Generic Zero-Knowledge and Multivariate Quadratic Systems

Alan Szepieniec and Bart Preneel

Dept. Electrical Engineering,
ESAT/COSIC, KU Leuven,
and imec, Belgium
{first-name}.{last-name}@esat.kuleuven.be

Abstract. Zero-knowledge proofs are a core building block for a broad range of cryptographic protocols. This paper introduces a generic zero-knowledge proof system capable of proving the correct computation of any circuit. Our protocol draws on recent advancements in multiparty computation and its security relies only on the underlying commitment scheme. Furthermore, we optimize this protocol for use with multivariate quadratic systems of polynomials, leading to provably secure signatures from multivariate quadratic systems, with keys that scale linearly and signatures that scale quadratically with the security parameter.

Keywords: zero-knowledge proof, post-quantum, signature, multivariate quadratic, provable security, multi-party computation

1 Introduction

One of the central tasks of cryptography is to provide users with the capability to produce evidence for particular claims while at the same time preserving privacy. In particular, *zero-knowledge proofs* enable one party, the prover, to convince another party, the verifier, beyond a shadow of a doubt that he knows secret information satisfying public criteria *without ever revealing it* [18].

There is an intrinsic link between zero-knowledge proofs and signature schemes: any sufficiently general construction for zero-knowledge proofs can be transformed into a digital signature scheme by using the Fiat-Shamir transform [13]. This relatively simple transformation has spawned a multitude of signature schemes, although there are many exceptions that do not arise from zero-knowledge constructions. Likewise, there exist many applications for zero-knowledge proofs beyond signature schemes, such as *e.g.* verifiable secret sharing, anonymous credentials, or anonymous mix networks.

Consequently, there is a large body of research attempting to build zero-knowledge proofs on top of minimalistic assumptions, such as the existence one-way functions. For example, it was shown early on by Goldreich *et al.* [17] that if such a one-way function exists, then there exist protocols for proving any **NP** statement in zero-knowledge. Aside from their implications on computational complexity theory, these minimalistic protocols are interesting from a cryptographic perspective because they offer a modicum of defense against unforeseeable advances in the cryptanalysis of any specific primitive: namely, the ability to switch to another. For example, when large-scale quantum computers are built, any protocol based on the discrete logarithm problem will be considered broken; however, any minimalistic protocol can retain security simply by switching to a primitive that is not broken by Shor's quantum algorithm [30].

In the same vein of zero-knowledge proofs on top of minimalistic assumptions, Ishai *et al.* [20] show an intrinsic link between zero-knowledge and *multiparty computation (MPC)*. In particular, any MPC protocol can be turned into an interactive or non-interactive zero-knowledge proof. Their construction relies on the prover's simulation of an MPC protocol "in the head", *i.e.*, by simulating all players in the protocol. The prover's first message to the verifier consists of a binding commitment to each player's contribution to the protocol. The verifier's challenge identifies the players whose protocol contributions are to be shown in the prover's second message. If the prover is cheating, his fraud will be laid bare with high probability by the verifier's random audit.

In addition to making only minimalistic assumptions, both Goldreich *et al.* and Ishai *et al.* propose protocols that are *generic*, *i.e.*, apply to any circuit or polynomial-time computation. In order for the protocol to be meaningfully zero-knowledge, this circuit or polynomial-time computation must itself be one-way. Otherwise, the attacker can obtain the secret preimage by applying the inverse function to the known output image.

One-way functions are by definition computable in polynomial time and the set of images associated with one family of one-way functions is a language in **NP**; the matching witness is given by the corresponding preimage. Consequently, the generic constructions for minimalistic zero-knowledge proofs are capable of proving knowledge of the preimage under any one-way function. The tempting combination of such a proof with a message-dependent non-interactive challenge trivially yields a provably secure signature scheme whose security relies on the difficulty of finding inverse images.

However, not any one-way function will do. Many cryptographically secure one-way functions such as hash functions are not designed with zero-knowledge proofs in mind. As a result, a generic zero-knowledge proof of knowledge of a preimage is likely to be bloated and difficult to generate. A one-way function must be tailored to the zero-knowledge proof for an efficient signature scheme.

