} \left(\alpha_{\lambda-j}^{(\ell')} - \sum_{\ell=t-t'+1}^{t-t'+j} \alpha_\lambda^{(\ell)} \alpha_{\ell-j}^{(\ell')} \right) \vec{a}_i^\top (\vec{k}^{\ell'} \odot \vec{b}_\gamma) - \sum_{\ell'=t-t'+1}^{t-j} \alpha_\lambda^{(\ell'+j)} \vec{a}_i^\top (\vec{k}^{\ell'} \odot \vec{b}_\gamma).$$

By Lem. 5.7 the coefficients in this sum vanish, so the entire sum is zero and Equation (14) holds.

The left-hand side of Equation (13) is not orthogonal to $\vec{a}_i$: We will show

$$1 = \vec{a}_i^\top \vec{b}_i - \sum_{\ell=t-t'+1}^{t} \alpha_j^{(\ell)} \vec{a}_i^\top (\vec{k}^{\ell-j} \odot \vec{b}_i). \tag{16}$$

Again by Equation (12)

$$\sum_{\ell=t-t'+1}^{t} \alpha_j^{(\ell)} \vec{a}_i^\top (\vec{k}^{\ell-j} \odot \vec{b}_i) = \sum_{\ell=t-t'+1}^{t-t'+j} \alpha_j^{(\ell)} \vec{a}_i^\top \left(\sum_{\ell'=t-t'+1}^{t} \alpha_{\ell-j}^{(\ell')} (\vec{k}^{\ell'} \odot \vec{b}_i) \right) + \sum_{\ell=t-t'+j+1}^{t} \alpha_j^{(\ell)} \vec{a}_i^\top (\vec{k}^{\ell-j} \odot \vec{b}_i),$$

and by Equation (11)

$$\vec{a}_i^\top \vec{b}_i = 1 + \sum_{\ell'=t-t'+1}^{t} \alpha_0^{(\ell')} \vec{a}_i^\top (\vec{k}^{\ell'} \odot \vec{b}_i).$$

Together we have

$$\begin{aligned}
&\vec{a}_i^\top \vec{b}_i - \sum_{\ell=t-t'+1}^{t} \alpha_j^{(\ell)} \vec{a}_i^\top (\vec{k}^{\ell-j} \odot \vec{b}_i) \\
=&1 + \sum_{\ell'=t-t'+1}^{t} \alpha_0^{(\ell')} \vec{a}_i^\top (\vec{k}^{\ell'} \odot \vec{b}_i) - \sum_{\ell=t-t'+1}^{t-t'+j} \alpha_j^{(\ell)} \vec{a}_i^\top \left(\sum_{\ell'=t-t'+1}^{t} \alpha_{\ell-j}^{(\ell')} (\vec{k}^{\ell'} \odot \vec{b}_i) \right) - \sum_{\ell=t-t'+j+1}^{t} \alpha_j^{(\ell)} \vec{a}_i^\top (\vec{k}^{\ell-j} \odot \vec{b}_i) \\
=&1 + \sum_{\ell'=t-t'+1}^{t} \left(\alpha_0^{(\ell')} - \sum_{\ell=t-t'+1}^{t-t'+j} \alpha_j^{(\ell)} \alpha_{\ell-j}^{(\ell')} \right) \vec{a}_i^\top (\vec{k}^{\ell'} \odot \vec{b}_i) - \sum_{\ell'=t-t'+1}^{t-j} \alpha_j^{(\ell'+j)} \vec{a}_i^\top (\vec{k}^{\ell'} \odot \vec{b}_i).
\end{aligned}$$

By Lem. 5.7 (set $\lambda = j$) the coefficients in this sum vanish, so the entire expression equals 1 and Equation (16) holds.

We now return to Equation (13): multiplying both sides by $\vec{a}_i^\top$ and using Equations (14) and (16) we arrive at $1 = 0$, a contradiction. □

We now return to Lem. 5.5. By Lem. 5.6, V is linearly dependent, so we have $\delta_{i,j} \in \mathbb{F}_p$ not all zero such that

$$\sum_{j=0}^{t-t'} \sum_{i=1}^{Q} \delta_{i,j} (\vec{m}_j \odot \vec{b}_i) = 0.$$

If a coefficient $\delta_{i,0}$ is nonzero then $\vec{m}_0 \odot \vec{b}_i \in \mathrm{Span}(V \setminus \{\vec{m}_0 \odot \vec{b}_i\})$, contradicting the Claim. Therefore

$$\sum_{j=1}^{t-t'} \sum_{i=1}^{Q} \delta_{i,j} (\vec{m}_j \odot \vec{b}_i) = 0.$$

By the same reasoning the Claim also implies $\delta_{i,1} = 0, \delta_{i,2} = 0, \ldots, \delta_{i,t-t'} = 0$, and we arrive at a contradiction. □

6 Separation Results

In this section we assume all reductions $\mathcal{R}$ are PPT reductions in the AGM; that is, $\mathcal{R}$ and the adversary $\mathcal{A}$ run by $\mathcal{R}$ are both algebraic. Furthermore, we allow $\mathcal{R}$ to run its adversary $\mathcal{A}$ as many times as it wants, but we do not allow $\mathcal{R}$ to choose the random coins of $\mathcal{A}$ or rewind $\mathcal{A}$ (otherwise $\mathcal{A}$ could fail with overwhelming probability). These requirements are not repeated in the theorem statements below.

We will use the meta-reduction methodology: given a reduction $\mathcal{R}$ from problem P_1 to problem P_2, we construct a meta-reduction $\mathcal{M}$ that uses $\mathcal{R}$ to unconditionally solve P_1 by simulating $\mathcal{R}$'s access to an adversary $\mathcal{A}$ for P_2. As in [BFL20], to correctly and cleanly argue about the probability distributions, (a) if $\mathcal{R}$ incorrectly simulates the security game to $\mathcal{A}$, then we let $\mathcal{A}$ output $\perp$ and abort, and (b) $\mathcal{M}$ completely simulates $\mathcal{R}$'s access to $\mathcal{A}$ and copies $\mathcal{R}$'s final output, even if $\mathcal{M}$ obtains enough information to cease interacting with $\mathcal{R}$ and solve the problem directly.

6.1 Separation Result for OMDL

Theorem 6.1. *For any $Q \geq 0$, suppose there is a reduction $\mathcal{R}$ from Q-OMDL to $(Q+1)$-OMDL. Then Q-OMDL is easy.*

Concretely, suppose $\mathcal{R}$ (T', ϵ')-solves Q-OMDL given access to an adversary $\mathcal{A}$ that (T, ϵ)-solves $(Q+1)$-OMDL. Then there is an (algebraic) meta-reduction $\mathcal{M}$ such that $\mathcal{M}^{\mathcal{R}}$ (T', ϵ')-solves Q-OMDL, as long as $T \geq (Q+1)[2(Q+3)\lambda + Q + 1]$ and $\epsilon \leq 1 - p^{-1} - O(p^{-2})$.

