

Generic Bounds for Multi-Instance Problems: The Strange Case of Inverse Diffie-Hellman

Akshima , Eike Kiltz , Aysan Nishaburi ,
Samin Nooripoor , and Emiel Wiedijk 
{akshima.akshima,eike.kiltz,aysan.nishaburi,
samin.nooripoor,emiel.wiedijk}@rub.de

Ruhr-Universität Bochum, Germany

September 28, 2026

Abstract. In a multi-instance problem, the adversary is given n independent problem instances and succeeds if it solves at least k of them. We study the generic hardness of a broad class of multi-instance problems over prime-order groups, including multi-instance Computational Diffie–Hellman (CDH), Square Diffie–Hellman (SDH), Inverse Diffie–Hellman (IDH), and Linear Kernel Diffie–Hellman (LKDH). As expected, we show that, in generic groups, solving multi-instance CDH and SDH is as hard as solving k independent discrete logarithm instances. In contrast, although the single-instance versions of CDH and IDH are known to be equivalent, we establish generic upper and lower bounds showing that multi-instance IDH (and LKDH) can be significantly easier to solve than multi-instance CDH in certain parameter regimes. All problems, however, offer $1/2(\log(p) + \log(k))$ bits of security. Our lower-bound techniques extend Yun’s information-theoretic hyperplane query model (EUROCRYPT 2015). At the core of our approach lies an inductive argument showing that, in order to establish a generic-group lower bound, it suffices to bound the adversary’s “zero-hyperplane-query” advantage in the hyperplane query model. For the multi-instance problems considered in this work, we derive such bounds using techniques from algebraic geometry. More precisely, we analyze the Krull dimension of suitable algebraic varieties, either by explicitly computing Gröbner bases or by upper bounding the number of algebraically independent elements in the associated coordinate rings.

Table of Contents

1	Introduction	3
1.1	Our contributions	3
1.2	Is Multi-Instance Inverse Diffie-Hellman Strange?	6
1.3	Related work	7
2	Preliminaries	8
3	Multi Instance Problems	8
3.1	Multi-Instance Problems	8
3.2	Generic Group Model	9
4	The Hyperplane Query Model	10
5	General Bounds in the Hyperplane Query Model	12
5.1	Relation to zero-query advantage	12
5.2	Useful bounds for applying Theorem 2	14
6	Zero-query advantage	15
6.1	(n, k) -SDH via Gröbner bases	16
6.2	(n, k) -CDH via Gröbner bases	19
6.3	(n, k) -IDH via algebraic independence	19
7	Multi-Instance bounds in the Generic Group Model	20
7.1	(n, k) -CDH and (n, k) -SDH	20
7.2	(n, k) -IDH	21
7.3	(n, k) -LKDH	21
7.4	Concrete bit security	22
A	Proof of Theorem 1	23
B	Proof of Lemma 1	25
C	Algebraic geometry	29
D	Multi-Instance CDH and LKDH	30
E	Proof of Lemma 6	31

1 Introduction

The two central computational problems in a cyclic group $\mathbb{G}$ of prime order p generated by g are the discrete logarithm (DLOG) problem and the Computational Diffie-Hellman (CDH) problem. The DLOG problem is defined as follows: given g^x for uniformly random $x \in \mathbb{Z}_p$, compute x . The CDH problem is: given (g^x, g^y) , compute the Diffie-Hellman key g^{xy} . Less well studied is the Inverse Diffie-Hellman (IDH) problem [SS01, BDZ03], which is defined as follows: given g^x , compute the inverse Diffie-Hellman key $g^{1/x}$. The two problems, IDH and CDH, are known to be (non-tightly) equivalent [SS01].

The best known generic algorithms for DLOG, CDH, and IDH are based on Pollard’s rho method and achieve running time $O(\sqrt{p})$. Shoup’s lower bound in the generic group model [Sho97] establishes the optimality of these algorithms in the generic setting.

MULTI-INSTANCE DISCRETE LOGARITHM AND DIFFIE-HELLMAN. For integers $k \leq n$, all problems admit natural k -out-of- n multi-instance variants in which the adversary is given n independent instances and must solve at least k of them. For example, the k -out-of- n multi-instance Computational Diffie-Hellman $((n, k)$ -CDH) problem is defined as follows: given $(g^{x_1}, g^{y_1}), \dots, (g^{x_n}, g^{y_n})$, compute k of the corresponding Diffie-Hellman values [AGK20].

It is well known that solving multiple discrete logarithm instances simultaneously can be substantially easier than solving them independently. The best known generic algorithm for the (n, k) -CDH problem runs in time $O(\sqrt{kp})$, and matching lower bounds are known in the generic group model [Yun15, AGK20, Hha25].

To establish generic lower bounds for the (n, n) -DLOG problem, Yun introduced the hyperplane query model [Yun15], an information-theoretic framework in which the adversary attempts to recover a hidden vector through adaptive hyperplane queries. Auerbach, Giacon, and Kiltz [AGK20] later employed the algebraic group model [FKL18] to derive generic lower bounds for the (n, k) -CDH problem.

Subsequently, Corrigan-Gibbs and Kogan [CK18], as well as Hhan [Hha25], developed compression-based techniques that, among other applications, simplify Yun’s lower-bound proof for the (n, n) -DLOG problem. Hhan posed as an open problem whether these techniques can be extended to obtain a direct lower bound for the (n, n) -CDH problem.

1.1 Our contributions

We first give a general definition of multi-instance problems over a prime-order group $\mathbb{G}$. Our definition covers multi-instance version of CDH and IDH, as well as Square Diffie-Hellman (SDH), and Linear Kernel Diffie-Hellman (LKDH). Our definitions also support corruption queries, where the adversary can query discrete logarithms, for example to obtain the Diffie-Hellman keys.

Next, we generalize the hyperplane query model to cover our multi-instance problems with corruptions and use it to prove generic group lower bounds for

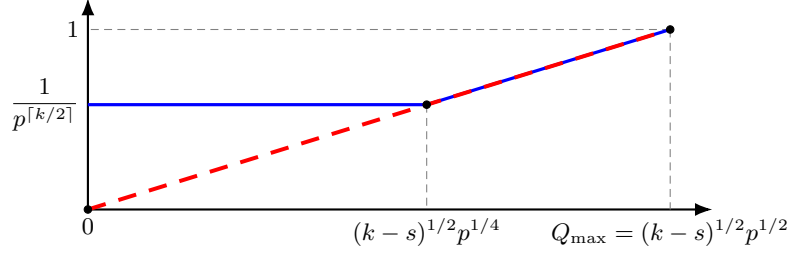


Fig. 1. Generic group lower bounds on the advantage functions for (n, k) -CDH and (n, k) -SDH (red), as well as for (n, k) -IDH and $(n, k+1)$ -LKDH (blue). The x -axis depicts the running time Q ; the y -axis depicts the advantage (log-log scale); and s denotes the number of discrete logarithm queries.

several multi-instance problems. Concretely, the generic group advantage of an adversary $\mathcal{A}$ against multi-instance CDH and SDH, making Q group operation and $s \leq k$ discrete logarithm (corruption) queries is as follows:

$$\forall P \in \{(n, k)\text{-CDH}, (n, k)\text{-SDH}\} : \text{Adv}^P(\mathcal{A}) \lesssim \left(\frac{Q^2}{(k-s)p} \right)^{k-s}. \quad (1)$$

However, for multi-instance IDH and LKDH, the advantage is:

$$\forall P \in \left\{ \begin{array}{l} (n, k)\text{-IDH}, \\ (n, k+1)\text{-LKDH} \end{array} \right\} : \text{Adv}^P(\mathcal{A}) \lesssim \begin{cases} \left(\frac{Q^2}{(k-s)p} \right)^{k-s} & Q \geq (k-s)^{1/2}p^{1/4} \\ \left(\frac{1}{p} \right)^{\lceil \frac{k-s}{2} \rceil} & \text{otherwise.} \end{cases} \quad (2)$$

Note that for $Q = (k-s)^{1/2}p^{1/4}$ queries, the advantages in the two cases (depending on Q) coincide. See also Figure 1 for a graphical representation of the two bounds (1) and (2). To state the upper bounds in (1) and (2), we use “ $\lesssim$ ” to suppress small constants within the exponents.¹

The core of our computations relies on concepts from algebraic geometry. We relate the advantage of an adversary to algebraic properties of a variety defined by the output of the adversary and a “winning polynomial” defined by the problem instance. Specifically, we bound the Krull dimension of this variety, either by computing a Gröbner basis or by bounding the number of algebraically independent elements in the coordinate ring. The framework is more general, however, and any method to upper bound the Krull dimension can be used to give upper bounds on the adversary’s success probability.

We now give technical details of our contributions.

GENERAL MULTI-INSTANCE PROBLEMS. We define k -out-of- n multi-instance problems relative to a winning polynomial W through the game $(n, k)\text{-MI}_W$. The polynomial W specifies the condition under which an instance is considered solved and is used to determine how many of the n instances the adversary successfully solves. Our

¹ A constant c in a term raised to the power $k-s$ contributes a factor c^{k-s} to the bound. Since this factor has only a marginal effect on the resulting bit-security level, we suppress it for simplicity.

formulation additionally allows the adversary to access a discrete logarithm oracle (up to s times). Our definition (Definition 1 of Section 3) covers multi-instance versions of CDH, IDH, SDH, as well as LKDH. The (n, n) -LKDH problem is the $\mathcal{L}_n$ -Kernel Diffie-Hellman problem from [MRV16, MRV15], where $\mathcal{L}_n$ is the linear Matrix Diffie-Hellman distribution [EHK⁺13].² We think that discrete logarithm queries are an interesting addition in the context of generic lower bounds, since they allow us to model interactive corruption of Diffie-Hellman keys.

HYPERPLANE QUERY MODEL. Our general hyperplane query model (n, k) -MI-HQM _{W} relative to a winning polynomial W is an information-theoretic framework in which the experiment samples a random secret vector x over $\mathbb{Z}_p$. The adversary is granted access to a hyperplane membership oracle $H_x(H_f)$. On input of a hyperplane H_f , the oracle returns 1 if and only if $x \in H_f$. Intuitively, hyperplane queries correspond to group queries in the generic group model. In addition, the adversary has access to a full evaluation oracle $\text{EVAL}_x(H_f)$, which, on input a hyperplane H_f , returns the value $\beta := f(x)$. Intuitively, evaluation queries correspond to discrete logarithm queries in the generic group model. Finally, the adversary outputs a list of polynomials and succeeds if these polynomials, evaluated at the secret x , satisfy a prescribed winning condition related to the winning polynomial W . We consider information-theoretic adversaries and measure their complexity both in terms of the total number q of oracle queries and the number $s \ll q$ of queries made specifically to the evaluation oracle. Our first main result (Theorem 1 of Section 4) shows that (n, k) -MI-HQM _{W} implies (n, k) -MI _{W} in the generic group model, where Q generic group queries can be simulated by $O(Q^2)$ hyperplane queries and s discrete logarithm queries can be simulated by s evaluation queries.

BOUNDS IN THE HYPERPLANE QUERY MODEL. Our second main result (Theorem 2 of Section 5) shows that the adversary’s advantage in the hyperplane query model (n, k) -MI-HQM _{W} can be related to its zero-query advantage. The zero-query advantage refers to the adversary’s advantage in the hyperplane query model when the adversary makes zero hyperplane and evaluation queries. At the core of the proof of Theorem 2 we use a “query-reduction” induction technique that recursively bounds the advantage of (n, k) -MI-HQM _{W} for any adversary making q queries by the advantages of (n, k) -MI-HQM _{W} and $(n - 1, k - 1)$ -MI-HQM _{W} for any adversary making $q - 1$ queries.

ZERO-QUERY ADVANTAGES. Finally, in Section 6 we compute the zero-query advantages for the considered multi-instance problems. Algebraically, the zero-query advantage is related to the Krull dimension of an algebraic variety related to the winning polynomial W . By the Lang–Weil theorem, the zero-query advantage is upper bounded by $\deg(W)^k \cdot p^{d-nm}$, where d is an upper bound on the Krull dimension of the algebraic variety and the integer $m \in \{1, 2\}$ is a parameter related to the winning polynomial.

For multi-instance CDH and SDH, we use Gröbner bases to upper bound the Krull dimension by $d_{\text{CDH}} = 2n - k$, and $d_{\text{SDH}} = n - k$. This yields a zero-query advantage of at most $(2/p)^k$ for both problems. Applying our framework then gives the generic-group bound (1). For IDH, bounding the Krull dimension requires a different approach. Exploiting the structure induced by the IDH equations, we upper bound the number of algebraically independent elements in the coordinate ring—which equals the Krull dimension—by $\lfloor n - k/2 \rfloor$. This weaker Krull-dimension bound accounts for the weaker generic bound (2). We do not directly show the hardness of LKDH using the zero-query

² Concretely, the (n, n) -LKDH problem is the $\mathcal{D}_{n+1, n}$ -Kernel Diffie-Hellman problem, where $\mathcal{D}_{n+1, n} = \mathcal{L}_n$ is the linear Matrix distribution [EHK⁺13].

approach. Instead, we infer the bound for (n, k) -LKDH by a reduction in the standard model, showing that (n, k) -IDH implies (n, k) -LKDH with one fewer query to a discrete logarithm oracle. Hence $(n, k + 1)$ -LKDH matches the generic lower bound of (n, k) -IDH. Again, using our framework, this implies the generic group bound (2).

TIME SPACE TRADEOFFS. We note that analysing multi-instance problems has proven useful in the study of time-space tradeoffs, which are meaningful as they characterize non-uniform security. The high level idea is that, if we can show no adversary can solve a (randomly and independently sampled) n instances of a problem with probability $\omega(\delta^n)$, then no adversary with n -bit advice can solve a single (random) instance of the problem with probability $\omega(\delta)$.

To the best of our knowledge, [Imp11] first introduced this approach as a variant of “constructive” Chernoff bounds. Recently, [ACDW20] used this concentration-based approach to study time-space tradeoffs for “short” collisions in Merkle-Damgård-based constructions. In an independent concurrent work, [CGLQ20] also showed this connection in (Q)ROM. Several works [GGKL21, AGL22, FGK22, GK22, FGK23, ADGL23, ABG⁺24] have since used this connection with multi-instance problems to obtain time-space lower bounds for problems in different idealized models.

1.2 Is Multi-Instance Inverse Diffie-Hellman Strange?

The bounds (1) and (2) indicate that the multi-instance IDH (and the multi-instance LKDH) problem exhibits qualitatively different behaviour from multi-instance CDH, despite the fact that the corresponding single-instance assumptions are equivalent. One may wonder whether our strange lower bound (2) is non-tight for small values of Q .

We emphasize that this is not the case. In fact, there exists a simple attack that matches the lower bound for the (n, k) -IDH problem in the regime $Q \leq k^{1/2}p^{1/4}$. Assume that $k \leq n$ is even, and consider the n instances $(g^{x_1}, \dots, g^{x_n})$. In case the product of the first two exponents is trivial, i.e., $x_1x_2 = 1$, the adversary can output the first two Diffie-Hellman inverses $g^{1/x_1} = g^{x_2}$ and $g^{1/x_2} = g^{x_1}$ without making a single computation. Generally, if $x_1x_2 = x_3x_4 = \dots x_{k-1}x_k = 1$, the adversary can output all k Diffie-Hellman inverses. The success probability of this attack equals the probability that all $k/2$ products are trivial, namely $1/p^{k/2}$. It is straightforward to verify that, for arbitrary (not necessarily even) values of k , the success probability of the attack is $1/p^{\lceil k/2 \rceil}$. Our lower bound shows that, for $Q \leq k^{1/2}p^{1/4}$, this trivial strategy is in fact optimal among all generic algorithms. The same attack extends to (n, k) -LKDH.

