Relations Between the Uniform MQ Assumption and Other Multivariate Assumptions

Zijun Zhuang[1][0009−0007−7729−5790], Yingjie Zhang[2][0000−0001−9745−0558], and Jintai Ding[3][0000−0003−1257−7598]

[1] Tsinghua University, 30 Shuangqing Rd, Haidian District, Beijing 100084, China
zhuangzj22@mails.tsinghua.edu.cn
[2] Beijing Institute of Mathematical Sciences and Applications, Yanqi Lake West Rd, Huairou District, Beijing 101408, China
zyj.crypto@outlook.com
[3] Xi'an Jiaotong-Liverpool University, 111 Ren'ai Rd, Suzhou Industrial Park, Suzhou 215123, China
jintai.ding@gmail.com

Abstract. The uniform multivariate quadratic (UMQ) assumption states that it is hard to find a zero of a uniformly generated MQ function. It is the average-case hardness assumption about the MQ problem. In this paper, we investigate the relations among the UMQ assumption, the MQ one-wayness (MQOW) assumption, and the MQ second-preimage resistance (MQSPR) assumption.

We show that UMQ and MQSPR tightly imply each other, and MQOW tightly implies UMQ. Then, we show that UMQ implies MQOW when $m \leq n + O(\log \lambda)$, where n is the number of variables, m is the number of MQ equations, and λ is the security parameter. In particular, when $m \leq n + O(1)$, this implication is tight.

As a corollary, we show that MQSPR implies MQOW under the same condition $m \leq n + O(\log \lambda)$, which is weaker than the compression condition $n = m + \omega(\log \lambda)$ required for the implication from SPR to OW for general function families. In particular, our result covers the square case $m = n$ as well as mildly overdetermined MQ systems satisfying $m = n + O(\log \lambda)$.

Keywords: Post-Quantum Cryptography · Multivariate Cryptography · Multivariate Assumptions · Reductions

1 Introduction

The multivariate quadratic (MQ) problem asks one to find a zero of an MQ function over a finite field. It is well-studied in multivariate cryptography [15]: on the one hand, it is **NP**-hard in the worst case [25]; on the other hand, it is widely believed to be hard on average for an extensive range of parameters, since research on algorithms for solving MQ systems [3,8,9,18–21,23,33,38] indicates that the best known algorithms have exponential complexity when $m = \Theta(n)$, where n is the number of variables and m is the number of MQ equations.

The MQ problem has many applications in cryptography, including pseudorandom generators [5,32], encryption schemes [16,27,37], signature schemes [4,6,12,29], commitment schemes [10], oblivious transfer protocols [31], zero-knowledge protocols [13,36], hash functions [7,17,39], and proof-of-work for blockchain [14]. Since there are many average-case assumptions related to the security of these constructions, we introduce some of them and investigate their relations in this paper.

MQ One-Wayness (MQOW) Assumption. One-way functions are fundamental cryptographic primitives [26,28,35]. The MQ one-wayness (MQOW) [12] assumption states that the MQ function family is one-way: given an MQ function and an output generated from an unknown input, it is computationally hard to find a preimage. Thus, when we consider the MQOW experiment, it is guaranteed that a preimage exists.

MQOW has been used to construct a variety of cryptographic primitives. Berbain et al. [5] constructed the QUAD pseudorandom generator, whose security is reduced to MQOW. Sakumoto et al. [36] constructed public-key identification schemes based on MQOW. Building on their identification protocol, Chen et al. [12] proposed MQDSS, the first provably secure multivariate signature scheme. Two works in 2024 [4,22] used the MPC-in-the-Head paradigms to construct signature schemes based on MQOW.

MQ Second-Preimage Resistance (MQSPR) Assumption. Second-preimage resistance (SPR) is a standard security property of cryptographic hash functions [34]. Since MQ functions can be used to construct hash functions [1, 7, 17], we consider the MQ second-preimage resistance (MQSPR) assumption. The MQSPR assumption states that the MQ function family is second-preimage resistant [36, 40]: given an MQ function and a preimage, it is computationally hard to find another preimage with the same image. From the perspective of solving an MQ system, MQSPR says that the given solution does not reveal enough information on other solutions.

MQSPR has been used to establish the security of some cryptosystems. Sakumoto et al. [36] used MQSPR to prove that the parallel repetition of their MQ-based identification protocol is secure against impersonation under active attacks. Zhuang et al. [40] used MQSPR and other properties of MQ functions to obtain a provably secure MQ-based one-time signature scheme in the standard model.

Uniform MQ (UMQ) Assumption. The uniform MQ (UMQ) assumption is the average-case hardness assumption about the MQ problem [13], which states that it is computationally hard to find a zero of an MQ function sampled uniformly at random. UMQ is different from MQOW, since a randomly generated MQ function may not have a zero.

UMQ has attracted growing attention in cryptography due to its applications in constructing non-interactive zero-knowledge (NIZK) arguments and ZAPs (two-message public-coin witness-indistinguishable proofs). Dao et al. (CRYPTO 2024) constructed NIZK arguments based on UMQ and other assumptions [13]. In their construction, UMQ is used to obtain correlation-intractable hash func-

tions [11]. Banerjee et al. (CRYPTO 2026) applied a generic NIZK-to-ZAP compiler to this NIZK construction, and obtained secure ZAP arguments for **NP** [2].

Their Differences and Relations. Although UMQ, MQOW, and MQSPR are all natural average-case assumptions related to the MQ problem, they involve different instance distributions and computational tasks. In UMQ, the MQ system is sampled uniformly at random and may have no solution. In MQOW, the system is generated together with an unknown solution. In MQSPR, the associated MQ system has a known solution (i.e., the given preimage), but the goal is to find another one. These differences make the reduction relations among the three assumptions nontrivial. Although it has been shown that MQOW implies MQSPR [40], clarifying the role of UMQ among these multivariate assumptions is important both for the theory of multivariate cryptography and for the construction of provably secure cryptographic schemes based on the MQ problem.

Contributions

Let $\mathcal{Q}_{n,m,q}$ denote the MQ function family consisting of all MQ functions from $\mathbb{F}_q^n$ to $\mathbb{F}_q^m$. Let $\mathrm{InSec}^{\mathsf{P}}(\mathcal{Q}_{n,m,q};t)$ denote the maximum success probability of any adversary running in time t against property P of $\mathcal{Q}_{n,m,q}$, and let $\mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(t)$ denote the maximum success probability of any algorithm running in time t and solving a UMQ instance. Let λ be the security parameter.

Our contributions are listed as follows:

- We show that UMQ and MQSPR tightly imply each other: for each $t > 0$, we have
$$\mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(t) \le \mathrm{InSec}^{\mathsf{SPR}}\big(\mathcal{Q}_{n+1,m,q};t'\big),$$
and
$$\mathrm{InSec}^{\mathsf{SPR}}\big(\mathcal{Q}_{n+1,m,q};t\big) \le q \cdot \mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(t'),$$
where $t' = t + \mathrm{poly}(n,m,\log q)$.
- We show that MQOW tightly implies UMQ: for each $t > 0$, we have
$$\mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(t) \le \frac{q}{q-1}\,\mathrm{InSec}^{\mathsf{OW}}(\mathcal{Q}_{n+1,m,q};t'),$$
where $t' = t + \mathrm{poly}(n,m,\log q)$.
- We show that when $m \le n + O(\log\lambda)$, UMQ implies MQOW: for each $t > 0$, we have
$$\mathrm{InSec}^{\mathsf{OW}}\big(\mathcal{Q}_{n+1,m,q};t\big) \le \big(q^{m-n} + q - q^{-n}\big)\,\mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(t'),$$
where $t' = t + \mathrm{poly}(n,m,\log q)$. When $m < n + O(1)$, this implication is tight.
- As a corollary, MQSPR implies MQOW under the condition $m \le n + O(\log\lambda)$, which is weaker than the compression condition $n = m + \omega(\log\lambda)$ required for the implication from SPR to OW for general function families [34]. In particular, our result covers the square case $m = n$ as well as mildly overdetermined systems satisfying $m = n + O(\log\lambda)$.

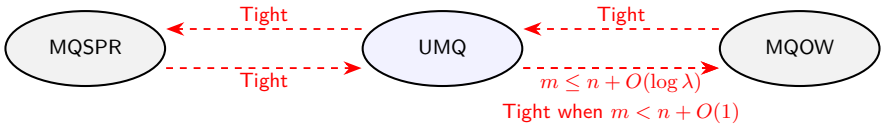

Figure 1. Red lines highlight our core technical contributions. An arrow $A \dashrightarrow B$ means that if the assumption A holds, then the assumption B holds. In other words, any efficient algorithm breaking assumption B can be transformed into an efficient algorithm breaking assumption A.

Figure 1 summarizes these contributions. Taken together, our results clarify the role of UMQ among several natural multivariate assumptions and provide a more unified foundation for the security of MQ-based cryptographic constructions.

Overview

We introduce the necessary preliminaries in Section 2, establish the equivalence between UMQ and MQSPR in Section 3, study the relations between UMQ and MQOW in Section 4, and conclude the paper in Section 5.

2 Preliminaries

For notation convenience, when we talk about computational assumptions, we set the number of multivariate quadratic (MQ) equations $m = m(\lambda)$, the number of variables $n = n(\lambda)$, the size of the underlying field $q = q(\lambda)$, and the running time $t = t(\lambda)$, where λ is the security parameter. We use "p.p.t." to refer to the term "probabilistic polynomial-time". For functions $g, h : \mathbb{N}^+ \to \mathbb{N}^+$, the notation $g(\lambda) = O(h(\lambda))$ means that there exists a constant $C > 0$ such that $g(\lambda) \leq C \cdot h(\lambda)$ for sufficiently large λ, and the notation $g(\lambda) = \omega(h(\lambda))$ means that $\lim_{\lambda \to +\infty} \frac{g(\lambda)}{h(\lambda)} = +\infty$. We use $[m]$ to denote the set $\{1, 2, ..., m\}$.

We consider the following keyed function family of MQ functions.

Definition 1 (MQ Function Family [40]). *We define the MQ function family to be*

$$\mathcal{Q}_{n,m,q} := \left\{ f_{\mathbf{A},\mathbf{b},\boldsymbol{c}} : \mathbb{F}_q^n \to \mathbb{F}_q^m \;\middle|\; \begin{array}{c} f_{\mathbf{A},\mathbf{b},\boldsymbol{c}}(\mathbf{x}) = \left(\mathbf{x}^\mathsf{T}\mathbf{A}_i\mathbf{x} + \mathbf{b}_i^\mathsf{T}\mathbf{x} + c_i\right)_{i=1}^m, \\ \textit{where } \mathbf{A}_i \in \mathbb{F}_q^{n\times n} \textit{ is upper triangular}, \\ \mathbf{b}_i \in \mathbb{F}_q^n,\ c_i \in \mathbb{F}_q, \quad i = 1, \ldots, m. \end{array} \right\},$$

where we use the coefficient tuple $k = (\mathbf{A}, \mathbf{b}, \boldsymbol{c}) = (\mathbf{A}_i, \mathbf{b}_i, c_i)_{i=1}^m \in \mathcal{K}_{n,m,q} \triangleq \mathcal{K} \triangleq \mathbb{F}_q^{N_{n,m}}$ *as the key, with* $N_{n,m} = \frac{1}{2}m(n+1)(n+2)$.

Now we formalize the MQ problem:

Definition 2 (MQ Problem [15]). *The MQ problem is defined as follows.*
Input: *Positive integers* n, m, q, *and an MQ function* $f \in \mathcal{Q}_{n,m,q}$.
Output: *A vector* $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ *such that* $f(\mathbf{x}) = \mathbf{0}$.

The MQ one-wayness (MQOW) assumption is the hardness assumption of the average-case MQ problem with a planted solution:

Definition 3 (MQOW Assumption [12]). *The* (n, m, q)*-MQOW assumption states that the MQ function family* $\mathcal{Q}_{n,m,q}$ *is one-way: given an MQ function* f *uniformly sampled from* $\mathcal{Q}_{n,m,q}$ *and an image vector* $\mathbf{y} = f(\mathbf{x})$*, where the unknown vector* $\mathbf{x}$ *is uniformly sampled from* $\mathbb{F}_q^n$*, it is computationally infeasible to find a preimage* $\mathbf{x}'$ *of* $\mathbf{y}$.

Equivalently, for each polynomial time t*, we have*

$$\mathrm{InSec}^{\mathsf{OW}}(\mathcal{Q}_{n,m,q}; t) := \sup_{\mathcal{A}} \mathrm{Succ}^{\mathsf{OW}}_{\mathcal{Q}_{n,m,q}}(\mathcal{A}) = \mathrm{negl}(\lambda),$$

where the supremum is taken over all p.p.t. algorithms running in time t*, and for each p.p.t. adversary* $\mathcal{A}$*, its success probability is defined to be*

$$\mathrm{Succ}^{\mathsf{OW}}_{\mathcal{Q}_{n,m,q}}(\mathcal{A}) := \Pr[f \xleftarrow{\$} \mathcal{Q}_{n,m,q},\ \mathbf{x} \xleftarrow{\$} \mathbb{F}_q^n,\ \mathbf{y} \leftarrow f(\mathbf{x}),\ \mathbf{x}' \leftarrow \mathcal{A}(f, \mathbf{y}) : \mathbf{x}' \in \mathbb{F}_q^n \ \wedge\ f(\mathbf{x}') = \mathbf{y}],$$

and the probability is over f*,* $\mathbf{x}$*, and* $\mathcal{A}$.

When $m = \Theta(n)$, research on specific algorithms indicates that the MQOW assumptions is widely believed to hold [3, 8, 9, 18–21, 23, 33, 38].

Definition 4 (MQ Second-Preimage Resistance (MQSPR) Assumption [36]). *The* (n, m, q)*-MQSPR assumption states that the MQ function family* $\mathcal{Q}_{n,m,q}$ *is second-preimage resistant: given an MQ function* f *uniformly sampled from* $\mathcal{Q}_{n,m,q}$ *and a vector* $\mathbf{x}$ *uniformly sampled from* $\mathbb{F}_q^n$*, it is computationally infeasible to find a second-preimage* $\mathbf{x}' \neq \mathbf{x}$ *of* $f(\mathbf{x})$.

Equivalently, for each polynomial time t*, we have*

$$\mathrm{InSec}^{\mathsf{SPR}}(\mathcal{Q}_{n,m,q}; t) := \sup_{\mathcal{A}} \mathrm{Succ}^{\mathsf{SPR}}_{\mathcal{Q}_{n,m,q}}(\mathcal{A}) = \mathrm{negl}(\lambda),$$

where the supremum is taken over all p.p.t. algorithms running in time t*, and for each p.p.t. adversary* $\mathcal{A}$*, its success probability is defined to be*

$$\mathrm{Succ}^{\mathsf{SPR}}_{\mathcal{Q}_{n,m,q}}(\mathcal{A}) := \Pr[f \xleftarrow{\$} \mathcal{Q}_{n,m,q},\ \mathbf{x} \xleftarrow{\$} \mathbb{F}_q^n,\ \mathbf{x}' \leftarrow \mathcal{A}(f, \mathbf{x}) : \mathbf{x}' \in \mathbb{F}_q^n \ \wedge\ \mathbf{x}' \neq \mathbf{x} \ \wedge\ f(\mathbf{x}') = f(\mathbf{x})],$$

where the probability is over f*,* $\mathbf{x}$*, and* $\mathcal{A}$.

Now we define the uniform MQ (UMQ) assumption. The "uniform" means the uniformnity of the coefficients.

Definition 5 (Uniform MQ (UMQ) Assumption [13]). *The (n, m, q)-UMQ assumption states that, no p.p.t. algorithm can solve the average-case search MQ problem: for each polynomial time t,*

$$\mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(t) := \sup_{\mathcal{A}} \mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(\mathcal{A}) = \mathrm{negl}(\lambda),$$

where the supremum is taken over all p.p.t. algorithms running in time t, and for each p.p.t. algorithm $\mathcal{A}$, its success probability is defined to be

$$\mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(\mathcal{A}) := \Pr[f \xleftarrow{\$} \mathcal{Q}_{n,m,q},\ \mathbf{x} \leftarrow \mathcal{A}(f) : \mathbf{x} \in \mathbb{F}_q^n \ \wedge \ f(\mathbf{x}) = \mathbf{0}],$$

where the probability is over f and $\mathcal{A}$.

3 Equivalence between UMQ and MQSPR

In this section, we give an equivalent assumption of MQSPR, and show that UMQ and MQSPR imply each other.

3.1 Zero-Constant MQ (ZCMQ): an equivalent assumption of MQSPR

The zero-constant MQ (ZCMQ) problem considers the MQ function without constants. To describe it properly, we denote $\mathcal{K}^{\mathsf{ZCMQ}}_{n,m,q} = \mathbb{F}_q^{N_{n,m}-m}$.