Contributions. We propose a zero-knowledge proof system which is generic in two senses: on the one hand, it is *minimalistic*, relying only on the random oracle model for security; and on the other, it allows the correct computation of *any arithmetic circuit* to be proven. Our protocol draws on recent advances in multiparty computation such as information-theoretically secure and linearly homomorphic MACs, as well as pre-computed multiplication triples for non-linear operations [5]. The only cryptographic primitive the protocol requires in practice is a collision-resistant hash function for secure commitments.

Moreover, we propose optimizations for this protocol that apply in special cases. For instance, the circuit whose correct computation is proven can contain *field inversion* gates; these require just one multiplication triple. Another example is the use of field extensions to *pack bits* and thus make more efficient use of the wires and gates in a given circuit. Most importantly, we extend the notion of multiplication triples to *bilinear triples* which enable the computation of any bilinear or quadratic form by representing it as a single gate.

We apply these last two optimizations to the case where the underlying circuit is a set of multivariate quadratic polynomials. In particular, we propose an optimized version of our protocol that is capable of proving knowledge of the preimage of *any multivariate quadratic function* using only *five* bilinear triples, resulting in protocol transcripts that scale quadratically in the security parameter, as opposed to the cubic scaling associated with a complete description of the quadratic system. While zero-knowledge proofs for multivariate quadratic systems are interesting in their own right, we focus on the non-interactive case in which the protocol boils down to a signature scheme. In contrast to other multivariate quadratic signature schemes, our signature scheme is provably secure — but at the cost of larger signatures.

In contrast to other generic zero-knowledge proof as well as identification schemes relying on post-quantum hard problems, our interactive protocols are proven secure against quantum adversaries. To this end, we invoke Unruh's notions of *quantum proof of knowledge* [31] as that which is to be proven, in addition to *collapsing* hash functions and *collapse-binding commitments* [34] as a mediate goal. In addition to that, we employ El Yousfi *et al.*'s $(2n+1)$-pass generalization of special soundness, called *special n-soundness* [1]. In a nutshell, collapse-binding commitments allow us to rewind the prover to just before the last message was sent. We change the last challenge of the verifier and obtain a different protocol transcript. Special n-soundness guarantees it is possible to extract the witness from these transcripts.

These strategies only apply to the interactive protocols. For non-interactivity we apply the Fiat-Shamir transform [13]; the transformed protocol is known to retain security in the classical random oracle model. Unfortunately, some proof techniques fail in the *quantum random oracle model* [6]. For instance, Dagdelen *et al.* show that if the underlying interactive protocol is actively secure and uses commitments that are independent of the witness, black box extractors cannot exist for the non-interactive protocol [10]. Ambainis *et al.* show that, relative to an oracle, some Fiat-Shamir-transformed protocols are classically-secure but quantumly-insecure, even if they rely only on quantumly-hard problems [2]. On the bright side, Unruh shows that it *is* possible to transform a specially sound protocol into a non-interactive version retaining quantum security by using extractable commitments and by forcing the protocol to adopt a split-and-choose strategy [32]. We prefer to use the much more message-size-friendly Fiat-Shamir transform with collapse-binding commitments and remark that our protocol is by design already a split-and-choose type protocol. Security is proven against classical adversaries only but we conjecture that it also holds in the quantum case. The same strategy is adopted by many, if not all, other post-quantum zero-knowledge-based signature schemes [1, 7–9, 21, 22, 24, 28, 29].

Related work. The graph-3-colorability protocol by Goldreich *et al.* [17] was the first generic (in both senses) zero-knowledge proof as graph-3-colorability is **NP**-complete. The first construction to explicitly target *cryptographic* applications (as opposed to *complexity-theoretic* theorems) was indeed the MPC-in-the-head protocol proposed by Ishai *et al.* [20]. In 2014, Ranelucci, Tapp and Zakarias [27] proposed another protocol that accomplishes the same in a different way, namely by emphasizing the cryptographic power of commitments to individual bits. Ranellucci *et al.* benchmark their scheme using the AES circuit, where they prove knowledge of the secret key that maps a known plaintext to a known ciphertext. More recently, Giacomelli *et al.* [15] define several improved variants of the MPC-in-the-head technique and benchmark their scheme using the circuits for SHA-1 and SHA-256, demonstrating their practical feasibility.