It remains an interesting question whether (n, k) -IDH is inherently “easier” to attack than (n, k) -CDH. The bounds (1) and (2) indicate that both problems require approximately $\sqrt{(k-s)p}$ group operations to be solved with constant success probability, where s is the number of discrete logarithm oracle calls. Moreover, even under the standard notion of bit security [BR09]³, multi-instance IDH still achieves $\frac{1}{2}(\log(p) + \log(k-s))$ bits of security. This matches the security level of multi-instance (n, k) -CDH, (n, k) -SDH, as well as $(n, k+1)$ -LKDH. We refer to Section 7 for the derivation of the corresponding bit-security levels.

³ A problem has κ bits of security if, for every adversary $\mathcal{A}$, it holds that $\text{Adv}(\mathcal{A})/\text{Time}(\mathcal{A}) \leq 2^{-\kappa}$.

1.3 Related work

We have already discussed Yun’s hyperplane query model [Yun15] that established generic lower bounds for the (n, n) -DLOG problem using combinatorial techniques. The high level idea in [Yun15] is to count the number of winning transcripts with q adaptive hyperplane queries where the winning transcript should have n “successful” queries and the probability of any query “succeeding” is $1/p$.

Later, the proof has been simplified using compression arguments [CK18, Hha25]. We expect that it is possible to obtain our results using Hhan’s insights on compression techniques from [Hha25] and the algebraic geometry techniques we use here. However, it seems that compression techniques will fail to link the advantage to the zero-query advantage and thus miss the important intuition we build from it.

Auerbach, Giacon, and Kiltz [AGK20] later employed the algebraic group model [FKL18] to derive generic lower bounds for the (n, k) -CDH problem. Their geometric analysis of hypersurface queries tracks dimensions and irreducible components of successive intersections, using Bézout’s theorem to control the reduction loss. Their subsequent algebraic reduction avoids analyzing the size of the solution set of a multivariate system of quadratic equations. In our work, algebraic geometry is used to analyze similar equations, yielding upper bounds on the zero-query advantage. Specifically, the zero-query advantage relates to the number of finite-field solutions to the equations defined by the adversary’s output and the winning condition. For this analysis, we use additional tools, including Lang-Weil style bounds, Gröbner bases, and algebraic-independence arguments.

In subsequent work, Auerbach, Hoffmann, and Pascual-Perez [AHPP23] develop a reduction-based framework for proving generic-group lower bounds via intermediate geometric-search problems. They study the hardness of several such search problems in the generic group model, both with and without preprocessing, and use reductions to obtain lower bounds for a range of assumptions, including d -strong discrete logarithm, d -strong Diffie–Hellman inversion, multi-instance CDH, and classes of Uber assumptions. However, their framework does not appear to yield tight bounds for all problems considered here. In particular, for (n, k) -IDH, the optimal generic bound has a different behavior from the CDH-style bound, and their method would not capture the tight small-query regime obtained by our analysis.

Our techniques at a high level are similar to the work of Akshima et al. [ABG⁺24] with some significant differences that we elaborate later. [ABG⁺24] obtained the optimal time-space lower bounds for the Decisional Diffie–Hellman (DDH) problem in the generic group model, improving the bounds from [CK18]. [ABG⁺24] showed that DDH is asymptotically as secure as the DLOG problem even for pre-computing adversaries. The work uses techniques from [GGKL21] and [Yun15] to reduce the problem to analyzing an n -XOR variant of the DDH problem (guess $\bigoplus_{i=1}^n b_i$ given n instances $(g^{x_1}, g^{y_1}, g^{u_1}), \dots, (g^{x_n}, g^{y_n}, g^{u_n})$ where $u_i = b_i r_i + (1 - b_i) x_i y_i$ for $x_i, y_i, r_i \xleftarrow{\$} \mathbb{Z}_p$ and $b_i \xleftarrow{\$} \{0, 1\}$) in Yun’s hyperplane query model. As a by-product of their proof, they showed that n -XOR variant of DDH is asymptotically as secure as (n, n) -DLOG.

In this work, we extend and generalize their induction-style analysis in the Hyperplane Query Model to search-type group problems with discrete-logarithm (evaluation) queries. The biggest difference from [ABG⁺24] and one of our significant contributions is that we establish an explicit connection to the zero-query advantage. We realize, unlike decision problems where the zero-query advantage is always $1/2$ (or 0 depending on the definition) and hence of no consequence, computing the zero-query advantage for our search problems is more involved and allows us to precisely pinpoint how the

asymptotic security of the IDH problem differs from other problems such as the DLOG problem and the CDH problem. Finally, our induction is different from [ABG⁺24] as we allow corruptions.

2 Preliminaries

For an integer $k \in \mathbb{N}$, we let $[k]$ be the set $\{1, \dots, k\}$. If x is an n -dimensional vector, we write $x = (x_1, \dots, x_n)$ for its components.

3 Multi Instance Problems

Throughout this paper we assume a fixed public group description $\mathcal{G} = (\mathbb{G}, p, g)$, where $\mathbb{G}$ is a group of prime order p and g is a generator.

STANDARD PROBLEMS. We recall several standard computational problems. The Computational Diffie-Hellman (CDH) problem is: given g^x and g^y compute g^{xy} . The Square Diffie-Hellman (SDH) problem [MW96] is: given g^x compute g^{x^2} . SDH is known to be equivalent to CDH [MW96]. The Inverse Diffie-Hellman (IDH) problem [SS01, BDZ03] is: given g^x compute $g^{\frac{1}{x}}$. IDH is known to be (non-tightly) equivalent to SDH⁴ and hence also to CDH [SS01]. Finally, the Simultaneous Diffie-Hellman (SimDH) problem [AHH21, PW17, ABB⁺20] is: given (g^{a_1}, g^{a_2}, g^x) , compute a tuple of the form $(Y, Y^{x/a_1}, Y^{x/a_2})$. SimDH implies CDH, whereas the converse implication is likely false [MRV16].

3.1 Multi-Instance Problems

The n -multi-instance CDH problem (n -CDH) is: given $(g^{x_1}, g^{y_1}), \dots, (g^{x_n}, g^{y_n})$, compute the n Diffie-Hellman tuples $g^{x_1 y_1}, \dots, g^{x_n y_n}$. For $k \leq n$, the (n, k) -CDH problem is to compute k of the Diffie-Hellman tuples [AGK20].

In this paragraph we generalize the (n, k) -CDH problem to arbitrary (n, k) multi-instance problems. A multi-instance winning polynomial is given by integers $m \geq 1$ and $\ell \geq 0$, together with a polynomial $W \in \mathbb{Z}_p[\hat{X}_1, \dots, \hat{X}_m, \hat{Y}_1, \dots, \hat{Y}_\ell, \hat{Z}]$. (In all our instantiations, we will have $m \in \{1, 2\}$ and $\ell \in \{0, 1\}$.) We require that the total degree of W in the variables $\hat{Y}_1, \dots, \hat{Y}_\ell, \hat{Z}$ is at most one. Given an instance $g^{x_i} \in \mathbb{G}^m$, we say that $(g^y, g^{z_i}) \in \mathbb{G}^\ell \times \mathbb{G}$ is a solution to the instance if $W(x_i, y, z_i) = 0$ and $y \neq 0$. For integers $k \leq n$, we define the (n, k) -MI_W problem as follows. The adversary is given n instances $g^{x_1}, \dots, g^{x_n} \in \mathbb{G}^m$ and must identify at least k distinct indices i , together with a common g^y and corresponding g^{z_i} 's, such that (g^y, g^{z_i}) is a solution to g^{x_i} for each index i .

To further generalize our framework, we allow the adversary access to a discrete log (dlog) oracle, which on input g^u returns u . An adversary that makes sufficiently many dlog queries will typically be able to win easily. Rather than explicitly bounding the number of such queries as part of the assumption, we capture their impact in the advantage bound, which depends on the number of dlog oracle calls made by the adversary.

⁴ However, a randomized-generator variant of IDH is tightly equivalent to SDH. The randomized-generator variant of IDH is: given random g and g^x compute $g^{\frac{1}{x}}$. This is the same as SDH: given g^x and $g = (g^x)^{1/x}$, compute $(g^x)^{1/x^2} = g^{1/x}$.

Definition 1 $((n, k)\text{-}\text{MI}_W)$. To an adversary $\mathcal{A}$, a multi-instance winning polynomial W , and integers $1 \leq k \leq n$ we associate the (n, k) -multi-instance advantage function

$$\text{Adv}^{(n, k)\text{-}\text{MI}_W}(\mathcal{A}) := \Pr[(n, k)\text{-}\text{MI}_W \Rightarrow 1],$$

where game $(n, k)\text{-}\text{MI}_W$ is shown in [Figure 2](#). Note that in [Line 03](#) the game checks whether for every $i \in \mathcal{I}$, $W(x_i, y, z_i) = 0$, where the variables $\hat{X} = (\hat{X}_1, \dots, \hat{X}_m)$ of the winning polynomial are instantiated with $x_i = (x_{i,1}, \dots, x_{i,m}) \in \mathbb{Z}_p^m$, the variables $\hat{Y} = (\hat{Y}_1, \dots, \hat{Y}_\ell)$ are instantiated with $y \in \mathbb{Z}_p^\ell$, and the variable $\hat{Z}$ is instantiated with $z_i \in \mathbb{Z}_p$. By the restriction on the degree of W this check can be done efficiently.

Let $(n, k)\text{-}\text{MI}_W^*$ be the $(n, k)\text{-}\text{MI}_W$ problem with zero queries to the DL oracle. In general, we are interested in W such that solving $(n, k)\text{-}\text{MI}_W^*$ with constant probability requires $\approx \sqrt{kp}$ many steps. That is, the running time required to solve k instances is that of solving k instances of the discrete logarithm problem. Consequently, with at most s calls to the discrete logarithm oracle, we therefore expect to require $\approx \sqrt{(k-s)p}$ steps to solve the $(n, k)\text{-}\text{MI}_W$ problem.

Game $(n, k)\text{-}\text{MI}_W$	Oracle $\text{DL}(g^u)$
00 for $i \in [n] : x_i \xleftarrow{\$} \mathbb{Z}_p^m$	04 return u
01 $(\mathcal{I}, g^y, g^z) \leftarrow \mathcal{A}^{\text{DL}(\cdot)}((g^{x_i})_{i \in [n]})$	
02 parse $y = (y_1, \dots, y_\ell) \in \mathbb{Z}_p^\ell, z = (z_1, \dots, z_n) \in \mathbb{Z}_p^n$	
03 return $\bigwedge_{i \in \mathcal{I}} [W(x_i, y, z_i) = 0] \wedge [y \neq 0] \wedge [\mathcal{I} \geq k]$	

Fig. 2. Game $(n, k)\text{-}\text{MI}_W$ for winning polynomial W , where the set $\mathcal{I} \subseteq [n]$ contains the indices of the correct solutions.

EXAMPLES OF MULTI-INSTANCE PROBLEMS. We now give some instantiations of the MI_W problem in [Table 1](#), including multi-instance versions of the Computational Diffie-Hellman (CDH), Inverse Diffie-Hellman (IDH), and Square Diffie-Hellman (SDH) problem. We define the resulting problems as $(n, k)\text{-}\text{CDH}$, $(n, k)\text{-}\text{SDH}$, and $(n, k)\text{-}\text{IDH}$. The $(n, n)\text{-}\text{LKDH}$ problem is the $\mathcal{L}_n$ -Kernel Diffie-Hellman problem from [\[MRV16\]](#), where $\mathcal{L}_n$ is the linear Matrix Diffie-Hellman assumption [\[EHK⁺13\]](#). Here we generalize it to the k -out-of- n variant $(n, k)\text{-}\text{LKDH}$. It can also be seen as a generalization of the SimDH problem (defined above) to the multi-instance setting. As examples, the $(n, k)\text{-}\text{CDH}$ and $(n, k)\text{-}\text{LKDH}$ problems are depicted in [Figures 5 and 6](#) of [Appendix D](#).

3.2 Generic Group Model

The Generic Group Model (GGM) [\[Sho97, Mau05\]](#) is an idealized framework in which group elements are represented by random labels. From the adversary's point of view, these labels hide the concrete representation of the group elements. Thus, the adversary cannot exploit any special structure of the group representation and can only use the group operation through an oracle.

Let $\mathcal{L}$ be a set of labels with $|\mathcal{L}| \geq p$ and let $\sigma : \mathbb{Z}_p \rightarrow \mathcal{L}$ be a random injective labeling function. We identify each group element $g^x \in \mathbb{G}$ with its label $\sigma(x) \in \mathcal{L}$. As

Name	m	ℓ	Winning polynomial $W(\hat{X}_1, \dots, \hat{X}_m, \hat{Y}_1, \dots, \hat{Y}_\ell, \hat{Z})$
(n, k) -CDH [AGK20]	2	0	$\hat{X}_1 \hat{X}_2 - \hat{Z}$
(n, k) -SDH	1	0	$\hat{X}_1^2 - \hat{Z}$
(n, k) -IDH	1	0	$\hat{Z} \hat{X}_1 - 1$
(n, k) -LKDH [MRV16]	1	1	$\hat{X}_1 \hat{Y}_1 - \hat{Z}$

Table 1. Multi-instance winning polynomials W defining the multi-instance problems (n, k) -CDH, (n, k) -SDH, (n, k) -IDH, and (n, k) -LKDH via Definition 1.

such, we call $x \in \mathbb{Z}_p$ the discrete logarithm of the label $\sigma(x)$. The generic group oracle

$$\mathcal{O}_\sigma : \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L} \cup \{\perp\}$$

takes two labels $X, Y \in \mathcal{L}$ as input and responds as follows:

- If there exist $x, y \in \mathbb{Z}_p$ such that $X = \sigma(x)$ and $Y = \sigma(y)$, then the oracle returns $\sigma(x + y \bmod p)$.
- If there do not exist such $x, y \in \mathbb{Z}_p$, then the oracle returns $\perp$.

A generic algorithm over the label set $\mathcal{L}$ for $\mathbb{Z}_p$ is a probabilistic algorithm $\mathcal{A}$ which receives as input a list of labels $\sigma(x_1), \dots, \sigma(x_\ell)$ and has oracle access to the generic group oracle $\mathcal{O}_\sigma$. The query complexity of $\mathcal{A}$, denoted by Q , is the total number of oracle calls it makes.

With this definition, we can naturally instantiate the (n, k) -MI $_W$ and the (n, k) -MI $_W^{\text{DL}}$ problems in the generic group model, where all group elements are exchanged as labels. Note that the discrete-logarithm oracle DL receives a label $\sigma(x)$ and returns the corresponding exponent x . Since this acts as inverse of the labeling function σ , we denote this oracle by $\mathcal{O}_{\sigma^{-1}}$.

4 The Hyperplane Query Model

An affine hyperplane of dimension r ,

$$H_f := \{x \in \mathbb{Z}_p^r \mid f(x) = 0\},$$

is defined by an affine function $f : \mathbb{Z}_p^r \rightarrow \mathbb{Z}_p$, where $f(X) = \sum_{i \in [r]} \alpha_i X_i + \alpha_0$. $X_1, \dots, X_r$ are the canonical coordinates of $\mathbb{Z}_p^r$.

Yun’s hyperplane query model [Yun15, ABG⁺24] is an information-theoretic framework that allows for clean lower-bound proof in the generic group model. In our adaptation of the model, the experiment samples a random secret vector $x = (x_1, \dots, x_n)$, where each component $x_i = (x_{i,1}, \dots, x_{i,m}) \in \mathbb{Z}_p^m$. The adversary is granted access to a hyperplane membership oracle $H_x(H_f)$, where H_f is an nm -dimensional hyperplane defined by the affine function $f(X) = \sum_{i,j} \alpha_{i,j} X_{i,j} + \alpha_0$. On input H_f , the oracle returns $\llbracket x \in H_f \rrbracket$, that is, 1 if and only if $x \in H_f$, or equivalently, if $f(x) = 0$. In addition, the adversary has access to a full evaluation oracle $\text{EVAL}_x(H_f)$, which, on input a hyperplane H_f , returns the value $\beta := f(x)$. Finally, the adversary outputs a list of polynomials in the indeterminates $X_{i,j}$ and succeeds if these polynomials,

evaluated at the secret vector x , satisfy a prescribed winning condition. In our setting, this condition is defined by the multi-instance winning polynomial W .