Proof. Given such a reduction $\mathcal{R}$, we define a meta-reduction $\mathcal{M}$ that uses $\mathcal{R}$ to solve Q-OMDL. While running $\mathcal{R}$, $\mathcal{M}$ needs to play the role of $\mathcal{R}$'s Q-OMDL challenger, as well as the $(Q+1)$-OMDL solver that $\mathcal{R}$ uses. $\mathcal{M}$ works as follows:

Meta-reduction $\mathcal{M}$:

1. On Q-OMDL challenge $(A_{-1}, A_0, \ldots, A_Q)$, $\mathcal{M}$ feeds the challenge to $\mathcal{R}$.
2. $\mathcal{M}$, as $\mathcal{R}$'s challenger, must answer (up to) Q discrete log oracle queries by $\mathcal{R}$. This can be easily simulated since $\mathcal{M}$ itself is part of the Q-OMDL game, so it can just forward the oracle queries of $\mathcal{R}$ to its own oracle, along with the algebraic representations of the query elements.
3. To simulate a run of the $(Q+1)$-OMDL solver,
   (a) Suppose that $\mathcal{R}$ runs the solver on challenge $(B_{-1}, B_0, \ldots, B_{Q+1})$[7]; since $\mathcal{R}$ is algebraic, it must also provide the algebraic representations $B_j = \prod_{i=-1}^{Q} A_i^{z_{i,j}}$ with each $z_{i,j} \in \mathbb{F}_p$ for $i = -1, \ldots, Q$ and $j = -1, \ldots, Q+1$. If $\mathcal{R}$ simulates the generator $B_{-1} = 1_G$, $\mathcal{M}$ outputs $\perp$ and aborts.
   (b) $\mathcal{M}$ can make $Q+1$ queries to DL (the $(Q+1)$-OMDL solver's discrete log oracle simulated by $\mathcal{R}$). $\mathcal{M}$ will choose $u_{j,i} \leftarrow \mathbb{F}_p$ for $i = 0, \ldots, Q$ and $j = 0, \ldots, Q+1$, and query $m_i = \mathsf{DL}(\prod_{j=0}^{Q+1} B_j^{u_{j,i}})$, providing algebraic representations $(0, u_{0,i}, \ldots, u_{Q+1,i})$. $\mathcal{M}$ checks if $\mathcal{R}$ simulates DL correctly, i.e., if $B_{-1}^{m_i} = \prod_{j=0}^{Q+1} B_j^{u_{j,i}}$ for all i; if the equality does not hold for any i, then $\mathcal{M}$ outputs $\perp$ and aborts.
   (c) Finally, $\mathcal{M}$ returns $\mathrm{dlog}_{B_{-1}}(B_0), \ldots, \mathrm{dlog}_{B_{-1}}(B_{Q+1})$ to $\mathcal{R}$ with some overwhelming probability (how this is achieved is described later).

[7] Note that in our definition of OMDL the group generator g is sampled by the game challenger, rather than fixed in advance; this allows $\mathcal{R}$ to set B_{-1} to be different from A_{-1}.

4. When $\mathcal{R}$ is finished, it is supposed to output $\mathrm{dlog}_{A_{-1}}(A_0), \ldots, \mathrm{dlog}_{A_{-1}}(A_Q)$, and $\mathcal{M}$ copies $\mathcal{R}$'s output.

Let $\mathbf{Z}$ be the $(Q+3) \times (Q+2)$ matrix

$$\mathbf{Z} = \begin{bmatrix} z_{-1,-1} & \cdots & z_{Q,-1} \\ \vdots & \ddots & \vdots \\ z_{-1,Q+1} & \cdots & z_{Q,Q+1} \end{bmatrix},$$

which is defined by $\mathcal{R}$ and known to $\mathcal{M}$ in step 3a. If $\mathcal{M}$ does not abort in step 3a, $B_{-1} \neq 1_G$ so the first row of $\mathbf{Z}$ is nonzero.

Lemma 6.2. *Let $B_{-1}^{e_i} = A_i, B_{-1}^{d_j} = B_j$ for $i = -1, \ldots, Q$ and $j = 0, \ldots, Q+1$. (So $\mathcal{M}$ needs to compute $(d_0, \ldots, d_{Q+1})$.) Then*

$$\mathbf{Z}\vec{e} = \mathbf{Z} \begin{bmatrix} e_{-1} \\ e_0 \\ \vdots \\ e_Q \end{bmatrix} = \begin{bmatrix} 1 \\ d_0 \\ \vdots \\ d_{Q+1} \end{bmatrix} = \vec{d}.$$

Proof. From step 3a we have $B_j = \prod_{i=-1}^{Q} A_i^{z_{i,j}}$, so

$$B_{-1}^{d_j} = B_j = \prod_{i=-1}^{Q} A_i^{z_{i,j}} = \prod_{i=-1}^{Q} B_{-1}^{e_i z_{i,j}} = B_{-1}^{e_{-1}z_{-1,j} + \cdots + e_Q z_{Q,j}}.$$

As j ranges from -1 to $Q+1$ (with $d_{-1} = 1$) we obtain the lemma. □

Next, let $\mathbf{U}$ be the $(Q+2) \times (Q+3)$ matrix

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & u_{0,0} & \cdots & u_{Q+1,0} \\ 0 & \vdots & \ddots & \vdots \\ 0 & u_{0,Q} & \cdots & u_{Q+1,Q} \end{bmatrix},$$

which is defined by $\mathcal{M}$ in step 3b.

Lemma 6.3. *Suppose $\mathcal{M}$ does not abort in step 3b. Then*

$$\mathbf{U}\mathbf{Z}\vec{e} = \mathbf{U}\vec{d} = \begin{bmatrix} 1 \\ m_0 \\ \vdots \\ m_Q \end{bmatrix} = \vec{m}.$$

Proof. By Lem. 6.2 $\mathbf{U}\mathbf{Z}\vec{e} = \mathbf{U}\vec{d}$; we now prove that $\mathbf{U}\vec{d} = \vec{m}$. From step 3b we have $B_{-1}^{m_i} = \prod_{j=0}^{Q+1} B_j^{u_{j,i}}$, so

$$B_{-1}^{m_i} = \prod_{j=0}^{Q+1} B_j^{u_{j,i}} = \prod_{j=0}^{Q+1} B_{-1}^{d_j u_{j,i}} = B_{-1}^{d_0 u_{0,i} + \cdots + d_{Q+1} u_{Q+1,i}}.$$

As i ranges from -1 to Q (with $m_{-1} = 1$) we obtain the lemma. □

Assuming $\mathcal{M}$ does not abort in step 3b, we now describe how $\mathcal{M}$ computes $(d_0, \ldots, d_{Q+1})$ in step 3c. If $\ker(\mathbf{Z}) \neq \ker(\mathbf{UZ})$ then $\mathcal{M}$ outputs $\perp$ and aborts. (Since $\ker(\mathbf{Z}) \subseteq \ker(\mathbf{UZ})$, $\mathcal{M}$ can check if $\ker(\mathbf{Z}) \neq \ker(\mathbf{UZ})$ by verifying that $\dim(\ker(\mathbf{Z})) \neq \dim(\ker(\mathbf{UZ}))$ via row reduction.) Otherwise $\mathcal{M}$ computes some $\vec{v}$ such that $\mathbf{UZ}\vec{v} = \vec{m}$ (since $\vec{e}$ is one such $\vec{v}$ by Lem. 6.3, such a $\vec{v}$ always exists and can be computed via row reduction), and then returns $(d_0, \ldots, d_{Q+1})$ as the last $Q+2$ entries of $\mathbf{Z}\vec{v}$.

Analysis of $\mathcal{M}$. To begin with, in $\mathcal{R}$'s view $\mathcal{M}$'s behavior in step 3 defines a $(Q+1)$-OMDL solver $\mathcal{A}$ *in the real* $(Q+1)$*-OMDL game*; in particular, the DL queries in step 3b are uniformly random and thus independent of $\mathcal{M}$'s view while simulating $\mathcal{R}$'s Q-OMDL challenger (e.g., independent of the matrix $\mathbf{Z}$). Below we show that $\mathcal{A}$ has advantage $1 - p^{-1} - O(p^{-2})$ *in the real* $(Q+1)$*-OMDL game*; this is the more difficult part of the overall analysis of $\mathcal{M}$. After that, it is clear that if the reduction $\mathcal{R}$ "works" for $\mathcal{A}$, then $\mathcal{M}$ solves Q-OMDL with runtime and probability equal to $\mathcal{R}$'s, since $\mathcal{M}$ merely passes inputs and outputs (including oracle queries) between $\mathcal{M}$'s own challenger and $\mathcal{R}$.