Definition 6 (ZCMQ Problem). *Suppose n, m, q are integers. The (n, m, q)-ZCMQ problem is defined as follows.*

Input: *A coefficient tuple without constants $(\mathbf{A}, \mathbf{b}) \in \mathcal{K}^{\mathsf{ZCMQ}}_{n,m,q}$.*
Output: *A non-zero vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ such that $f_{\mathbf{A},\mathbf{b},\mathbf{0}}(\mathbf{x}) = \mathbf{0}$.*

Definition 7 (ZCMQ Assumption [1]). *The (n, m, q)-ZCMQ assumption states that, no p.p.t. algorithm can solve the average-case (n, m, q)-ZCMQ problem efficiently: for each polynomial time t,*

$$\mathrm{Succ}^{\mathsf{ZCMQ}}_{n,m,q}(t) := \sup_{\mathcal{A}} \mathrm{Succ}^{\mathsf{ZCMQ}}_{n,m,q}(\mathcal{A}) = \mathrm{negl}(\lambda),$$

where the supremum is taken over all p.p.t. algorithms running in time t, and for each p.p.t. algorithm $\mathcal{A}$, its success probability is defined to be

$$\mathrm{Succ}^{\mathsf{ZCMQ}}_{n,m,q}(\mathcal{A}) := \Pr[(\mathbf{A}, \mathbf{b}) \xleftarrow{\$} \mathcal{K}^{\mathsf{ZCMQ}}_{n,m,q}, \mathbf{x} \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{b}) : \mathbf{x} \in \mathbb{F}_q^n \setminus \{\mathbf{0}\} \ \wedge \ f_{\mathbf{A},\mathbf{b},\mathbf{0}}(\mathbf{x}) = \mathbf{0}],$$

where the probability is over $(\mathbf{A}, \mathbf{b})$ and $\mathcal{A}$.

Lemma 1 (ZCMQ $\Rightarrow$ MQSPR [1]). *Under the (n, m, q)-ZCMQ assumption, the (n, m, q)-MQSPR assumption holds: for each t, we have*

$$\mathrm{InSec}^{\mathsf{SPR}}(\mathcal{Q}_{n,m,q}; t) \leq \mathrm{Succ}^{\mathsf{ZCMQ}}_{n,m,q}(t'),$$

where $t' = t + \mathrm{poly}(n, m, \log q)$.

Algorithm 1 The MQSPR adversary $\mathcal{B}^{\mathcal{A}}$

Input: An (n, m, q)-MQSPR challenge $f = f_{\mathbf{A},\mathbf{b},\mathbf{c}} \in \mathcal{Q}_{n,m,q}$, a vector $\mathbf{x} \xleftarrow{\$} \mathbb{F}_q^n$, and oracle access to an (n, m, q)-ZCMQ solver $\mathcal{A}$
Output: A second preimage $\mathbf{x}' \in \mathbb{F}_q^n$ such that $\mathbf{x}' \neq \mathbf{x}$ and $f(\mathbf{x}') = f(\mathbf{x})$, or $\perp$

for $i = 1, 2, \ldots, m$ **do**
  Set $\widetilde{\mathbf{A}}_i \leftarrow \mathbf{A}_i$, $\widetilde{\mathbf{b}}_i \leftarrow \mathbf{b}_i + (\mathbf{A}_i + \mathbf{A}_i^{\mathsf{T}})\mathbf{x}$
end for
Run $\mathcal{A}$ on the constructed (n, m, q)-ZCMQ instance $(\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i)_{i=1}^m$
$\mathbf{z} \leftarrow \mathcal{A}\big((\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i)_{i=1}^m\big)$
if $\mathbf{z} = \perp$ **or** $\mathbf{z} = \mathbf{0}$ **or** $\mathbf{z}^{\mathsf{T}}\widetilde{\mathbf{A}}_i\mathbf{z} + \widetilde{\mathbf{b}}_i^{\mathsf{T}}\mathbf{z} \neq 0$ for some $i \in [m]$ **then**
  return $\perp$
end if
$\mathbf{x}' \leftarrow \mathbf{x} + \mathbf{z}$
if $\mathbf{x}' = \mathbf{x}$ **or** $f(\mathbf{x}') \neq f(\mathbf{x})$ **then**
  return $\perp$
end if
return $\mathbf{x}'$

The proof intuition of Lemma 1 is to consider the differential $\Delta = \mathbf{x} - \mathbf{x}'$ and transform the given MQSPR instance into a ZCMQ instance in Δ. This transformation can be reused in the proof of MQSPR $\Rightarrow$ ZCMQ.

Proposition 1 (MQSPR $\Rightarrow$ ZCMQ). *Under the (n, m, q)-MQSPR assumption, the (n, m, q)-ZCMQ assumption holds: for each t, we have*

$$\mathrm{Succ}_{n,m,q}^{\mathsf{ZCMQ}}(t) \leq \mathrm{InSec}^{\mathsf{SPR}}(\mathcal{Q}_{n,m,q}; t'),$$

where $t' = t + \mathrm{poly}(n, m, \log q)$.

Proof. Let $\mathcal{A}$ be a p.p.t. (n, m, q)-ZCMQ solver running in time at most t. We construct an (n, m, q)-MQSPR adversary $\mathcal{B}^{\mathcal{A}}$ in Algorithm 1.

Overview of $\mathcal{B}^{\mathcal{A}}$. $\mathcal{B}^{\mathcal{A}}$ considers the differential map $\widetilde{f}(\mathbf{z}) := f(\mathbf{x}+\mathbf{z}) - f(\mathbf{x})$. This map has zero constant term and can be supplied to $\mathcal{A}$ as a ZCMQ instance. If $\mathcal{A}$ returns a nonzero root $\mathbf{z}$ of $\widetilde{f}$, then $\mathcal{B}^{\mathcal{A}}$ outputs $\mathbf{x}' = \mathbf{x} + \mathbf{z}$.

Correctness analysis. For every $i \in [m]$, write $f_i(\mathbf{u}) = \mathbf{u}^{\mathsf{T}}\mathbf{A}_i\mathbf{u} + \mathbf{b}_i^{\mathsf{T}}\mathbf{u} + c_i$. For any $\mathbf{z} \in \mathbb{F}_q^n$, we have

$$\begin{aligned} f_i(\mathbf{x}+\mathbf{z}) - f_i(\mathbf{x}) &= \mathbf{z}^{\mathsf{T}}\mathbf{A}_i\mathbf{z} + \big(\mathbf{b}_i + (\mathbf{A}_i + \mathbf{A}_i^{\mathsf{T}})\mathbf{x}\big)^{\mathsf{T}}\mathbf{z} \\ &= \mathbf{z}^{\mathsf{T}}\widetilde{\mathbf{A}}_i\mathbf{z} + \widetilde{\mathbf{b}}_i^{\mathsf{T}}\mathbf{z} = \widetilde{f}_i(\mathbf{z}). \end{aligned}$$

Suppose that $\mathcal{A}$ succeeds on the ZCMQ instance $(\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i)_{i=1}^m$ and outputs $\mathbf{z}$. Then $\mathbf{z} \neq \mathbf{0}$ and $\widetilde{f}(\mathbf{z}) = \mathbf{0}$. Hence we have $f(\mathbf{x}+\mathbf{z}) = f(\mathbf{x})$, and the vector $\mathbf{x}' = \mathbf{x} + \mathbf{z}$ outputed by $\mathcal{B}^{\mathcal{A}}$ is a second preimage of $f(\mathbf{x})$.

Distribution of the constructed ZCMQ instance. We now show that $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{b}}) = (\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i)_{i=1}^m$ is a uniform ZCMQ instance. Fix an arbitrary $\mathbf{x} \in \mathbb{F}_q^n$. Consider the map $\phi_{\mathbf{x}} : \mathcal{K}_{n,m,q}^{\mathsf{ZCMQ}} \to \mathcal{K}_{n,m,q}^{\mathsf{ZCMQ}}$, $(\mathbf{A}_i, \mathbf{b}_i)_{i=1}^m \mapsto (\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i)_{i=1}^m = (\mathbf{A}_i, \mathbf{b}_i + (\mathbf{A}_i + \mathbf{A}_i^\mathsf{T})\mathbf{x})_{i=1}^m$. This map is a bijection, with inverse $(\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i)_{i=1}^m \mapsto (\mathbf{A}_i, \mathbf{b}_i)_{i=1}^m = (\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i - (\widetilde{\mathbf{A}}_i + \widetilde{\mathbf{A}}_i^\mathsf{T})\mathbf{x})_{i=1}^m$. Since $(\mathbf{A}_i, \mathbf{b}_i)_{i=1}^m$ is uniform over $\mathcal{K}_{n,m,q}^{\mathsf{ZCMQ}}$, $(\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i)_{i=1}^m$ is uniform over $\mathcal{K}_{n,m,q}^{\mathsf{ZCMQ}}$ and is a uniform ZCMQ instance.

Success probability. Let E_{ZCMQ} denote the event that $\mathcal{A}$ succeeds on the constructed ZCMQ instance, i.e., $E_{\mathsf{ZCMQ}} := \left\{\mathbf{z} \in \mathbb{F}_q^n \setminus \{\mathbf{0}\} \ \wedge \ \widetilde{f}(\mathbf{z}) = \mathbf{0}\right\}$. By the correctness analysis, when E_{ZCMQ} occurs, $\mathcal{B}^{\mathcal{A}}$ outputs $\mathbf{x}' = \mathbf{x} + \mathbf{z} \neq \mathbf{x}$ and succeeds. Conversely, by Algorithm 1, $\mathcal{B}^{\mathcal{A}}$ returns a candidate second preimage only after verifying $\mathbf{z} \neq \mathbf{0}$ and $\widetilde{f}(\mathbf{z}) = \mathbf{0}$. Thus the success event of $\mathcal{B}^{\mathcal{A}}$ is E_{ZCMQ} and we have

$$\mathrm{Succ}_{n,m,q}^{\mathsf{ZCMQ}}(\mathcal{A}) = \Pr[E_{\mathsf{ZCMQ}}] = \mathrm{Succ}_{\mathcal{Q}_{n,m,q}}^{\mathsf{SPR}}\left(\mathcal{B}^{\mathcal{A}}\right) \leq \mathrm{InSec}^{\mathsf{SPR}}\left(\mathcal{Q}_{n,m,q}; t'\right).$$

Since $\mathcal{B}^{\mathcal{A}}$ invokes $\mathcal{A}$ once, and all remaining operations take $\mathrm{poly}(n, m, \log q)$ time, $\mathcal{B}^{\mathcal{A}}$ runs in time $t' = t + \mathrm{poly}(n, m, \log q)$. Taking the supremum over all algorithms $\mathcal{A}$ running in time at most t yields

$$\mathrm{Succ}_{n,m,q}^{\mathsf{ZCMQ}}(t) \leq \mathrm{InSec}^{\mathsf{SPR}}\left(\mathcal{Q}_{n,m,q}; t'\right),$$

completing the proof. □

3.2 MQSPR ⇒ UMQ

Now we show that MQSPR implies UMQ. From Section 3.1 we know that it suffices to show that ZCMQ implies UMQ.

Theorem 1 (ZCMQ ⇒ UMQ). *Under the $(n+1, m, q)$-ZCMQ assumption, the (n, m, q)-UMQ assumption holds: for each t, we have*

$$\mathrm{Succ}_{n,m,q}^{\mathsf{UMQ}}(t) \leq \mathrm{Succ}_{n+1,m,q}^{\mathsf{ZCMQ}}(t'),$$

where $t' = t + \mathrm{poly}(n, m, \log q)$.

Proof idea. The key is to transform an $(n+1, m, q)$-ZCMQ instance into an (n, m, q)-UMQ instance. This can be done by restricting one variable (we choose the last variable) to be some non-zero constant (we choose 1) in the $(n+1, m, q)$-ZCMQ instance.

Proof. Let $\mathcal{A}$ be a p.p.t. algorithm that runs in time t, and solves (n, m, q)-UMQ instances with success probability $\epsilon = \mathrm{Succ}_{n,m,q}^{\mathsf{UMQ}}(\mathcal{A})$. We construct an algorithm $\mathcal{B} = \mathcal{B}^{\mathcal{A}}$ that can solve $(n+1, m, q)$-ZCMQ instances. We construct $\mathcal{B}^{\mathcal{A}}$ is shown in Algorithm 2.

Algorithm 2 The ZCMQ solver $\mathcal{B}^{\mathcal{A}}$

Input: A random $(n+1,m,q)$-ZCMQ instance $(\mathbf{A}_i,\mathbf{b}_i)_{i=1}^m \in \mathcal{K}_{n+1,m,q}^{\mathsf{ZCMQ}}$, and oracle access to an (n,m,q)-UMQ solver $\mathcal{A}$

Output: A nonzero solution $\mathbf{x}^* \in \mathbb{F}_q^{n+1}$ to the ZCMQ instance, or $\perp$

1: **for** $i = 1, 2, \ldots, m$ **do**
2: Parse the upper-triangular matrix and the linear coefficient vector as $\mathbf{A}_i = \begin{pmatrix} \widetilde{\mathbf{A}}_i & \mathbf{u}_i \\ \mathbf{0}^\mathsf{T} & a_i \end{pmatrix}, \mathbf{b}_i = \begin{pmatrix} \mathbf{v}_i \\ d_i \end{pmatrix}$, where $\widetilde{\mathbf{A}}_i \in \mathbb{F}_q^{n\times n}$ is upper triangular, $\mathbf{u}_i, \mathbf{v}_i \in \mathbb{F}_q^n$, and $a_i, d_i \in \mathbb{F}_q$.
3: Set $\widetilde{\mathbf{b}}_i \leftarrow \mathbf{u}_i + \mathbf{v}_i, \widetilde{c}_i \leftarrow a_i + d_i$.
4: **end for**
5: Let $\widetilde{f}$ be the MQ function defined by $(\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i, \widetilde{c}_i)_{i=1}^m$
6: Run $\mathcal{A}(\widetilde{f})$ and receive its output $\mathbf{x}^* \leftarrow \mathcal{A}(\widetilde{f})$
7: **if** $\mathbf{x}^* = \perp$ **or** $(\mathbf{x}^*)^\mathsf{T}\widetilde{\mathbf{A}}_i\mathbf{x}^* + \widetilde{\mathbf{b}}_i^\mathsf{T}\mathbf{x}^* + \widetilde{c}_i \neq 0$ for some $i \in [m]$ **then**
8: **return** $\perp$
9: **end if**
10: **return** $\widetilde{\mathbf{x}}^* \leftarrow \begin{pmatrix} \mathbf{x}^* \\ 1 \end{pmatrix}$

Overview of $\mathcal{B}^{\mathcal{A}}$. After receiving an $(n+1,m,q)$-ZCMQ instance, $\mathcal{B}^{\mathcal{A}}$ set the last variable to be 1, and obtain an (n,m,q)-UMQ instance. Then $\mathcal{B}$ invokes $\mathcal{A}$, receives the solution vector from $\mathcal{A}$, lifts it to be the solution of the original $(n+1,m,q)$-ZCMQ instance, and outputs it.

Correctness analysis. Let the input to $\mathcal{B}^{\mathcal{A}}$ be a uniformly random $(n+1,m,q)$-ZCMQ instance $(\mathbf{A}_i,\mathbf{b}_i)_{i=1}^m$. For every $i \in [m]$, write $\mathbf{A}_i = \begin{pmatrix} \widetilde{\mathbf{A}}_i & \mathbf{u}_i \\ \mathbf{0}^\mathsf{T} & a_i \end{pmatrix}, \mathbf{b}_i = \begin{pmatrix} \mathbf{v}_i \\ d_i \end{pmatrix}$, $\widetilde{\mathbf{b}}_i = \mathbf{u}_i + \mathbf{v}_i$, and $\widetilde{c}_i = a_i + d_i$. Denote the i-th polynomial in this ZCMQ instance to be $G_i(\mathbf{x}, y)$, where $\mathbf{x} \in \mathbb{F}_q^n$ and $y \in \mathbb{F}_q$, and we obtain

$$G_i(\mathbf{x},y) = \begin{pmatrix} \mathbf{x} \\ y \end{pmatrix}^\mathsf{T} \mathbf{A}_i \begin{pmatrix} \mathbf{x} \\ y \end{pmatrix} + \mathbf{b}_i^\mathsf{T} \begin{pmatrix} \mathbf{x} \\ y \end{pmatrix} = \mathbf{x}^\mathsf{T}\widetilde{\mathbf{A}}_i\mathbf{x} + y\mathbf{u}_i^\mathsf{T}\mathbf{x} + \mathbf{v}_i^\mathsf{T}\mathbf{x} + a_i y^2 + d_i y.$$

Then, $\mathcal{B}$ plugs in $y = 1$ and gets a polynomial in $\mathbf{x}$:

$$\widetilde{f}_i(\mathbf{x}) := G_i(\mathbf{x},1) = \mathbf{x}^\mathsf{T}\widetilde{\mathbf{A}}_i\mathbf{x} + (\mathbf{u}_i+\mathbf{v}_i)^\mathsf{T}\mathbf{x} + (a_i+d_i) = \mathbf{x}^\mathsf{T}\widetilde{\mathbf{A}}_i\mathbf{x} + \widetilde{\mathbf{b}}_i^\mathsf{T}\mathbf{x} + \widetilde{c}_i.$$

Thus $(\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i, \widetilde{c}_i)_{i=1}^m$ defines a search-MQ instance $\widetilde{f}$. Now we show that it is uniformly distributed.