We consider information-theoretic adversaries and measure their complexity both in terms of the total number q of oracle queries and the number $s \ll q$ of queries made specifically to the evaluation oracle EVAL_x .

Definition 2 ((n, k) -MI-HQM $_W$). *Let W be a winning polynomial and let $1 \leq k \leq n$ be integers. Then for any hyperplane query algorithm $\mathcal{B}$, we define the (n, k) -multi-instance-HQM advantage function*

$$\text{Adv}^{(n, k)\text{-MI-HQM}_W}(\mathcal{B}) := \Pr[(n, k)\text{-MI-HQM}_W \Rightarrow 1]$$

where the game (n, k) -MI-HQM $_W$ is shown in Figure 3. The adversary returns the index set $\mathcal{I}$, together with polynomials $Y = (Y_1, \dots, Y_\ell)$, $Z = (Z_1, \dots, Z_n)$, where $Y_i, Z_i \in \mathbb{Z}_p[X_{1,1}, \dots, X_{n,m}]$. For every $i \in \mathcal{I}$, the game checks whether $W(x_i, Y(x), Z_i(x)) = 0$, where variables $\hat{X} = (\hat{X}_1, \dots, \hat{X}_m)$ of the winning polynomial are instantiated with $x_i = (x_{i,1}, \dots, x_{i,m}) \in \mathbb{Z}_p^m$, the variables $\hat{Y} = (\hat{Y}_1, \dots, \hat{Y}_\ell)$ are instantiated with $Y(x) \in \mathbb{Z}_p^\ell$, and the variable $\hat{Z}$ is instantiated with $Z_i(x) \in \mathbb{Z}_p$.

Game (n, k) -MI-HQM $_W$	Oracle $H_x(H_f)$
00 for $i \in [n] : x_i \xleftarrow{\$} \mathbb{Z}_p^m$	04 return $\llbracket x \in H_f \rrbracket$
01 $x = (x_1, \dots, x_n)$	
02 $(\mathcal{I}, Y, Z) \leftarrow \mathcal{B}^{H_x, \text{EVAL}_x}$	Oracle $\text{EVAL}_x(H_f)$
03 return $\bigwedge_{i \in \mathcal{I}} \llbracket W(x_i, Y(x), Z_i(x)) = 0 \rrbracket$	05 Parse $f(X) = \sum \alpha_{i,j} X_{i,j} + \alpha_0$
$\wedge \llbracket \mathcal{I} \geq k \rrbracket \wedge \llbracket Y(x) \neq 0 \rrbracket$	06 $\beta := f(x) \in \mathbb{Z}_p$
	07 return β

Fig. 3. Game (n, k) -MI-HQM $_W$ for winning polynomial W , where the set $\mathcal{I} \subseteq [n]$ contains the indices of the correct solutions.

Example instantiations of this definition are given in Figures 5 and 6 of Appendix D.

We now show that (n, k) -MI-HQM $_W$ implies (n, k) -MI $_W$ in the generic group model, where Q generic group queries can be simulated by $O(Q^2)$ hyperplane queries and s discrete logarithm queries can be simulated by s evaluation queries. The main idea is as follows: the reduction associates every query to the GGM oracle to a linear polynomial in the input. Two polynomials P_1, P_2 receive the same label, if $P_1(x) = P_2(x)$ for the secret point x , which the reduction can check by querying $H_x(H_{P_1-P_2})$ to its HQM oracle. For simplicity of notation, from here on we overload H to denote both the hyperplane and the associated affine function.

Theorem 1 (GGM to HQM). *Let W be a multi-instance winning polynomial, and let $k \leq n$ be integers. For any generic algorithm $\mathcal{A}$ that makes at most Q oracle queries (including at most s queries to the discrete-logarithm oracle), there exists a hyperplane query algorithm $\mathcal{B}$ that makes a total of at most $2 \cdot (3Q + mn + \ell + n + 1)^2$ queries to H_x (and EVAL_x with at most s queries to EVAL_x) such that*

$$\text{Adv}^{(n, k)\text{-MI-HQM}_W}(\mathcal{B}) = \text{Adv}^{(n, k)\text{-MI}_W}(\mathcal{A}).$$

We note that the proof follows closely the proof of Lemma 2 in [ABG⁺24]. We essentially need to show that $\mathcal{B}$ can additionally simulate the discrete-logarithms queries of $\mathcal{A}$ by querying its own evaluation oracle EVAL_x without causing any inconsistency. Hence, we defer the proof of Theorem 1 to Appendix A.

5 General Bounds in the Hyperplane Query Model

We define the best possible advantage of winning the (n, k) -MI-HQM_W game for an adversary making at most q queries (s of which to EVAL_x) as

$$\text{Adv}^W(n, k, q, s) := \max_{\mathcal{B} \in \mathbf{A}(n, k, q, s)} \text{Adv}^{(n, k)\text{-MI-HQM}_W}(\mathcal{B}),$$

where $\mathbf{A}(n, k, q, s)$ is the set of all adversaries running in game (n, k) -MI-HQM_W and making at most q queries (s of which to EVAL_x). The special case of $\text{Adv}^W(n, k, 0, 0)$ is called the zero-query advantage, which will play an important role in deriving our bounds. For simplicity of notation to avoid case distinctions in the induction we define $\text{Adv}^W(n, k, q, s) = 0$, when $\min\{q, s\} < 0$.

5.1 Relation to zero-query advantage

The following theorem relates the advantage $\text{Adv}^{(n, k)\text{-MI-HQM}_W}(\mathcal{A})$ of an adversary $\mathcal{A}$ making s evaluation queries to the weighted sum of zero-query advantages $\text{Adv}(n', k', 0, 0)$ for $n' \leq n - s$ and $k' \leq k - s$.

Theorem 2. *Let $1 \leq k \leq n$ be integers and W be a winning polynomial. Then, for any adversary $\mathcal{A}$ making at most q queries (s of which to EVAL_x) we have*

$$\text{Adv}^{(n, k)\text{-MI-HQM}_W}(\mathcal{A}) \leq \sum_{i=0}^{k-s} \binom{q}{i} \left(\frac{2}{p}\right)^i \text{Adv}(n - i - s, k - i - s, 0, 0).$$

The adversary is allowed to make either a hyperplane query or an evaluation query. To prove this theorem, we make a distinction between whether the first query is a hyperplane query or an evaluation query. Then for the case when adversary makes a hyperplane query, its response can be either 1 or 0.

- Intuitively, if the adversary makes a hyperplane query H_x that returns 1, the dimension of the hidden point x is reduced by at most one. The adversary learns one equation about x , namely $f(x) = 0$, for some affine function f . After this query, there exist still $n - 1$ degrees of freedom in x .
- If H_x returns 0 then the information learned by the adversary is also bounded by essentially one query less or the amount it learns with a hyperplane query that returns 1.

The adversary is also allowed to make evaluation queries. An evaluation query may reduce the degrees of freedom in x by at most one, while the analysis of a hyperplane query is more involved. We have the following core lemma which we prove in Appendix B.

Lemma 1. *For any winning polynomial W ,*

$$\text{Adv}^W(n, k, q, s) \leq \max \left\{ \text{Adv}^W(n, k, q - 1, s) + \frac{2}{p} \text{Adv}^W(n - 1, k - 1, q - 1, s), \text{Adv}^W(n - 1, k - 1, q - 1, s - 1) \right\}.$$

Proof (of Theorem 2). We prove

$$\text{Adv}(n, k, q, s) \leq \sum_{i=0}^{k-s} \binom{q}{i} \left(\frac{2}{p}\right)^i \text{Adv}(n-i-s, k-i-s, 0, 0)$$

by induction over $k+q$. The base cases for $k=0$ and $q=0$ are trivially true. Now we imagine the inequality holds for all (k', q') such that $q' + k' < q + k$. By Lemma 1 we have

$$\text{Adv}(n, k, q, s) \leq \max \left\{ \begin{array}{l} \text{Adv}(n, k, q-1, s) + \frac{2}{p} \text{Adv}(n-1, k-1, q-1, s), \\ \text{Adv}(n-1, k-1, q-1, s-1) \end{array} \right\}.$$

Combining this with our induction step gives us

$$\begin{aligned} \text{Adv}(n, k, q, s) &\leq \max \left\{ \begin{array}{l} \sum_{i=0}^{k-s} \binom{q-1}{i} \left(\frac{2}{p}\right)^i \text{Adv}(n-i-s, k-i-s, 0, 0) \\ + \frac{2}{p} \sum_{i=0}^{k-s-1} \binom{q-1}{i} \left(\frac{2}{p}\right)^i \text{Adv}(n-i-s, k-i-s-1, 0, 0), \\ \sum_{i=0}^{k-s} \binom{q-1}{i} \left(\frac{2}{p}\right)^i \text{Adv}(n-i-s, k-i-s, 0, 0) \end{array} \right\} \\ &\leq \sum_{i=0}^{k-s} \binom{q-1}{i} \left(\frac{2}{p}\right)^i \text{Adv}(n-i-s, k-i-s, 0, 0) \\ &\quad + \frac{2}{p} \sum_{i=0}^{k-s-1} \binom{q-1}{i} \left(\frac{2}{p}\right)^i \text{Adv}(n-i-s, k-i-s-1, 0, 0) \\ &= \sum_{i=0}^{k-s} \binom{q-1}{i} \left(\frac{2}{p}\right)^i \text{Adv}(n-i-s, k-i-s, 0, 0) \\ &\quad + \sum_{i=1}^{k-s} \binom{q-1}{i-1} \left(\frac{2}{p}\right)^i \text{Adv}(n-i-s-1, k-i-s, 0, 0) \\ &\leq \sum_{i=0}^{k-s} \binom{q-1}{i} \left(\frac{2}{p}\right)^i \text{Adv}(n-i-s, k-i-s, 0, 0) \\ &\quad + \sum_{i=1}^{k-s} \binom{q-1}{i-1} \left(\frac{2}{p}\right)^i \text{Adv}(n-i-s, k-i-s, 0, 0) \\ &= \sum_{i=1}^{k-s} \left(\frac{2}{p}\right)^i \text{Adv}(n-i-s, k-i-s, 0, 0) \cdot \left[\binom{q-1}{i} + \binom{q-1}{i-1} \right] \\ &\quad + \text{Adv}(n-s, k-s, 0, 0) \\ &= \sum_{i=1}^{k-s} \left(\frac{2}{p}\right)^i \binom{q}{i} \text{Adv}(n-i-s, k-i-s, 0, 0) + \text{Adv}(n-s, k-s, 0, 0) \\ &= \sum_{i=0}^{k-s} \left(\frac{2}{p}\right)^i \binom{q}{i} \text{Adv}(n-i-s, k-i-s, 0, 0). \end{aligned}$$

This proves [Theorem 2](#). □

5.2 Useful bounds for applying [Theorem 2](#)

We now prove a simple technical lemma that helps in the application of [Theorem 2](#).

Lemma 2. *If there exists constants $0 \leq \theta \leq 1$ and $c \geq 1$ such that for every $k \in [n]$, $\text{Adv}(n, k, 0, 0) \leq \left(\frac{c}{p^\theta}\right)^k$. Then,*

$$\text{Adv}(n, k, q, s) \leq \begin{cases} \left(\frac{e^2 q}{(k-s)p}\right)^{k-s} & \text{if } q \geq c(k-s)p^{1-\theta} \\ \left(\frac{ce^2}{p^\theta}\right)^{k-s} & \text{otherwise.} \end{cases}$$

In particular, for $\theta = 1$ and $q \geq c(k-s)$ we have $\text{Adv}(n, k, q, s) \leq \left(\frac{ce^2 q}{(k-s)p}\right)^{k-s}$.

Proof. By [Theorem 2](#) we have

$$\text{Adv}(n, k, q, s) \leq \sum_{i=0}^{k-s} \binom{q}{i} \left(\frac{2}{p}\right)^i \text{Adv}(n-i-s, k-i-s, 0, 0).$$

Therefore by plugging in $\text{Adv}(n-i-s, k-i-s, 0, 0) \leq \left(\frac{c}{p^\theta}\right)^{k-i-s}$ we get

$$\text{Adv}(n, k, q, s) \leq (cp^{-\theta})^{k-s} \cdot \sum_{i=0}^{k-s} \binom{q}{i} (2c^{-1}p^{\theta-1})^i.$$

Assuming $q \geq c(k-s)p^{1-\theta}$, we have

$$\begin{aligned} \text{Adv}(n, k, q, s) &= \left(\frac{q}{(k-s)p}\right)^{k-s} \left(\frac{c(k-s)p^{1-\theta}}{q}\right)^{k-s} \cdot \sum_{i=0}^{k-s} \binom{q}{i} (2c^{-1}p^{\theta-1})^i \\ &\leq \left(\frac{q}{(k-s)p}\right)^{k-s} \cdot \sum_{i=0}^{k-s} \binom{q}{i} (2c^{-1}p^{\theta-1})^i \left(\frac{c(k-s)p^{1-\theta}}{q}\right)^i \end{aligned}$$

since $c(k-s)p^{1-\theta}/q \leq 1$. We continue with

$$\begin{aligned} &= \left(\frac{q}{(k-s)p}\right)^{k-s} \cdot \sum_{i=0}^q \binom{q}{i} \left(\frac{2k-2s}{q}\right)^i \\ &= \left(\frac{q}{(k-s)p}\right)^{k-s} \cdot \left(1 + \frac{2k-2s}{q}\right)^q \\ &\leq \left(\frac{q}{(k-s)p}\right)^{k-s} \cdot e^{\frac{2k-2s}{q} \cdot q} = \left(\frac{q}{(k-s)p}\right)^{k-s} \cdot e^{2(k-s)}. \end{aligned}$$

Now, assume $q \leq c(k-s)p^{1-\theta}$.

$$\begin{aligned} \text{Adv}(n, k, q, s) &\leq (cp^{-\theta})^{k-s} \sum_{i=0}^q \binom{q}{i} (2c^{-1}p^{\theta-1})^i \\ &= (cp^{-\theta})^{k-s} \left(1 + \frac{2}{cp^{1-\theta}}\right)^q \leq (cp^{-\theta})^{k-s} e^{2q/cp^{1-\theta}}. \end{aligned}$$

Combining this bound with $q \leq c(k-s)p^{1-\theta}$ yields the claimed bound. $\square$

6 Zero-query advantage

In this section, we derive upper bounds on the zero-query advantage $\text{Adv}^W(n, k, 0, 0)$. We first establish a general theorem, based on techniques from algebraic geometry, that upper bounds this advantage for arbitrary winning polynomials. We then apply this theorem to the winning polynomials corresponding to the Square Diffie-Hellman, Computational Diffie-Hellman, and Inverse Diffie-Hellman problems.

In the zero-query setting, the adversary is reduced to a program that simply computes an output that is statistically independent of the secret x . Concretely, the adversary outputs an index set $\mathcal{I}$ of size $|\mathcal{I}| = k$ and polynomials $Y_1(X), \dots, Y_\ell(X)$ and $Z_1(X), \dots, Z_n(X)$. The adversary wins if the winning polynomial evaluated on the random x evaluates to zero for all $i \in \mathcal{I}$. Formally we want to upper bound

$$\varepsilon_W^{Y, Z, \mathcal{I}} := \Pr_{x \leftarrow \mathbb{F}_p^{nm}} [\forall i \in \mathcal{I} : W(x_{i,1}, \dots, x_{i,m}, Y_1(x), \dots, Y_\ell(x), Z_i(x)) = 0]. \quad (3)$$

where $x = (x_{1,1}, \dots, x_{n,m})$, $Y = (Y_1, \dots, Y_\ell)$ and $Z = (Z_1, \dots, Z_n)$.