At CRYPTO 2011, Sakumoto *et al.* proposed an interactive zero-knowledge proof system for proving knowledge of the preimage associated with the output image under a public multivariate quadratic function [29]. Their protocol applies the split-and-choose pattern to the polar form of the system of polynomials, thus convincing the verifier that the opened shares testify to the validity of the claim while hiding its secret information-theoretically. This construction gives rise to a provably secure identification scheme with a probability of successful impersonation of 2^{-30}. To reduce this number to a level that is acceptable for signatures, the protocol must be repeated several times. Therefore, the protocol's relatively low communication cost of some 26 565 bits for a single execution only represents a fraction of the size of the signature. In contrast, our scheme incurs a communication cost of 29 600 bits for the same security level of 80 bits. Moreover, as it is a signature scheme by design it does not require any repetition.

More recently, Chen *et al.* present an optimized implementation of the signature scheme based on the Sakumoto *et al.* protocol targeting a 128 bit security level against quantum computers [9]. Their signatures clock in at 327 616 bits whereas ours are 356 608 bits in size for the same security level or even 414 592 bits if we mimic their parameter choices. While our scheme seems to be outperformed for the purpose of post-quantum signatures with provable security based on the MQ problem, we argue that the contributions of this paper still stand. For instance, the novel constructions introduced in this paper may inspire further improvements that do in the end break the Chen *et al.* record; or inspire improvements in other fields altogether. Moreover, our scheme is naturally flexible: it allows the user to prove knowledge of the solution to an MQ problem and *link this information to other circuit satisfiability problems*. It is not clear whether constructions based on Sakumoto *et al.*'s identification scheme are capable of such composability.

2 Preliminaries

Arithmetic Circuits. Every function $\mathcal{H} : \{0,1\}^n \to \{0,1\}^m$ of fixed input and output length can be represented as an arithmetic circuit over any finite field $\mathbb{F}_q$. An *arithmetic circuit* is a directed acyclic graph $(\mathcal{G}, \mathcal{E})$ whose edges or *wires* represent elements in $\mathbb{F}_q$ and whose vertices or *gates* represent operations on them. We employ *bit packing* when one such field element represents more than one bit. In addition to fields, arithmetic circuits may be defined over many other algebraic structures. In particular, for our purposes we consider arithmetic fields over the ring of univariate polynomials $\mathbb{F}_q[x]$, where a wire's value is represented by the polynomial's value at $x = 0$.

Negligible. A function $\epsilon : \mathbb{N} \to \mathbb{R}_{>0}$ is *negligible* if for all polynomials $p(x) \in \mathbb{R}[x]$ there is an $N \in \mathbb{N}$ such that for all $x > N$, $\epsilon(x)$ drops faster than the reciprocal of $|p(x)|$. Formally, we need only consider the dominant monomial of $p(x)$:

$$\forall c > 1 \,.\, \exists N \in \mathbb{N} \,.\, \forall \kappa > N \,.\, \epsilon(\kappa) \leq \frac{1}{\kappa^c} \enspace .$$

From here on, any reference to negligible functions drops the quantifiers from the notation. However, they are still implicitly present whenever asymptotical security notions, the *security parameter* κ, or the exponent c appear.