Distribution of the constructed search-MQ instance. For each $i \in [m]$, since the coefficients $\widetilde{\mathbf{A}}_i$, $\mathbf{u}_i$, $\mathbf{v}_i$, a_i, and d_i are mutually independent and uniformly distributed over their respective coefficient spaces, we know that $\widetilde{\mathbf{A}}_i$, $\widetilde{\mathbf{b}}_i = \mathbf{u}_i + \mathbf{v}_i$, $\widetilde{c}_i = a_i + d_i$ are mutually independent and uniformly distributed over their respective coefficient spaces. Since coefficient tuples associated with different indices $i \in [m]$ are independent, $(\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i, \widetilde{c}_i)_{i=1}^m$ is uniformly distributed over $\mathcal{K}_{n,m,q}$, and $\widetilde{f}$ is an (n,m,q)-UMQ instance.

Success probability. Suppose that $\mathcal{A}$ returns a valid solution $\mathbf{x}^*$ of the derived UMQ instance $\widetilde{f}$. Then, for every $i \in [m]$, we have $G_i(\mathbf{x}^*, 1) = \widetilde{f}_i(\mathbf{x}^*) = 0$. Since $\widetilde{\mathbf{x}}^* = \begin{pmatrix} \mathbf{x}^* \\ 1 \end{pmatrix} \neq \mathbf{0}$, $\widetilde{\mathbf{x}}^*$ is a nonzero solution to the original ZCMQ instance $(\mathbf{A}_i, \mathbf{b}_i)_{i=1}^m$, and the success probability of $\mathcal{B}^{\mathcal{A}}$ is $\epsilon' = \epsilon$.

Constructing the (n, m, q)-UMQ instance and verifying the output require $\mathrm{poly}(n, m, \log q)$ additional time, thus $\mathcal{B}^{\mathcal{A}}$ runs in time at most $t' = t + \mathrm{poly}(n, m, \log q)$, and we obtain $\mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(\mathcal{A}) = \epsilon' \leq \mathrm{Succ}^{\mathsf{ZCMQ}}_{n+1,m,q}(t')$. Taking the supremum over all algorithms $\mathcal{A}$ running in time at most t, we obtain

$$\mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(t) \leq \mathrm{Succ}^{\mathsf{ZCMQ}}_{n+1,m,q}(t'),$$

completing the proof. □

Combining MQSPR ⇒ ZCMQ (Proposition 1) and ZCMQ ⇒ UMQ (Theorem 1), we have

Theorem 2 (MQSPR ⇒ UMQ). *Under the $(n+1, m, q)$-MQSPR assumption, the (n, m, q)-UMQ assumption holds: for each t, we have*

$$\mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(t) \leq \mathrm{InSec}^{\mathsf{SPR}}\big(\mathcal{Q}_{n+1,m,q}; t'\big),$$

where $t' = t + \mathrm{poly}(n, m, \log q)$.

Remark 1 (Tightness). Since the security loss is 1, our reduction is tight.

3.3 UMQ ⇒ MQSPR

Now we show that UMQ implies MQSPR. From Section 3.1 we know that it suffices to show that UMQ implies ZCMQ.

We introduce the following notation for convenience: for a square matrix $\mathbf{s}$, let $\mathsf{UpperTri}(\mathbf{s})$ denote the upper-triangular matrix representing the same quadratic form, namely

$$\mathsf{UpperTri}(\mathbf{s})_{rr} = S_{rr}, \qquad \mathsf{UpperTri}(\mathbf{s})_{rs} = S_{rs} + S_{sr} \quad (r < s).$$

Equivalently, for every vector $\mathbf{z}$, we have $\mathbf{z}^{\mathsf{T}}\mathsf{UpperTri}(\mathbf{s})\mathbf{z} = \mathbf{z}^{\mathsf{T}}\mathbf{s}\mathbf{z}$.

Theorem 3 (UMQ ⇒ ZCMQ). *Under the (n, m, q)-UMQ assumption, the $(n+1, m, q)$-ZCMQ assumption holds: for each t, we have*

$$\mathrm{Succ}^{\mathsf{ZCMQ}}_{n+1,m,q}(t) \leq (q - q^{-n}) \cdot \mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(t'),$$

where $t' = t + \mathrm{poly}(n, m, \log q)$

Proof idea. The idea is to choose one particular coordinate, add some randomness and transform a UMQ instance into a ZCMQ instance. We hope that the ZCMQ solver returns some "good" solution, which can be used to solve the UMQ instance. However, we can not control the output of the ZCMQ solver to satisfy our needs. To solve this problem, we utilize a random invertible linear transform to the variables and "blind" the ZCMQ solver.

Proof. Suppose we are given a p.p.t. algorithm $\mathcal{A}$ running in time t, solving an $(n+1, m, q)$-ZCMQ instance with success probability ϵ. We construct an (n, m, q)-UMQ solver $\mathcal{B}^{\mathcal{A}}$ in Algorithm 3.

Algorithm 3 The UMQ solver $\mathcal{B}^{\mathcal{A}}$

Input: An (n, m, q)-UMQ instance f, with coefficient tuples $(\mathbf{A}_i, \mathbf{b}_i, c_i)_{i=1}^m$, and oracle access to an $(n+1, m, q)$-ZCMQ solver $\mathcal{A}$

Output: A valid solution $\mathbf{x}^* \in \mathbb{F}_q^n$ to the UMQ instance, or $\perp$

for $i = 1, 2, \ldots, m$ **do**
  Sample a masking vector $\mathbf{u}_i \xleftarrow{\$} \mathbb{F}_q^n$ and a scalar $d_i \xleftarrow{\$} \mathbb{F}_q$
  Construct the $(n+1) \times (n+1)$ matrix $\widetilde{\mathbf{A}}_i \leftarrow \begin{pmatrix} \mathbf{A}_i & \mathbf{u}_i \\ \mathbf{0}^\mathsf{T} & c_i - d_i \end{pmatrix}$
  Construct the $(n+1)$-dimensional vector $\widetilde{\mathbf{b}}_i \leftarrow \begin{pmatrix} \mathbf{b}_i - \mathbf{u}_i \\ d_i \end{pmatrix}$
end for
Sample an invertible matrix $\mathbf{M} \xleftarrow{\$} \mathsf{GL}_{n+1}(\mathbb{F}_q)$
for $i = 1, 2, \ldots, m$ **do**
  $\mathbf{A}'_i \leftarrow \mathsf{UpperTri}(\mathbf{M}^\mathsf{T} \widetilde{\mathbf{A}}_i \mathbf{M})$
  $\mathbf{b}'_i \leftarrow \mathbf{M}^\mathsf{T} \widetilde{\mathbf{b}}_i$
end for
Run $\mathcal{A}$ on the constructed $(n+1, m, q)$-ZCMQ instance $(\mathbf{A}'_i, \mathbf{b}'_i)_{i=1}^m$
$\mathbf{w}^* \leftarrow \mathcal{A}((\mathbf{A}'_i, \mathbf{b}'_i)_{i=1}^m)$
if $\mathbf{w}^* = \perp$ **or** $\mathbf{w}^* = \mathbf{0}$ **or** $(\mathbf{w}^*)^\mathsf{T} \mathbf{A}'_i \mathbf{w}^* + (\mathbf{b}'_i)^\mathsf{T} \mathbf{w}^* \neq 0$ for some $i \in [m]$ **then**
  return $\perp$
end if
Compute the mapped vector $\widetilde{\mathbf{x}}^* \leftarrow \mathbf{M}\mathbf{w}^*$
Parse $\widetilde{\mathbf{x}}^*$ as $\begin{pmatrix} \mathbf{x}^* \\ y^* \end{pmatrix}$, where $\mathbf{x}^* \in \mathbb{F}_q^n$ and $y^* \in \mathbb{F}_q$
if $y^* = 1$ **then**
  return $\mathbf{x}^*$
else
  return $\perp$
end if

Overview of $\mathcal{B}^{\mathcal{A}}$. At a high level, $\mathcal{B}$ pads the UMQ challenge into an intermediate ZCMQ instance, randomizes it by a linear change of variables, invokes $\mathcal{A}$, and reverses the map $\mathbf{M}$ to extract a valid solution to the original UMQ instance.

Correctness of the padding step. Let the intermediate variable vector be $\widetilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ y \end{pmatrix} \in \mathbb{F}_q^{n+1}$. For each $i \in [m]$, define the intermediate polynomial $\widetilde{p}_i(\mathbf{x}, y) := \begin{pmatrix} \mathbf{x} \\ y \end{pmatrix}^\mathsf{T} \widetilde{\mathbf{A}}_i \begin{pmatrix} \mathbf{x} \\ y \end{pmatrix} + \widetilde{\mathbf{b}}_i^\mathsf{T} \begin{pmatrix} \mathbf{x} \\ y \end{pmatrix}$. By the construction of $\widetilde{\mathbf{A}}_i$ and $\widetilde{\mathbf{b}}_i$, we have

$$\widetilde{p}_i(\mathbf{x}, y) = \mathbf{x}^\mathsf{T} \mathbf{A}_i \mathbf{x} + y\, \mathbf{u}_i^\mathsf{T} \mathbf{x} + (c_i - d_i) y^2 + (\mathbf{b}_i - \mathbf{u}_i)^\mathsf{T} \mathbf{x} + d_i y,$$

and $\widetilde{p}_i(\mathbf{x}, 1) = \mathbf{x}^\mathsf{T} \mathbf{A}_i \mathbf{x} + \mathbf{b}_i^\mathsf{T} \mathbf{x} + c_i$, so any solution of the intermediate system whose last coordinate is 1 yields a solution of the original UMQ instance.

Distribution of the constructed instance. For each $i \in [m]$, $\mathcal{B}$ transforms the independent and uniform random variables $(\mathbf{A}_i, \mathbf{b}_i, c_i, \mathbf{u}_i, d_i)$ into the intermediate set $(\mathbf{A}_i, \mathbf{b}_i - \mathbf{u}_i, c_i - d_i, \mathbf{u}_i, d_i)$.

Consider the map $\phi : \mathcal{K}_{n,m,q} \times (\mathbb{F}_q^n \times \mathbb{F}_q)^m \to \mathcal{K}_{n+1,m,q}^{\mathsf{ZCMQ}}, (\mathbf{A}_i, \mathbf{b}_i, c_i, \mathbf{u}_i, d_i)_{i=1}^m \mapsto (\widehat{\mathbf{A}}_i, \widehat{\mathbf{b}}_i, \widehat{c}_i, \widehat{\mathbf{u}}_i, \widehat{d}_i)_{i=1}^m = (\mathbf{A}_i, \mathbf{b}_i - \mathbf{u}_i, c_i - d_i, \mathbf{u}_i, d_i)_{i=1}^m$, then it is a bijection with inverse $(\widehat{\mathbf{A}}_i, \widehat{\mathbf{b}}_i, \widehat{c}_i, \widehat{\mathbf{u}}_i, \widehat{d}_i)_{i=1}^m \mapsto (\widehat{\mathbf{A}}_i, \widehat{\mathbf{b}}_i + \widehat{\mathbf{u}}_i, \widehat{c}_i + \widehat{d}_i, \widehat{\mathbf{u}}_i, \widehat{d}_i)_{i=1}^m$. Since $(\mathbf{A}_i, \mathbf{b}_i, c_i, \mathbf{u}_i, d_i)_{i=1}^m$ is uniform over $\mathcal{K}_{n,m,q} \times (\mathbb{F}_q^n \times \mathbb{F}_q)^m$, $(\mathbf{A}_i, \mathbf{b}_i - \mathbf{u}_i, c_i - d_i, \mathbf{u}_i, d_i)_{i=1}^m$ is uniform over $\mathcal{K}_{n+1,m,q}^{\mathsf{ZCMQ}}$, and $(\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i)_{i=1}^m$ is a uniform ZCMQ instance.

Then, the algorithm applies a linear transformation to variables $\widetilde{\mathbf{x}} = \mathbf{M}\mathbf{w}$. Since $\mathbf{M} \in \mathsf{GL}_{n+1}(\mathbb{F}_q)$ is invertible, this transformation induces a bijection on $\mathcal{K}_{n+1,m,q}^{\mathsf{ZCMQ}}$ (which can be checked by considering the coefficients of the corresponding quadratic polynomials). Since $(\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i)_{i=1}^m$ is a uniform ZCMQ instance, $(\mathbf{A}'_i, \mathbf{b}'_i)_{i=1}^m$ is a uniform $(n+1, m, q)$-ZCMQ instance.

Success probability. Assume $\mathcal{A}$ successfully returns a non-zero solution $\mathbf{w}^* \in \mathbb{F}_q^{n+1} \setminus \{\mathbf{0}\}$ for the transformed system $(\widetilde{\mathbf{A}}'_i, \widetilde{\mathbf{b}}'_i)_{i=1}^m$ (which happens with probability ϵ). $\mathcal{B}$ then maps this solution back to the intermediate coordinates: $\widetilde{\mathbf{x}}^* = \mathbf{M}\mathbf{w}^* = \begin{pmatrix} \mathbf{x}^* \\ y^* \end{pmatrix}$.

Now we analyze the adversary's view to determine the distribution of $\widetilde{\mathbf{x}}^*$. Let $I' = (\mathbf{A}'_i, \mathbf{b}'_i)_{i=1}^m$ and $\widetilde{I} = (\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i)_{i=1}^m$. By the distribution argument above, $\widetilde{I}$ is uniformly distributed over $\mathcal{K}_{n+1,m,q}^{\mathsf{ZCMQ}}$, and it is independent of $\mathbf{M} \xleftarrow{\$} \mathsf{GL}_{n+1}(\mathbb{F}_q)$.

For any fixed $\mathbf{M}_0 \in \mathsf{GL}_{n+1}(\mathbb{F}_q)$, the map $\widetilde{I} \mapsto I$ induced by $\mathbf{M}_0$ is a bijection on $\mathcal{K}_{n+1,m,q}^{\mathsf{ZCMQ}}$. Therefore, for every fixed instance $I_0 \in \mathcal{V}_{n+1}^m$, we have

$$\Pr[I = I_0 \mid \mathbf{M} = \mathbf{M}_0] = \frac{1}{|\mathcal{V}_{n+1}|^m},$$

which is independent of $\mathbf{M}_0$. Bayes' rule then gives

$$\Pr[\mathbf{M} = \mathbf{M}_0 \mid I = I_0] = \Pr[\mathbf{M} = \mathbf{M}_0]$$

for every $\mathbf{M}_0 \in \mathsf{GL}_{n+1}(\mathbb{F}_q)$, meaning that conditioned on the adversary's view I, the distribution of $\mathbf{M}$ remains uniform over $\mathsf{GL}_{n+1}(\mathbb{F}_q)$.

Let R denote the internal randomness of $\mathcal{A}$, which is independent of all other randomness, and write $\mathbf{w}^* = \mathcal{A}(I; R)$. Conditioned on fixed values of I and R, the output $\mathbf{w}^*$ is fixed, while $\mathbf{M}$ remains uniformly distributed over $\mathsf{GL}_{n+1}(\mathbb{F}_q)$. Since the event $E := \{\mathcal{A} \text{ succeeds}\}$ is determined by I and R, $\mathbf{M}$ remains uniformly distributed over $\mathsf{GL}_{n+1}(\mathbb{F}_q)$ conditioned on E.