Analysis of $\mathcal{A}$. First suppose $\mathcal{R}$ correctly simulates the $(Q+1)$-OMDL game to $\mathcal{A}$. Then $\mathcal{A}$ does not abort in steps 3a,3b, so $\mathcal{A}$ aborts if and only if $\ker(\mathbf{Z}) \neq \ker(\mathbf{UZ})$, in step 3c. Suppose $\mathcal{A}$ doesn't abort. The set of $\vec{v}$ such that $\mathbf{UZ}\vec{v} = \vec{m}$ is $\vec{e} + \ker(\mathbf{UZ})$ by Lem. 6.3. By assumption $\ker(\mathbf{Z}) = \ker(\mathbf{UZ})$ so $\vec{v} = \vec{e} + \vec{k}$ for some $\vec{k} \in \ker(\mathbf{Z})$. Therefore $\mathbf{Z}\vec{v} = \mathbf{Z}\vec{e} + \mathbf{Z}\vec{k} = \mathbf{Z}\vec{e}$, so by Lem. 6.2 the last $Q+2$ entries (i.e., excluding $d_{-1} = 1$) are the correct values of d_j, and $\mathcal{A}$ succeeds.

Next, we show that $\mathcal{A}$ aborts with negligible probability. Since $\mathbf{U}$ and $\mathbf{Z}$ are chosen independently, we bound the probability of $\ker(\mathbf{Z}) \neq \ker(\mathbf{UZ})$ where $\mathbf{Z}$ is a fixed matrix with a nonzero first row. Note that $\ker(\mathbf{Z}) \neq \ker(\mathbf{UZ})$ if and only if $\mathrm{im}(\mathbf{Z}) \cap \ker(\mathbf{U}) \neq \{0\}$. Since $\mathrm{im}(\mathbf{Z})$ has dimension at most $Q+2$, and the probability of $\mathrm{im}(\mathbf{Z}) \cap \ker(\mathbf{U}) \neq \{0\}$ is maximized when $\mathrm{im}(\mathbf{Z})$ is as large as possible, it suffices to consider the case $\mathrm{im}(\mathbf{Z}) = H(\vec{h})$ for some $\vec{h} \neq \vec{0}$. Let $\vec{u}_i$ be the i-th row of $\mathbf{U}$, $1 \leq i \leq Q+2$. Since $\ker(\mathbf{U}) = H(\vec{u}_1, \ldots, \vec{u}_{Q+2})$ we have $\mathrm{im}(\mathbf{Z}) \cap \ker(\mathbf{U}) = H(\vec{h}, \vec{u}_1, \ldots, \vec{u}_{Q+2})$.

To lighten notation, define

$$S_1 = \{\vec{h}\},\ S_i = \{\vec{h}, \vec{u}_1, \ldots, \vec{u}_{i-1}\} \text{ if } i > 1.$$

Applying Lem. 3.5 to $H(\vec{h}, \vec{u}_1, \ldots, \vec{u}_{Q+2})$ gives

$$\Pr[\mathrm{im}(\mathbf{Z}) \cap \ker(\mathbf{U}) = \{0\}] = \Pr[\vec{h} \neq 0] \prod_{i=1}^{Q+2} \Pr[\vec{u}_i \notin \mathrm{Span}(S_i) \mid \dim(S_i) = i]. \tag{17}$$

Note that $\vec{h} \neq 0$ by assumption, and $\vec{u}_1, \vec{h}$ are independent: if $\vec{u}_1 = \lambda\vec{h}, \lambda \neq 0$ then

$$\mathrm{im}(\mathbf{Z}) = H(\vec{h}) = \{\vec{v} \in \mathbb{F}_p^{Q+3} \mid (\vec{v})_1 = 0\},$$

which contradicts the assumption $\mathbf{Z}$ has a nonzero first row. Therefore

$$\Pr[\vec{u}_1 \notin \mathrm{Span}(S_1) \mid \dim(S_1) = 1] = 1$$

so the $i = 1$ term of Equation (17) is 1. For $i > 1$, if $\vec{u}_i \in \mathrm{Span}(S_i)$ then there are coefficients $\{\alpha_j\}_{j=1}^{i-1}$ and β such that $\vec{u}_i = \sum_{j=1}^{i-1} \alpha_j \vec{u}_j + \beta\vec{h}$. Take the first entry of the vectors in the equation; since

$$(\vec{u}_j)_1 = \begin{cases} 0 & j \neq 1 \\ 1 & j = 1 \end{cases},$$

we have

$$0 = \alpha_1 + \beta(\vec{h})_1. \tag{18}$$

There are p^i choices of $\{\alpha_j\}_{j=1}^{i-1}$ and β, but by Equation (18) there are only p^{i-1} possible choices of $\vec{u}_i$ contained in $\mathrm{Span}(S_i)$. Since $(\vec{u_i})_1 = 0$ and $(\vec{u_i})_2, \ldots, (\vec{u_i})_{Q+3} \in \mathbb{F}_p$, there are p^{Q+2} total choices for $\vec{u}_i$, so

$$\Pr[\vec{u}_i \notin \mathrm{Span}(S_i) \mid \dim(S_i) = i] = 1 - \frac{p^{i-1}}{p^{Q+2}} = 1 - p^{i-Q-3}.$$

Putting it all together,

$$\Pr[\mathrm{im}(\mathbf{Z}) \cap \ker(\mathbf{U}) = \{0\}] = \prod_{i=2}^{Q+2}(1 - p^{i-Q-3}) = \prod_{i=1}^{Q+1}(1 - p^{-i}) = 1 - p^{-1} - O(p^{-2}).$$

Overall, we have shown that $\mathcal{A}$'s success probability is at least $1 - p^{-1} + O(p^{-2})$. If $\mathcal{R}$ incorrectly simulates the $(Q+1)$-OMDL game to $\mathcal{A}$, either $B_{-1} = 1_G$ or the discrete log oracle DL is not implemented correctly. Then $\mathcal{A}$ aborts in step 3a or step 3b respectively.

Regarding $\mathcal{A}$'s runtime, in step 3b $\mathcal{A}$ makes $Q+1$ queries to DL, each of which involves $Q+2$ exponentiations and $Q+1$ multiplications; after that, $\mathcal{A}$ checks consistency of the answers, which involves $Q+1$ exponentiations. So the total number of group operations is up to $(Q+1)[2(Q+2)\lambda + Q + 1] + (Q+1) \cdot 2\lambda = (Q+1)[2(Q+3)\lambda + Q + 1]$. Step 3c does not involve any group operations and is thus "free". □

6.2 Separation Result for OMDL and Q-DL

Theorem 6.4. *For any $Q \geq 0$, suppose there is a reduction $\mathcal{R}$ from Q-OMDL to 1-DL. Then Q-OMDL is easy.*

Concretely, suppose $\mathcal{R}$ (T', ϵ')-solves Q-OMDL given access to an adversary $\mathcal{A}$ that (T, ϵ)-solves 1-DL. Then there is an (algebraic) meta-reduction $\mathcal{M}$ such that $\mathcal{M}^{\mathcal{R}}$ (T', ϵ')-solves Q-OMDL, as long as $T \geq (2Q^2 + 6Q + 4)\lambda + (Q+1)^2$ and $\epsilon \leq 1 - 4p^{-1} - O(p^{-2})$.

Proof. Given such a reduction $\mathcal{R}$, we define a meta-reduction $\mathcal{M}$ that uses $\mathcal{R}$ to solve Q-OMDL. While running $\mathcal{R}$, $\mathcal{M}$ needs to play the role of $\mathcal{R}$'s Q-OMDL challenger, as well as the 1-DL solver that $\mathcal{R}$ uses. $\mathcal{M}$ works as follows:

Meta-reduction $\mathcal{M}$:

1. On Q-OMDL challenge $(A_{-1}, A_0, \ldots, A_Q)$, $\mathcal{M}$ feeds the challenge to $\mathcal{R}$. Let $A_i = A_{-1}^{x_i}$ for $i = 0, \ldots, Q$ and $\vec{x} = (x_0, \ldots, x_Q)$.