Zero Knowledge. An *interactive proof system* for a language $\mathcal{L} \in \mathbf{NP}$ is a protocol between a pair of interactive Turing machines (called the *prover* $\mathcal{P}$ and *verifier* $\mathcal{V}$, respectively) whose common input is a string $\ell \in \{0,1\}^*$. The secret information for the prover is a witness $v \in \{0,1\}^*$ that certifies that $\ell \in \mathcal{L}$, *i.e.*, $\mathcal{L}(\ell, v) = 1$. At the end of the protocol, the verifier outputs a single bit, denoted $\langle \mathcal{P}(v), \mathcal{V} \rangle(\ell)$, which is 1 if he accepts and 0 if he rejects. The *transcript* consists of all messages sent between the two parties and the distribution of transcripts is denoted $\mathcal{T}\langle \mathcal{P}(v), \mathcal{V} \rangle(\ell)$. We aim to satisfy three properties [4,16]:

1. *Completeness.* For every $\ell \in \mathcal{L}$ and matching certificate v, an honest prover will likely convince an honest verifier:

$$\forall \ell \in \{0,1\}^*, v \in \{0,1\}^* \,.\, \mathcal{L}(\ell, v) = 1 \Rightarrow \Pr[\langle \mathcal{P}(v), \mathcal{V} \rangle(\ell) = 1] \geq \frac{2}{3} \enspace .$$

2. *Soundness.* For every $\ell \notin \mathcal{L}$ no prover $\mathcal{B}$ is likely to convince the verifier:

$$\forall \ell \notin \mathcal{L} \,.\, \forall \mathcal{B} \,.\, \Pr[\langle \mathcal{B}, \mathcal{V} \rangle(\ell) = 1] \leq \frac{1}{3} \enspace .$$

   The probability $\frac{1}{3}$ is an arbitrary cutoff point. More generally, the proof has *soundness error* σ whenever $\Pr[\langle \mathcal{B}, \mathcal{V} \rangle(\ell) = 1] \leq \sigma$ and as long as σ is not negligibly close to 1 the notion is meaningful because repeating the protocol will send the soundness error to a negligible quantity in the security parameter.

2*. In some cases we want a stronger version of zero-knowledge proofs, namely zero-knowledge *proofs of knowledge* which prove not only that $\ell \in \mathcal{L}$ but also that $\mathcal{P}$ "knows" a witness v and in this case we write the proved claim concisely as $\mathsf{ZKPoK}\{(v) \ : \ \mathcal{L}(\ell, v) = 1\}$ or even $\mathsf{ZKQPoK}\{\cdot\}$ if the prover is a quantum computer. This notion extends the soundness property by requiring the existence of a probabilistic (respectively, quantum) polynomial-time Turing machine $\mathcal{E}^{\mathcal{P}}$ ("the extractor") with black-box oracle access to $\mathcal{P}$, capable of outputting a witness v for ℓ with a probability related to $\mathcal{P}$'s probability of convincing $\mathcal{V}$:

$$\exists \mathcal{E} \,.\, \forall \mathcal{P} \,.\, \Pr[\langle \mathcal{P}, \mathcal{V} \rangle(\ell) = 1] \geq 1 - \sigma \ \Rightarrow \ \Pr[\mathcal{E}^{\mathcal{P}}(\ell) = v] \geq 1 - \sigma \ .$$

Here σ represents the *knowledge error.* If $\mathcal{P}$ and $\mathcal{E}$ are quantum computers, black box oracle access is defined following Unruh [31]: The prover's computations before and between sending and receiving messages are described by exponentially-large unitary matrices $\mathfrak{P}_0, \ldots, \mathfrak{P}_n$ acting on a quantum state $|\Psi\rangle$ and where the first acts on the all-zero-qubits state $|\Psi\rangle = |0\rangle$. The extractor can apply these unitaries as well as their inverses to his own quantum register $|\Phi\rangle$.

2**. An even stronger variant called *special n-soundness* [1] may apply when the transcript of the protocol consists of $2n + 1$ messages of which every second one is from the verifier; this property is satisfied whenever a probabilistic polynomial-time extractor $\mathcal{E}$ is able to extract the witness v from any pair of transcripts $T_1 \neq T_2$ that prove the same claim and differ only in the last two messages.

$$\begin{aligned} &\exists \mathcal{E} \,.\, \forall T_1, T_2 \sim \mathcal{T}\langle \mathcal{P}(v), \mathcal{V} \rangle(\ell) \,. \\ &\qquad \Big(\big(\forall i \in \{1, \ldots, 2n-1\} \,.\ T_1[i] = T_2[i]\big) \wedge T_1 \neq T_2 \Big) \ \Rightarrow \ \mathcal{E}(T_1, T_2) = v \ . \end{aligned}$$

This notion generalizes *special soundness* which applies when $n = 1$.