For the non-zero vector $\mathbf{w}^* \in \mathbb{F}_q^{n+1}$, uniformity of $\mathbf{M}$ over $\mathsf{GL}_{n+1}(\mathbb{F}_q)$ makes $\widetilde{\mathbf{x}}^* = \mathbf{M}\mathbf{w}^*$ uniform over $\mathbb{F}_q^{n+1} \setminus \{\mathbf{0}\}$. Hence

$$\Pr[y^* = 1 \mid \mathcal{A} \text{ succeeds}] = \frac{|\{(\mathbf{x}, 1) : \mathbf{x} \in \mathbb{F}_q^n\}|}{|\mathbb{F}_q^{n+1} \setminus \{\mathbf{0}\}|} = \frac{q^n}{q^{n+1} - 1}.$$

When $\mathcal{A}$ succeeds and $y^* = 1$, $\mathbf{w}^*$ is a non-zero solution of the ZCMQ instance I, and $\widetilde{\mathbf{x}}^* = \begin{pmatrix} \mathbf{x}^* \\ 1 \end{pmatrix}$ is a non-zero solution of the ZCMQ instance $\widetilde{I}$. The correctness of the padding step implies that:

$$(\mathbf{x}^*)^\mathsf{T}\mathbf{A}_i\mathbf{x}^* + \mathbf{b}_i^\mathsf{T}\mathbf{x}^* + c_i = 0, \qquad \text{for all } i \in [m].$$

In this event, $\mathcal{B}$ successfully outputs $\mathbf{x}^*$ as a solution of the original UMQ instance $(\mathbf{A}_i, \mathbf{b}_i, c_i)_{i=1}^m$. The overall success probability is $\epsilon' = \epsilon \cdot q^n/(q^{n+1}-1)$. Since $\mathrm{Succ}^{\mathsf{ZCMQ}}_{n+1,m,q}(\mathcal{A}) = \epsilon = \frac{q^{n+1}-1}{q^n}\epsilon' \le (q - q^{-n}) \cdot \mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(t')$, and the running time t' is bounded by t plus the minor polynomial overhead of matrix multiplications: $t' = t + \mathrm{poly}(n, m, \log q)$, taking the supremum over all p.p.t algorithm $\mathcal{A}$ running in time t completes the proof. □

Combining ZCMQ ⇒ MQSPR (Lemma 1) and UMQ ⇒ ZCMQ (Theorem 3), we have

Theorem 4 (UMQ ⇒ MQSPR). *Under the (n, m, q)-UMQ assumption, the $(n+1, m, q)$-MQSPR assumption holds: for each t, we have*

$$\mathrm{InSec}^{\mathsf{SPR}}\big(\mathcal{Q}_{n+1,m,q}; t\big) \le (q - q^{-n}) \cdot \mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(t') \le q \cdot \mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(t'),$$

where $t' = t + \mathrm{poly}(n, m, \log q)$[4].

Remark 2 (Tightness). The reduction loss is $q - q^{-n} \le q$. Since q is a small constant in multivariate cryptography, our reduction is tight.

Remark 3. Since [1] shows that ZCMQ tightly implies MQ universal one-wayness (MQUOW), we can combine it with the tight implication UMQ ⇒ ZCMQ (Theorem 3) to show that (n, m, q)-UMQ tightly implies $(n+1, m, q)$-MQUOW.

4 Relations between MQOW and UMQ

In this section, we discuss the relations between MQOW and UMQ. We show that MQOW implies UMQ, and that UMQ conditionally implies MQOW.

4.1 MQOW ⇒ UMQ

Now we show that MQOW implies UMQ.

Theorem 5 (MQOW ⇒ UMQ). *Under the $(n+1, m, q)$-MQOW assumption, the (n, m, q)-UMQ assumption holds: for each t, we have*

$$\mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(t) \le \frac{q}{q-1}\,\mathrm{InSec}^{\mathsf{OW}}(\mathcal{Q}_{n+1,m,q}; t'),$$

where $t' = t + \mathrm{poly}(n, m, \log q)$.

[4] We note that [36] stated this result with a proof sketch but without a concrete security bound. The sketch assumes, without justification, that the last component of the SPR adversary's output difference equals to 1. To the best of our knowledge, we give the first complete proof with a concrete security bound.

Algorithm 4 The MQOW adversary $\mathcal{B}^{\mathcal{A}}$

Input: An $(n+1,m,q)$-MQOW challenge $(f,\mathbf{y})$, where $f \xleftarrow{\$} \mathcal{Q}_{n+1,m,q}$ has the coefficient tuple $(\mathbf{A}_i,\mathbf{b}_i,c_i)_{i=1}^m \in \mathcal{K}_{n+1,m,q}$, and $\mathbf{y} = f(\mathbf{x})$ for some unknown vector $\mathbf{x} \xleftarrow{\$} \mathbb{F}_q^{n+1}$; oracle access to an (n,m,q)-UMQ solver $\mathcal{A}$

Output: A preimage $\mathbf{x}' \in \mathbb{F}_q^{n+1}$ satisfying $f_k(\mathbf{x}') = \mathbf{y}$, or $\perp$

for $i = 1,\ldots,m$ **do**
  Parse $\mathbf{A}_i = \begin{pmatrix} \mathbf{A}_i' & \mathbf{r}_i \\ \mathbf{0}^\mathsf{T} & a_i \end{pmatrix}$, $\mathbf{b}_i = \begin{pmatrix} \mathbf{s}_i \\ \beta_i \end{pmatrix}$, where $\mathbf{A}_i' \in \mathbb{F}_q^{n\times n}$ is upper triangular, $\mathbf{r}_i,\mathbf{s}_i \in \mathbb{F}_q^n$, and $a_i,\beta_i \in \mathbb{F}_q$
end for
Sample $\alpha \xleftarrow{\$} \mathbb{F}_q$
for $i = 1,\ldots,m$ **do**
  Set $\mathbf{b}_i' \leftarrow \mathbf{s}_i + \alpha\mathbf{r}_i$
  Set $c_i' \leftarrow a_i\alpha^2 + \beta_i\alpha + c_i - y_i$
end for
Let $k_\alpha' \leftarrow (\mathbf{A}_i',\mathbf{b}_i',c_i')_{i=1}^m$ and $f_{k_\alpha'}$ be the MQ function defined by k_α'
Query $\mathcal{A}$ on $f_{k_\alpha'}$ and let $\mathbf{z} \leftarrow \mathcal{A}(f_{k_\alpha'})$
if $\mathbf{z} \in \mathbb{F}_q^n$ and $\mathbf{z}^\mathsf{T}\mathbf{A}_i'\mathbf{z} + \mathbf{b}_i'^\mathsf{T}\mathbf{z} + c_i' = 0$ for every $i \in [m]$ **then**
  Set $\mathbf{x}' \leftarrow \begin{pmatrix} \mathbf{z} \\ \alpha \end{pmatrix}$
  return $\mathbf{x}'$
else
  return $\perp$
end if

Proof idea. The key observation is that, after fixing the value of one variable, an MQOW instance can be transformed into a UMQ instance under some conditions.

Proof. Suppose we are given a p.p.t. algorithm $\mathcal{A}$ running in time t, solving a (n,m,q)-UMQ instance with success probability ϵ. Now we construct an algorithm $\mathcal{B} = \mathcal{B}^{\mathcal{A}}$ to break the $(n+1,m,q)$-MQOW assumption. We construct $\mathcal{B}^{\mathcal{A}}$ in Algorithm 4.

Overview of $\mathcal{B}^{\mathcal{A}}$. $\mathcal{B}^{\mathcal{A}}$ samples $\alpha \xleftarrow{\$} \mathbb{F}_q$, fixes the last input coordinate to α, and obtains the polynomials $h_{i,\alpha}(\mathbf{z}) := f_i\left(\begin{pmatrix} \mathbf{z} \\ \alpha \end{pmatrix}\right) - y_i, i \in [m]$. $\mathcal{B}^{\mathcal{A}}$ views it as a search MQ instance k_α^* and sends it to $\mathcal{A}$. The solution returned by $\mathcal{A}$ can be lifted to a solution of the original MQOW challenge $(k,\mathbf{y})$.

Correctness analysis. If $\mathcal{A}$ returns a vector $\mathbf{z}$ such that $h_{i,\alpha}(\mathbf{z}) = 0$ for all $i \in [m]$, then $\mathbf{x}' = \begin{pmatrix} \mathbf{z} \\ \alpha \end{pmatrix}$ satisfies $f_i\left(\begin{pmatrix} \mathbf{z} \\ \alpha \end{pmatrix}\right) = y_i$ for all $i \in [m]$, and hence $f_k(\mathbf{x}') = \mathbf{y}$. Thus, $\mathcal{B}^{\mathcal{A}}$ returns a preimage of $\mathbf{y}$ under f.

Distribution of k'_α. Firstly, we denote the unknown input in the MQOW challenge as $\mathbf{x} = \begin{pmatrix}\mathbf{u}\\ v\end{pmatrix}$, where $\mathbf{u} \in \mathbb{F}_q^n$ and $v \in \mathbb{F}_q$. We now show that, conditioned on the event that $\alpha \neq v$, the search MQ instance k'_α is uniformly distributed over $\mathcal{K}_{n,m,q}$.

For every $i \in [m]$, denote $\mathbf{A}_i = \begin{pmatrix}\mathbf{A}'_i & \mathbf{r}_i\\ \mathbf{0}^\mathsf{T} & a_i\end{pmatrix}$, and $\mathbf{b}_i = \begin{pmatrix}\mathbf{s}_i\\ \beta_i\end{pmatrix}$, where $\mathbf{A}_i$ is an $n \times n$ upper triangular matrix over $\mathbb{F}_q^n$, $\mathbf{r}_i, \mathbf{s}_i \in \mathbb{F}_q^n, a_i, \beta_i \in \mathbb{F}_q$. A simple calculation shows that

$$y_i = \mathbf{u}^\mathsf{T}\mathbf{A}'_i\mathbf{u} + v\mathbf{r}_i^\mathsf{T}\mathbf{u} + a_i v^2 + \mathbf{s}_i^\mathsf{T}\mathbf{u} + \beta_i v + c_i.$$

Hence, for each fixed $\alpha \in \mathbb{F}_q$, we have

$$\begin{aligned} h_{i,\alpha}(\mathbf{z}) &= \mathbf{z}^\mathsf{T}\mathbf{A}'_i\mathbf{z} + (\mathbf{s}_i + \alpha\mathbf{r}_i)^\mathsf{T}\mathbf{z} + a_i\alpha^2 + \beta_i\alpha + c_i - y_i,\\ &= \mathbf{z}^\mathsf{T}\mathbf{A}'_i\mathbf{z} + (\mathbf{s}_i + \alpha\mathbf{r}_i)^\mathsf{T}\mathbf{z} + a_i\alpha^2 + \beta_i\alpha + c_i\\ &\quad - (\mathbf{u}^\mathsf{T}\mathbf{A}'_i\mathbf{u} + v\mathbf{r}_i^\mathsf{T}\mathbf{u} + a_i v^2 + \mathbf{s}_i^\mathsf{T}\mathbf{u} + \beta_i v + c_i)\\ &= \mathbf{z}^\mathsf{T}\mathbf{A}'_i\mathbf{z} + (\mathbf{s}_i + \alpha\mathbf{r}_i)^\mathsf{T}\mathbf{z}\\ &\quad + (\alpha - v)[a_i(\alpha + v) + \beta_i] - \mathbf{u}^\mathsf{T}\mathbf{A}'_i\mathbf{u} - (v\mathbf{r}_i + \mathbf{s}_i)^\mathsf{T}\mathbf{u} \end{aligned}$$

and $k'_\alpha = (\mathbf{A}'_i, \mathbf{b}'_i, c'_i)_{i=1}^m = (\mathbf{A}'_i,\ \mathbf{s}_i + \alpha\mathbf{r}_i,\ (\alpha - v)[a_i(\alpha + v) + \beta_i] - \mathbf{u}^\mathsf{T}\mathbf{A}'_i\mathbf{u} - (v\mathbf{r}_i + \mathbf{s}_i)^\mathsf{T}\mathbf{u})_{i=1}^m$. Now we show that, conditioned on any fixed $(\mathbf{x}, \alpha) = (\mathbf{u}, v, \alpha)$ satisfying $\alpha \neq v$, k'_α is uniform over $\mathcal{K}_{n,m,q}$.

Consider the map $\phi_{\mathbf{x},\alpha} : \mathcal{K}_{n+1,m,q} \to \mathcal{K}_{n,m,q}, (\mathbf{A}_i, \mathbf{b}_i, c_i)_{i=1}^m = (\mathbf{A}'_i, \mathbf{r}_i, a_i, \mathbf{s}_i, \beta_i, c_i)_{i=1}^m \mapsto (\widehat{\mathbf{A}}_i, \widehat{b}_i, \widehat{c}_i)_{i=1}^m = (\mathbf{A}'_i,\ \mathbf{s}_i + \alpha\mathbf{r}_i,\ (\alpha - v)[a_i(\alpha + v) + \beta_i] - \mathbf{u}^\mathsf{T}\mathbf{A}'_i\mathbf{u} - (v\mathbf{r}_i + \mathbf{s}_i)^\mathsf{T}\mathbf{u})_{i=1}^m$. When $\alpha \neq v$, it is a surjection, since for arbitrary $(\widehat{\mathbf{A}}_i, \widehat{b}_i, \widehat{c}_i)_{i=1}^m$, we can fix arbitrary $\mathbf{r}_i, a_i, c_i$ and set

$$\mathbf{s}_i = \widehat{b}_i - \alpha\mathbf{r}_i, \quad \beta_i = \frac{1}{\alpha - v}(\widehat{c}_i + (\mathbf{u}^\mathsf{T}\widehat{\mathbf{A}}_i\mathbf{u} + (v\mathbf{r}_i + \mathbf{s}_i)^\mathsf{T}\mathbf{u})) - a_i(\alpha + v),$$

and then we have $\phi_{\mathbf{x},\alpha}((\widehat{\mathbf{A}}_i, \mathbf{r}_i, a_i, \mathbf{s}_i, \beta_i, c_i)_{i=1}^m) = (\widehat{\mathbf{A}}_i, \widehat{b}_i, \widehat{c}_i)_{i=1}^m$. Since $\phi_{\mathbf{x},\alpha}$ is also linear, by the First Isomorphism Theorem for vector spaces, each vector in $\mathcal{K}_{n,m,q}$ has the same number of preimages. Since $(\mathbf{A}_i, \mathbf{b}_i, c_i)_{i=1}^m$ is uniform over $\mathcal{K}_{n+1,m,q}$, k'_α is uniform over $\mathcal{K}_{n,m,q}$ and is an (n, m, q)-UMQ instance.

Success Probability. Since α is sampled independently and uniformly from $\mathbb{F}_q$, the event $\alpha \neq v$ occurs with probability $1 - 1/q$. Conditioned on $\alpha \neq v$, the instance submitted to $\mathcal{A}$ is a uniformly random UMQ instance, and therefore $\mathcal{A}$ succeeds with probability ϵ. It follows that $\mathrm{Succ}^{\mathsf{OW}}_{\mathcal{Q}_{n+1,m,q}}(\mathcal{B}^{\mathcal{A}}) \geq \Pr[\alpha \neq v]\epsilon = \left(1 - \frac{1}{q}\right)\epsilon$, and hence

$$\mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(\mathcal{A}) = \epsilon \leq \frac{q}{q-1}\,\mathrm{Succ}^{\mathsf{OW}}_{\mathcal{Q}_{n+1,m,q}}(\mathcal{B}^{\mathcal{A}}).$$

The $\mathcal{B}^{\mathcal{A}}$ invokes $\mathcal{A}$ once, and all remaining operations require $\mathrm{poly}(n, m, \log q)$ time. Hence, the running time of $\mathcal{B}^{\mathcal{A}}$ is $t' = t + \mathrm{poly}(n, m, \log q)$ and we have $\mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(\mathcal{A}) \leq \frac{q}{q-1}\,\mathrm{InSec}^{\mathsf{OW}}(\mathcal{Q}_{n+1,m,q}; t')$. Taking the supremum over all algorithms $\mathcal{A}$ running in time at most t completes the proof. □

Remark 4 (Tightness). The reduction loss is $q/(q-1) \leq 2$, hence our reduction is tight.

Combining ZCMQ $\Rightarrow$ MQSPR (Lemma 1), UMQ $\Rightarrow$ ZCMQ (Theorem 3) and MQOW $\Rightarrow$ UMQ (Theorem 5), we obtain another proof of MQOW $\Rightarrow$ MQSPR different from [40]:

Corollary 1 (MQOW $\Rightarrow$ MQSPR). *Under the (n, m, q)-MQOW assumption, the (n, m, q)-MQSPR assumption holds: for each t, we have*

$$\mathrm{InSec}^{\mathsf{SPR}}(\mathcal{Q}_{n,m,q}; t) \leq \frac{q^2 - q^{2-n}}{q-1} \cdot \mathrm{InSec}^{\mathsf{OW}}(\mathcal{Q}_{n,m,q}; t'),$$

where $t' = t + \mathrm{poly}(n, m, \log q)$.

Remark 5. Since [1] shows that ZCMQ tightly implies MQUOW, we can combine it with the tight implications UMQ $\Rightarrow$ ZCMQ (Theorem 3) and MQOW $\Rightarrow$ UMQ (Theorem 5) and show that (n, m, q)-MQOW tightly implies (n, m, q)-MQUOW.

4.2 Conditional Reduction: UMQ $\Rightarrow$ MQOW

This subsection shows that UMQ implies MQOW when $m \leq n + O(\log \lambda)$.