While the win condition is easy to define, obtaining an upper bound of Equation (3) requires tools from algebraic geometry. We assume the reader is familiar with basic concepts of algebraic geometry, such as degree and Krull dimension. If not, we refer the reader to a textbook such as [Har77, CLO25]. We recall the required theorems in Appendix C. We use the degree convention from [Hei83] that includes the degree of lower-dimension components. When K is an algebraically closed field we can define a variety as the set of common roots for polynomials $P_1, \dots, P_t \in K[X_1, \dots, X_r]$. Formally,

$$V(P_1, \dots, P_t) := \{x \in K^r \mid P_1(x) = \dots = P_t(x) = 0\}.$$

For an ideal $I \subseteq K[X_1, \dots, X_r]$ we can extend this definition to $V(I)$ for the common set of all roots of I over K^r . Since varieties are defined over an algebraically closed field, we define them over $\overline{\mathbb{F}}_p$, the algebraic closure of $\mathbb{F}_p$. We write $X := (X_{1,1}, \dots, X_{n,m})$. We then define the (affine) variety over $\overline{\mathbb{F}}_p^{nm}$ as

$$V_W^{Y, Z, \mathcal{I}} := V\left((W(X_{i,1}, \dots, X_{i,m}, Y_1(X), \dots, Y_\ell(X), Z_i(X)))_{i \in \mathcal{I}}\right).$$

Note that the winning probability (3) can be rewritten as

$$\varepsilon_W^{Y, Z, \mathcal{I}} = \Pr_{x \in \overline{\mathbb{F}}_p^{nm}} \left[x \in V_W^{Y, Z, \mathcal{I}} \right].$$

We show how to approximate this probability by upper bounding the (Krull) dimension of this variety $V_W^{Y, Z, \mathcal{I}}$.

Theorem 3. Let $W \in \mathbb{F}_p[\hat{X}_{i,1}, \dots, \hat{X}_{i,m}, \hat{Y}, \hat{Z}_i]$ be a winning polynomial as defined in Section 3.1. Assume that for all linear polynomials

$$Y_1(X), \dots, Y_\ell(X), Z_1(X), \dots, Z_n(X) \in \mathbb{F}_p[X_{1,1}, \dots, X_{n,m}]$$

and all index sets $\mathcal{I} \subseteq [n]$ of size $|\mathcal{I}| = k$, we have

$$\dim V_W^{Y,Z,\mathcal{I}} \leq d,$$

where $\dim$ is the Krull dimension of the variety. Then

$$\text{Adv}^W(n, k, 0, 0) \leq \frac{\deg(W)^k \cdot p^d}{p^{nm}}.$$

Proof. This statement is a consequence of the Lang–Weil-style bound (Theorem 6). Without loss of generality, the adversary $\mathcal{A}$ is deterministic. Then $\mathcal{A}$ outputs linear polynomials

$$Y_1(X), \dots, Y_\ell(X), Z_1(X), \dots, Z_n(X) \in \mathbb{F}_p[X_{1,1}, \dots, X_{n,m}],$$

and an index set $\mathcal{I}$, which are independent of the secret point x as $\mathcal{A}$ makes no queries. Therefore $V_W^{Y,Z,\mathcal{I}}$ is well-defined. For all i , the expression

$$W(X_{i,1}, \dots, X_{i,m}, Y_1(X), \dots, Y_\ell(X), Z_i(X)) \in \mathbb{F}_p[X_{1,1}, \dots, X_{n,m}]$$

is a polynomial of degree at most $\deg(W)$. The total number of solutions of the system of equations is bounded as follows

$$|V_W^{Y,Z,\mathcal{I}} \cap \mathbb{F}_p^{nm}| \leq \deg(W)^k \cdot p^d,$$

by Theorem 6. Furthermore, the total number of secret points $x := (x_{1,1}, \dots, x_{n,m}) \in \mathbb{F}_p^{nm}$ is p^{nm} . The adversary wins when $x \in V_W^{Y,Z,\mathcal{I}}$, giving

$$\text{Adv}^W(n, k, 0, 0) = \max_{Y,Z,\mathcal{I}} \Pr_x [x \in V_W^{Y,Z,\mathcal{I}}] \leq \frac{\deg(W)^k \cdot p^d}{p^{nm}}$$

as required. $\square$

To show the hardness of a multi-instance problem, we need to upper bound the dimension of the variety $V_W^{Y,Z,\mathcal{I}}$ for arbitrary linear $Y_1(X), \dots, Y_\ell(X), Z_1(X), \dots, Z_n(X) \in \mathbb{F}_p[X_{1,1}, \dots, X_{n,m}]$. We will show two different techniques to upper bound this dimension.

6.1 (n, k) -SDH via Gröbner bases

In this subsection we establish the zero-query advantage of Square Diffie-Hellman. Let $W_{\text{SDH}}(\hat{X}_1, \hat{Z}) = \hat{X}_1^2 - \hat{Z}$ denote the winning polynomial for Square Diffie-Hellman, cf. Table 1. The main result is the following lemma.

Lemma 3. For all $n \geq k \geq 1$, $\text{Adv}^{W_{\text{SDH}}}(n, k, 0, 0) \leq (2/p)^k$.

Proof. In order to prove this lemma, we need to upper bound the Krull dimension of the variety $V_{Z, \mathcal{I}}^{W_{\text{SDH}}}$ for arbitrary $Z, \mathcal{I}$. We will show that

$$\dim V_{Z, \mathcal{I}}^{W_{\text{SDH}}} \leq n - k.$$

The upper bound on $\text{Adv}^{W_{\text{SDH}}}(n, k, 0, 0)$ now follows by [Theorem 3](#). We postpone the upper bound on this dimension to [Lemma 5](#). $\square$

To simplify notation write $X_i = X_{i,1}$ during this section. We fix a subset $\mathcal{I} \subseteq [n]$ with $|\mathcal{I}| = k$ and consider, for $i \in \mathcal{I}$, the evaluations of the winning polynomial

$$g_i(X) := X_i^2 - Z_i(X) \in \mathbb{F}_p[X_1, \dots, X_n].$$

Define the ideal over the algebraic closure

$$I_{\text{SDH}} := \langle g_i \mid i \in \mathcal{I} \rangle \subseteq \overline{\mathbb{F}}_p[X_1, X_2, \dots, X_n]$$

so that $V_{Z, \mathcal{I}}^{W_{\text{SDH}}} = V(I_{\text{SDH}})$. The argument now has two steps. First, we prove that the natural generators

$$G_{\text{SDH}} = \{g_i \mid i \in \mathcal{I}\} \tag{4}$$

form a Gröbner basis of I_{SDH} , in the sense of [\[CLO25, Def. 2.5.5\]](#). Once this fact is established, the leading term ideal, as defined in [\[CLO25, Def. 2.5.1\]](#), preserves the Krull dimension so we can argue

$$\dim V_{Z, \mathcal{I}}^{W_{\text{SDH}}} = \dim V(\langle \text{LT}(I_{\text{SDH}}) \rangle) = \dim V((X_i^2)_{i \in \mathcal{I}}),$$

where LT denotes the leading term with respect to the fixed monomial order [\[CLO25, Def. 2.2.7\]](#). Throughout this subsection, we work with the graded lexicographic order $>_{\text{grlex}}$ of [\[CLO25, Def. 2.2.5\]](#).

Lemma 4. *The set $G_{\text{SDH}} = \{g_i \mid i \in \mathcal{I}\}$ is a Gröbner basis of I_{SDH} with respect to the graded lexicographic order $>_{\text{grlex}}$. Moreover,*

$$\langle \text{LT}(I_{\text{SDH}}) \rangle = \langle X_i^2 \mid i \in \mathcal{I} \rangle.$$

Proof (of [Lemma 4](#)). Since each $Z_i(X)$ is linear and $>_{\text{grlex}}$ is a graded monomial order, the quadratic term X_i^2 has larger total degree than every term of $Z_i(X)$. Hence, for every $i \in \mathcal{I}$,

$$\text{LT}(g_i) = X_i^2.$$

We now prove that G_{SDH} is a Gröbner basis of I_{SDH} . By Buchberger's criterion from [Theorem 5](#), it suffices to show that $S(g_i, g_j)$ has remainder zero on division by G_{SDH} for every pair of distinct indices $i, j \in \mathcal{I}$. Here the S -polynomial [\[CLO25, Def. 2.6.4\]](#) is defined as

$$S(g_i, g_j) := \frac{\text{lcm}(\text{LM}(g_i), \text{LM}(g_j))}{\text{LT}(g_i)} g_i - \frac{\text{lcm}(\text{LM}(g_i), \text{LM}(g_j))}{\text{LT}(g_j)} g_j.$$

Here LM denotes the leading monomial and lcm denotes the least common multiple of leading monomials [\[CLO25, Def. 2.2.7, Def. 2.6.4\]](#). The remainder is taken with respect to the multivariate division algorithm [\[CLO25, Thm. 2.3.3\]](#). Fix distinct $i, j \in \mathcal{I}$. Since $\text{LT}(g_i) = X_i^2$, and $\text{LT}(g_j) = X_j^2$, we have

$$\text{lcm}(\text{LM}(g_i), \text{LM}(g_j)) = X_i^2 X_j^2.$$

Therefore, by the definition of the S -polynomial,

$$S(g_i, g_j) = X_j^2 g_i - X_i^2 g_j.$$

Substituting $g_i = X_i^2 - Z_i(X)$ and $g_j = X_j^2 - Z_j(X)$, we obtain

$$\begin{aligned} S(g_i, g_j) &= X_j^2(X_i^2 - Z_i(X)) - X_i^2(X_j^2 - Z_j(X)) \\ &= X_i^2 Z_j(X) - X_j^2 Z_i(X). \end{aligned}$$

We claim that this polynomial has remainder zero on division by G_{SDH} . Indeed, we have

$$\begin{aligned} Z_j(X)g_i - Z_i(X)g_j &= Z_j(X)(X_i^2 - Z_i(X)) - Z_i(X)(X_j^2 - Z_j(X)) \\ &= X_i^2 Z_j(X) - X_j^2 Z_i(X) \\ &= S(g_i, g_j). \end{aligned}$$

As $Z_i(X)$ and $Z_j(X)$ are linear, this shows $S(g_i, g_j)$ has remainder zero on division by G_{SDH} . Since this holds for every pair of distinct indices $i, j \in \mathcal{I}$, Buchberger's criterion from [Theorem 5](#) implies that G_{SDH} is a Gröbner basis of I_{SDH} . Finally, because G_{SDH} is a Gröbner basis and $\text{LT}(g_i) = X_i^2$ for every $i \in \mathcal{I}$, we get

$$\langle \text{LT}(I_{\text{SDH}}) \rangle = \langle \text{LT}(g_i) \mid i \in \mathcal{I} \rangle = \langle X_i^2 \mid i \in \mathcal{I} \rangle.$$

This proves the lemma. $\square$

Lemma 5. *Let $\mathcal{I} \subseteq [n]$ with $|\mathcal{I}| = k$. Then the variety $V(I_{\text{SDH}})$ has dimension $n - k$.*

Proof. By [Lemma 4](#), we have

$$\langle \text{LT}(I_{\text{SDH}}) \rangle = \langle X_i^2 \mid i \in \mathcal{I} \rangle.$$

We first compute the dimension of the variety of this monomial ideal. For each generator $m_i = X_i^2$ with $i \in \mathcal{I}$, the corresponding set M_i from [Proposition 1](#) is

$$M_i = \{i\}.$$

Thus, by the definition of $\mathcal{M}$ in [Proposition 1](#), a subset $J \subseteq [n]$ belongs to $\mathcal{M}$ precisely when

$$J \cap \{i\} \neq \emptyset \quad \text{for every } i \in \mathcal{I}.$$

But $J \cap \{i\} \neq \emptyset$ holds if and only if $i \in J$. Hence the above condition is equivalent to

$$\mathcal{I} \subseteq J.$$

Therefore, among all subsets $J \in \mathcal{M}$, the smallest possible choice is $J = \mathcal{I}$. Since $|\mathcal{I}| = k$, we get

$$\min\{|J| \mid J \in \mathcal{M}\} = k.$$

Applying [Proposition 1](#) to the monomial ideal $\langle X_i^2 \mid i \in \mathcal{I} \rangle \subset \overline{\mathbb{F}}_p[X_1, \dots, X_n]$, we obtain

$$\dim V(\langle \text{LT}(I_{\text{SDH}}) \rangle) = n - k.$$

Since the initial ideal has the same dimension as the original ideal, it follows that

$$\dim V(I_{\text{SDH}}) = \dim V(\langle \text{LT}(I_{\text{SDH}}) \rangle) = n - k.$$

This completes the proof. $\square$

6.2 (n, k) -CDH via Gröbner bases

Similar to the last subsection, we establish the zero-query advantage of Computational Diffie-Hellman. Let $W_{\text{CDH}}(\hat{X}_1, \hat{X}_2, \hat{Z}) = \hat{X}_1 \hat{X}_2 - \hat{Z}$ denote the winning polynomial for Computational Diffie-Hellman, cf. [Table 1](#).

Lemma 6. *For all $1 \leq k \leq n$, $\text{Adv}^{W_{\text{CDH}}}(n, k, 0, 0) \leq (2/p)^k$.*

The proof is analogous to the case of SDH and is therefore deferred to [Appendix E](#).

6.3 (n, k) -IDH via algebraic independence

When trying to bound the zero-query advantage for Inverse Diffie-Hellman, analysing the variety generated by the win polynomials is trickier than in the SDH/CDH cases. Surprisingly, the advantage for IDH is significantly higher than for SDH/CDH. Let $W_{\text{IDH}}(\hat{X}_1, \hat{Z}) = \hat{X}_1 \hat{Z} - 1$ denote the winning polynomial for Inverse Diffie-Hellman, cf. [Table 1](#).

Lemma 7. *For all $1 \leq k \leq n$, $\text{Adv}^{W_{\text{IDH}}}(n, k, 0, 0) \leq (4/p)^{\lceil k/2 \rceil}$.*

Proof. For arbitrary $Y_1(X), \dots, Y_\ell(X), Z_1(X), \dots, Z_n(X) \in \mathbb{F}_p[X_{1,1}, \dots, X_{n,1}]$, we will show in [Lemma 8](#) that

$$\dim V_{Z, \mathcal{I}}^{W_{\text{IDH}}} \leq n - k/2.$$

Since the dimension is an integer, this gives

$$\dim V_{Z, \mathcal{I}}^{W_{\text{IDH}}} \leq \lfloor n - k/2 \rfloor.$$

Then by [Theorem 3](#) we get

$$\text{Adv}^{W_{\text{IDH}}}(n, k, 0, 0) \leq \frac{\deg(W_{\text{IDH}})^k \cdot p^{\lfloor n - k/2 \rfloor}}{p^n} = \frac{4^{k/2}}{p^{\lceil k/2 \rceil}} \leq (4/p)^{\lceil k/2 \rceil},$$

which concludes the proof. $\square$

Again during this section we will write $X_i := X_{i,1}$ to simplify notation. To analyse the dimension $V_{Z, \mathcal{I}}^{W_{\text{IDH}}}$ we need the ideal

$$I_{\text{IDH}} = \langle X_i Z_i(X) - 1 \mid i \in \mathcal{I} \rangle \subseteq \overline{\mathbb{F}}_p[X_1, \dots, X_n].$$

However, the natural generators of I_{IDH} do not necessarily form a Gröbner basis for I_{IDH} . While it would be possible to compute a Gröbner basis, as we do for SDH/CDH, we give a different proof strategy that upper bounds the dimension of $V(I_{\text{IDH}})$.

Lemma 8. *Let $Z_1(X), \dots, Z_n(X) \in \mathbb{F}_p[X_1, \dots, X_n]_{\leq 1}$ be arbitrary linear polynomials and let $\mathcal{I} = \{s_1, \dots, s_k\}$ be an index set of size k . Define the ideal*

$$I_{\text{IDH}} = \langle Z_i(X) X_i - 1 \mid i \in \mathcal{I} \rangle,$$

and let $V_{\text{IDH}} = V(I_{\text{IDH}})$. Then $\dim V_{\text{IDH}} \leq n - k/2$.