3. *Zero knowledge.* For all probabilistic polynomial-time verifiers $\mathcal{V}$ there exists simulator $\mathcal{S}_{\mathcal{V}}$ capable of producing a transcript $T \leftarrow \mathcal{S}_{\mathcal{V}}(\ell)$ (or equivalently $T \sim \mathcal{S}_{\mathcal{V}}(\ell)$) of the protocol without knowledge of v (and indeed, regardless of whether $\ell \in \mathcal{L}$), such that T is indistinguishable from authentic transcripts of protocol executions between $\mathcal{P}(v)$ and $\mathcal{V}$. The restriction of the same notion to only verifiers that follow the protocol specifications is named *honest-verifier zero knowledge.* We consider two variants of indistinguishability: *perfect indistinguishability*, which holds if the distributions of counterfeit and authentic transcripts are identical:

$$\mathcal{S}(\ell) = \mathcal{T}\langle \mathcal{P}(v), \mathcal{V} \rangle(\ell) \ ;$$

and *computational indistinguishability*, which holds if there exists no probabilistic (or quantum) polynomial-time Turing machine $\mathcal{D}$ who has more than a negligible advantage over a random guess in distinguishing authentic from counterfeit transcripts:

$$\forall \mathcal{D} \,.\, \big|\Pr[\mathcal{D}(T_a) = 1 | T_a \sim \mathcal{T}\langle \mathcal{P}(v), \mathcal{V} \rangle(\ell)] - \Pr[\mathcal{D}(T_s) = 1 \,|\, T_s \sim \mathcal{S}(\ell)]\big| \leq \frac{1}{|\ell|^c} \ .$$

Commitment scheme. An important primitive frequently used in zero-knowledge proofs is a *commitment scheme.* This subprotocol consists of two phases: a *commit* phase, in which the prover binds himself to a choice without revealing it; and a *reveal* phase, in which the prover reveals a value that was previously committed to. As both phases represent messages sent by the prover, we denote by $\mathsf{com}(v;r)$ and $\mathsf{rev}(v;r)$ the commitment and opening of a message v, bearing in mind that the randomness r may be omitted from this notation. The verifier's role is to receive the value v, or to reject (*i.e.*, output $\perp$) if either message is malicious. This should be possible regardless of the randomness used: $\mathcal{V}(\mathsf{com}(v;r), \mathsf{rev}(v;r)) = v$. We require two more properties from a commitment scheme:

1. *Hiding.* Commitments are semantically secure. That is to say, for all quantum polynomial-time distinguishers $\mathcal{D}$ and for all pairs of values (v_1, v_2), $\mathcal{D}$ has a negligible advantage over a random guess in distinguishing the messages' commitments:
$$\forall \mathcal{D} \,.\, \forall v_1, v_2 \,.\, \big|\Pr[\mathcal{D}(\mathsf{com}(v_1)) = 1] - \Pr[\mathcal{D}(\mathsf{com}(v_2)) = 1]\big| \leq \frac{1}{(|v_1| + |v_2|)^c} \;\; .$$
2. *Binding.* All quantum polynomial-time cheating provers $\mathcal{B}$ have a negligible probability of producing a message $\mathcal{B}(v,r) \neq \mathsf{rev}(v,r)$ that will cause the verifier to receive and accept an alternative value $v' \neq v$:
$$\forall \mathcal{B} \,.\, \Pr[v \neq \mathcal{V}(\mathsf{com}(v;r), \mathcal{B}(v,r)) \neq \perp] \leq \frac{1}{(|v| + |r|)^c} \;\; .$$