Theorem 6 (Conditional UMQ $\Rightarrow$ MQOW). *Suppose $m \leq n + O(\log \lambda)$. Then, under the (n, m, q)-UMQ assumption, the $(n+1, m, q)$-MQOW assumption holds: for each $t > 0$, we have*

$$\mathrm{InSec}^{\mathsf{OW}}\left(\mathcal{Q}_{n+1,m,q}; t\right) \leq \left(q^{m-n} + q - q^{-n}\right) \mathrm{Succ}^{\mathsf{UMQ}}_{n,m,q}(t'),$$

where $t' = t + \mathrm{poly}(n, m, \log q)$.

Proof idea. We transform an (n, m, q)-UMQ instance into an $(n+1, m, q)$-MQOW instance. According to whether the returned preimage is equal to or different from the planted preimage, we design two possible reductions. We then combine the two reductions with an appropriate mixing probability.

For each $(n+1, m, q)$-MQOW adversary $\mathcal{A}$, we define its success event to be $\mathsf{Good}_{\mathsf{OW}}$, and two probabilities according to whether the planted preimage is returned:

$$\mathsf{Good}_{\mathsf{OW}} := \{f \xleftarrow{\$} \mathcal{Q}_{n+1,m,q}, \mathbf{X} \xleftarrow{\$} \mathbb{F}_q^{n+1}, \mathbf{Y} \leftarrow f(\mathbf{X}), \mathbf{U} \leftarrow \mathcal{A}(f, \mathbf{Y}) : \mathbf{U} \in \mathbb{F}_q^{n+1} \ \wedge \ f(\mathbf{U}) = f(\mathbf{X})\},$$

$$\epsilon_{=} := \Pr_{f,\mathbf{X},\mathcal{A}}[\mathsf{Good}_{\mathsf{OW}} \ \wedge \ \mathbf{U} = \mathbf{X}], \qquad \epsilon_{\neq} := \Pr_{f,\mathbf{X},\mathcal{A}}[\mathsf{Good}_{\mathsf{OW}} \ \wedge \ \mathbf{U} \neq \mathbf{X}].$$

Clearly we have $\mathrm{Succ}^{\mathsf{OW}}_{\mathcal{Q}_{n+1,m,q}}(\mathcal{A}) = \Pr_{f,\mathbf{X},\mathcal{A}}[\mathsf{Good}_{\mathsf{OW}}] = \epsilon_{=} + \epsilon_{\neq}$.

Algorithm 5 The branch-I UMQ solver $\mathcal{B}_{\mathsf{I}}^{\mathcal{A}}$

Input: A (n, m, q)-UMQ instance $f \in \mathcal{Q}_{n,m,q}$, whose coefficient tuple is $(\mathbf{A}_i, \mathbf{b}_i, c_i)_{i=1}^m$; oracle access to an $(n+1, m, q)$-MQOW adversary $\mathcal{A}$
Output: A zero $\mathbf{z} \in \mathbb{F}_q^n$ satisfying $f(\mathbf{z}) = \mathbf{0}$, or $\perp$

Sample $\mathbf{y} \xleftarrow{\$} \mathbb{F}_q^m$
for $i = 1, \ldots, m$ **do**
  Sample $\mathbf{u}_i \xleftarrow{\$} \mathbb{F}_q^n$ and $v_i, w_i \xleftarrow{\$} \mathbb{F}_q$
  Set $\widetilde{\mathbf{A}}_i \leftarrow \begin{pmatrix} \mathbf{A}_i & \mathbf{u}_i \\ \mathbf{0}^\mathsf{T} & w_i \end{pmatrix}$, $\widetilde{\mathbf{b}}_i \leftarrow \begin{pmatrix} \mathbf{b}_i \\ v_i \end{pmatrix}$, and $\widetilde{c}_i \leftarrow c_i + y_i$
end for
Let $g \in \mathcal{Q}_{n+1,m,q}$ be the MQ function with coefficient tuple $(\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i, \widetilde{c}_i)_{i=1}^m$
Sample $\pi \xleftarrow{\$} \mathrm{AGL}_{n+1}(\mathbb{F}_q)$
Set $\widehat{g} \leftarrow g \circ \pi^{-1}$
Query $\mathcal{A}$ on $(\widehat{g}, \mathbf{y})$ and let $\mathbf{w} \leftarrow \mathcal{A}(\widehat{g}, \mathbf{y})$
if $\mathbf{w} \in \mathbb{F}_q^{n+1}$ **then**
  Parse $\pi^{-1}(\mathbf{w}) = (\mathbf{z}, s)$
  if $s = 0$ and $f(\mathbf{z}) = \mathbf{0}$ **then**
    return $\mathbf{z}$
  end if
end if
return $\perp$

Branch I: the MQOW solver returns the planted preimage

For notation convenience, we write a point of $\mathbb{F}_q^{n+1}$ as $(\mathbf{z}, s)$ with $\mathbf{z} \in \mathbb{F}_q^n$ and $s \in \mathbb{F}_q$, write $\mathrm{AGL}_{n+1}(\mathbb{F}_q)$ for the group of invertible affine linear maps on $\mathbb{F}_q^{n+1}$, and denote $\mu = q^{n-m}$. Clearly we have $\mu \geq 1/\operatorname{poly}(\lambda) \iff m \leq n + O(\log \lambda)$.

Proposition 2. *Suppose $\mathcal{A}$ is an $(n+1, m, q)$-MQOW adversary running in time t with success probability $\epsilon = \epsilon_= + \epsilon_{\neq}$. Then there exists an (n, m, q)-UMQ solver $\mathcal{B}_{\mathsf{I}}^{\mathcal{A}}$ running in time $t' = t + \operatorname{poly}(n, m, \log q)$ with success probability*

$$\mathrm{Succ}_{n,m,q}^{\mathsf{UMQ}}\left(\mathcal{B}_{\mathsf{I}}^{\mathcal{A}}\right) \geq \mu\epsilon_=.$$

Proof. We construct $\mathcal{B}_{\mathsf{I}}^{\mathcal{A}}$ in Algorithm 5.
Overview of $\mathcal{B}_{\mathsf{I}}^{\mathcal{A}}$. Given an UMQ instance f, $\mathcal{B}_{\mathsf{I}}^{\mathcal{A}}$ samples a random vector $\mathbf{y}$, pads a variable and some coefficients, yielding an MQ function $g \in \mathcal{Q}_{n+1,m,q}$ such that $g(\mathbf{z}, 0) = f(\mathbf{z}) + \mathbf{y}$. Hence, each zero $\mathbf{x}$ of f satisfies $g(\mathbf{x}, 0) = \mathbf{y}$. Then, $\mathcal{B}_{\mathsf{I}}^{\mathcal{A}}$ applies a random invertible affine linear map to the variables and create an $(n+1, m, q)$-MQOW challenge. After $\mathcal{A}$ returns a preimage of $\mathbf{y}$, $\mathcal{B}_{\mathsf{I}}^{\mathcal{A}}$ inverts it back, gets $(\mathbf{z}, s)$ and checks whether $s = 0$ and $f(\mathbf{z}) = \mathbf{0}$.
Correctness analysis. Denote the coefficient tuple of f as $(\mathbf{A}_i, \mathbf{b}_i, c_i)_{i=1}^m$. For $\mathbf{y} = (y_1, \ldots, y_m)$, $\mathbf{u}_i \in \mathbb{F}_q^n$, and $v_i, w_i \in \mathbb{F}_q$, define $\widetilde{\mathbf{A}}_i := \begin{pmatrix} \mathbf{A}_i & \mathbf{u}_i \\ \mathbf{0}^\mathsf{T} & w_i \end{pmatrix}, \widetilde{\mathbf{b}}_i := \begin{pmatrix} \mathbf{b}_i \\ v_i \end{pmatrix}, \widetilde{c}_i := c_i + y_i$. Let $g \in \mathcal{Q}_{n+1,m,q}$ be the MQ function whose coefficient

tuple is $(\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i, \widetilde{c}_i)_{i=1}^m$. Then

$$g(\mathbf{z}, s) = f(\mathbf{z}) + \mathbf{y} + \left(s(\mathbf{u}_i^\mathsf{T}\mathbf{z} + v_i) + w_i s^2\right)_{i=1}^m,$$

and hence $g(\mathbf{z}, 0) = f(\mathbf{z}) + \mathbf{y}$. For an invertible affine linear map π on $\mathbb{F}_q^{n+1}$, we have $\widehat{g} = g \circ \pi^{-1} \in \mathcal{Q}_{n+1,m,q}$.

Suppose $\mathcal{B}_{\mathsf{I}}^{\mathcal{A}}$ outputs a vector $\mathbf{z}_0$. Then from the checking step (Line 12) we know that $\mathbf{z}_0$ must satisfy $\mathbf{w} = \pi(\mathbf{z}_0, 0)$ and $\mathbf{w}$ is the output of $\mathcal{A}$. When $\mathcal{A}$ succeeds, $\widehat{g}(\mathbf{w}) = \mathbf{y}$ and hence $f(\mathbf{z}_0) = \mathbf{0}$. Hence, $\mathbf{z}_0$ is a zero of the input UMQ instance f.

Conditional distribution of the constructed MQOW instance. Fix an arbitrary $\mathbf{x}_0 \in \mathbb{F}_q^n$. Now we are going to show that conditioned on the event $E_{\mathbf{x}_0} := \{f(\mathbf{x}_0) = \mathbf{0}\}$, the constructed $(\widehat{g}, \mathbf{y})$ is a valid MQOW instance.

Consider the evaluation map $\mathrm{ev}_{\mathbf{x}_0} : \mathcal{Q}_{n,m,q} \to \mathbb{F}_q^m, P \mapsto P(\mathbf{x}_0)$. Since the constant in $\mathcal{Q}_{n,m,q}$ can be arbitrary chosen, we know that $\mathrm{ev}_{\mathbf{x}_0}$ is surjective. Since $\mathrm{ev}_{\mathbf{x}_0}$ is a linear map, by the First Isomorphism Theorem, the kernel of $\mathrm{ev}_{\mathbf{x}_0}$ satisfies $|\ker(\mathrm{ev}_{\mathbf{x}_0})| = |\mathcal{Q}_{n,m,q}|/|\mathbb{F}_q^m| = q^{N_{n,m}-m}$. Again by considering the constant coefficients of P, we have $\Pr_f[E_{\mathbf{x}_0}] = q^{-m}$, and for every $h_0 \in \mathcal{Q}_{n,m,q}$, we have

$$\Pr_f[f = h_0 \mid E_{\mathbf{x}_0}] = \frac{1}{|\ker(\mathrm{ev}_{\mathbf{x}_0})|}\, \mathbf{1}\{h_0(\mathbf{x}_0) = \mathbf{0}\}\,.$$

In fact, when $h_0(\mathbf{x}_0) = \mathbf{0}$, the conditional probability is $q^{-N_{n,m}}/q^{-m} = q^{-(N_{n,m}-m)}$; otherwise the indicator is zero. Hence, conditioned on the event $E_{\mathbf{x}_0}$, the random variable f is uniformly distributed over $\ker(\mathrm{ev}_{\mathbf{x}_0})$.

• Step 1. We show that conditioned on $E_{\mathbf{x}_0}$, the pair $(g, \mathbf{y})$ has the same distribution as $(g^\star, g^\star(\mathbf{p}_{\mathbf{x}_0}))$, where $g^\star \xleftarrow{\$} \mathcal{Q}_{n+1,m,q}$, and $\mathbf{p}_{\mathbf{x}_0} := (\mathbf{x}_0, 0) \in \mathbb{F}_q^{n+1}$. In fact, the construction of g defines a map $\phi : \ker(\mathrm{ev}_{\mathbf{x}_0}) \times \mathbb{F}_q^m \times (\mathbb{F}_q^n \times \mathbb{F}_q \times \mathbb{F}_q)^m \to \mathcal{Q}_{n+1,m,q}$, and we want to show that it is a bijection. It can be calculated that $|\ker(\mathrm{ev}_{\mathbf{x}_0}) \times \mathbb{F}_q^m \times (\mathbb{F}_q^n \times \mathbb{F}_q \times \mathbb{F}_q)^m| = q^{N_{n,m}} = |\mathcal{Q}_{n+1,m,q}|$, so it suffices to show that ϕ is a surjection.

For each $h \in \mathcal{Q}_{n+1,m,q}$, we have

$$h_i(\mathbf{z}, s) = P_i(\mathbf{z}) + s(\widetilde{\mathbf{u}}_i^\mathsf{T}\mathbf{z} + \widetilde{v}_i) + \widetilde{w}_i s^2,$$

where $P_i \in \mathcal{Q}_{n,m,q}, \widetilde{\mathbf{u}}_i \in \mathbb{F}_q^n, \widetilde{v}_i, \widetilde{w}_i \in \mathbb{F}_q, i = 1, 2, ..., m$. For each fixed $\mathbf{y}_0 \in \mathbb{F}_q^m$. Under the event $E_{\mathbf{x}_0}$, the event $\{g = h, \mathbf{y} = \mathbf{y}_0\}$ is possible if and only if $\mathbf{y}_0 = h(\mathbf{p}_{\mathbf{x}_0})$. When $\mathbf{y}_0 = h(\mathbf{p}_{\mathbf{x}_0})$, we define $f_h : \mathbb{F}_q^{n+1} \to \mathbb{F}_q^m$, $f_h(\mathbf{z}, s) := h(\mathbf{z}, s) - y_0$. We can observe that $f(\mathbf{z}) := f_h(\mathbf{z}, 0)$ satisfies $f(\mathbf{x}_0) = \mathbf{0}$, and $f \in \ker(\mathrm{ev}_{\mathbf{x}_0})$. Then $\phi(f, \mathbf{y}_0, (\widetilde{\mathbf{u}}_i, \widetilde{v}_i, \widetilde{w}_i)_{i=1}^m) = h$. This shows that ϕ is a surjection and hence a bijection.

When $\mathbf{y}_0 = h(\mathbf{p}_{\mathbf{x}_0})$, since $f, \mathbf{y}, (\mathbf{u}_i, v_i, w_i)_{i=1}^m$ are independent, and $E_{\mathbf{x}_0}$ depends only on f, we have

$$\begin{aligned}
&\Pr_{f,\mathbf{y},(\mathbf{u}_i,v_i,w_i)_{i=1}^m}[g = h, \mathbf{y} = \mathbf{y}_0 \mid E_{\mathbf{x}_0}] \\
&= \Pr_f[f = f_h \mid E_{\mathbf{x}_0}] \Pr_{\mathbf{y}}[\mathbf{y} = \mathbf{y}_0] \Pr_{(\mathbf{u}_i,v_i,w_i)_{i=1}^m}[(\mathbf{u}_i, v_i, w_i)_{i=1}^m = (\widetilde{\mathbf{u}}_i, \widetilde{v}_i, \widetilde{w}_i)_{i=1}^m] \\
&= \frac{1}{q^{N_{n,m}-m}} \cdot \frac{1}{q^m} \cdot \frac{1}{q^{m(n+2)}} = \frac{1}{q^{N_{n+1,m}}} = \frac{1}{|\mathcal{Q}_{n+1,m,q}|}.
\end{aligned}$$

Therefore, for every $h \in \mathcal{Q}_{n+1,m,q}$ and $\mathbf{y}_0 \in \mathbb{F}_q^m$,

$$\Pr_{f,\mathbf{y},(\mathbf{u}_i,v_i,w_i)_{i=1}^m}[g = h, \mathbf{y} = \mathbf{y}_0 \mid E_{\mathbf{x}_0}] = \frac{1}{|\mathcal{Q}_{n+1,m,q}|} \mathbf{1}\{\mathbf{y}_0 = h(\mathbf{p}_{\mathbf{x}_0})\}.$$

In other words, conditioned on $E_{\mathbf{x}_0}$, the pair $(g, \mathbf{y})$ has the same distribution as $(g^\star, g^\star(\mathbf{p}_{\mathbf{x}_0}))$, where $g^\star \xleftarrow{\$} \mathcal{Q}_{n+1,m,q}$.

• Step 2. Now we add the randomness of π and show that conditioned on the event $E_{\mathbf{x}_0}$, the tuple $(\widehat{g}, \mathbf{y} = \widehat{g}(\mathbf{x}))$ is a valid $(n+1, m, q)$-MQOW instance, where $\mathbf{x} := \pi(\mathbf{p}_{\mathbf{x}_0}), \widehat{g} := g \circ \pi^{-1}$.