2. $\mathcal{M}$, as $\mathcal{R}$'s challenger, must answer (up to) Q discrete log oracle queries by $\mathcal{R}$. Say Q' of them are made before $\mathcal{R}$ runs the 1-DL solver, and let the queries be on $V_1, \ldots, V_{Q'}$ with algebraic representations $V_j = \prod_{i=-1}^{Q} A_i^{v_{i,j}}$ for $j = 1, \ldots, Q'$. Define polynomials $L_j(X_0, \ldots, X_Q) = \sum_{i=-1}^{Q} v_{i,j} X_i$ (denote $X_{-1} = 1$), so $V_j = A_{-1}^{L_j(\vec{x})}$. $\mathcal{M}$ stores L_j in a sequence $\mathcal{L}$, but excludes those that can be linearly expressed by existing polynomials in $\mathcal{L}$; together with $r_j = L_j(\vec{x})$ in a separate sequence $\mathcal{L}'$. Concretely, $\mathcal{M}$ initializes $\mathcal{L} = \{L_0^* = 1\}$ and $\mathcal{L}' = \{r_0^* = 1\}$. When $\mathcal{R}$ makes a discrete log oracle query on V_j,

- If $L_j \in \mathrm{Span}(\mathcal{L})$, i.e., $L_j = \sum_{j'=0}^{\ell-1} \alpha_{j'} L^*_{j'}$ for some $\alpha_{j'} \in \mathbb{F}_p$ (where $\ell = |\mathcal{L}|$), then $\mathcal{M}$ can compute $r_j = \mathrm{dlog}_{A_{-1}}(V_j) = \sum_{j'=0}^{\ell-1} \alpha_{j'} r^*_{j'}$ by itself.
- Otherwise $\mathcal{M}$ queries its own discrete log oracle on V_j, receiving r_j as the result. $\mathcal{M}$ sets $L^*_\ell := L_j, r^*_\ell := r_j$ and adds them to $\mathcal{L}, \mathcal{L}'$, respectively.

Either way, $\mathcal{M}$ returns r_j to $\mathcal{R}$.

3. To simulate a run of the 1-DL solver,

   (a) Suppose that $\mathcal{R}$ runs the solver on challenge (B_{-1}, B_0, B_1); since $\mathcal{R}$ is algebraic, it must also provide the algebraic representations $B_j = \prod_{i=-1}^{Q} A_i^{z_{i,j}}$ for $j = -1, 0, 1$. If $\mathcal{R}$ simulates the group generator $B_{-1} = 1_G$, $\mathcal{M}$ outputs $\perp$ and aborts. If $B_0 = B_1 = 1_G$, $\mathcal{M}$ outputs 0. Otherwise define $P_j(X_0, \ldots, X_Q) = \sum_{i=-1}^{Q} z_{i,j} X_i$, so $B_j = A_{-1}^{P_j(\vec{x})}$; and $S = P_1 P_{-1} - P_0^2$.
   Let $\mathcal{I}_n$ be the ideal generated by $\{L^*_k(\vec{X}) - r^*_k\}_{k=1}^{n-1}$ for some n. Note that the generators of $\mathcal{I}_n$ are linearly independent as $\mathcal{L}$ is linearly independent (guaranteed in step 2), and that $\vec{x} \in V(\mathcal{I}_n)$ for any n. Define $[R]_n$ to be the canonical representative (see Sect. 3.1) of R in $\mathbb{F}_p[X_0, \ldots, X_Q]/\mathcal{I}_n$.

   (b) If $[S]_\ell = 0$ we have $[P_1]_\ell = ([P_0]_\ell/[P_{-1}]_\ell)^2 [P_{-1}]_\ell$; by Lem. 3.4 this implies $[P_0]_\ell/[P_{-1}]_\ell = c$ for some $c \in \mathbb{F}_p$ ($[P_{-1}]_\ell \neq 0$ since $B_{-1} \neq 1_G$). $\mathcal{M}$ returns c to $\mathcal{R}$.

   The case when $[S]_\ell \neq 0$ is handled as follows.

   (c) If $\ell \leq Q$, $\mathcal{M}$ samples $\{v^*_{i,j}\} \leftarrow \mathbb{F}_p$ for $i = -1, \ldots, Q, j = \ell, \ldots, Q$ and queries its discrete log oracle DL on $V^*_j = \prod_{i=-1}^{Q} A_i^{v^*_{i,j}}$. ($\mathcal{M}$ queried DL $\ell - 1$ times in step 2, so its total number of queries is Q.) Define $L^*_j(X_0, \ldots, X_Q) = \sum_{i=-1}^{Q} v^*_{i,j} X_i$ so $V^*_j = A_{-1}^{L^*_j(\vec{x})}$; $\mathcal{M}$ adds L^*_j to $\mathcal{L}$. If $\mathcal{L}$ is linearly dependent or $[S]_{Q+1} = 0$, $\mathcal{M}$ outputs $\perp$ and aborts.

   (d) Since $\mathcal{I}_{Q+1}$ is generated by Q independent elements, there is a nonpivotal variable X_t such that $[S]_{Q+1}$ is a polynomial only in X_t. For each root x'_t of $[S]_{Q+1}$, $\mathcal{M}$ substitutes $X_t = x'_t$ into the equations defining $\mathcal{I}_{Q+1}$, recovering the full solution $\vec{x}'$ via row reduction. $\mathcal{M}$ then computes $u' = P_0(\vec{x}')/P_{-1}(\vec{x}')$ and checks whether $B_{-1}^{u'} = B_0, B_{-1}^{(u')^2} = B_1$. If this holds for some u', $\mathcal{M}$ returns u' to $\mathcal{R}$. Otherwise $\mathcal{M}$ outputs $\perp$ and aborts.

4. After the 1-DL solver is finished, $\mathcal{R}$ might make some additional (up to $Q - Q'$) discrete log oracle queries and further invocations of the 1-DL solver.

   (a) If $\mathcal{M}$ returned in step 3b, $\mathcal{M}$ can continue to forward $\mathcal{R}$'s discrete log oracle queries to its own DL oracle, since $\mathcal{M}$ has only made DL queries also made by $\mathcal{R}$. $\mathcal{M}$ simulates subsequent invocations of the 1-DL solver as in step 3.

   (b) If $\mathcal{M}$ returned in step 3d, $\mathcal{M}$ has "used up" all of its DL queries, so it cannot make any more. However, in this case $\mathcal{M}$ has recovered $\vec{x}$, so it can now answer $\mathcal{R}$'s discrete log oracle query on V_j by returning $L_j(\vec{x})$, and can simulate any invocation of the 1-DL solver by returning $u = P_0(\vec{x})/P_{-1}(\vec{x})$.

5. When $\mathcal{R}$ is finished, it is supposed to output $\mathrm{dlog}_{A_{-1}}(A_0), \ldots, \mathrm{dlog}_{A_{-1}}(A_Q)$, and $\mathcal{M}$ copies $\mathcal{R}$'s output.

Analysis of $\mathcal{M}$. As in the proof of Thm. 6.1, in $\mathcal{R}$'s view $\mathcal{M}$'s behavior in step 3 defines a 1-DL solver $\mathcal{A}$ *in the real* 1*-DL game*. Below we show that $\mathcal{A}$ has advantage $1 - 4p^{-1} - O(p^{-2})$ *in the real* 1*-DL game*. After that, it is clear that if the reduction $\mathcal{R}$ "works" for $\mathcal{A}$, then $\mathcal{M}$ solves Q-OMDL with runtime and probability equal to $\mathcal{R}$'s, since $\mathcal{M}$ merely passes inputs and outputs between $\mathcal{M}$'s own challenger and $\mathcal{R}$, and can answer $\mathcal{R}$'s oracle queries correctly in steps 2 and 4. (Note that steps 2 and 4 do not involve any group operations and are thus "free".)