Proof. Let $d := \dim V_{\text{IDH}}$. If $d \leq n - k$, then the theorem holds, as $n - k \leq n - k/2$. We may therefore assume $d > n - k$. By [Theorem 4](#), there exist d algebraically independent elements $X_{i_1}, \dots, X_{i_d} \in \mathbb{F}_p[V_{\text{IDH}}]$, indexed by the set $D := \{i_1, \dots, i_d\}$. Here $\mathbb{F}_p[V_{\text{IDH}}] := \mathbb{F}_p[X_1, \dots, X_n]/I(V_{\text{IDH}})$, the coordinate ring. Define the index set

$$T := \mathcal{I} \cap D = \{j_1, \dots, j_t\}.$$

Because $d > n - k$, we have $|T| \geq d - (n - k) > 0$. We will show that

$$R := (1, X_{i_1}, \dots, X_{i_d}, Z_{j_1}(X), \dots, Z_{j_t}(X))$$

are linearly independent in

$$W := \text{Span}_{\mathbb{F}_p}(1, X_1, \dots, X_n) \subseteq \mathbb{F}_p[V_{\text{IDH}}]$$

We will use the equality $Z_i(X)X_i = 1 \in \mathbb{F}_p[V_{\text{IDH}}]$. If a linear combination of elements in R is zero, then

$$\begin{aligned} & \sum_{i \in D} a_i X_i + \sum_{i \in T} b_i Z_i(X) + c = 0 \\ \implies & \sum_{i \in D} a_i X_i \prod_{j \in T} X_j + \sum_{i \in T} b_i \prod_{j \in T} X_j Z_i(X) + c \prod_{j \in T} X_j = 0 \quad (\text{by } \cdot \prod_{j \in T} X_j) \\ \implies & \sum_{i \in D} a_i X_i \prod_{j \in T} X_j + \sum_{i \in T} b_i \prod_{j \in T, i \neq j} X_j + c \prod_{j \in T} X_j = 0 \quad (\text{as } Z_i(X)X_i = 1) \end{aligned}$$

shows that all coefficients must be zero by algebraic independence of $X_{i_1}, \dots, X_{i_d}$. Note that $\dim W \leq n + 1$ as it is generated by $n + 1$ elements. Then finally,

$$d + (d - (n - k)) + 1 \leq |D| + |T| + 1 \leq \dim W \leq n + 1,$$

so $d \leq n - k/2$ as required. $\square$

7 Multi-Instance bounds in the Generic Group Model

7.1 (n, k) -CDH and (n, k) -SDH

Combining the bounds for the zero-query advantage from [Section 6.2](#) ([Section 6.1](#)) and [Lemma 2](#) (specifically the case where $\theta = 1$ and $c = 2$) gives us bounds for the advantage of (n, k) -CDH ((n, k) -SDH) in the hyperplane query model. We can then use [Theorem 1](#) to achieve bounds in the generic group model. This gives us for $P \in \{(n, k)\text{-CDH}, (n, k)\text{-SDH}\}$:

$$\text{Adv}^P(\mathcal{A}) \leq \left(\frac{2e^2 \cdot (3Q + 3n + 1)^2}{(k - s)p} \right)^{k-s} \approx \left(\frac{Q^2}{(k - s)p} \right)^{k-s},$$

for every adversary $\mathcal{A}$ in the generic group model against (n, k) -CDH ((n, k) -SDH) that makes at most Q queries to the group-operation oracle and s queries to the DL oracle. This bound is tight as it matches the known upper bound for solving k discrete logarithms [[Yun15](#)].

7.2 (n, k) -IDH

Combining the bounds for the zero-query advantage from [Section 6.3](#) and [Lemma 2](#) gives us bounds for the advantage of (n, k) -IDH in the hyperplane query model. We can then use [Theorem 1](#) to achieve bounds in the generic group model. This gives us

$$\text{Adv}^{(n,k)\text{-IDH}}(\mathcal{A}) \leq \begin{cases} \left(\frac{2e^2 \cdot (3Q+2n+1)^2}{(k-s)p} \right)^{k-s} & \text{if } Q \geq \frac{1}{3} \left((2k-2s)^{1/2} p^{1/4} \right) - 2n - 1, \\ \left(\frac{4e^2}{p} \right)^{\lceil \frac{k-s}{2} \rceil} & \text{otherwise,} \end{cases}$$

for any adversary $\mathcal{A}$ in the generic group model against (n, k) -IDH that makes at most Q and s queries to the group-operation and DL oracle, respectively. This bound is tight. As we have sketched in the introduction there exists an adversary $\mathcal{A}$ against (n, k) -IDH with small running time and advantage $\text{Adv}^{(n,k)\text{-IDH}}(\mathcal{A}) = 1/p^{\lceil k/2 \rceil}$.

7.3 (n, k) -LKDH

To achieve bounds for the (n, k) -LKDH problem we use the following lemma. Furthermore, this demonstrates that the DL oracle is a useful tool for establishing reductions between our multi-instance problems. In particular, it allows us to reduce the hardness of (n, k) -LKDH to that of (n, k) -IDH at the cost of a single additional query to the oracle DL.

Lemma 9. *For any adversary $\mathcal{A}$ against (n, k) -LKDH making s queries to the DL oracle, there exists an adversary $\mathcal{B}$ against (n, k) -IDH running in similar time making $s + 1$ queries to DL such that*

$$\text{Adv}^{(n,k)\text{-IDH}}(\mathcal{B}) = \text{Adv}^{(n,k)\text{-LKDH}}(\mathcal{A}).$$

Proof. Let $\mathcal{B}$ be as defined in [Figure 4](#), where $\mathcal{A}$'s queries to DL are simulated with $\mathcal{B}$'s own DL oracle. With $z_i = \log_g(Z_i)$ and $z'_i = \log_g(Z'_i) = z_i/y$, we have

$$\begin{aligned} \text{Adv}^{(n,k)\text{-LKDH}}(\mathcal{A}) &= \Pr \left[\bigwedge_{i \in \mathcal{I}} z_i x_i - y = 0 \wedge y \neq 0 \right] \\ &= \Pr \left[\bigwedge_{i \in \mathcal{I}} z'_i x_i - 1 = 0 \right] = \text{Adv}^{(n,k)\text{-IDH}}(\mathcal{B}). \end{aligned}$$

Furthermore, $\mathcal{B}$ makes $s + 1$ queries to its oracle DL. This concludes the proof. $\square$

This gives us

$$\text{Adv}^{(n,k)\text{-LKDH}}(\mathcal{A}) \leq \begin{cases} \left(\frac{2e^2 \cdot (3Q+2n+2)^2}{(k-s-1)p} \right)^{k-s-1} & \text{if } Q \geq \frac{1}{3} \left((2k-2s-2)^{1/2} p^{1/4} \right) - 2n - 2 \\ \left(\frac{4e^2}{p} \right)^{\lceil \frac{k-s-1}{2} \rceil} & \text{otherwise,} \end{cases}$$

for any adversary $\mathcal{A}$ in the generic group model against (n, k) -LKDH that makes at most Q and s queries to the group-operation and DL oracle, respectively.

$\mathcal{B}^{\text{DL}(\cdot)}((g^{x_i})_{i \in [n]})$ 00 $(\mathcal{I}, Y, (Z_i)_{i \in [n]}) \leftarrow \mathcal{A}^{\text{DL}(\cdot)}((g^{x_i})_{i \in [n]})$ 01 $y = \text{DL}(Y)$ 02 if $y = 0$: abort 03 for $i \in [n]$: $Z'_i := Z_i^{1/y}$ 04 return $(\mathcal{I}, (Z'_i)_{i \in [n]})$

Fig. 4. Adversary $\mathcal{B}$ in reduction from (n, k) -IDH to (n, k) -LKDH.

7.4 Concrete bit security

Following [BR09], a problem has κ bits of security if, for every adversary $\mathcal{A}$ with $\text{Time}(\mathcal{A}) \leq 2^\kappa$, it holds that $\text{SR}(\mathcal{A}) := \text{Adv}(\mathcal{A})/\text{Time}(\mathcal{A}) \leq 2^{-\kappa}$. We now compute the concrete level of bit security for our multi-instance problems.

Suppose that the Krull dimension of the algebraic variety is at most $nm - \theta k$, for some constant $0 \leq \theta \leq 1$. Combining the methods from Section 6 with Lemma 2 and Theorem 1, we obtain (ignoring constants) the generic group bound

$$\text{Adv}^P(\mathcal{A}) \leq \begin{cases} \left(\frac{Q^2}{(k-s)p} \right)^{k-s} & Q \geq (k-s)^{1/2} p^{\frac{1-\theta}{2}} \\ \left(\frac{1}{p} \right)^{\theta(k-s)} & \text{otherwise.} \end{cases} \quad (5)$$

If $\theta = 1$ the bound is always dominated by the top case. An easy computation shows that hence it supports $\frac{1}{2}(\log(p) + \log(k-s))$ bits of security. This is the case for (n, k) -CDH and (n, k) -SDH.

If $\theta < 1$, the bound is sometimes dominated by the bottom case, namely for small values of $Q \leq (k-s)^{1/2} p^{(1-\theta)/2}$. In this regime, a straightforward calculation shows that the resulting bit-security level is

$$\min \left\{ (k-s)\theta \log(p), \frac{1}{2}(\log(p) + \log(k-s)) \right\}. \quad (6)$$

Fortunately, for $\theta \geq 1/2$, this minimum is still attained by $\frac{1}{2}(\log(p) + \log(k-s))$. This is the case for (n, k) -IDH and $(n, k+1)$ -LKDH, which explains why both problems achieve $\frac{1}{2}(\log(p) + \log(k-s))$ bits of security. For $\theta < 1/2$, however, the bit-security level would decrease, at least for small values of k . In this sense, the claimed bit-security level for (n, k) -IDH and $(n, k+1)$ -LKDH is a close call.

Acknowledgments

We would like to thank Akin Ünal and Maximilian von Consbruch for mentioning the attack on (n, k) -IDH. Aysan Nishaburi, Samin Nooripoor and Emiel Wiedijk were supported by the European Union (ERC AdG REWORC – 101054911). Eike Kiltz and Akshima were supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC 2092 CASA – 390781972, and by the European Union (ERC AdG REWORC – 101054911).

A Proof of Theorem 1

$\mathcal{B}$ simulates $\mathcal{A}$ and outputs according to the output of $\mathcal{A}$. $\mathcal{B}$ maintains the following structures:

- **List₁**: each entry in this list is of the form (P, im) where P is a polynomial of degree 1 in $X_1, \dots, X_n$ and $\text{im} \in \mathcal{L}$.
- **Set₁**: set of preimages sampled “lazily” in the simulation.
- **Set₂**: global set for the image set of σ in $\mathcal{L}$. This means $|\text{Set}_2| = p$.

List **List₁** and set **Set₁** are empty at the beginning of the simulation. $\mathcal{B}$ initializes **Set₂** as the image set. $\mathcal{B}$ can simulate $\mathcal{A}$ with access to enough randomness to lazily sample (i.e., generate input-output pairs on the fly) a random injective function consistently using its hyperplane oracle queries and **List₁**.

Next, we describe the simulation using two subroutines, namely **FindImage** and **FindPreimage**, that will be defined later. Note that the following simulation is exactly the same as in Lemma 2 in [ABG⁺24], except for the addition of simulating the corruption queries of $\mathcal{A}$.

1. Call the subroutine **FindImage**(H) for every polynomial

$$H \in \{1, X_{1,1}, \dots, X_{1,m}, \dots, X_{n,1}, X_{n,m}\}$$

to generate the challenges $\hat{\sigma}(1)$ and

$$\hat{\sigma}(x) = \hat{\sigma}(X_{1,1}), \dots, \hat{\sigma}(X_{1,m}), \dots, \hat{\sigma}(X_{n,1}), \dots, \hat{\sigma}(X_{n,m}).$$

2. Run the generic adversary $\mathcal{A}$ on $\hat{\sigma}(1), \hat{\sigma}(x)$ and answer its queries as follows:
 - (a) For query $\mathcal{O}_\sigma(i_1, i_2)$: If $i_1 \notin \text{Set}_2$ or $i_2 \notin \text{Set}_2$ return $\perp$. Otherwise, return **FindImage**(**FindPreimage**(i_1) + **FindPreimage**(i_2)).
 - (b) For query $\mathcal{O}_{\sigma^{-1}}(i)$: If $i \notin \text{Set}_2$, return $\perp$. Otherwise, return $\text{Eval}_x(\text{FindPreimage}(i))$.
3. For each element t in the output from $\mathcal{A}$: return **FindPreimage**(t)

Now, we define the subroutines.

FindImage(H):

1. If $H_x(H - P) = 1$ for some $(P, \text{im}) \in \text{List}_1$, return im .
2. Otherwise, $\text{im} \leftarrow_{\$} \text{Set}_2 \setminus \mathcal{I}$ where $\mathcal{I} = \{a : (*, a) \in \text{List}_1\}$, add (H, im) to **List₁** and return im .

FindPreimage(i):

1. If there is an entry of the form (P, i) in **List₁**, return P .
2. $\beta \leftarrow_{\$} \mathbb{Z}_p \setminus \text{Set}_1$.
3. If there exists some $(P, \text{im}) \in \text{List}_1$ such that $H_x(P - \beta) = 1$, set $\text{Set}_1 = \text{Set}_1 \cup \{\beta\}$ and go to step 2. Otherwise, go to step 4.
4. Set $\text{Set}_1 = \text{Set}_1 \cup \{\beta\}$, add (β, i) to **List₁** and return β .

Correctness. Let $\mathcal{J}$ be a randomly sampled image set in $\mathcal{L}$ such that $|\mathcal{J}| = p$. Let $\sigma : \mathbb{Z}_p \rightarrow \mathcal{J}$ be a random encoding used in the “real” GGM setting. We will assume that σ is lazily sampled. We need to show that $\mathcal{B}$ “correctly” simulates $\mathcal{A}$ and generates the outputs correctly for $\text{Adv}(\mathcal{B}) = \text{Adv}(\mathcal{A})$ to hold. Essentially, we need to show the following for a fixed x , image set $\mathcal{J}$ in $\mathcal{L}$ and setting of $\text{Set}_2 = \mathcal{J}$:

1. for simulating any i^{th} challenge to $\mathcal{A}$, the output from the subroutine **FindImage** is identically distributed to the response from σ given the response to the first $(i - 1)$ challenges are equal in both scenarios
2. for simulating any i^{th} query of $\mathcal{A}$, the response from 2(a) (or 2(b)) is identically distributed to the response from $\mathcal{O}_\sigma$ (or $\mathcal{O}_{\sigma^{-1}}$), given the responses to the previous $(i - 1)$ queries are equal in the two scenarios
3. the output of $\mathcal{B}$ satisfies the winning polynomial when the labels outputted by $\mathcal{A}$ satisfy the winning polynomial

Let's begin from the $i = 0$ challenge. We can say that the responses from both **FindImage** and σ are identically distributed since both return a random value in $\mathcal{J}$. For any arbitrary $i \geq 1$ challenge, let us assume that for any $1 \leq j < i$, $\mathcal{B}$ correctly simulates the j^{th} challenge. Given that the responses to the first $(i - 1)$ challenges are equal in the two scenarios, it means that List_1 and the list of defined input-output pairs for σ are equal. Now to generate the i^{th} challenge, the subroutine **FindImage** first checks whether $H = X_{\lceil i/m \rceil, i - m \cdot (\lceil i/m \rceil - 1)} = X_{\lceil j/m \rceil, j - m \cdot (\lceil j/m \rceil - 1)}$ for any $j < i$, i.e., $H_x(H - P) = 1$ for any $(P, \text{im}) \in \text{List}_1$. If there is such a P , the subroutine returns im . Otherwise, the subroutine samples randomly from $\mathcal{J} \setminus \{a : (*, a) \in \text{List}_1\}$ and returns that. This is identically distributed to $\sigma(x[i])$ when List_1 and the list of defined input-output pairs for σ are equal. Thus, by induction we have shown 1.