For each fixed $h \in \mathcal{Q}_{n+1,m,q}$, $\mathbf{u}_0 \in \mathbb{F}_q^{n+1}$, and $\mathbf{y}_0 \in \mathbb{F}_q^m$, since π is independent of f (and hence $E_{\mathbf{x}_0}$), we use the total probability formula and obtain

$$\begin{aligned}
&\Pr_{f,\mathbf{y},(\mathbf{u}_i,v_i,w_i)_{i=1}^m,\pi}[\widehat{g} = h, \mathbf{x} = \mathbf{u}_0, \mathbf{y} = \mathbf{y}_0 \mid E_{\mathbf{x}_0}] \\
&= \sum_{\substack{\varphi \in \mathrm{AGL}_{n+1}(\mathbb{F}_q): \\ \varphi(\mathbf{p}_{\mathbf{x}_0}) = \mathbf{u}_0}} \Pr_\pi[\pi = \varphi] \cdot \Pr_{f,\mathbf{y},(\mathbf{u}_i,v_i,w_i)_{i=1}^m}[g = h \circ \varphi, \mathbf{y} = \mathbf{y}_0 \mid E_{\mathbf{x}_0}].
\end{aligned}$$

The number of invertible affine linear maps φ satisfying $\varphi(\mathbf{p}_{\mathbf{x}_0}) = \mathbf{u}_0$ is (in fact, each invertible matrix $L \in \mathrm{GL}_{n+1}(\mathbb{F}_q)$ induces a unique constant vector $\mathbf{t} = \mathbf{u}_0 - L\mathbf{p}_{\mathbf{x}_0} \in \mathbb{F}_q^{n+1}$)

$$|\{\varphi \in \mathrm{AGL}_{n+1}(\mathbb{F}_q) : \varphi(\mathbf{p}_{\mathbf{x}_0}) = \mathbf{u}_0\}| = |\mathrm{GL}_{n+1}(\mathbb{F}_q)|.$$

Since $|\mathrm{AGL}_{n+1}(\mathbb{F}_q)| = |\mathrm{GL}_{n+1}(\mathbb{F}_q)|q^{n+1}$, we obtain

$$\Pr_{f,\mathbf{y},(\mathbf{u}_i,v_i,w_i)_{i=1}^m,\pi}[\widehat{g} = h, \mathbf{x} = \mathbf{u}_0, \mathbf{y} = \mathbf{y}_0 \mid E_{\mathbf{x}_0}] = \frac{1}{|\mathcal{Q}_{n+1,m,q}|q^{n+1}} \mathbf{1}\{\mathbf{y}_0 = h(\mathbf{u}_0)\}.$$

This shows that conditioned on the event $E_{\mathbf{x}_0}$, the tuple $(\widehat{g}, \mathbf{y} = \widehat{g}(\mathbf{x}))$ is a valid $(n+1, m, q)$-MQOW instance.

Success probability. From now on all probabilities are over $f, \mathbf{y}, (\mathbf{u}_i, v_i, w_i)_{i=1}^m, \pi, \mathcal{A}$. Let $\mathbf{w} = \mathcal{A}(\widehat{g}, \mathbf{y})$ and define the success event of $\mathcal{A}$ to be $E_{\mathsf{MQOW}} = \{\mathcal{A} \text{ succeeds}\} = \{\widehat{g}(\mathbf{w}) = \mathbf{y}\}$. Then we have

$$\Pr[E_{\mathsf{MQOW}} \mid E_{\mathbf{x}_0}] = \epsilon, \text{ and } \Pr[E_{\mathsf{MQOW}} \wedge \mathbf{w} = \pi(\mathbf{p}_{\mathbf{x}_0}) \mid E_{\mathbf{x}_0}] = \epsilon_=.$$

Now for each $\mathbf{x}_0 \in \mathbb{F}_q^n$, define $F_{\mathbf{x}_0} := E_{\mathbf{x}_0} \wedge E_{\mathsf{MQOW}} \wedge \{\mathbf{w} = \pi(\mathbf{p}_{\mathbf{x}_0})\}$. Since the algorithm $\mathcal{A}$ produces one vector $\mathbf{w}$, and the map π is injective, we know that

the events $\{F_{\mathbf{x}_0}\}_{\mathbf{x}_0\in\mathbb{F}_q^n}$ are pairwise disjoint. When the event $F_{\mathbf{x}_0}$ happens, the $\mathcal{B}_{\mathsf{I}}^{\mathcal{A}}$ returns the valid zero $\mathbf{x}_0$. Therefore

$$\mathrm{Succ}_{n,m,q}^{\mathsf{UMQ}}(\mathcal{B}_{\mathsf{I}}^{\mathcal{A}}) \geq \sum_{\mathbf{x}_0\in\mathbb{F}_q^n} \Pr[F_{\mathbf{x}_0}] = \sum_{\mathbf{x}_0\in\mathbb{F}_q^n} \Pr[E_{\mathbf{x}_0}]\epsilon_{=} = q^n q^{-m}\epsilon_{=} = \mu\epsilon_{=}.$$

Since sampling the coefficients and π, applying π^{-1} and π, and verifying the output take $\mathrm{poly}(n, m, \log q)$ additional time, the running time of $\mathcal{B}_{\mathsf{I}}^{\mathcal{A}}$ is $t' = t + \mathrm{poly}(n, m, \log q)$. This completes the proof. □

Branch II: the MQOW solver returns a different preimage

For notation convenience, for every $\pi \in \mathrm{AGL}_{n+1}(\mathbb{F}_q)$, write its unique decomposition as $\pi(\mathbf{z}) = L_\pi\mathbf{z} + \mathbf{t}_\pi, L_\pi \in \mathrm{GL}_{n+1}(\mathbb{F}_q), \mathbf{t}_\pi \in \mathbb{F}_q^{n+1}$. We denote $\alpha = q^n/(q^{n+1}-1)$.

Proposition 3. *Suppose $\mathcal{A}$ is an $(n+1, m, q)$-MQOW adversary running in time t with success probability $\epsilon = \epsilon_{=} + \epsilon_{\neq}$. Then there exists an (n, m, q)-UMQ solver $\mathcal{B}_{\mathsf{II}}^{\mathcal{A}}$ running in time $t' = t + \mathrm{poly}(n, m, \log q)$ with success probability*

$$\mathrm{Succ}_{n,m,q}^{\mathsf{UMQ}}(\mathcal{B}_{\mathsf{II}}^{\mathcal{A}}) \geq \alpha\epsilon_{\neq}.$$

Proof. We construct $\mathcal{B}_{\mathsf{II}}^{\mathcal{A}}$ in Algorithm 6.

Overview of $\mathcal{B}_{\mathsf{II}}^{\mathcal{A}}$. $\mathcal{B}_{\mathsf{II}}^{\mathcal{A}}$ chooses the coefficients v_i such that $g(\boldsymbol{\xi}) = \mathbf{y}$, where $\boldsymbol{\xi} = (\mathbf{0}, 1) \notin H_0 := \{(\mathbf{z}, 0) : \mathbf{z} \in \mathbb{F}_q^n\}$. Then it uses an invertible affine linear map π to change the variables and hide the H_0. If the $\mathcal{A}$ returns a valid preimage different from the planted point $\pi(\boldsymbol{\xi})$ and this preimage lies in the hidden hyperplane $H = \pi(H_0)$, it yields a zero of f.

Correctness analysis. For $\mathbf{y} = (y_i)_{i=1}^m \in \mathbb{F}_q^m$, $\mathbf{u}_i \in \mathbb{F}_q^n$, and $w_i \in \mathbb{F}_q$, set $v_i := -c_i - w_i, \widetilde{\mathbf{A}}_i := \begin{pmatrix}\mathbf{A}_i & \mathbf{u}_i\\ \mathbf{0}^\mathsf{T} & w_i\end{pmatrix}, \widetilde{\mathbf{b}}_i := \begin{pmatrix}\mathbf{b}_i\\ v_i\end{pmatrix}, \widetilde{c}_i := c_i + y_i$. Suppose $g \in \mathcal{Q}_{n+1,m,q}$ has the coefficient tuple $(\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i, \widetilde{c}_i)_{i=1}^m$, then

$$g(\mathbf{z}, s) = f(\mathbf{z}) + \mathbf{y} + \left(s(\mathbf{u}_i^\mathsf{T}\mathbf{z} - c_i - w_i) + w_i s^2\right)_{i=1}^m,$$

so $g(\mathbf{z}, 0) = f(\mathbf{z}) + \mathbf{y}$ and $g(\boldsymbol{\xi}) = \mathbf{y}$.

Suppose $\mathcal{B}_{\mathsf{II}}^{\mathcal{A}}$ outputs a vector $\mathbf{z}_0$. Then from the checking step (Line 12) we know that $\mathbf{z}_0$ must satisfy $\mathbf{w} = \pi(\mathbf{z}_0, 0)$ and $\mathbf{w}$ is the output of $\mathcal{A}$. When $\mathcal{A}$ succeeds, $\widehat{g}(\mathbf{w}) = \mathbf{y}$ and hence $f(\mathbf{z}_0) = \mathbf{0}$. Hence, $\mathbf{z}_0$ is a zero of the input UMQ instance f.

Besides, since $\boldsymbol{\xi} = (\mathbf{0}, 1) \neq (\mathbf{z}_0, 0)$, and $\pi(\boldsymbol{\xi}) \neq \mathbf{w}$, $\mathcal{A}$ returns a different preimage in the case that $\mathcal{B}_{\mathsf{II}}^{\mathcal{A}}$ outputs a vector $\mathbf{z}_0$.

Distribution of the constructed MQOW instance. Now we show that the tuple $(\widehat{g}, \mathbf{y}) = (\widehat{g}, \widehat{g}(\mathbf{x}))$ is a valid $(n+1, m, q)$-MQOW instance, where $\mathbf{x} = \pi(\boldsymbol{\xi})$.

• Step 1. For each $\mathbf{y}_0 \in \mathbb{F}_q^m$, we show that conditioned on the event $\{\mathbf{y} = \mathbf{y}_0\}$, g is uniformly distributed over $\mathcal{F}_{\boldsymbol{\xi},\mathbf{y}_0} := \{h \in \mathcal{Q}_{n+1,m,q} : h(\boldsymbol{\xi}) = \mathbf{y}_0\}$. Firstly,

Algorithm 6 The branch-II UMQ solver $\mathcal{B}_{\mathsf{II}}^{\mathcal{A}}$

Input: A (n, m, q)-UMQ instance $f \in \mathcal{Q}_{n,m,q}$, whose coefficient tuple is $(\mathbf{A}_i, \mathbf{b}_i, c_i)_{i=1}^m$; oracle access to an $(n+1, m, q)$-MQOW adversary $\mathcal{A}$

Output: A zero $\mathbf{z} \in \mathbb{F}_q^n$ satisfying $f(\mathbf{z}) = \mathbf{0}$, or $\perp$

Sample $\mathbf{y} \xleftarrow{\$} \mathbb{F}_q^m$
for $i = 1, \ldots, m$ **do**
  Sample $\mathbf{u}_i \xleftarrow{\$} \mathbb{F}_q^n$ and $w_i \xleftarrow{\$} \mathbb{F}_q$
  Set $v_i \leftarrow -c_i - w_i$
  Set $\widetilde{\mathbf{A}}_i \leftarrow \begin{pmatrix} \mathbf{A}_i & \mathbf{u}_i \\ \mathbf{0}^\mathsf{T} & w_i \end{pmatrix}$, $\widetilde{\mathbf{b}}_i \leftarrow \begin{pmatrix} \mathbf{b}_i \\ v_i \end{pmatrix}$, and $\widetilde{c}_i \leftarrow c_i + y_i$
end for
Let $g \in \mathcal{Q}_{n+1,m,q}$ be the function with coefficient tuple $(\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{b}}_i, \widetilde{c}_i)_{i=1}^m$
Sample $\pi \xleftarrow{\$} \mathrm{AGL}_{n+1}(\mathbb{F}_q)$
Set $\widehat{g} \leftarrow g \circ \pi^{-1}$
Query $\mathcal{A}$ on $(\widehat{g}, \mathbf{y})$ and let $\mathbf{w} \leftarrow \mathcal{A}(\widehat{g}, \mathbf{y})$
if $\mathbf{w} \in \mathbb{F}_q^{n+1}$ **then**
  Parse $\pi^{-1}(\mathbf{w}) = (\mathbf{z}, s)$
  if $s = 0$ and $f(\mathbf{z}) = \mathbf{0}$ **then**
    return $\mathbf{z}$
  end if
end if
return $\perp$

consider the linear map $\mathrm{ev}_{\boldsymbol{\xi}} : \mathcal{Q}_{n+1,m,q} \to \mathbb{F}_q^m, f \mapsto f(\boldsymbol{\xi})$. Since the constant in $\mathcal{Q}_{n+1,m,q}$ can be arbitrary chosen, we know that $\mathrm{ev}_{\boldsymbol{\xi}}$ is surjective. By the First Isomorphism Theorem, we know that $|\ker(\mathrm{ev}_{\boldsymbol{\xi}})| = |\mathcal{Q}_{n+1,m,q}|/q^m$, and that $\mathcal{F}_{\boldsymbol{\xi},\mathbf{y}_0}$ is the coset $\mathbf{y}_0 + \ker(\mathrm{ev}_{\boldsymbol{\xi}})$, so $|\mathcal{F}_{\boldsymbol{\xi},\mathbf{y}_0}| = |\ker(\mathrm{ev}_{\boldsymbol{\xi}})| = |\mathcal{Q}_{n+1,m,q}|/q^m$.

Now we consider the map $\phi_{\mathbf{y}_0} : \mathcal{Q}_{n,m,q} \times (\mathbb{F}_q^n \times \mathbb{F}_q)^m \to \mathcal{F}_{\boldsymbol{\xi},\mathbf{y}_0}, (f, (\mathbf{u}_i, w_i)_{i=1}^m) \mapsto g$, where $g(\mathbf{z}, s) = f(\mathbf{z}) + \mathbf{y}_0 + \left(s(\mathbf{u}_i^\mathsf{T}\mathbf{z} - c_i - w_i) + w_i s^2\right)_{i=1}^m$, and c_i is the constant of f_i (i.e., $f_i(\mathbf{0}) = c_i$). Now we show that $\phi_{\mathbf{y}_0}$ is a bijection. Since $|\mathcal{Q}_{n,m,q} \times (\mathbb{F}_q^n \times \mathbb{F}_q)^m| = |\mathcal{Q}_{n+1,m,q}|/q^m = |\mathcal{F}_{\boldsymbol{\xi},\mathbf{y}_0}|$, and it suffices to show that $\phi_{\mathbf{y}_0}$ is a surjection.

For each $h \in \mathcal{F}_{\boldsymbol{\xi},\mathbf{y}_0}$, we have $\widetilde{h}(\mathbf{z}, s) := h(\mathbf{z}, s) - \mathbf{y}_0$ satisfies $\widetilde{h}(\mathbf{0}, 1) = 0$. Since $\widetilde{h}_i(\mathbf{z}, s) = \widetilde{f}_i(\mathbf{z}) + s(\widetilde{\mathbf{u}}_i^\mathsf{T}\mathbf{z} + \widetilde{v}_i) + \widetilde{w}_i s^2$ for some $\widetilde{f} \in \mathcal{Q}_{n,m,q}, \widetilde{\mathbf{u}}_i \in \mathbb{F}_q^n, \widetilde{v}_i, \widetilde{w}_i \in \mathbb{F}_q$, we have $\widetilde{f}_i(\mathbf{0}) + \widetilde{v}_i + \widetilde{w}_i = 0$, and hence $\widetilde{v}_i = -\widetilde{f}_i(\mathbf{0}) - \widetilde{w}_i$. Then we have $\phi_{\mathbf{y}_0}(\widetilde{f}, (\widetilde{\mathbf{u}}_i, \widetilde{w}_i)_{i=1}^m) = h$. This shows that $\phi_{\mathbf{y}_0}$ is surjective.

Since $\phi_{\mathbf{y}_0}$ is a bijection, and $(f, (\mathbf{u}_i, w_i)_{i=1}^m)$ is uniformly random over $\mathcal{Q}_{n,m,q} \times (\mathbb{F}_q^n \times \mathbb{F}_q)^m$, we know that, conditioned on the event $\{\mathbf{y} = \mathbf{y}_0\}$, g is uniformly random over $\mathcal{F}_{\boldsymbol{\xi},\mathbf{y}_0}$.

• Step 2. We show that $(g, \mathbf{y})$ follows the same distribution of $(g^\star, \mathbf{y}^\star)$, where $g^\star \xleftarrow{\$} \mathcal{Q}_{n+1,m,q}$ and $\mathbf{y}^\star := g^\star(\boldsymbol{\xi})$. In fact, by step 1, for every fixed $\mathbf{y}_0$ and every $h \in \mathcal{Q}_{n+1,m,q}$,

$$\Pr_{f,(\mathbf{u}_i,w_i)_{i=1}^m}[g = h \mid \mathbf{y} = \mathbf{y}_0] = \frac{q^m}{|\mathcal{Q}_{n+1,m,q}|}\,\mathbf{1}\{h(\boldsymbol{\xi}) = \mathbf{y}_0\}\,,$$

and

$$\begin{aligned}\Pr_{f,\mathbf{y},(\mathbf{u}_i,w_i)_{i=1}^m}[g=h,\mathbf{y}=\mathbf{y}_0] &= \Pr_{f,(\mathbf{u}_i,w_i)_{i=1}^m}[g=h \mid \mathbf{y}=\mathbf{y}_0]\Pr_{\mathbf{y}}[\mathbf{y}=\mathbf{y}_0]\\ &= \frac{1}{|\mathcal{Q}_{n+1,m,q}|}\,\mathbf{1}\{\mathbf{y}_0=h(\boldsymbol{\xi})\}\,.\end{aligned}$$

• Step 3. Now we add the randomness of π and determine the distribution of $(\widehat{g},\mathbf{y})=(\widehat{g},\widehat{g}(\mathbf{x}))$.