Analysis of $\mathcal{A}$. First suppose that $\mathcal{R}$ simulates the 1-DL game to $\mathcal{A}$ correctly, i.e., $\mathcal{R}$ submits a valid 1-DL instance $(B_{-1} \neq 1_G, B_0 = B_{-1}^u, B_1 = B_0^u)$. Since $B_{-1} \neq 1_G$, $\mathcal{A}$ does not output $\perp$ in step 3a. If $u = 0$, then $B_0 = B_1 = 1_G$ so $\mathcal{A}$ outputs 0 in step 3a and succeeds in the 1-DL game; below we assume $u \neq 0$. Since $B_j = A_{-1}^{P_j(\vec{x})}$ (see step 3a) we have $u = P_0(\vec{x})/P_{-1}(\vec{x}) = P_1(\vec{x})/P_0(\vec{x})$, so $S(\vec{x}) = 0$ ($P_{-1}(\vec{x}), P_0(\vec{x}) \neq 0$ since $B_{-1} \neq 1_G$ and $u \neq 0$).

If $\mathcal{A}$ enters step 3b, i.e., $[S]_\ell = 0$, then $u = P_0(\vec{x})/P_{-1}(\vec{x}) = [P_0]_\ell(\vec{x})/[P_{-1}]_\ell(\vec{x}) = c$, so $\mathcal{A}$ succeeds.

If $\mathcal{A}$ enters step 3c, it may abort if $\mathcal{L}$ is linearly dependent or $[S]_{Q+1} = 0$; let these events be D, E respectively. To show $\Pr[D \vee E]$ is negligible, note

$$\Pr[D \vee E] = \Pr[D] + \Pr[E \wedge \bar{D}] = \Pr[D] + \Pr[E \mid \bar{D}] \Pr[\bar{D}] \leq \Pr[D] + \Pr[E \mid \bar{D}]$$

so we upper-bound $\Pr[D]$ and $\Pr[E \mid \bar{D}]$. Define $S_j = \{L_0^*, \ldots, L_j^*\}$; $\Pr[D]$ is then given by applying Lem. 3.5 to $\mathcal{L}$:

$$\Pr[\dim(S_Q) = Q + 1] = \Pr[L_0^* \neq 0] \prod_{j=0}^{Q-1} \Pr[L_{j+1}^* \notin \mathrm{Span}(S_j) \mid \dim(S_j) = j + 1].$$

The first item is 1 (since $L_0^* = 1 \neq 0$), and by $\mathcal{M}$'s simulation of the discrete log oracle, $\dim(S_{\ell-1}) = \ell$, so if $\ell = Q + 1$ the product is 1. Otherwise we are left with

$$\prod_{j=\ell-1}^{Q-1} \Pr[L_{j+1}^* \notin \mathrm{Span}(S_j) \mid \dim(S_j) = j + 1].$$

For the j-th term, if $L_{j+1}^* \in \mathrm{Span}(S_j)$ there are coefficients $\{\alpha_{j'}\}_{j'=0}^{j}$ such that $L_{j+1}^* = \sum_{j'=0}^{i} \alpha_{j'} L_{j'}^*$. Therefore there are p^{j+1} choices of $\{\alpha_{j'}\}_{j'=0}^{j}$ for $L_{j+1}^* \in \mathrm{Span}(S_j)$ out of p^{Q+2} total choices. By linear independence there are p^{j+1} choices of $L_{j+1}^* \in \mathrm{Span}(S_j)$, so the j-th term is $1 - p^{j+1}/p^{Q+2} = 1 - p^{j-Q-1}$; then

$$\prod_{j=\ell-1}^{Q-1} \Pr[L_{j+1}^* \notin \mathrm{Span}(S_j) \mid \dim(S_j) = j + 1] = \prod_{j=\ell-1}^{Q-1} (1 - p^{j-Q-1}) = \prod_{j=2}^{Q-\ell+2} (1 - p^{-j}) = 1 - p^{-2} - O(p^{-3}).$$

Next, we show $\Pr[E \mid \bar{D}]$ is negligible; that is, assuming $\mathcal{L}$ is linearly independent, the probability that $[S]_{Q+1} = 0$ is negligible. This probability is maximized when ℓ is the smallest, so it suffices to consider the case $\ell = 1$. For $t = 1, \ldots, Q$, let E_t be the event that $[S]_t \neq 0$ but $[S]_{t+1} = 0$; by assumption $[S]_1 = S \neq 0$ so $E = \vee_{t=1}^{Q} E_t$. We now bound $\Pr[E_t \mid \bar{D} \wedge \wedge_{i=1}^{t-1} \bar{E}_i]$ for each t.

Since $\wedge_{i=1}^{t-1} \bar{E}_i$ occurs we have $[S]_t \neq 0$. Let $U_t(\vec{X}) = L_t^*(\vec{X}) - r_t^*$; if E_t also occurs then $[S]_t \equiv 0 \pmod{\langle [U_t]_t \rangle}$, so $[U_t]_t$ divides $[S]_t$.

Lemma 6.5. *For each choice of* U_t*, there are* $< 2p^{t-1}$ *choices of* U_t' *such that* $U_t \equiv U_t' \pmod{\mathcal{I}_t}$.

Proof. In the trivial case when $t = 1$ and $\mathcal{I}_1 = \{0\}$ we must have $U_t' = U_t$, and $1 < 2p^{1-1} = 2$. We now assume $t > 1$ so $\{L_i^*(\vec{X}) - r_i^* = 0\}_{i=1}^{t-1}$ is nonempty: these equations can be written as

$$\mathbf{M}\vec{X} = \vec{c} \text{ for some } \mathbf{M} \in \mathbb{F}_p^{(t-1)\times(Q+1)}, \vec{c} \in \mathbb{F}_p^{Q+1}.$$

Write $U_t(\vec{X}) = \vec{a} \cdot \vec{X} + a, U_t'(\vec{X}) = \vec{b} \cdot \vec{X} + b$ for some $\vec{a}, \vec{b} \in \mathbb{F}_p^{Q+1}; a, b \in \mathbb{F}_p$ and define

$$\vec{x}_0 = \begin{cases} \vec{0}, & \text{if } \vec{c} = \vec{0} \\ \vec{x}, & \text{otherwise.} \end{cases}$$

In either case $\mathbf{M}\vec{x}_0 = \vec{c}$, so

$$V(\mathcal{I}_t) = \{\vec{y} \in \mathbb{F}_p^{Q+1} \mid \mathbf{M}\vec{y} = \vec{c}\} = \vec{x}_0 + \ker(\mathbf{M}).$$

Since $U_t \equiv U_t' \pmod{\mathcal{I}_t}$ we have $U_t(\vec{x}_0 + \vec{k}) = U_t'(\vec{x}_0 + \vec{k})$ for any $\vec{k} \in \ker(\mathbf{M})$. Plugging this in and rearranging gives

$$(\vec{a} - \vec{b}) \cdot (\vec{x}_0 + \vec{k}) = b - a.$$

Since $\mathcal{L}$ is linearly independent, $\{L_i^*(\vec{X}) - r_i^*\}_{i=1}^{t-1}$ is as well, so $\dim(\ker(\mathbf{M})) = Q + 2 - t$. We then choose $\vec{k}_1, \ldots, \vec{k}_{Q-t+2}$ to be a basis of $\ker(\mathbf{M})$, obtaining $Q - t + 3$ constraints on $\vec{a} - \vec{b}$ as $\vec{k}$ ranges over $\vec{0}, \vec{k}_1, \ldots, \vec{k}_{Q-t+2}$. To examine the linear independence of these constraints, suppose there is $\alpha_0, \alpha_1, \ldots, \alpha_{Q-t+2}$ such that

$$\alpha_0\vec{x}_0 + \sum_{i=1}^{Q-t+2} \alpha_i(\vec{x}_0 + \vec{k}_i) = 0,$$
$$\sum_{i=0}^{Q-t+2} \alpha_i(b - a) = 0.$$