Now to show 2, let's look at the i^{th} query for $1 \leq i \leq Q$. Again assume that List_1 and the list of defined input-output pairs for σ are equal up to $(i - 1)$ query. Then there are two possible scenarios:

- The i^{th} query is $\mathcal{O}_\sigma(t_1, t_2)$. If $t_1 \notin \mathcal{J}$ or $t_2 \notin \mathcal{J}$, then $\mathcal{O}_\sigma$ returns $\perp$ without any updates to the table of σ . And so does the simulation and no updates are made to List_1 . Otherwise the simulation calls subroutine **FindPreimage** on t_1 and t_2 . Let's consider **FindPreimage**(t_1) first. If $t_1 \in \mathcal{J} \setminus \{a : (*, a) \in \text{List}_1\}$ then **FindPreimage** adds (α, t_1) to List_1 such that α is randomly distributed in $\mathbb{Z}_p \setminus \mathcal{I}$ where $\mathcal{I}$ is the set of preimages from entries in List_1 . Using the assumption this means this action is identically distributed to $\sigma^{-1}(t_1)$. By a similar argument **FindPreimage**(t_2) is identically distributed to $\sigma^{-1}(t_2)$. More importantly, List_1 and the list of defined input-output pairs in σ are identically distributed. We need to show **FindImage**(**FindPreimage**(t_1) + **FindPreimage**(t_2)) is identically distributed to $\mathcal{O}(t_1, t_2)$ given List_1 and list of σ are equal up to $i - 1$ query and identically distributed after **FindPreimage**(t_1), **FindPreimage**(t_2) and $\sigma^{-1}(t_1)$, $\sigma^{-1}(t_2)$ respectively. If t_1, t_2 were already in List , then the subroutine **FindImage** checks whether $\alpha_1 + \alpha_2 = P$ for any $(P, \text{lm}) \in \text{List}_1$. If such an entry exists it returns lm otherwise randomly samples from $\mathcal{J} \setminus \{a : (*, a) \in \text{List}_1\}$.

If t_1 and t_2 were already in List_1 it means there were entries (L_1, t_1) and (L_2, t_2) in List_1 . Then the subroutine **FindImage** checks whether $L_1(x) + L_2(x) = P$ for any $(P, \text{lm}) \in \text{List}_1$. If such an entry exists it returns lm otherwise randomly samples from $\mathcal{J} \setminus \{a : (*, a) \in \text{List}_1\}$. This is exactly how $\mathcal{O}_\sigma$ responds to query and hence after responding to the query List_1 should be identically distributed to the list of defined input-output pairs of σ .

- The i^{th} query is a discrete-logarithm query $\mathcal{O}_{\sigma^{-1}}(t)$. Again if $t \notin \mathcal{J}$, $\mathcal{O}_{\sigma^{-1}}$ should return $\perp$ and that's exactly how the simulation responds. If $t \in \mathcal{J}$, $\mathcal{O}_{\sigma^{-1}}$ searches for an entry of the form (α, t) in its list. If such an entry is defined it responds with α . Otherwise it adds an entry (α, t) to its list such that $\alpha \in \mathbb{Z}_p \setminus \mathcal{I}$ where $\mathcal{I}$ is the

set of preimages from the list of defined input-output pairs in σ . The simulation calls the subroutine **FindPreimage** which also checks whether $(P, t) \in \text{List}_1$. Now there are 3 possibilities. Such an entry exists in List_1 and P is a constant β in $\mathbb{Z}_p$. Then essentially the simulation responds with β .

Second possibility is such $(P, t) \in \text{List}_1$ and P is polynomial. Then $\text{EVAL}_x(P)$ will return $P(x)$. As by assumption List_1 and list of defined input-output pair in σ are equal up to $i - 1$ query, the responses in these 2 possibilities should be identically distributed in the simulation and $\mathcal{O}_{\sigma-1}$.

Let's look at the third possibility that there is no entry $(*, t)$ in List_1 . Then subroutine **FindPreimage** adds (α, t) to List_1 such that $\alpha \in \mathbb{Z}_p \setminus \{a : (a, *) \in \text{List}_1\}$. (In such a case EVAL_x returns α .) As argued before the response from simulation should be identically distributed to $\mathcal{O}_{\sigma-1}$ adding an entry as List_1 is assumed to be equal to the list of defined input-output of σ .

Finally, we show 3. If $\mathcal{A}$ outputs labels that it has received as challenge or response, for each label t there must exist a corresponding entry (P, t) in List_1 . Then the polynomial P from such an entry must satisfy the winning polynomial whenever $\mathcal{A}$ wins. Hence, $\mathcal{B}$ wins. Otherwise, whenever $\mathcal{A}$ outputs a 'new' label, $\mathcal{B}$ outputs a randomly distributed β in $\mathbb{Z}_p \setminus \mathcal{I}$ where $\mathcal{I}$ is the set of preimages from entries in List_1 . Then, the probability that $\mathcal{B}$'s output for the instance satisfies the winning polynomial is the same as the probability $\mathcal{A}$ succeeds on the instance.

Query Complexity. It is easy to see that $\mathcal{B}$ queries EVAL_x exactly once to simulate each discrete-logarithm query of $\mathcal{A}$. As $\mathcal{A}$ makes at most s discrete-logarithm queries to $\mathcal{O}_{\sigma-1}$, $\mathcal{B}$ needs to make just as many queries to EVAL_x . In addition, simulation of each discrete-logarithm query requires fewer invocations of the subroutines compared to a query to $\mathcal{O}_\sigma$. So, it suffices to count the number of H_x queries required to simulate Q queries of $\mathcal{A}$ to $\mathcal{O}_\sigma$.

$\mathcal{B}$ makes $mn + 1$ calls to **FindImage** subroutine to generate the challenges for $\mathcal{A}$, with each call adding at most 1 new pair to List_1 . For each $\mathcal{O}_\sigma$ query of $\mathcal{A}$, $\mathcal{B}$ must make a total of 3 calls to the subroutines, **FindImage** and **FindPreimage**, and adds at most 3 new pairs to List_1 . Then to generate its output $\mathcal{B}$ makes $\ell + n$ calls to the **FindPreimage** subroutine. So, the size of List_1 is at most $3Q + mn + \ell + n + 1$ at any point in the simulation. In addition, the size of Set_1 is at most $3Q + mn + \ell + n + 1$. This is because Set_1 can have at most as many elements as in List_1 .

$\mathcal{B}$ needs to make $|\text{List}_1|$ queries to H_x for each call to subroutine **FindImage**. For each call to **FindPreimage**, $\mathcal{B}$ is required to make $|\text{Set}_1| + |\text{List}_1|$ queries to H_x . So, overall $\mathcal{B}$ makes at most $2 \cdot (3Q + mn + \ell + n + 1)^2$ queries to H_x .

B Proof of Lemma 1

Let $\text{EVENT}_{\text{eval}}$ and EVENT_{HS} be the events that the first query of the adversary is an evaluation and hyperplane query, respectively. Additionally, let H_1 be the first hyperplane query, then $H_x(H_1)$ denotes its response. For simplicity of notation write Adv for Adv^W in the proof. Let $\mathcal{A}$ be the optimal adversary for (n, k) -MI-HQM given q queries, including s evaluation queries. Without loss of generality $\mathcal{A}$ is deterministic

Recall $\mathbf{A}(n, k, q, s)$ is the set of adversaries in the hyperplane query model, solving MI-HQM_W in at most q queries, (of which at most s to EVAL_x)

$$\begin{aligned}\Delta_{\mathbb{H}} &:= \max_{\mathcal{B} \in \mathbf{A}(n, k, q, s)} \Pr[(n, k) \text{- MI-HQM}^{\mathcal{B}} \Rightarrow 1 \wedge \text{EVENT}_{\mathbb{H}\mathbb{S}} \wedge H_x(H_1) = 1] \\ \bar{\Delta}_{\mathbb{H}} &:= \max_{\mathcal{B} \in \mathbf{A}(n, k, q, s)} \Pr[(n, k) \text{- MI-HQM}^{\mathcal{B}} \Rightarrow 1 \wedge \text{EVENT}_{\mathbb{H}\mathbb{S}} \wedge H_x(H_1) = 0] \\ \Delta_{\mathbb{C}} &:= \max_{\mathcal{B} \in \mathbf{A}(n, k, q, s)} \Pr[(n, k) \text{- MI-HQM}^{\mathcal{B}} \Rightarrow 1 \wedge \text{EVENT}_{\text{eval}}]\end{aligned}$$

We need the following claims.

Claim. $\Delta_{\mathbb{H}}(n, k, q, s) \leq \frac{1}{p} \text{Adv}(n-1, k-1, q-1, s)$.

Proof. Let $\mathcal{C}_1 \in \mathbf{A}(n, k, q, s)$ be an adversary attaining the maximum in $\Delta_{\mathbb{H}}(n, k, q, s)$. Hence

$$\Delta_{\mathbb{H}}(n, k, q, s) = \Pr[(n, k) \text{- MI-HQM}^{\mathcal{C}_1} \Rightarrow 1 \wedge \text{EVENT}_{\mathbb{H}\mathbb{S}} \wedge H_x(H_1) = 1].$$

In the (n, k) -MI-HQM game for $\mathcal{C}_1$, let $H_1 = H_f$ be the first query of $\mathcal{C}_1$ under the event $\text{EVENT}_{\mathbb{H}\mathbb{S}}$, where H_f is defined by the affine function

$$f(X_1, \dots, X_n) = \alpha_0 + \sum_{i=1}^n \sum_{j=1}^m \alpha_{i,j} X_{i,j}. \quad (7)$$

By conditional probability,

$$\begin{aligned}\Pr[(n, k) \text{- MI-HQM}^{\mathcal{C}_1} \Rightarrow 1 \wedge \text{EVENT}_{\mathbb{H}\mathbb{S}} \wedge H_x(H_1) = 1] \\ = \Pr[\text{EVENT}_{\mathbb{H}\mathbb{S}} \wedge H_x(H_1) = 1] \cdot \Pr[(n, k) \text{- MI-HQM}^{\mathcal{C}_1} \Rightarrow 1 \mid \text{EVENT}_{\mathbb{H}\mathbb{S}} \wedge H_x(H_1) = 1].\end{aligned}$$

Let the secret point of $\mathcal{C}_1$ be $x \in \mathbb{Z}_p^{mn}$. Conditioned on $\text{EVENT}_{\mathbb{H}\mathbb{S}}$, the first query is the hyperplane query H_1 defined by the nonzero affine polynomial f in Equation (7). Since f is nonzero affine and the secret x is uniform over $\mathbb{Z}_p^{mn}$, Schwartz-Zippel gives

$$\Pr[H_x(H_1) = 1 \wedge \text{EVENT}_{\mathbb{H}\mathbb{S}}] \leq \Pr[H_x(H_1) = 1] \leq \frac{1}{p}. \quad (8)$$

It remains to bound the conditional success probability. We show that

$$\Pr[(n, k) \text{- MI-HQM}^{\mathcal{C}_1} \Rightarrow 1 \mid \text{EVENT}_{\mathbb{H}\mathbb{S}} \wedge H_x(H_1) = 1] \leq \text{Adv}(n-1, k-1, q-1, s).$$

We prove the bound by reduction. We construct an adversary $\mathcal{C}_2 \in \mathbf{A}(n-1, k-1, q-1, s)$ for the $(n-1, k-1)$ -MI-HQM problem. Let $x' = (x'_1, \dots, x'_{n-1})$ be the secret point of $\mathcal{C}_2$ such that $x'_i = (x'_{i,1}, \dots, x'_{i,m}) \in \mathbb{Z}_p^m$. The adversary $\mathcal{C}_2$ simulates $\mathcal{C}_1$ in the (n, k) -MI-HQM game conditioned on $\text{EVENT}_{\mathbb{H}\mathbb{S}} \wedge H_x(H_1) = 1$. We denote the simulated secret for $\mathcal{C}_1$ by $x = (x_1, \dots, x_{n-1})$, where $x_i = (x_{i,1}, \dots, x_{i,m}) \in \mathbb{Z}_p^m$. Note that WLOG the simulation identifies $x_i = x'_i$ for all $i \in [n-1]$, and fixes the remaining coordinates $x_{n,j}$ for $j \neq j^*$. Since $\mathcal{C}_2$ operates in the $(n-1, k-1)$ -MI-HQM game, every query it forwards must be an affine polynomial in $X_{1,1}, \dots, X_{n-1,m}$. We now explain how $\mathcal{C}_2$ uses the first query of $\mathcal{C}_1$ to transform all subsequent queries into this form.

- **First query.** Since we condition on $\text{EVENT}_{\mathbb{H}\mathbb{S}} \wedge H_x(H_1) = 1$ in the (n, k) -MI-HQM game, the reduction returns 1 on the first query H_f for the function f defined in

[Equation \(7\)](#). We relabel instances so that $\alpha_{n,j^*} \neq 0$ for some $j^* \in [m]$; hence $f(X) = 0$ gives

$$X_{n,j^*} = -\alpha_{n,j^*}^{-1} \left(\alpha_0 + \sum_{i=1}^{n-1} \sum_{j=1}^m \alpha_{i,j} X_{i,j} + \sum_{j \neq j^*} \alpha_{n,j} X_{n,j} \right). \quad (9)$$

- **Remaining queries.** For any later query $H_{\hat{f}}$ by $\mathcal{C}_1$, let $\hat{f}$ be the corresponding affine function

$$\hat{f}(X_1, \dots, X_n) = \hat{\alpha}_0 + \sum_{i=1}^n \sum_{j=1}^m \hat{\alpha}_{i,j} X_{i,j}.$$

The reduction substitutes X_{n,j^*} using [Equation \(9\)](#), and replaces each $X_{n,j}$ for $j \neq j^*$ by the sampled values $x_{n,j}$ which are known to the reduction. This gives an affine polynomial $\hat{f}'$ in the variables $X_{1,1}, \dots, X_{n-1,m}$. Then $\mathcal{C}_2$ forwards $H_{\hat{f}'}$ to the corresponding oracle and returns the answer to $\mathcal{C}_1$.

Since the first query is answered without using an oracle query, $\mathcal{C}_2$ makes at most $q - 1$ queries in total and at most s evaluation queries.

Finally, suppose $\mathcal{C}_1$ outputs $(\mathcal{I}, Y, Z)$ and wins the simulated (n, k) -MI-HQM problem under the condition $\text{EVENT}_{\text{HS}} \wedge H_x(H_1) = 1$. The adversary $\mathcal{C}_2$ removes the simulated instance n and outputs $\mathcal{I}' := \mathcal{I} \setminus \{n\}$, together with the polynomials obtained from Y and Z_i for $i \in \mathcal{I}'$ by applying the same substitution as above. We denote these substituted polynomials by Y' and Z'_i .