Similar to the calculation in Step 2 of the branch I, we know that for each $h\in\mathcal{Q}_{n+1,m,q}$, $\mathbf{x}_0\in\mathbb{F}_q^{n+1}$, and $\mathbf{y}_0\in\mathbb{F}_q^m$,

$$\begin{aligned}&\Pr_{f,\mathbf{y},(\mathbf{u}_i,w_i)_{i=1}^m,\pi}[\widehat{g}=h,\mathbf{x}=\mathbf{x}_0,\mathbf{y}=\mathbf{y}_0]\\ &\quad=\sum_{\substack{\varphi\in\mathrm{AGL}_{n+1}(\mathbb{F}_q):\\ \varphi(\boldsymbol{\xi})=\mathbf{x}_0}}\Pr_{\pi}[\pi=\varphi]\cdot\Pr_{f,\mathbf{y},(\mathbf{u}_i,w_i)_{i=1}^m}[g=h\circ\varphi,\mathbf{y}=\mathbf{y}_0]\\ &\quad=\frac{1}{|\mathcal{Q}_{n+1,m,q}|q^{n+1}}\,\mathbf{1}\{\mathbf{y}_0=h(\mathbf{x}_0)\}\,.\end{aligned}$$

Thus $\widehat{g}$ and $\mathbf{x}$ are independent and uniform, and $\mathbf{y}=\widehat{g}(\mathbf{x})$ holds with probability one. This shows that the tuple $(\widehat{g},\mathbf{y})=(\widehat{g},\widehat{g}(\mathbf{x}))$ is a valid $(n+1,m,q)$-MQOW instance.

Conditional probability of the event that the returned vector of $\mathcal{A}$ lies in the hidden hyperplane. Denote $\mathbf{w}=\mathcal{A}(\widehat{g},\mathbf{y})$. By the correctness analysis, we know that if $\mathbf{w}\in H=\pi(H_0)$ is a preimage of $\mathbf{y}$ under $\widehat{g}$ then $\mathcal{B}_{\mathsf{II}}^{\mathcal{A}}$ successfully outputs a zero of f.

Arbitrarily fix $h\in\mathcal{Q}_{n+1,m,q},\mathbf{x}_0\in\mathbb{F}_q^{n+1},\mathbf{y}_0\in\mathbb{F}_q^m,\mathbf{w}_0\in\mathbb{F}_q^{n+1}$. Suppose $\mathbf{y}_0=h(\mathbf{x}_0)$ (we say that the tuple $(h,\mathbf{x}_0,\mathbf{y}_0)$ is compatible if $\mathbf{y}_0=h(\mathbf{x}_0)$). We show that

$$\Pr_{f,\mathbf{y},(\mathbf{u}_i,w_i)_{i=1}^m,\pi,\mathcal{A}}[\mathbf{w}\in H\mid\widehat{g}=h,\mathbf{x}=\mathbf{x}_0,\mathbf{y}=\mathbf{y}_0,\mathbf{w}=\mathbf{w}_0]=\frac{q^n}{q^{n+1}-1}=\alpha.$$

• Step 1. Arbitrarily fix $h\in\mathcal{Q}_{n+1,m,q},\mathbf{x}_0\in\mathbb{F}_q^{n+1}$, and set $\mathbf{y}_0=h(\mathbf{x}_0)$. We show that conditioned on the event $(\widehat{g},\mathbf{x},\mathbf{y})=(h,\mathbf{x}_0,\mathbf{y}_0)$, π is uniform over the set $\mathcal{E}_{\boldsymbol{\xi},\mathbf{x}_0}:=\{\varphi\in\mathrm{AGL}_{n+1}(\mathbb{F}_q):\varphi(\boldsymbol{\xi})=\mathbf{x}_0\}$.

Similar to the calculation in Step 2 of the branch I, we have $|\mathcal{E}_{\boldsymbol{\xi},\mathbf{x}_0}|=|\,\mathrm{GL}_{n+1}(\mathbb{F}_q)|$. For each $\varphi\in\mathcal{E}_{\boldsymbol{\xi},\mathbf{x}_0}$, use the Bayes' rule, we obtain

$$\begin{aligned}&\Pr_{f,\mathbf{y},(\mathbf{u}_i,w_i)_{i=1}^m,\pi}[\pi=\varphi|g=h,\mathbf{x}=\mathbf{x}_0,\mathbf{y}=\mathbf{y}_0]\\ &=\frac{\Pr_{f,\mathbf{y},(\mathbf{u}_i,w_i)_{i=1}^m}[\widehat{g}=h,\mathbf{x}=\mathbf{x}_0,\mathbf{y}=\mathbf{y}_0|\pi=\varphi]\cdot\Pr_{\pi}[\pi=\varphi]}{\Pr_{f,\mathbf{y},(\mathbf{u}_i,w_i)_{i=1}^m,\pi}[g=h,\mathbf{x}=\mathbf{x}_0,\mathbf{y}=\mathbf{y}_0]}\\ &=\frac{\Pr_{f,\mathbf{y},(\mathbf{u}_i,w_i)_{i=1}^m}[g=h\circ\varphi,\mathbf{y}=\mathbf{y}_0]\cdot|\,\mathrm{AGL}_{n+1}(\mathbb{F}_q)|^{-1}}{(|\mathcal{Q}_{n+1,m,q}|q^{n+1})^{-1}}\\ &=\frac{1}{|\,\mathrm{GL}_{n+1}(\mathbb{F}_q)|}\\ &=\frac{1}{|\mathcal{E}_{\boldsymbol{\xi},\mathbf{x}_0}|}.\end{aligned}$$

This proves our claim.

• Step 2. We determine the conditional distribution of the hidden hyperplane $H = \pi(H_0)$ under the event $\{\widehat{g} = h, \mathbf{x} = \mathbf{x}_0, \mathbf{y} = \mathbf{y}_0\}$, where $(h, \mathbf{x}_0, \mathbf{y}_0)$ is compatible.

As we have shown in Step 2 of the branch I, Sampling a map $\pi = L_\pi + \mathbf{t}_\pi$ from $\mathcal{E}_{\boldsymbol{\xi},\mathbf{x}_0}$, where L_π and $\mathbf{t}_\pi$ are the linear part and the constant part respectively, is equivalent to sampling $L_\pi \in \mathrm{GL}_{n+1}(\mathbb{F}_q)$ and then defining $t_\pi = \mathbf{x}_0 - L_\pi\boldsymbol{\xi}$. Consequently, for each $\ell \in \mathrm{GL}_{n+1}(\mathbb{F}_q)$ and compatible $(h, \mathbf{x}_0, \mathbf{y}_0)$, we have

$$\Pr_{f,\mathbf{y},(\mathbf{u}_i,w_i)_{i=1}^m,\pi}[L_\pi = \ell \mid \widehat{g} = h, \mathbf{x} = \mathbf{x}_0, \mathbf{y} = \mathbf{y}_0] = \frac{1}{|\,\mathrm{GL}_{n+1}(\mathbb{F}_q)|}.$$

In other words, conditioned on the event $\{(\widehat{g}, \mathbf{x}, \mathbf{y}) = (h, \mathbf{x}_0, \mathbf{y}_0)\}$, L_π is uniform over $\mathrm{GL}_{n+1}(\mathbb{F}_q)$.

Let $\mathbf{e}_{n+1} = (0, \ldots, 0, 1)^\mathsf{T}$ and define $\boldsymbol{\eta}^\mathsf{T} := \mathbf{e}_{n+1}^\mathsf{T} L_\pi^{-1}$. By the conditional uniformity of L_π established above, $\boldsymbol{\eta}$ is uniform over $\mathbb{F}_q^{n+1} \setminus \{\mathbf{0}\}$. Together with $\mathbf{x} = L_\pi\boldsymbol{\xi} + \mathbf{t}_\pi$ and $\mathbf{e}_{n+1}^\mathsf{T}\boldsymbol{\xi} = 1$, we know that $H = \pi(H_0)$ satisfies

$$\mathbf{z} \in H \iff \mathbf{e}_{n+1}^\mathsf{T} L_\pi^{-1}(\mathbf{z} - \mathbf{t}_\pi) = 0 \iff \boldsymbol{\eta}^\mathsf{T}(\mathbf{z} - \mathbf{x}) = -1,$$

Thus

$$H = \{\mathbf{z} \in \mathbb{F}_q^{n+1} : \boldsymbol{\eta}^\mathsf{T}(\mathbf{z} - \mathbf{x}) = -1\}. \qquad (\star)$$

Every nonzero $\boldsymbol{\eta}$ defines an affine hyperplane avoiding $\mathbf{x}$, and every affine hyperplane avoiding $\mathbf{x}$ has a unique representation in $(\star)$. Therefore, conditioned on the event that $(\widehat{g}, \mathbf{x}, \mathbf{y}) = (h, \mathbf{x}_0, \mathbf{y}_0)$ is compatible, H is uniform over all affine hyperplanes avoiding $\mathbf{x}$.

• Step 3. We add the randomness of $\mathbf{w} \leftarrow \mathcal{A}(\widetilde{g}, \mathbf{y})$ and calculate the probability of the event $\mathbf{w} \in H$ under the condition that $g = h, \mathbf{x} = \mathbf{x}_0, \mathbf{y} = \mathbf{y}_0, \mathbf{w} = \mathbf{w}_0$, where $(h, \mathbf{x}_0, \mathbf{y}_0)$ is compatible.

Fix $h \in \mathcal{Q}_{n+1,m,q}, \mathbf{x}_0, \mathbf{w}_0 \in \mathbb{F}_q^{n+1}, \mathbf{y}_0 = \widehat{g}(\mathbf{x}_0)$, and suppose that $\Pr_{\mathcal{A}}[\mathcal{A}(h, \mathbf{y}_0) = \mathbf{w}_0] > 0$. The probability of the event $\{\mathbf{w} = \mathbf{w}_0\}$, conditioned on $(\widehat{g}, \mathbf{x}, \mathbf{y}) = (h, \mathbf{x}_0, \mathbf{y}_0)$, depends only on the visible pair $(h, \mathbf{y}_0)$ and is independent of π. Define the events $C := \{\widehat{g} = h,\ \mathbf{x} = \mathbf{x}_0,\ \mathbf{y} = \mathbf{y}_0\}, W := \{\mathbf{w} = \mathbf{w}_0\}$, and suppose $\vartheta := \Pr_{\mathcal{A}}[\mathcal{A}(h, \mathbf{y}_0) = \mathbf{w}_0] > 0$. Conditioned on C and on any value $L_\pi = \ell$, the input supplied to $\mathcal{A}$ is still the fixed pair $(h, \mathbf{y}_0)$. Hence $\Pr_{\mathcal{A}}[W \mid C, L_\pi = \ell] = \vartheta$, which is independent of ℓ. Hence (from now on all probabilities are over $f, \mathbf{y}, (\mathbf{u}_i, w_i)_{i=1}^m, \pi, \mathcal{A}$)

$$\begin{aligned}\Pr[W \mid C] &= \sum_{\ell' \in \mathrm{GL}_{n+1}(\mathbb{F}_q)} \Pr[W \mid C, L_\pi = \ell']\Pr[L_\pi = \ell' \mid C] \\ &= \sum_{\ell' \in \mathrm{GL}_{n+1}(\mathbb{F}_q)} \vartheta \cdot \frac{1}{|\,\mathrm{GL}_{n+1}(\mathbb{F}_q)|} = \vartheta.\end{aligned}$$

For every $\ell \in \mathrm{GL}_{n+1}(\mathbb{F}_q)$, use Bayes' rule and we obtain

$$\begin{aligned}\Pr[L_\pi = \ell \mid C, W] &= \frac{\Pr[W \mid C, L_\pi = \ell]\ \Pr[L_\pi = \ell \mid C]}{\Pr[W \mid C]} \\ &= \frac{\vartheta \cdot |\,\mathrm{GL}_{n+1}(\mathbb{F}_q)|^{-1}}{\vartheta} = \frac{1}{|\,\mathrm{GL}_{n+1}(\mathbb{F}_q)|},\end{aligned}$$

which shows that conditioned on the events C and W, the random variable L_π is still uniform over $\mathrm{GL}_{n+1}(\mathbb{F}_q)$. Consequently, $\boldsymbol{\eta}^\mathsf{T} = \mathbf{e}_{n+1}^\mathsf{T} L_\pi^{-1}$ is uniform over $\mathbb{F}_q^{n+1} \setminus \{\mathbf{0}\}$ under the condition C and W. If $\mathbf{w}_0 \neq \mathbf{x}_0$, the condition $\mathbf{w}_0 \in H$ is equivalent to $\boldsymbol{\eta}^\mathsf{T}(\mathbf{w}_0 - \mathbf{x}_0) = -1$. Since this affine linear equation in $\boldsymbol{\eta}$ has q^n nonzero solutions, we have

$$\Pr[\mathbf{w} \in H \mid \widehat{g} = h, \mathbf{x} = \mathbf{x}_0, \mathbf{y} = \mathbf{y}_0, \mathbf{w} = \mathbf{w}_0] = \frac{q^n}{q^{n+1} - 1} = \alpha.$$

Success probability. Let $\mathbf{w} := \mathcal{A}(\widehat{g}, \mathbf{y})$ and define the success event of $\mathcal{A}$ to be $E_{\mathsf{MQOW}} = \{\mathcal{A} \text{ succeeds}\} = \{\widehat{g}(\mathbf{w}) = \mathbf{y}\}$. Let $S_{\neq} = E_{\mathsf{MQOW}} \wedge \{\mathbf{w} \neq \mathbf{x}\}$. Then we have $\Pr[S_{\neq}] = \epsilon_{\neq}$.

We briefly show that Algorithm 6 succeeds when $\mathcal{A}$ outputs a different preimage $\mathbf{w}$ that lies in the hidden hyperplane H. On the one hand, if $\mathbf{w} \in H = \pi(H_0)$, then $\pi^{-1}(\mathbf{w}) = (\mathbf{z}, 0)$ for some $\mathbf{z} \in \mathbb{F}_q^n$; the condition $\widehat{g}(\mathbf{w}) = \mathbf{y}_0$ then implies $f(\mathbf{z}) = \mathbf{0}$. On the other hand, the planted preimage $\mathbf{x} = \pi(\boldsymbol{\xi})$ does not lie in H, since $\boldsymbol{\xi} \notin H_0$. Hence, we have

$$\left\{\mathcal{B}_{\mathsf{II}}^{\mathcal{A}} \text{ succeeds}\right\} = S_{\neq} \cap \{\mathbf{w} \in H\}\,.$$

Let $\mathcal{T}_{\neq}$ be the set of all quadruples $\tau = (h, \mathbf{x}_0, \mathbf{y}_0, \mathbf{w}_0)$ satisfying $\mathbf{y}_0 = h(\mathbf{x}_0), \mathbf{w}_0 \neq \mathbf{x}_0, h(\mathbf{w}_0) = \mathbf{y}_0$, and $\Pr[T_\tau] > 0$, where $T_\tau := \{\widehat{g} = h,\ \mathbf{x} = \mathbf{x}_0,\ \mathbf{y} = \mathbf{y}_0,\ \mathbf{w} = \mathbf{w}_0\}$. The events $\{T_\tau\}_{\tau \in \mathcal{T}_{\neq}}$ are pairwise disjoint and partition $S_{\neq}$. Therefore, we obtain

$$\begin{aligned}\mathrm{Succ}_{n,m,q}^{\mathsf{UMQ}}\left(\mathcal{B}_{\mathsf{II}}^{\mathcal{A}}\right) &= \Pr[S_{\neq} \wedge \{\mathbf{w} \in H\}] = \sum_{\tau \in \mathcal{T}_{\neq}} \Pr[T_\tau \wedge \{\mathbf{w} \in H\}] \\ &= \sum_{\tau \in \mathcal{T}_{\neq}} \Pr[\mathbf{w} \in H \mid T_\tau] \Pr[T_\tau] = \sum_{\tau \in \mathcal{T}_{\neq}} \alpha \Pr[T_\tau] = \alpha \Pr[S_{\neq}] = \alpha\epsilon_{\neq}.\end{aligned}$$

Since sampling the coefficients and π, applying π^{-1} and π, and verifying the output take $\mathrm{poly}(n, m, \log q)$ additional time, the running time of $\mathcal{B}_{\mathsf{II}}^{\mathcal{A}}$ is $t' = t + \mathrm{poly}(n, m, \log q)$, completing the proof. □

Proof of Theorem 6

The design of the combined algorithm is given in Algorithm 7. In short, $\mathcal{B}^{\mathcal{A}}$ runs $\mathcal{B}_{\mathsf{I}}^{\mathcal{A}}$ with probability ρ and runs $\mathcal{B}_{\mathsf{II}}^{\mathcal{A}}$ with probability $1-\rho$, where $\rho\mu = (1-\rho)\alpha = \frac{\mu\alpha}{\mu+\alpha}$.