By the second equation, either $\sum_{i=0}^{Q-t+2} \alpha_i = 0$ or $b = a$. If $b \neq a$, we have

$$\left(\sum_{i=0}^{Q-t+2} \alpha_i\right)\vec{x}_0 + \sum_{i=1}^{Q-t+2} \alpha_i\vec{k}_i = 0 \overset{(a)}{\Rightarrow} \sum_{i=1}^{Q-t+2} \alpha_i\vec{k}_i = 0 \overset{(b)}{\Rightarrow} \alpha_1 = \cdots = \alpha_{Q-t+2} = 0 \overset{(a)}{\Rightarrow} \alpha_0 = 0,$$

where (a) follows from $\sum_{i=0}^{Q+2-t} \alpha_i = 0$ and (b) follows from independence of $\{\vec{k}_i\}$, so there are $Q - t + 3$ independent constraints. If $b = a$ and $\sum_{i=0}^{Q-t+2} \alpha_i = 0$ the same logic holds, but if $b = a$ and $\sum_{i=0}^{Q-t+2} \alpha_i \neq 0$,

$$\left(\sum_{i=0}^{Q-t+2} \alpha_i\right)\vec{x}_0 = -\sum_{i=1}^{Q-t+2} \alpha_i\vec{k}_i \Rightarrow \vec{x}_0 = -\left(\sum_{i=0}^{Q-t+2} \alpha_i\right)^{-1}\left(\sum_{i=1}^{Q-t+2} \alpha_i\vec{k}_i\right),$$

which implies $\vec{x}_0 \in \ker(\mathbf{M})$, so $\vec{x}_0 = \vec{0}$ by the definition of $\vec{x}_0$. In this case there are $Q - t + 2$ linearly independent constraints by the independence of $\{\vec{k}_i\}$.

Consequently when $b \neq a$, $\vec{a} - \vec{b}$ is contained in a $(Q + 1) - (Q - t + 3) = (t - 2)$-dimensional affine subspace of $\mathbb{F}_p^{Q+1}$, and thus there are $\leq p^{t-2}$ possible choices for $\vec{a} - \vec{b}$; otherwise the subspace is $\leq (t - 1)$-dimensional, and there are $\leq p^{t-1}$ choices of $\vec{a} - \vec{b}$. All together, there are at most

$$(p - 1)p^{t-2} + p^{t-1} = p^{t-2}(2p - 1) < 2p^{t-1}$$

choices of U_t' in total. □

Since $[S]_t \neq 0$ and has degree at most 2, there are at most $2p$ choices of $[U_t]_t$ that will divide $[S]_t$ (the two coprime linear factors of $[S]_t$ and all of their scalar multiples). By Lem. 6.5 there are $< 2p(2p^{t-1}) = 4p^t$ possible choices for U_t such that $[U_t]_t$ divides $[S]_t$. Since there are p^{Q+1} total choices for U_t and the choice of U_t is independent of $[S]_t$,

$$\Pr\left[E_t \mid \bar{D} \wedge \bigwedge_{i=1}^{t-1} \bar{E}_i\right] < \frac{4p^t}{p^{Q+1}} = 4p^{-(Q-t+1)}.$$

We then have

$$\Pr[\bar{E} \mid \bar{D}] = \Pr\left[\bigwedge_{t=1}^{Q} \bar{E}_t \mid \bar{D}\right] = \prod_{t=1}^{Q} \Pr\left[\bar{E}_t \mid \bar{D} \wedge \bigwedge_{i=1}^{t-1} \bar{E}_i\right] >$$
$$\prod_{t=1}^{Q}(1 - 4p^{-(Q-t+1)}) = \prod_{t=1}^{Q}(1 - 4p^{-t}) = 1 - 4p^{-1} - O(p^{-2}),$$

so $\Pr[E \mid \bar{D}] \leq 4p^{-1} + O(p^{-2})$.

Adding up $\Pr[D]$ and $\Pr[E \mid \bar{D}]$, the probability that $\mathcal{A}$ aborts in step 3c is at most

$$(p^{-2} + O(p^{-3})) + (4p^{-1} + O(p^{-2})) = 4p^{-1} + O(p^{-2}).$$

Finally, if $\mathcal{A}$ does not abort in step 3c, then in step 3d $0 = S(\vec{x}) = [S]_{Q+1}(x_t)$ so x_t is a root of $[S]_{Q+1}$. Therefore $\mathcal{A}$ will compute $\vec{x}$ and $u = P_0(\vec{x})/P_{-1}(\vec{x}) = P_1(\vec{x})/P_0(\vec{x})$. The check on u passes and $\mathcal{A}$ returns u, i.e., $\mathcal{A}$ succeeds in the 1-DL game. We conclude that $\mathcal{A}$'s advantage is at least $1 - 4p^{-1} - O(p^{-2})$.

The remaining case is that $\mathcal{R}$ simulates the 1-DL game to $\mathcal{A}$ incorrectly, i.e., either $B_{-1} = 1_G$ or $\mathrm{dlog}_{B_{-1}}(B_0) \neq \mathrm{dlog}_{B_0}(B_1)$; in this case we show that $\mathcal{M}$ will abort. If the former, $\mathcal{M}$ aborts in step 3a. If the latter, note that $\mathrm{dlog}_{B_{-1}}(B_0) \neq \mathrm{dlog}_{B_0}(B_1)$ is equivalent to $S(\vec{x}) \neq 0$. Therefore $0 \neq S(\vec{x}) = [S]_\ell(\vec{x})$, so $[S]_\ell \neq 0$ and $\mathcal{M}$ does not hit step 3b. Therefore either $\mathcal{M}$ aborts in step 3c, or $\mathcal{M}$ makes it to step 3d: in the latter case, the check on u' will never pass so $\mathcal{M}$ also aborts.

Regarding $\mathcal{A}$'s runtime, in step 3c $\mathcal{A}$ makes up to $Q+1$ queries to DL, each of which involves $Q+2$ exponentiations and $Q+1$ multiplications; after that, in step 3d $\mathcal{A}$ checks consistency of the answers, which involves 2 exponentiations. So the total number of group operations is up to $(Q+1)[2(Q+2)\lambda + Q + 1] + 2 \cdot (2\lambda) = (2Q^2 + 6Q + 4)\lambda + (Q+1)^2$. □

Remark 6.6. *Asymptotic-wise, Thm. 6.1 is stronger than what's needed for a separation result for OMDL: a successful reduction should work for any PPT $(Q+1)$-OMDL solver that succeeds with* non-negligible *probability, while our result even rules out reductions that are much weaker, i.e., those that only work for PPT $(Q+1)$-OMDL solvers whose success probability is* overwhelming. *The same goes for Thm. 6.4.*

7 Future Work

While our work covers all group-based one-more assumptions in the literature, there are some potential future directions that remain unexplored. Bauer, Fuchsbauer, and Plouviez [BFP21] note it appears unlikely that OMDL can be shown to be hard in the GGM via standard applications of the Schwartz–Zippel lemma, the technique by which all other existing proofs of hardness in the GGM proceed. Instead, they use a different proof technique, and suggest that this difference may be linked to the result of [BFL20] that the hardness of Q-OMDL cannot be concluded from the

hardness of Q'-DL in the AGM (for any $Q \geq 0, Q' \geq 1$). Our work lends additional credence to this intuition, showing that Q-OMDL and Q'-DL define infinite and incomparable hierarchies of problems in the AGM. We believe an intriguing direction for future work is to explore to what extent results in the AGM function as "meta-theorems" in the GGM. It seems probable that a problem is part of the Q-DL hierarchy in the AGM if and only if it admits a "standard" proof of hardness in the GGM: how might this be formalized and proven?