Since $\mathcal{C}_1$ wins, after removing the simulated instance n there are still at least $k - 1$ correct instances. The same substitution used in the simulation turns the output of $\mathcal{C}_1$ into polynomials Y' and Z'_i satisfying $W(x'_i, Y'(x'), Z'_i(x')) = 0$ for all $i \in \mathcal{I}'$, and preserves the condition $Y'(x') \neq 0$. Hence $\mathcal{C}_2$ wins the $(n - 1, k - 1)$ -MI-HQM problem. Hence

$$\Pr[(n, k)\text{-MI-HQM}^{\mathcal{C}_1} \Rightarrow 1 \mid \text{EVENT}_{\text{HS}} \wedge H_x(H_1) = 1] \leq \text{Adv}(n - 1, k - 1, q - 1, s). \quad (10)$$

Combining [Equation \(10\)](#) with [Equation \(8\)](#), we obtain

$$\begin{aligned} \Delta_{\text{H}}(n, k, q, s) &= \Pr[(n, k)\text{-MI-HQM}^{\mathcal{C}_1} \Rightarrow 1 \wedge \text{EVENT}_{\text{HS}} \wedge H_x(H_1) = 1] \\ &= \Pr[\text{EVENT}_{\text{HS}} \wedge H_x(H_1) = 1] \cdot \Pr[(n, k)\text{-MI-HQM}^{\mathcal{C}_1} \Rightarrow 1 \mid \text{EVENT}_{\text{HS}} \wedge H_x(H_1) = 1] \\ &\leq \frac{1}{p} \text{Adv}(n - 1, k - 1, q - 1, s). \end{aligned}$$

This completes the proof of the claim. $\square$

Claim. $\bar{\Delta}_{\text{H}}(n, k, q, s) \leq \text{Adv}(n, k, q - 1, s) + \Delta_{\text{H}}(n, k, q, s)$.

Proof. For any adversary $\mathcal{B} \in \text{HQM}(n, k, q, s)$, let $\mathcal{B}_0$ and $\mathcal{B}_1 \in \text{HQM}(n, k, q - 1, s)$ be the sub-algorithms that $\mathcal{B}$ runs when the reply to its first query is 0 and 1, respectively. Let the hyperplane H_1 denote the first query of $\mathcal{B}$. Clearly $\mathcal{B}_0$ and $\mathcal{B}_1$ both make one query fewer and we have

$$\begin{aligned} \Pr[(n, k)\text{-MI-HQM}^{\mathcal{B}_0} \Rightarrow 1 \wedge H_x(H_1) = 0] \\ = \Pr[(n, k)\text{-MI-HQM}^{\mathcal{B}} \Rightarrow 1 \wedge \text{EVENT}_{\text{HS}} \wedge H_x(H_1) = 0] \end{aligned}$$

Additionally by law of total probability, we can write:

$$\begin{aligned} & \Pr[(n, k) - \text{MI-HQM}^{\mathcal{B}_0} \Rightarrow 1] \\ &= \Pr[(n, k) - \text{MI-HQM}^{\mathcal{B}_0} \Rightarrow 1 \wedge H_x(H_1) = 0] + \Pr[(n, k) - \text{MI-HQM}^{\mathcal{B}_0} \Rightarrow 1 \wedge H_x(H_1) = 1] \end{aligned}$$

This implies

$$\begin{aligned} & \Pr[(n, k) - \text{MI-HQM}^{\mathcal{B}} \Rightarrow 1 \wedge \text{EVENT}_{\text{HS}} \wedge H_x(H_1) = 0] \\ &= \Pr[(n, k) - \text{MI-HQM}^{\mathcal{B}_0} \Rightarrow 1 \wedge H_x(H_1) = 0] \\ &\leq \Pr[(n, k) - \text{MI-HQM}^{\mathcal{B}_0} \Rightarrow 1] + \Pr[(n, k) - \text{MI-HQM}^{\mathcal{B}_0} \Rightarrow 1 \wedge H_x(H_1) = 1] \end{aligned}$$

Let $\mathcal{C}$ be the adversary that achieves $\bar{\Delta}_{\text{H}}(n, k, q, s)$ and let $H_1, \mathcal{C}_0$ and $\mathcal{C}_1$ be defined analogously as above. Then using the definitions, we can write:

$$\begin{aligned} \bar{\Delta}_{\text{H}}(n, k, q, s) &= \Pr[(n, k) - \text{MI-HQM}^{\mathcal{C}} \Rightarrow 1 \wedge \text{EVENT}_{\text{HS}} \wedge H_x(H_1) = 0] \\ &\leq \Pr[(n, k) - \text{MI-HQM}^{\mathcal{C}_0} \Rightarrow 1] + \Pr[(n, k) - \text{MI-HQM}^{\mathcal{C}_0} \Rightarrow 1 \wedge H_x(H_1) = 1] \\ &\leq \text{Adv}(n, k, q - 1, s) + \Delta_{\text{H}}(n, k, q, s). \end{aligned}$$

This completes the proof of the claim. $\square$

Claim. For every $s, q \geq 1$, $\Delta_{\text{C}}(n, k, q, s) \leq \text{Adv}(n - 1, k - 1, q - 1, s - 1)$.

Proof. Let $\mathcal{B} \in \mathbf{A}(n, k, q, s)$ be the optimal adversary attaining $\Delta_{\text{C}}(n, k, q, s)$ and E_1 denote its first evaluation query. Thus,

$$\Delta_{\text{C}}(n, k, q, s) = \Pr[(n, k) - \text{MI-HQM}^{\mathcal{B}} \Rightarrow 1 \wedge \text{EVENT}_{\text{eval}}].$$

Let's denote $E_1 = \sum_{i=1}^n \sum_{j=1}^m \alpha_{i,j} X_{i,j} + \alpha_0$. Then we show there exists an adversary $\mathcal{C} \in \text{HQM}(n - 1, k - 1, q - 1, s - 1)$ such that $\Pr[(n - 1, k - 1) - \text{MI-HQM}^{\mathcal{C}} \Rightarrow 1] = \Pr[(n, k) - \text{MI-HQM}^{\mathcal{B}} \Rightarrow 1 \wedge \text{EVENT}_{\text{eval}}]$.

We show how to construct such an adversary $\mathcal{C}$. It simulates $\mathcal{B}$ given $\text{EVENT}_{\text{eval}}$.

Let the secret point of $\mathcal{C}$ be $x = (x_1, \dots, x_{n-1})$ with $x_i = (x_{i,1}, \dots, x_{i,m}) \in \mathbb{Z}_p^m$. Say $\mathcal{C}$ simulates secret point $\hat{x} = (\hat{x}_1, \dots, \hat{x}_n)$ such that $\hat{x}_i = (\hat{x}_{i,1}, \dots, \hat{x}_{i,m}) \in \mathbb{Z}_p^m$.

Without loss of generality, we can assume $\alpha_{n,j^*} \neq 0$ for some $j^* \in [m]$ in E_1 . Then $\mathcal{C}$ simulates the secret point of $\mathcal{B}$ as follows:

- For $i \in [n - 1]$: $\hat{x}_i = x_i$
- $\beta \xleftarrow{\$} \mathbb{Z}_p$ and set $E_1 = \sum_{i=1}^n \sum_{j=1}^m \alpha_{i,j} X_{i,j} + \alpha_0 = \beta$. Independently sample $\hat{x}_{n,j} \xleftarrow{\$} \mathbb{Z}_p$ for all $j \in [m] \setminus \{j^*\}$. Then we use these $\hat{x}_{n,j}$ and $E_1 = \beta$ to obtain an expression for X_{n,j^*} in terms of $(X_{i,j})_{i \in [n-1], j \in [m]}$.

$\mathcal{C}$ simulates all queries of $\mathcal{B}$, starting from returning β for E_1 . For each of the remaining $s - 1$ evaluation queries from $\mathcal{B}$, $\mathcal{C}$ queries its evaluation oracle by replacing X_{n,j^*} with the expression obtained above and replacing $X_{n,j} = \hat{x}_{n,j}$ for every $j \in [m] \setminus \{j^*\}$. Similarly $\mathcal{C}$ simulates the hyperplane queries of $\mathcal{B}$ by querying its H_x .

Finally, suppose $\mathcal{B}$ outputs $(\mathcal{I}, Y, Z)$ and wins the simulated $(n, k) - \text{MI-HQM}$ problem under the condition $\text{EVENT}_{\text{eval}} \wedge E_1$. The adversary $\mathcal{C}$ removes the simulated instance n and outputs $\mathcal{I}' := \mathcal{I} \setminus \{n\}$, together with the polynomials obtained from Y and Z_i for $i \in \mathcal{I}'$ by applying the same substitution as above. We denote these substituted polynomials by Y' and Z'_i .

Since $\mathcal{B}$ wins, after removing the simulated instance n there are still at least $k - 1$ correct instances. The same substitution used in the simulation turns the output of $\mathcal{B}$ into polynomials Y' and Z'_i satisfying $W(x_i, Y'(x), Z'_i(x)) = 0$ for all $i \in \mathcal{I}'$, and preserves the condition $Y'(x) \neq 0$. Hence $\mathcal{C}$ wins the $(n - 1, k - 1)$ -MI-HQM problem. Hence,

$$\begin{aligned}\Delta_{\mathcal{C}}(n, k, q, s) &= \Pr[(n, k) - \text{MI-HQM}^{\mathcal{B}} \Rightarrow 1 \wedge \text{EVENT}_{\text{eval}}] \\ &= \Pr[(n - 1, k - 1) - \text{MI-HQM}^{\mathcal{C}} \Rightarrow 1] \\ &\leq \text{Adv}(n - 1, k - 1, q - 1, s - 1)\end{aligned}$$

This completes the proof of the claim. $\square$

Now we can conclude the proof of the lemma by

$$\begin{aligned}\text{Adv}(n, k, q, s) &= \Pr[(n, k) - \text{MI-HQM}^{\mathcal{A}} \Rightarrow 1] \\ &\leq \max \left\{ \begin{array}{l} \Pr[(n, k) - \text{MI-HQM}^{\mathcal{A}} \Rightarrow 1 \wedge \text{EVENT}_{\text{eval}}], \\ \Pr[(n, k) - \text{MI-HQM}^{\mathcal{A}} \Rightarrow 1 \wedge \text{EVENT}_{\text{HS}}] \end{array} \right\} \\ &\leq \max \left\{ \begin{array}{l} \Pr[(n, k) - \text{MI-HQM}^{\mathcal{A}} \Rightarrow 1 \wedge \text{EVENT}_{\text{HS}} \wedge H_x(H_1) = 1] \\ + \Pr[(n, k) - \text{MI-HQM}^{\mathcal{A}} \Rightarrow 1 \wedge \text{EVENT}_{\text{HS}} \wedge H_x(H_1) = 0], \\ \Pr[(n, k) - \text{MI-HQM}^{\mathcal{A}} \Rightarrow 1 \wedge \text{EVENT}_{\text{eval}}] \end{array} \right\} \\ &\leq \max \{ \Delta_{\mathcal{H}}(n, k, q, s) + \bar{\Delta}_{\mathcal{H}}(n, k, q, s), \Delta_{\mathcal{C}}(n, k, q, s) \} \\ &\leq \max \{ \text{Adv}(n, k, q - 1, s) + 2 \cdot \Delta_{\mathcal{H}}(n, k, q, s), \Delta_{\mathcal{C}}(n, k, q, s) \} \\ &\leq \max \left\{ \begin{array}{l} \text{Adv}(n, k, q - 1, s) + \frac{2}{p} \text{Adv}(n - 1, k - 1, q - 1, s), \\ \text{Adv}(n - 1, k - 1, q - 1, s - 1) \end{array} \right\}.\end{aligned}$$

This proves the lemma.

C Algebraic geometry

We assume some basic background in algebraic geometry and Gröbner basis theory, and therefore do not repeat all standard definitions here. We refer the reader to textbooks such as [Har77, CLO25]. In this section, we fix the notation used later and state the results from [CLO25] that we need. In the following let K be an algebraically closed field.

Definition 3 ([CLO25, Def. 1.4.2]). Let $f_1, \dots, f_s \in K[X_1, \dots, X_n]$. We define

$$\langle f_1, \dots, f_s \rangle := \left\{ \sum_{i=1}^s h_i f_i \mid h_1, \dots, h_s \in K[X_1, \dots, X_n] \right\}.$$

This is an ideal of $K[X_1, \dots, X_n]$, called the ideal generated by $f_1, \dots, f_s$.

Theorem 4 ([CLO25, Thm. 9.5.2]). Let V be an affine variety with $d = \dim V$. Then d is the maximum number of elements $X_{i_1}, \dots, X_{i_d} \in K[V] := K[X_1, \dots, X_n]/I(V)$ such that $X_{i_1}, \dots, X_{i_d}$ are algebraically independent.

Proposition 1 ([CLO25, Prop. 9.1.3]). *Let $I = \langle m_1, \dots, m_t \rangle \subseteq K[X_1, \dots, X_n]$ be a monomial ideal, where each m_j is a monomial. For each $j \in [t]$, define*

$$M_j := \{i \in [n] \mid X_i \text{ divides } m_j\},$$

and let

$$\mathcal{M} := \{J \subseteq [n] \mid J \cap M_j \neq \emptyset \text{ for all } j \in [t]\}.$$

Then:

1. if $J = \{i_1, \dots, i_r\} \in \mathcal{M}$, then $V(X_{i_1}, \dots, X_{i_r}) \subseteq V(I)$;
2. $\dim V(I) = n - \min\{|J| \mid J \in \mathcal{M}\}$.

Theorem 5 (Buchberger’s Criterion [CLO25, Thm. 2.6.6]). *Let $I \subseteq K[X_1, \dots, X_n]$ be a polynomial ideal, and let $G = \{g_1, \dots, g_t\}$ be a basis of I . Choose an ordering of G and regard it as the ordered tuple $G = (g_1, \dots, g_t)$. Then G is a Gröbner basis of I if and only if for every pair $i \neq j$, the remainder on division of $S(g_i, g_j)$ by G is zero.*

Furthermore, we will include the Lang–Weil theorem, which is not stated as such in the classic textbooks. We use the degree convention from [Hei83] that includes the degree of lower-dimension components. It gives an upper bound of the zero-set of a system of polynomials over a finite field. Lang–Weil can be viewed as a generalization of both Schwartz–Zippel (where $s = 1$), and Bézout’s theorem (where $d = 0$). For the following Theorem, let $\overline{\mathbb{F}}_p$ be the algebraic closure of $\mathbb{F}_p$.

Theorem 6 (Lang–Weil style bound [LW54, Tao12] [DKL14, Claim 7.2]). *Let $V(f_1, \dots, f_s) \subseteq \overline{\mathbb{F}}_p^m$ be a variety for $f_1, \dots, f_s \in \overline{\mathbb{F}}_p[X_1, \dots, X_m]$ with $\dim V = d$. Then for the finite field $\mathbb{F}_p$*

$$|V \cap \mathbb{F}_p^m| \leq \prod_{i=1}^s \deg(f_i) \cdot p^d.$$

Proof. This theorem is a slight restatement of [DKL14, Claim 7.2]. They prove $|V \cap \mathbb{F}_p^m| \leq rp^d$ where r is the *degree* of the variety. Combining this with

$$\deg(V(f_1, \dots, f_s)) \leq \prod_{i=1}^s \deg(V(f_i)) = \prod_{i=1}^s \deg(f_i)$$

from [Hei83, Theorem 1, Remark 2 (3)] proves the theorem. $\square$

D Multi-Instance CDH and LKDH

In this section we explicitly define the (n, k) -CDH, (n, k) -CDH-HQM (Figure 5), (n, k) -LKDH and (n, k) -LKDH-HQM (Figure 6) games, where oracle DL is defined in Figure 2 and oracles H_x , EVAL_x are defined in Figure 3. The winning advantage of an adversary is as defined in Definition 1 and Definition 2.

Game (n, k) -CDH	Game (n, k) -CDH-HQM
00 for $i \in [n]$: $(x_i, y_i) \xleftarrow{\$} \mathbb{Z}_p^2$	03 for $i \in [n]$: $(x_i, y_i) \xleftarrow{\$} \mathbb{Z}_p^2$
01 $(\mathcal{I}, Z_1, \dots, Z_n) \leftarrow \mathcal{A}^{\text{DL}}((g^{x_i}, g^{y_i})_{i \in [n]})$	04 $x = (x_1, y_1, \dots, x_n, y_n)$
02 return $\bigwedge_{i \in \mathcal{I}} [Z_i = g^{x_i y_i}] \wedge [\mathcal{I} \geq k]$	05 $(\mathcal{I}, Z_1, \dots, Z_n) \leftarrow \mathcal{A}^{\text{H}_x, \text{EVAL}_x}$
	06 return $\bigwedge_{i \in \mathcal{I}} [Z_i(x) = x_i y_i] \wedge [\mathcal{I} \geq k]$

Fig. 5. The (n, k) -CDH game (left) and the version in the Hyperplane Query Model, (n, k) -CDH-HQM (right).