The correctness of Algorithm 7 has been shown in Proposition 2 and 3.

Algorithm 7 The combined UMQ solver $\mathcal{B}^{\mathcal{A}}$

Input: A (n, m, q)-UMQ function $f \in \mathcal{Q}_{n,m,q}$; oracle access to an $(n+1, m, q)$-MQOW adversary $\mathcal{A}$
Output: A zero $\mathbf{z} \in \mathbb{F}_q^n$ satisfying $f(\mathbf{z}) = \mathbf{0}$, or $\perp$
1: Set $\rho \leftarrow \alpha/(\mu + \alpha)$
2: Sample $b \leftarrow \mathsf{in}$ with probability ρ; otherwise set $b \leftarrow \mathsf{out}$
3: **if** $b = \mathsf{in}$ **then**
4: **return** $\mathcal{B}_{\mathsf{I}}^{\mathcal{A}}(f)$ using Algorithm 5
5: **else**
6: **return** $\mathcal{B}_{\mathsf{II}}^{\mathcal{A}}(f)$ using Algorithm 6
7: **end if**

Now we calculate the success probability of $\mathcal{B}^{\mathcal{A}}$:

$$\mathrm{Succ}_{n,m,q}^{\mathsf{UMQ}}(\mathcal{B}^{\mathcal{A}}) \geq \rho\mu\epsilon_{=} + (1-\rho)\alpha\epsilon_{\neq} = \frac{\mu\alpha}{\mu+\alpha}(\epsilon_{=} + \epsilon_{\neq}) = \frac{\mu\alpha}{\mu+\alpha}\,\epsilon.$$

Equivalently,

$$\epsilon \leq \left(\frac{1}{\mu} + \frac{1}{\alpha}\right) \mathrm{Succ}_{n,m,q}^{\mathsf{UMQ}}(\mathcal{B}^{\mathcal{A}}) = \left(q^{m-n} + q - q^{-n}\right) \mathrm{Succ}_{n,m,q}^{\mathsf{UMQ}}(\mathcal{B}^{\mathcal{A}}),$$

The $\mathcal{B}^{\mathcal{A}}$ invokes $\mathcal{A}$ once, and all remaining operations require $\mathrm{poly}(n, m, \log q)$ time, hence the running time of $\mathcal{B}^{\mathcal{A}}$ is $t' = t + \mathrm{poly}(n, m, \log q)$, and

$$\mathrm{Succ}_{\mathcal{Q}_{n+1,m,q}}^{\mathsf{OW}}(\mathcal{A}) \leq \left(q^{m-n} + q - q^{-n}\right) \mathrm{Succ}_{n,m,q}^{\mathsf{UMQ}}(t').$$

Taking the supremum over all adversaries $\mathcal{A}$ running in time at most t yields

$$\mathrm{InSec}^{\mathsf{OW}}(\mathcal{Q}_{n+1,m,q}; t) \leq \left(q^{m-n} + q - q^{-n}\right) \mathrm{Succ}_{n,m,q}^{\mathsf{UMQ}}(t'),$$

completing the proof. □

Remark 6 (Tightness). When $m \leq n$, the reduction loss is bounded by $2q$, and our reduction is tight; more generally, when $m \leq n + O(1)$, i.e., $m \leq n + C$ for some constant $C > 0$, the reduction loss is bounded by $q + q^C$, and our reduction is tight.

Remark 7. When $m > n + O(\log \lambda)$, the (n, m, q)-UMQ assumption may not imply the $(n+1, m, q)$-MQOW assumption. For instance, when $m = \Theta(n^2)$, an (n, m, q)-UMQ instance has no solution with overwhelming probability [24], making (n, m, q)-UMQ assumption trivially holds. By contrast, an $(n+1, m, q)$-MQOW instance can be solved efficiently using linearization or relinearization techniques [30], so the $(n+1, m, q)$-MQOW assumption does not hold.

Combining MQSPR ⇒ UMQ (Theorem 2) and conditional UMQ ⇒ MQOW (Theorem 6), we have

Corollary 2 (Conditional MQSPR $\Rightarrow$ MQOW). *Suppose $m \leq n+O(\log \lambda)$. Under the (n, m, q)-MQSPR assumption, the (n, m, q)-MQOW assumption holds: for each t, we have*

$$\mathrm{InSec}^{\mathsf{OW}}(\mathcal{Q}_{n,m,q}; t) \leq \left(q^{m-n} + q - q^{-n}\right) \cdot \mathrm{InSec}^{\mathsf{SPR}}(\mathcal{Q}_{n,m,q}; t'),$$

where $t' = t + \mathrm{poly}(n, m, \log q)$.

Remark 8. For a general keyed function family $\mathcal{G}_\lambda = \{g_\lambda : \{0,1\}^n \to \{0,1\}^m \mid k \in \mathcal{K}_\lambda\}$, the implication SPR $\Rightarrow$ OW requires $\mathcal{G}_\lambda$ to be sufficiently compressing: $n = m + \omega(\log \lambda)$ [34]. For the MQ function family $\mathcal{Q}_{n,m,q}$, however, Corollary 2 shows that this compression requirement can be relaxed to $m \leq n + O(\log \lambda)$. In particular, our result covers the square case $m = n$ as well as mildly overdetermined systems satisfying $m = n + O(\log \lambda)$.

5 Conclusion

This paper clarifies the relations among UMQ, MQOW, and MQSPR. Specifically, we prove that UMQ and MQSPR are equivalent, that MQOW implies UMQ, and that UMQ implies MQOW under some condition. These results identify MQOW as a foundational assumption underlying both UMQ and MQSPR in multivariate cryptography.

References

1. Applebaum, B., Haramaty-Krasne, N., Ishai, Y., Kushilevitz, E., Vaikuntanathan, V.: Low-Complexity Cryptographic Hash Functions. In: Papadimitriou, C.H. (ed.) 8th Innovations in Theoretical Computer Science Conference (ITCS 2017). Leibniz International Proceedings in Informatics (LIPIcs), vol. 67, pp. 7:1–7:31. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2017). https://doi.org/10.4230/LIPIcs.ITCS.2017.7
2. Banerjee, A., Waters, B., Wu, D.J.: From NIZK arguments to ZAPs, generically. Cryptology ePrint Archive, Paper 2026/886 (2026), https://eprint.iacr.org/2026/886
3. Bellini, E., Makarim, R.H., Sanna, C., Verbel, J.: An estimator for the hardness of the MQ problem. In: Batina, L., Daemen, J. (eds.) Progress in Cryptology – AFRICACRYPT 2022. Lecture Notes in Computer Science, vol. 13503, pp. 323–347. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-17433-9_14
4. Benadjila, R., Feneuil, T., Rivain, M.: MQ on my mind: Post-quantum signatures from the non-structured multivariate quadratic problem. In: 9th IEEE European Symposium on Security and Privacy, EuroS&P 2024, Vienna, Austria, July 8–12, 2024. pp. 468–485. IEEE (2024). https://doi.org/10.1109/EuroSP60621.2024.00032
5. Berbain, C., Gilbert, H., Patarin, J.: QUAD: A practical stream cipher with provable security. In: Vaudenay, S. (ed.) Advances in Cryptology – EUROCRYPT 2006. Lecture Notes in Computer Science, vol. 4004, pp. 109–128. Springer, Berlin, Heidelberg (2006). https://doi.org/10.1007/11761679_8
6. Beullens, W.: MAYO: Practical post-quantum signatures from oil-and-vinegar maps. In: AlTawy, R., Hülsing, A. (eds.) Selected Areas in Cryptography. pp. 355–376. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-030-99277-4_17
7. Billet, O., Robshaw, M.J.B., Peyrin, T.: On building hash functions from multivariate quadratic equations. In: Pieprzyk, J., Ghodosi, H., Dawson, E. (eds.) Information Security and Privacy – ACISP 2007. Lecture Notes in Computer Science, vol. 4586, pp. 82–95. Springer, Berlin, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73458-1_7
8. Björklund, A., Kaski, P., Williams, R.: Solving Systems of Polynomial Equations over GF(2) by a Parity-Counting Self-Reduction. In: Baier, C., Chatzigiannakis, I., Flocchini, P., Leonardi, S. (eds.) 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019). Leibniz International Proceedings in Informatics (LIPIcs), vol. 132, pp. 26:1–26:13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2019). https://doi.org/10.4230/LIPIcs.ICALP.2019.26, http://drops.dagstuhl.de/opus/volltexte/2019/10602
9. Bouillaguet, C., Chen, H.C., Cheng, C.M., Chou, T., Niederhagen, R., Shamir, A., Yang, B.Y.: Fast exhaustive search for polynomial systems in $\mathbb{F}_2$. In: Mangard, S., Standaert, F.X. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2010. pp. 203–218. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
10. Bouillaguet, C., Feneuil, T., Maire, J., Rivain, M., Sauvage, J., Vergnaud, D.: Multivariate commitments and signatures with efficient protocols. Cryptology ePrint Archive, Paper 2025/2035 (2025), https://eprint.iacr.org/2025/2035
11. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. Journal of the ACM **51**(4), 557–594 (Jul 2004). https://doi.org/10.1145/1008731.1008734, https://doi.org/10.1145/1008731.1008734

12. Chen, M.S., Hülsing, A., Rijneveld, J., Samardjiska, S., Schwabe, P.: From 5-pass $\mathcal{MQ}$-based identification to $\mathcal{MQ}$-based signatures. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology – ASIACRYPT 2016. Lecture Notes in Computer Science, vol. 10032, pp. 135–165. Springer, Berlin, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_5
13. Dao, Q., Jain, A., Jin, Z.: Non-interactive zero-knowledge from LPN and MQ. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology – CRYPTO 2024. Lecture Notes in Computer Science, vol. 14928. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-68400-5_10, https://doi.org/10.1007/978-3-031-68400-5_10
14. Ding, J.: A new proof of work for blockchain based on random multivariate quadratic equations. In: Zhou, J., Deng, R., Li, Z., Majumdar, S., Meng, W., Wang, L., Zhang, K. (eds.) Applied Cryptography and Network Security Workshops. Lecture Notes in Computer Science, vol. 11605, pp. 97–107. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29729-9_5
15. Ding, J., Petzoldt, A., Schmidt, D.S.: Multivariate public key cryptosystems. Springer Nature (2020)
16. Ding, J., Schmidt, D.: Cryptanalysis of hfev and internal perturbation of hfe. In: Vaudenay, S. (ed.) Public Key Cryptography – PKC 2005, Lecture Notes in Computer Science, vol. 3386, pp. 288–301. Springer, Berlin, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30580-4_20
17. Ding, J., Yang, B.Y.: Multivariates polynomials for hashing. In: Pei, D., Yung, M., Lin, D., Wu, C. (eds.) Information Security and Cryptology. pp. 358–371. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
18. Dinur, I.: Improved algorithms for solving polynomial systems over GF(2) by multiple parity-counting. In: Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms. pp. 2550–2564. SODA '21, Society for Industrial and Applied Mathematics (2021). https://doi.org/10.1137/1.9781611976465.151
19. Faugère, J.C.: A new efficient algorithm for computing gröbner bases (f4). Journal of Pure and Applied Algebra **139**(1–3), 61–88 (1999). https://doi.org/10.1016/S0022-4049(99)00005-5
20. Faugère, J.C.: A new efficient algorithm for computing gröbner bases without reduction to zero (f5). In: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation. pp. 75–83. ISSAC '02, Association for Computing Machinery, New York, NY, USA (2002). https://doi.org/10.1145/780506.780516
21. Faugère, J.C., Safey El Din, M., Verron, T.: On the complexity of computing gröbner bases for weighted homogeneous systems. Journal of Symbolic Computation **76**, 107–141 (2016). https://doi.org/10.1016/j.jsc.2015.12.001
22. Feneuil, T.: Building MPCitH-based signatures from MQ, MinRank, and Rank SD. In: Pöpper, C., Batina, L. (eds.) Applied Cryptography and Network Security. Lecture Notes in Computer Science, vol. 14583. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-54770-6_16, https://doi.org/10.1007/978-3-031-54770-6_16
23. Furue, H., Nakamura, S., Takagi, T.: Improving thomae-wolf algorithm for solving underdetermined multivariate quadratic polynomial problem. In: Cheon, J.H., Tillich, J.P. (eds.) Post-Quantum Cryptography – PQCrypto 2021. Lecture Notes in Computer Science, vol. 12841, pp. 87–106. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-81293-5_4

24. Fusco, G., Bach, E.: Phase transition of multivariate polynomial systems. In: Cai, J.Y., Cooper, S.B., Zhu, H. (eds.) Theory and Applications of Models of Computation. Lecture Notes in Computer Science, vol. 4484, pp. 632–645. Springer, Berlin, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72504-6_58, https://doi.org/10.1007/978-3-540-72504-6_58
25. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, San Francisco (1979)
26. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing **28**(4), 1364–1396 (1999). https://doi.org/10.1137/S0097539793244708
27. Huang, Y.J., Liu, F.H., Yang, B.Y.: Public-key cryptography from new multivariate quadratic assumptions. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) Public Key Cryptography – PKC 2012. Lecture Notes in Computer Science, vol. 7293, pp. 190–205. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30057-8_12
28. Impagliazzo, R., Luby, M.: One-way functions are essential for complexity-based cryptography. In: 30th Annual Symposium on Foundations of Computer Science. pp. 230–235. IEEE (1989). https://doi.org/10.1109/SFCS.1989.63483
29. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: Stern, J. (ed.) Advances in Cryptology — EUROCRYPT '99. pp. 206–222. Springer Berlin Heidelberg, Berlin, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_15
30. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE public key cryptosystem by relinearization. In: Wiener, M. (ed.) Advances in Cryptology — CRYPTO '99. Lecture Notes in Computer Science, vol. 1666, pp. 19–30. Springer, Berlin, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_2, https://doi.org/10.1007/3-540-48405-1_2
31. Kundu, N., Debnath, S.K., Mishra, D.: 1-out-of-2: Post-quantum oblivious transfer protocols based on multivariate public key cryptography. Sādhanā **45**, 209 (2020). https://doi.org/10.1007/s12046-020-01447-6, https://doi.org/10.1007/s12046-020-01447-6
32. Liu, F.H., Lu, C.J., Yang, B.Y.: Secure PRNGs from specialized polynomial maps over any $\mathbb{F}_q$. In: Buchmann, J., Ding, J. (eds.) Post-Quantum Cryptography. Lecture Notes in Computer Science, vol. 5299, pp. 181–202. Springer, Berlin, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88403-3_13
33. May, A., Ostuzzi, M., Ressler, H.: Just guess: Improved (quantum) algorithm for the underdetermined MQ problem. In: Daemen, J., Thomé, E. (eds.) Advances in Cryptology – EUROCRYPT 2026. Lecture Notes in Computer Science, vol. 16544, pp. 352–382. Springer, Cham (2026). https://doi.org/10.1007/978-3-032-25327-9_12
34. Rogaway, P., Shrimpton, T.: Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In: International workshop on fast software encryption. pp. 371–388. Springer (2004)
35. Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing. pp. 387–394. ACM (1990). https://doi.org/10.1145/100216.100269
36. Sakumoto, K., Shirai, T., Hiwatari, H.: Public-key identification schemes based on multivariate quadratic polynomials. In: Rogaway, P. (ed.) Advances in Cryptology – CRYPTO 2011. Lecture Notes in Computer Science, vol. 6841,

pp. 706–723. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_40
37. Tao, C., Xiang, H., Petzoldt, A., Ding, J.: Simple matrix – a multivariate public key cryptosystem (mpkc) for encryption. Finite Fields and Their Applications **35**, 352–368 (2015). https://doi.org/10.1016/j.ffa.2015.06.001, https://www.sciencedirect.com/science/article/pii/S1071579715000593
38. Thomae, E., Wolf, C.: Solving underdetermined systems of multivariate quadratic equations revisited. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) Public Key Cryptography – PKC 2012. Lecture Notes in Computer Science, vol. 7293, pp. 156–171. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30057-8_10
39. Wang, H., Zhang, H., Wu, Q., Zhang, Y., Li, C., Zhang, X.: Design theory and method of multivariate hash function. Science China Information Sciences **53**(10), 1977–1987 (2010). https://doi.org/10.1007/s11432-010-4062-1
40. Zhuang, Z., Zhang, Y., Ding, J.: Trapdoor-free multivariate one-time signatures in the standard model. Cryptology ePrint Archive, Paper 2025/952 (2025), https://eprint.iacr.org/2025/952