2*. A stronger variant of binding particularly relevant for quantum computers is *collapse-binding*, defined along the same lines as collapsing hash functions [34]. Consider a pair of quantum algorithms $\mathcal{A}$ and $\mathcal{B}$, where $\mathcal{A}$ outputs three quantum states M, S, U and one classical message h, and where $\mathcal{B}$ takes the three quantum states and outputs a single classical bit. Such a pair is *valid* if whenever M and U are measured in the computational basis, thus giving $m \leftarrow \mathsf{M}(M)$ and $u \leftarrow \mathsf{M}(U)$, then $\mathcal{V}(h,u) = m$. Consider the following two games:
$$\begin{aligned}
&\mathsf{Game1} : (S,M,U,h) \leftarrow \mathcal{A}(1^\kappa); \quad m \leftarrow \mathsf{M}(M); \quad b \leftarrow \mathcal{B}(1^\kappa, S, M, U) \\
&\mathsf{Game2} : (S,M,U,h) \leftarrow \mathcal{A}(1^\kappa); \qquad\qquad\qquad\quad\; b \leftarrow \mathcal{B}(1^\kappa, S, M, U) \;\; .
\end{aligned}$$
Then a commitment scheme $(\mathsf{com}, \mathsf{rev}, \mathcal{V})$ is *collapse-binding* if every quantum polynomial-time adversary $(\mathcal{A}, \mathcal{B})$ has at most a negligible advantage in distinguishing the two games:
$$\forall (\mathcal{A}, \mathcal{B}) \,.\, |\Pr[b = 1 \,|\, \mathsf{Game1}] - \Pr[b = 1 \,|\, \mathsf{Game2}]| \leq 1/\kappa^c \;\; .$$

Our results require only a generic hiding and collapse-binding commitment scheme. For the purpose of estimating operational costs we use the canonical commitment scheme which is defined by
$$\begin{aligned}
&\mathsf{com} : \{0,1\}^* \times \{0,1\}^\kappa \to \{0,1\}^\kappa : (v;r) \mapsto \mathsf{RO}(v\|r) \\
&\mathsf{rev} : \{0,1\}^* \times \{0,1\}^\kappa \to \{0,1\}^* : (v,r) \mapsto v\|r \;\; .
\end{aligned}$$

Unruh shows that this commitment scheme is collapse-binding in the quantum random oracle model and remains collapse-binding even if the random oracle (RO) is instantiated by a collapsing hash function. Furthermore, the same author gives evidence that popular hash functions such as SHA2-256 may well be collapsing already, even though the definition is relatively new [33, 34].

Signature scheme. A public key signature scheme is defined as a triple of polynomial-time algorithms $(\mathcal{G}, \mathcal{S}, \mathcal{V})$. The probabilistic key generation algorithm produces a secret and public key: $\mathcal{G}(1^\kappa) = (\mathsf{sk}, \mathsf{pk})$; the possibly probabilistic signature generation algorithm produces a signature: $s = \mathcal{S}(\mathsf{sk}, m) \in \{0,1\}^*$. The verification algorithm takes the public key, the message and the signature and decides if the signature is valid $\mathcal{V}(\mathsf{pk}, m, s) \in \{0,1\}$, which it should be if and only if the signature matches both the public key and the message.

Security is defined with respect to the Existential Unforgeability under Chosen Message Attack (EUF-CMA) game [19] between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$, both probabilistic polynomial-time Turing machines. The challenger generates a key pair and sends the public key to the adversary. The adversary is allowed to make a polynomial number of queries $m_i, i \in \{1, \ldots, q\}, q \leq \kappa^c$, which the challenger signs using the secret key and sends back: $s_i \leftarrow \mathcal{S}(\mathsf{sk}, m_i)$. At the end of the game, the adversary must produce a pair of values (m', s') where m' was not queried before: $m' \notin \{m_i\}_{i=1}^q$. The adversary wins if $\mathcal{V}(\mathsf{pk}, m', s') = 1$.