Game (n, k) -LKDH	Game (n, k) -LKDH-HQM
00 for $i \in [n]$: $x_i \xleftarrow{\$} \mathbb{Z}_p$	03 for $i \in [n]$: $x_i \xleftarrow{\$} \mathbb{Z}_p$
01 $(\mathcal{I}, Y, Z_1, \dots, Z_n) \leftarrow \mathcal{A}^{\text{DL}}((g^{x_i})_{i \in [n]})$	04 $x = (x_1, \dots, x_n)$
02 return $\bigwedge_{i \in \mathcal{I}} [Z_i = Y^{x_i}] \wedge [Y \neq 1_{\mathbb{G}}] \wedge [\mathcal{I} \geq k]$	05 $(\mathcal{I}, Y, Z_1, \dots, Z_n) \leftarrow \mathcal{A}^{\text{H}_x, \text{EVAL}_x}$
	06 return $\bigwedge_{i \in \mathcal{I}} [Z_i(x) = x_i Y(x)] \wedge [Y(x) \neq 0] \wedge [\mathcal{I} \geq k]$

Fig. 6. The (n, k) -LKDH game (left) and its version in the Hyperplane Query Model, (n, k) -LKDH-HQM (right).

E Proof of Lemma 6

Proof. The proof will work similarly to the hardness of Square Diffie-Hellman (see Lemma 3). Note that compared to Square Diffie-Hellman, $V_{Z, \mathcal{I}}^{\text{CDH}}$ is defined over the polynomial ring $\mathbb{F}_p[X_{1,1}, X_{1,2}, X_{2,1}, \dots, X_{n,2}]$ in $2n$ variables. In Lemma 11 we will show that

$$\dim V_{Z, \mathcal{I}}^{\text{CDH}} = 2n - k,$$

so by Theorem 3 it follows that

$$\text{Adv}^{W_{\text{CDH}}}(n, k, 0, 0) \leq \frac{\deg(W_{\text{CDH}})^k \cdot p^{2n-k}}{p^{2n}} = \left(\frac{2}{p}\right)^k,$$

as required. $\square$

We fix a subset $\mathcal{I} \subseteq [n]$ with $|\mathcal{I}| = k$ and consider the evaluations of the winning polynomial

$$h_i(X) := X_{i,1}X_{i,2} - Z_i(X)$$

for $i \in \mathcal{I}$. Here $Z_i(X)$ is the linear polynomial representing the adversary's output for the i -th instance in the zero-query setting. Define the ideal

$$I_{\text{CDH}} := \langle h_i \mid i \in \mathcal{I} \rangle \subseteq \mathbb{F}_p[X_{1,1}, \dots, X_{n,2}].$$

We consider the variety $V(I_{\text{CDH}})$, which is the common zero set of the generators of I_{CDH} . As in the (n, k) -SDH case, we first show that $H_{\text{CDH}} := \{h_i \mid i \in \mathcal{I}\}$ forms a Gröbner basis for I_{CDH} and then compute

$$\dim V(I_{\text{CDH}}) = \dim V(\langle \text{LT}(I_{\text{CDH}}) \rangle) = \dim V((X_{i,1}X_{i,2})_{i \in \mathcal{I}}),$$

as the leading term ideal preserves the dimension. Throughout this subsection, we again use the graded lexicographic order $>_{\text{grlex}}$.

Lemma 10. *The set*

$$H_{\text{CDH}} = \{h_i \mid i \in \mathcal{I}\}$$

is a Gröbner basis of I_{CDH} with respect to $>_{\text{grlex}}$. Moreover,

$$\langle \text{LT}(I_{\text{CDH}}) \rangle = \langle X_{i,1}X_{i,2} \mid i \in \mathcal{I} \rangle.$$

Proof (of Lemma 10). The proof is identical to the proof of Lemma 4, with the quadratic term X_i^2 replaced by $X_{i,1}X_{i,2}$.

For every $i \in \mathcal{I}$, since $Z_i(X)$ is linear and $>_{\text{grlex}}$ is graded, we have

$$\text{LT}(h_i) = X_{i,1}X_{i,2}.$$

Thus the leading terms of the natural generators of I_{CDH} are $X_{i,1}X_{i,2}$ for $i \in \mathcal{I}$. It remains only to check Buchberger's criterion. For distinct $i, j \in \mathcal{I}$, using the same argument as in the (n, k) -SDH case, we get

$$S(h_i, h_j) = X_{i,1}X_{i,2}Z_j(X) - X_{j,1}X_{j,2}Z_i(X).$$

Indeed,

$$S(h_i, h_j) = Z_j(X)h_i - Z_i(X)h_j.$$

Hence every $S(h_i, h_j)$ has remainder zero on division by H_{CDH} . By Buchberger's criterion, H_{CDH} is a Gröbner basis of I_{CDH} .

Finally, because H_{CDH} is a Gröbner basis and $\text{LT}(h_i) = X_{i,1}X_{i,2}$ for every $i \in \mathcal{I}$, we get

$$\langle \text{LT}(I_{\text{CDH}}) \rangle = \langle \text{LT}(h_i) \mid i \in \mathcal{I} \rangle = \langle X_{i,1}X_{i,2} \mid i \in \mathcal{I} \rangle.$$

This proves the lemma. $\square$

Lemma 11. *Let $\mathcal{I} \subseteq [n]$ with $|\mathcal{I}| = k$. Then*

$$\dim V(I_{\text{CDH}}) = 2n - k.$$

Proof. By Lemma 10,

$$\langle \text{LT}(I_{\text{CDH}}) \rangle = \langle X_{i,1}X_{i,2} \mid i \in \mathcal{I} \rangle.$$

This is a monomial ideal in $2n$ variables. For each $i \in \mathcal{I}$, the generator $X_{i,1}X_{i,2}$ involves the pair $\{X_{i,1}, X_{i,2}\}$. Hence, in the notation of Proposition 1, a minimal set J must choose one variable from each of the k pairs. Therefore

$$\min\{|J| \mid J \in \mathcal{M}\} = k.$$

Applying Proposition 1, and using the fact that the initial ideal preserves the Krull dimension gives

$$\dim V(I_{\text{CDH}}) = \dim V(\langle \text{LT}(I_{\text{CDH}}) \rangle) = 2n - k,$$

which completes the proof. $\square$

References

- ABB⁺20. Michel Abdalla, Manuel Barbosa, Tatiana Bradley, Stanislaw Jarecki, Jonathan Katz, and Jiayu Xu. Universally composable relaxed password authenticated key exchange. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 278–307. Springer, Cham, August 2020. doi:10.1007/978-3-030-56784-2_10. (Cited on p. 8).
- ABG⁺24. Akshima, Tyler Besselman, Siyao Guo, Zhiye Xie, and Yuping Ye. Tight time-space tradeoffs for the decisional Diffie-Hellman problem. In Bojan Mohar, Igor Shinkar, and Ryan O’Donnell, editors, *56th ACM STOC*, pages 1739–1749. ACM Press, June 2024. doi:10.1145/3618260.3649752. (Cited on pp. 6, 7, 8, 10, 12, and 23).
- ACDW20. Akshima, David Cash, Andrew Drucker, and Hoeteck Wee. Time-space tradeoffs and short collisions in merkle-damgård hash functions. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 157–186. Springer, Cham, August 2020. doi:10.1007/978-3-030-56784-2_6. (Cited on p. 6).
- ADGL23. Akshima, Xiaoqi Duan, Siyao Guo, and Qipeng Liu. On time-space lower bounds for finding short collisions in Sponge hash functions. In Guy N. Rothblum and Hoeteck Wee, editors, *TCC 2023, Part III*, volume 14371 of *LNCS*, pages 237–270. Springer, Cham, November / December 2023. doi:10.1007/978-3-031-48621-0_9. (Cited on p. 6).
- AGK20. Benedikt Auerbach, Federico Giacon, and Eike Kiltz. Everybody’s a target: Scalability in public-key encryption. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 475–506. Springer, Cham, May 2020. doi:10.1007/978-3-030-45727-3_16. (Cited on pp. 3, 7, 8, and 10).
- AGL22. Akshima, Siyao Guo, and Qipeng Liu. Time-space lower bounds for finding collisions in merkle-Damgård hash functions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 192–221. Springer, Cham, August 2022. doi:10.1007/978-3-031-15982-4_7. (Cited on p. 6).
- AHH21. Michel Abdalla, Björn Haase, and Julia Hesse. Security analysis of CPace. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 711–741. Springer, Cham, December 2021. doi:10.1007/978-3-030-92068-5_24. (Cited on p. 8).
- AHPP23. Benedikt Auerbach, Charlotte Hoffmann, and Guillermo Pascual-Perez. Generic-group lower bounds via reductions between geometric-search problems: With and without preprocessing. In Guy N. Rothblum and Hoeteck Wee, editors, *TCC 2023, Part III*, volume 14371 of *LNCS*, pages 301–330. Springer, Cham, November / December 2023. doi:10.1007/978-3-031-48621-0_11. (Cited on p. 7).
- BDZ03. Feng Bao, Robert H. Deng, and Huafei Zhu. Variations of Diffie-Hellman problem. In Sihan Qing, Dieter Gollmann, and Jianying Zhou, editors, *ICICS 03*, volume 2836 of *LNCS*, pages 301–312. Springer, Berlin, Heidelberg, October 2003. doi:10.1007/978-3-540-39927-8_28. (Cited on pp. 3 and 8).
- BR09. Mihir Bellare and Thomas Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for Waters’ IBE

- scheme. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 407–424. Springer, Berlin, Heidelberg, April 2009. doi:10.1007/978-3-642-01001-9_24. (Cited on pp. 6 and 22).
- CGLQ20. Kai-Min Chung, Siyao Guo, Qipeng Liu, and Luowen Qian. Tight quantum time-space tradeoffs for function inversion. In *61st FOCS*, pages 673–684. IEEE Computer Society Press, November 2020. doi:10.1109/FOCS46700.2020.00068. (Cited on p. 6).
- CK18. Henry Corrigan-Gibbs and Dmitry Kogan. The discrete-logarithm problem with preprocessing. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 415–447. Springer, Cham, April / May 2018. doi:10.1007/978-3-319-78375-8_14. (Cited on pp. 3 and 7).
- CLO25. David A. Cox, John Little, and Donal O’Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Undergraduate Texts in Mathematics. Springer Nature Switzerland, Cham, 2025. doi:10.1007/978-3-031-91841-4. (Cited on pp. 15, 17, 29, and 30).
- DKL14. Zeev Dvir, János Kollár, and Shachar Lovett. Variety Evasive Sets. *computational complexity*, 23(4):509–529, December 2014. doi:10.1007/s00037-013-0073-9. (Cited on p. 30).
- EHK⁺13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Berlin, Heidelberg, August 2013. doi:10.1007/978-3-642-40084-1_8. (Cited on pp. 5 and 9).
- FGK22. Cody Freitag, Ashrujit Ghoshal, and Ilan Komargodski. Time-space tradeoffs for Sponge hashing: Attacks and limitations for short collisions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 131–160. Springer, Cham, August 2022. doi:10.1007/978-3-031-15982-4_5. (Cited on p. 6).
- FGK23. Cody Freitag, Ashrujit Ghoshal, and Ilan Komargodski. Optimal security for keyed hash functions: Avoiding time-space tradeoffs for finding collisions. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part IV*, volume 14007 of *LNCS*, pages 440–469. Springer, Cham, April 2023. doi:10.1007/978-3-031-30634-1_15. (Cited on p. 6).
- 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. doi:10.1007/978-3-319-96881-0_2. (Cited on pp. 3 and 7).
- GGKL21. Nick Gravin, Siyao Guo, Tsz Chiu Kwok, and Pinyan Lu. Concentration bounds for almost k -wise independence with applications to non-uniform security. In Dániel Marx, editor, *32nd SODA*, pages 2404–2423. ACM-SIAM, January 2021. doi:10.1137/1.9781611976465.143. (Cited on pp. 6 and 7).
- GK22. Ashrujit Ghoshal and Ilan Komargodski. On time-space tradeoffs for bounded-length collisions in merkle-Damgård hashing. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 161–191. Springer, Cham, August 2022. doi:10.1007/978-3-031-15982-4_6. (Cited on p. 6).

- Har77. Robin Hartshorne. *Algebraic Geometry*, volume 52 of *Graduate Texts in Mathematics*. Springer, New York, NY, 1977. doi:10.1007/978-1-4757-3849-0. (Cited on pp. 15 and 29).
- Hei83. Joos Heintz. Definability and fast quantifier elimination in algebraically closed fields. *Theoretical Computer Science*, 24(3):239–277, August 1983. doi:10.1016/0304-3975(83)90002-6. (Cited on pp. 15 and 30).
- Hha25. Minki Hhan. A new approach to generic lower bounds - classical/quantum MDL, quantum factoring, and more. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part VII*, volume 15607 of *LNCS*, pages 345–374. Springer, Cham, May 2025. doi:10.1007/978-3-031-91098-2_13. (Cited on pp. 3 and 7).
- Imp11. Russell Impagliazzo. Relativized Separations of Worst-Case and Average-Case Complexities for NP. In *2011 IEEE 26th Annual Conference on Computational Complexity*, pages 104–114, June 2011. doi:10.1109/CCC.2011.34. (Cited on p. 6).
- LW54. Serge Lang and André Weil. Number of Points of Varieties in Finite Fields. *American Journal of Mathematics*, 76(4):819–827, 1954. arXiv:2372655, doi:10.2307/2372655. (Cited on p. 30).
- Mau05. Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, Berlin, Heidelberg, December 2005. doi:10.1007/11586821_1. (Cited on p. 9).
- MRV15. Paz Morillo, Carla Ràfols, and Jorge L. Villar. Matrix computational assumptions in multilinear groups. Cryptology ePrint Archive, Paper 2015/353, 2015. URL: <https://eprint.iacr.org/2015/353>, doi:10.1007/978-3-662-53887-6_27. (Cited on p. 5).
- MRV16. Paz Morillo, Carla Ràfols, and Jorge Luis Villar. The kernel matrix Diffie-Hellman assumption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 729–758. Springer, Berlin, Heidelberg, December 2016. doi:10.1007/978-3-662-53887-6_27. (Cited on pp. 5, 8, 9, and 10).
- MW96. Ueli M. Maurer and Stefan Wolf. Diffie-Hellman oracles. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 268–282. Springer, Berlin, Heidelberg, August 1996. doi:10.1007/3-540-68697-5_21. (Cited on p. 8).
- PW17. David Pointcheval and Guilin Wang. VTBPEKE: Verifier-based two-basis password exponential key exchange. In Ramesh Karri, Ozgur Sinanoglu, Ahmad-Reza Sadeghi, and Xun Yi, editors, *ASIACCS 17*, pages 301–312. ACM Press, April 2017. doi:10.1145/3052973.3053026. (Cited on p. 8).
- 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. doi:10.1007/3-540-69053-0_18. (Cited on pp. 3 and 9).
- SS01. Ahmad-Reza Sadeghi and Michael Steiner. Assumptions related to discrete logarithms: Why subtleties make a real difference. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 244–261. Springer, Berlin, Heidelberg, May 2001. doi:10.1007/3-540-44987-6_16. (Cited on pp. 3 and 8).
- Tao12. Terence Tao. The Lang-Weil bound, September 2012. (Cited on p. 30).

- Yun15. Aaram Yun. Generic hardness of the multiple discrete logarithm problem. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 817–836. Springer, Berlin, Heidelberg, April 2015. [doi:10.1007/978-3-662-46803-6_27](https://doi.org/10.1007/978-3-662-46803-6_27). (Cited on pp. 3, 7, 10, and 20).