References

[Adl79] Leonard Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In *20th Annual Symposium on Foundations of Computer Science (SFCS 1979)*, pages 55–60. IEEE Computer Society, 1979.

[AMMM18] Shashank Agrawal, Peihan Miao, Payman Mohassel, and Pratyay Mukherjee. PASTA: PASsword-based threshold authentication. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 2042–2059. ACM Press, October 2018.

[BB08] Dan Boneh and Xavier Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, April 2008.

[BFL20] Balthazar Bauer, Georg Fuchsbauer, and Julian Loss. A classification of computational assumptions in the algebraic group model. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 121–151. Springer, Cham, August 2020.

[BFP21] Balthazar Bauer, Georg Fuchsbauer, and Antoine Plouviez. The one-more discrete logarithm assumption in the generic group model. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 587–617. Springer, Cham, December 2021.

[BMV08] Emmanuel Bresson, Jean Monnerat, and Damien Vergnaud. Separation results on the "one-more" computational problems. In Tal Malkin, editor, *CT-RSA 2008*, volume 4964 of *LNCS*, pages 71–87. Springer, Berlin, Heidelberg, April 2008.

[BN06] Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 390–399. ACM Press, October / November 2006.

[BNN04] Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Security proofs for identity-based identification and signature schemes. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 268–286. Springer, Berlin, Heidelberg, May 2004.

[BNPS02] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The power of RSA inversion oracles and the security of Chaum's RSA-based blind signature scheme. In Paul F. Syverson, editor, *FC 2001*, volume 2339 of *LNCS*, pages 319–338. Springer, Berlin, Heidelberg, February 2002.

[Bol03] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 31–46. Springer, Berlin, Heidelberg, January 2003.

[BP02] Mihir Bellare and Adriana Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 162–177. Springer, Berlin, Heidelberg, August 2002.

[Che06] Jung Hee Cheon. Security analysis of the strong Diffie-Hellman problem. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 1–11. Springer, Berlin, Heidelberg, May / June 2006.

[DL78] Richard A DeMillo and Richard J Lipton. A probabilistic remark on algebraic program testing. *Information processing letters*, 7(4):193–195, 1978.

[FKL18] Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Cham, August 2018.

[FPS20] Georg Fuchsbauer, Antoine Plouviez, and Yannick Seurin. Blind Schnorr signatures and signed ElGamal encryption in the algebraic group model. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 63–95. Springer, Cham, May 2020.

[GJK+24] Yanqi Gu, Stanislaw Jarecki, Pawel Kedzior, Phillip Nazarian, and Jiayu Xu. Threshold PAKE with security against compromise of all servers. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part V*, volume 15488 of *LNCS*, pages 66–100. Springer, Singapore, December 2024.

[HLY09] Jung Yeon Hwang, Dong Hoon Lee, and Moti Yung. Universal forgery of the identity-based sequential aggregate signature scheme. In Wanqing Li, Willy Susilo, Udaya Kiran Tupakula, Reihaneh Safavi-Naini, and Vijay Varadharajan, editors, *ASIACCS 09*, pages 157–160. ACM Press, March 2009.

[JKK14] Stanislaw Jarecki, Aggelos Kiayias, and Hugo Krawczyk. Round-optimal password-protected secret sharing and T-PAKE in the password-only model. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 233–253. Springer, Berlin, Heidelberg, December 2014.

[JKKX16] Stanislaw Jarecki, Aggelos Kiayias, Hugo Krawczyk, and Jiayu Xu. Highly-efficient and composable password-protected secret sharing (or: How to protect your bitcoin wallet online). In *2016 IEEE European Symposium on Security and Privacy*, pages 276–291. IEEE Computer Society Press, March 2016.

[JKKX17] Stanislaw Jarecki, Aggelos Kiayias, Hugo Krawczyk, and Jiayu Xu. TOPPSS: Cost-minimal password-protected secret sharing based on threshold OPRF. In Dieter Gollmann, Atsuko Miyaji, and Hiroaki Kikuchi, editors, *ACNS 17International Conference on Applied Cryptography and Network Security*, volume 10355 of *LNCS*, pages 39–58. Springer, Cham, July 2017.

[JM24] Joseph Jaeger and Deep Inder Mohan. Generic and algebraic computation models: When AGM proofs transfer to the GGM. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part V*, volume 14924 of *LNCS*, pages 14–45. Springer, Cham, August 2024.

[KMMM23] Aniket Kate, Easwar Vivek Mangipudi, Siva Maradana, and Pratyay Mukherjee. FlexiRand: Output private (distributed) VRFs and application to blockchains. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *ACM CCS 2023*, pages 1776–1790. ACM Press, November 2023.

[KU11] Kiran S Kedlaya and Christopher Umans. Fast polynomial factorization and modular composition. *SIAM Journal on Computing*, 40(6):1767–1802, 2011.

[LNÖ25] Anja Lehmann, Phillip Nazarian, and Cavit Özbay. Stronger security for threshold blind signatures. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 335–364. Springer, 2025.

[LRSW99] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In Howard M. Heys and Carlisle M. Adams, editors, *SAC 1999*, volume 1758 of *LNCS*, pages 184–199. Springer, Berlin, Heidelberg, August 1999.

[NRS21] Jonas Nick, Tim Ruffing, and Yannick Seurin. MuSig2: Simple two-round Schnorr multi-signatures. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 189–221, Virtual Event, August 2021. Springer, Cham.

[Rot22] Lior Rotem. Revisiting the uber assumption in the algebraic group model: Fine-grained bounds in hidden-order groups and improved reductions in bilinear groups. In *3rd Conference on Information-Theoretic Cryptography*, 2022.

[Sch24] Sven Schäge. New limits of provable security and applications to ElGamal encryption. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part IV*, volume 14654 of *LNCS*, pages 255–285. Springer, Cham, May 2024.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Berlin, Heidelberg, May 1997.

[TCR+22] Nirvan Tyagi, Sofía Celi, Thomas Ristenpart, Nick Sullivan, Stefano Tessaro, and Christopher A. Wood. A fast and simple partially oblivious PRF, with applications. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 674–705. Springer, Cham, May / June 2022.

[Zha22] Mark Zhandry. To label, or not to label (in generic groups). In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 66–96. Springer, Cham, August 2022.

[ZZC+14] Jiang Zhang, Zhenfeng Zhang, Yu Chen, Yanfei Guo, and Zongyang Zhang. Black-box separations for one-more (static) CDH and its generalization. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 366–385. Springer, Berlin, Heidelberg, December 2014.

[ZZK22] Cong Zhang, Hong-Sheng Zhou, and Jonathan Katz. An analysis of the algebraic group model. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part IV*, volume 13794 of *LNCS*, pages 310–322. Springer, Cham, December 2022.

A Proof of Lems. 5.6 and 5.7

Since Lem. A.1 and Lem. 5.6 are almost identical to [JKKX17, Lemmas 1 and 2], we provide proofs for them primarily for clarity of exposition, as we fix several typos and other minor mistakes in the original proofs.

Lemma A.1 ([JKKX17, Lemma 1])**.** *Let $u, n \in \mathbb{Z}_{>0}$. Then there is no $\vec{q} = (q_1, \ldots, q_n) \in \mathbb{Z}^n_{\geq 0}$ such that*

1. *$w \geq Qu$;*
2. *$q_i \leq Q$ for all $1 \leq i \leq n$,*

for $w = W(\vec{q})$ and $Q = C_u(\vec{q}) + 1$.

Proof. Proof by induction on Q. If $Q = 1$ then $C_u(\vec{q}) = 0$ so there are at most $u - 1$ nonzero entries of q. If Item 2 holds then $w \leq u - 1$ so Item 1 cannot hold.