A signature scheme is defined to be secure in the EUF-CMA model if for all adversaries $\mathcal{A}$, the probability of winning is negligible:

$$\forall \mathcal{A} \,.\, \Pr\left[\begin{array}{l} \mathcal{V}(\mathsf{pk}, m', s') = 1 \\ \wedge\, m' \notin \{m_i\}_{i=1}^q \end{array} \middle| \begin{array}{l} (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathcal{G}(1^\kappa) \\ (\{m_i, s_i\}_{i=1}^{q<\kappa^c}, m', s') \leftarrow \langle \mathcal{C}(\mathsf{sk}), \mathcal{A}\rangle(\mathsf{pk}) \end{array}\right] \leq \frac{1}{\kappa^c} \;.$$

Multivariate quadratic systems. A subset of arithmetic circuits of particular interest to this paper are multivariate quadratic (MQ) systems, consisting of circuits of algebraic degree two. By representing the outputs algebraically as polynomial functions of the inputs, the problem of finding satisfying inputs reduces to the MQ problem:

MQ Problem: Given a set $\mathcal{H} \in (\mathbb{F}_q[x_1, \ldots, x_n])^m$ of m multivariate quadratic polynomials in n variables; find a solution $\mathbf{x} = (x_1, \ldots, x_n)^\mathsf{T} \in \mathbb{F}_q^n$ such that $\mathcal{H}(\mathbf{x}) = \mathbf{0}$.

The associated *MQ Assumption* states that no quantum polynomial-time Turing machine exists that solves uniformly random MQ Problem instances with $m \approx n$. In addition to being **NP**-hard [14], the MQ Problem is empirically hard-on-average for these parameters. Moreover, MQ cryptography is itself a well-established branch of post-quantum public-key cryptography whose cryptosystems depend on the conjectured average-case hardness of this problem [12].

A frequently occurring theme in MQ cryptography is the notion of an extension field $\mathbb{F}_{q^n}$ of $\mathbb{F}_q$, which is constructed by choosing an irreducible degree-n polynomial $f(z) \in \mathbb{F}_q[z]$ and working in the ring of polynomials modulo $f(z)$, *i.e.*, $\mathbb{F}_{q^n} \cong \mathbb{F}_q[z]/\langle f(z)\rangle$. The embedding function $\varphi : \mathbb{F}_q^n \to \mathbb{F}_{q^n}$ maps vectors of base field elements to extension field elements according to $\varphi((a_0, \ldots, a_{n-1})^\mathsf{T}) = a_0 + a_1 z + \cdots a_{n-1} z^{n-1} \in \mathbb{F}_{q^n}$. The same function can be applied to the vector of indeterminates $\mathbf{x}$, in which case we get a single extension field indeterminate $\mathcal{X} = \varphi(\mathbf{x})$. If $m = n$, the system of multivariate polynomials over the base field $\mathcal{H}(\mathbf{x}) \in (\mathbb{F}_q[\mathbf{x}])^m$ can be represented by a single *univariate* polynomial over the extension field $\mathcal{H}(\mathcal{X}) \in \mathbb{F}_{q^n}[\mathcal{X}]$.

Moreover, because $\mathcal{H}(\mathbf{x})$ is quadratic over the base field, $\mathcal{H}(\mathcal{X})$ is *q-quadratic*, meaning that it can be described by coefficients $\gamma_{i,j}, \beta_i, \alpha \in \mathbb{F}_{q^n}$ through the expression $\mathcal{H}(\mathcal{X}) = \sum_{i=0}^{n-1} \sum_{j=i}^{n-1} \gamma_{i,j} \mathcal{X}^{q^i + q^j} + \sum_{i=0}^{n-1} \beta_i \mathcal{X}^{q^i} + \alpha$. We use bold-and-calligraphic capital letters such as $\boldsymbol{\mathcal{X}}$ to indicate that we identify $\mathcal{X} = \varphi(\mathbf{x})$ with the vector $\boldsymbol{\mathcal{X}} = (\mathcal{X}, \mathcal{X}^q, \ldots, \mathcal{X}^{q^{n-1}})^\mathsf{T}$. By doing so we can identify the homogeneous part $\hat{\mathcal{H}}(\mathcal{X})$ of $\mathcal{H}(\mathcal{X})$ with a quadratic form: $\hat{\mathcal{H}}(\mathcal{X}) = \boldsymbol{\mathcal{X}}^\mathsf{T} \mathfrak{H} \boldsymbol{\mathcal{X}}$ where $\mathfrak{H} \in \mathbb{F}_{q^n}^{n \times n}$ is uniquely defined up to an alternating matrix term, *i.e.*, $\mathfrak{H} + A \equiv \mathfrak{H}$ iff $A^\mathsf{T} = -A$. If the unique upper-triangular representation of $\mathfrak{H}$ is chosen, it contains the $\gamma_{i,j}$ coefficients.