Now we show if the claim is false for Q it's false for $Q - 1$. Suppose $\vec{q}$ is a counterexample to the claim with $Q = C_u(\vec{q}) + 1$. Then $\vec{q}$ has at most $u - 1$ entries $\geq Q$; otherwise we'd have $C_u(\vec{q}) \geq Q$ by decreasing these entries Q times. Let $(\vec{q})'$ be $\vec{q}$ with the largest u entries decreased by 1, and $w' = W((\vec{q})')$. By assumption $w \geq Qu$ and $q_i \leq Q$, so $w' = w - u \geq Qu - u = (Q-1)u, q'_i \leq Q - 1$ and $C_u((\vec{q})') = Q - 2$ by construction. Therefore $(\vec{q})'$ is a counterexample for $Q - 1$. □

Proof of Lem. 5.6. Let $\mathbf{M} \in \mathbb{F}_p^{w \times Q(t-t'+1)}$ be the matrix whose columns are vectors in V: it is sufficient to show $\operatorname{rank}(\mathbf{M}) < Q(t - t' + 1)$. For $r \in [n]$ there are q_r coordinates of $\vec{k}$ with entry r. Consider the corresponding rows in $\mathbf{M}$ and denote this $q_r \times Q(t - t' + 1)$ submatrix as $\mathbf{M}_r$. Note that the columns of $\mathbf{M}_r$ are the vectors

$$\left(r^j - \sum_{\ell=t-t'+1}^{t} \alpha_j^{(\ell)} r^\ell \right) [\vec{b}_i] \text{ for } j \in \{0, \ldots, t - t'\}, i \in [Q]$$

where $[\vec{b}_i]$ is the vector $\vec{b}_i$ restricted to the rows of $\mathbf{M}$ with entry r. Therefore $\operatorname{rank}(\mathbf{M}_r) \leq Q$: all columns with $i = i_0$ are multiples of $[\vec{b}_{i_0}]$. For any $q_r > Q$ we then select Q rows of $\mathbf{M}_r$ that span the row space of $\mathbf{M}_r$ to form $\mathbf{M}'_r$. For any $q_r \leq Q$ set $\mathbf{M}'_r = \mathbf{M}_r$. Let q'_r be the number of rows of $\mathbf{M}'_r$ (so $q'_r \leq Q$) and let $w' = W((\vec{q})')$. Now let $\mathbf{M}' \in \mathbb{F}_p^{w' \times Q(t-t'+1)}$ be the concatenation of $\mathbf{M}'_1, \ldots, \mathbf{M}'_n$. By construction the row space of $\mathbf{M}'$ equals the row space of $\mathbf{M}$, so $\operatorname{rank}(\mathbf{M}') = \operatorname{rank}(\mathbf{M})$.

Additionally, $C_{t-t'+1}((\vec{q})') = C_{t-t'+1}(\vec{q}) = Q - 1$. To see this, by construction $C_{t-t'+1}((\vec{q})') \leq Q - 1$. On the other hand, take $\vec{v}_1, \ldots, \vec{v}_{Q-1} \in \mathcal{V}_{t-t'+1}$ such that $\sum_{i=1}^{Q-1} \vec{v}_i \leq \vec{q}$. Each entry of $\sum_{i=1}^{Q-1} \vec{v}_i$ is at most $Q - 1$ so $\sum_{i=1}^{Q-1} \vec{v}_i \leq (\vec{q})'$ since $(\vec{q})'$ is the vector $\vec{q}$ with entries $> Q$ decreased to Q. Therefore $C_{t-t'+1}((\vec{q})') \geq Q - 1$.

By Lem. A.1 we have $w' < Q(t - t' + 1)$, so $\operatorname{rank}(\mathbf{M}) = \operatorname{rank}(\mathbf{M}') \leq w' < Q(t - t' + 1)$. □

Proof of Lem. 5.7. Recall from Sect. 5.2 that $\{\vec{f}^u, \vec{f}^{u+1}, \ldots, \vec{f}^{u+t'-1}\}$ is linearly independent for any $u \geq 0$, and

$$\vec{f}^j = \sum_{\ell=t-t'+1}^{t} \alpha_j^{(\ell)} \vec{f}^\ell \text{ for } 0 \leq j \leq t - t'.$$

By the linear independence of $\{\vec{f}^{(t-t'+j)+1}, \vec{f}^{(t-t'+j)+2}, \ldots, \vec{f}^{t+j}\}$ it suffices to show

$$\underbrace{\sum_{\ell'=t-t'+1}^{t} \alpha_{\lambda-j}^{(\ell')} \vec{f}^{\ell'+j}}_{E} = \sum_{\ell'=t-t'+1}^{t-j} \left(\alpha_{\lambda}^{(\ell'+j)} + \sum_{\ell=t-t'+1}^{t-t'+j} \alpha_{\lambda}^{(\ell)} \alpha_{\ell-j}^{(\ell')} \right) \vec{f}^{\ell'+j} + \sum_{\ell'=t-j+1}^{t} \left(\sum_{\ell=t-t'+1}^{t-t'+j} \alpha_{\lambda}^{(\ell)} \alpha_{\ell-j}^{(\ell')} \right) \vec{f}^{\ell'+j}$$

$$= \underbrace{\sum_{\ell'=t-t'+1}^{t-j} \alpha_{\lambda}^{(\ell'+j)} \vec{f}^{\ell'+j}}_{F_1} + \underbrace{\sum_{\ell'=t-t'+1}^{t-j} \left(\sum_{\ell=t-t'+1}^{t-t'+j} \alpha_{\lambda}^{(\ell)} \alpha_{\ell-j}^{(\ell')} \right) \vec{f}^{\ell'+j} + \sum_{\ell'=t-j+1}^{t} \left(\sum_{\ell=t-t'+1}^{t-t'+j} \alpha_{\lambda}^{(\ell)} \alpha_{\ell-j}^{(\ell')} \right) \vec{f}^{\ell'+j}}_{F_2}.$$

We have

$$E = \vec{f}^{j} \odot \sum_{\ell'=t-t'+1}^{t} \alpha_{\lambda-j}^{(\ell')} \vec{f}^{\ell'} = \vec{f}^{j} \odot \vec{f}^{\lambda-j} = \vec{f}^{\lambda}.$$

Additionally,

$$F_1 = \sum_{\ell'=t-t'+1+j}^{t} \alpha_{\lambda}^{(\ell')} \vec{f}^{\ell'} = \vec{f}^{\lambda} - \sum_{\ell'=t-t'+1}^{t-t'+j} \alpha_{\lambda}^{(\ell')} \vec{f}^{\ell'}.$$

Finally,

$$\begin{aligned} F_2 &= \sum_{\ell=t-t'+1}^{t-t'+j} \left(\sum_{\ell'=t-t'+1}^{t-j} \alpha_{\lambda}^{(\ell)} \alpha_{\ell-j}^{(\ell')} \vec{f}^{\ell'+j} + \sum_{\ell'=t-j+1}^{t} \alpha_{\lambda}^{(\ell)} \alpha_{\ell-j}^{(\ell')} \vec{f}^{\ell'+j} \right) \\ &= \sum_{\ell=t-t'+1}^{t-t'+j} \left(\sum_{\ell'=t-t'+1}^{t} \alpha_{\lambda}^{(\ell)} \alpha_{\ell-j}^{(\ell')} \vec{f}^{\ell'+j} \right) = \sum_{\ell=t-t'+1}^{t-t'+j} \alpha_{\lambda}^{(\ell)} \vec{f}^{j} \odot \left(\sum_{\ell'=t-t'+1}^{t} \alpha_{\ell-j}^{(\ell')} \vec{f}^{\ell'} \right) \\ &= \sum_{\ell=t-t'+1}^{t-t'+j} \alpha_{\lambda}^{(\ell)} \vec{f}^{j} \odot \vec{f}^{\ell-j} = \sum_{\ell=t-t'+1}^{t-t'+j} \alpha_{\lambda}^{(\ell)} \vec{f}^{\ell}, \end{aligned}$$

so $F_1 + F_2 = \vec{f}^{\lambda} = E$. □