3 Generic Zero-Knowledge

3.1 Overview

The function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^*$ to be computed is given as an arithmetic circuit of gates whose inputs and outputs are connected acyclically by wires. Each wire represents an element of the finite field $\mathbb{F}_q$, where for practical purposes q is on the order of several hundred but in principle q has no upper bound. Both addition and multiplication are represented by gates that have two inputs and one output. Furthermore, constant gates have zero inputs and one output and guarantee that their output assumes a given value.

Each wire f is associated with a univariate polynomial $f(x) \in \mathbb{F}_q[x]$ with $\mathsf{deg}(f) \leq d$. The *value* associated with the wire corresponds to the evaluation of the polynomial in zero. The value of f, *i.e.*, $f(0)$, remains information-theoretically hidden from the verifier as long as the verifier knows at most d points on $f(x)$ and $x = 0$ is not among them. If this is the case, then we will denote the wire by $\langle f \rangle$ whereas we will use f to indicate that its value is known by the verifier.

The evaluations of $f(x)$ in d points $x_1, \ldots, x_d$ are MAC values of $f(0)$ and the points $x_1, \ldots, x_d$ are their corresponding keys. The verifier can verify this MAC value as soon as he discovers the full description of $f(x)$ by evaluating the polynomial in the points that make up the MAC keys. The key principle that makes the protocol work is that the prover is ignorant of the MAC keys; the only way he can ensure that necessary relations hold for all possible MAC values is to ensure that the same relations hold for the full polynomials. However, since the verifier knows only d MAC values, he is not in possession of any information on the secret value $f(0)$.

This MAC is homomorphic for addition. Given two wires $\langle f \rangle$ and $\langle g \rangle$ where the evaluations of the associated polynomials in $x_1, \ldots, x_d$ are known by the verifier, the output of the addition gate is easily obtained by both participants. The associated polynomial of $\langle f + g \rangle$ is $f(x) + g(x)$ and hence the MAC value is computed by element-wise addition. Moreover, addition of a known constant is cost-free as the constant is simply added to the wire's value and MACs. Multiplication by a constant is similarly cost-free.

A slightly more difficult problem is multiplication, as the product of two polynomials with degree at most d is of degree up to $2d$. Thankfully, Beaver proposes an elegant solution by way of *multiplication triples* [3], which has since been applied to great effect in the online phase of multiparty computation protocols [5, 11]. Given a triple of wires $\langle a \rangle, \langle b \rangle, \langle c \rangle$ for which it is known that $a(0)b(0) = c(0)$, we can efficiently compute $\langle fg \rangle$ from $\langle f \rangle$ and $\langle g \rangle$ as follows. We open $\langle f - a \rangle$ and $\langle g - b \rangle$ and then compute $(f(0) - a(0))\langle b \rangle$ and $(g(0) - b(0))\langle a \rangle$. Finally, we compute $\langle fg \rangle = \langle c \rangle + (f(0) - a(0))\langle b \rangle + (g(0) - b(0))\langle a \rangle + (f(0) - a(0))(g(0) - b(0))$. No information on $f(0), g(0)$ or $f(0)g(0)$ is leaked. Of course, once a multiplication triple $\langle a \rangle, \langle b \rangle, \langle c \rangle$ has been used, it cannot be reused and hence every multiplication consumes one multiplication triple. The good news is that the multiplication triples are independent of the gate inputs and hence can be generated beforehand.

This leaves us with the problem of proving that a particular multiplication triple $\langle a \rangle, \langle b \rangle, \langle c \rangle$ is indeed a correct one and not the malicious design of a cheating prover. We accomplish this by emulating Bendlin *et al.* [5] and sacrificing another multiplication triple $\langle p \rangle, \langle q \rangle, \langle r \rangle$. In particular, we use the second triple to compute $\langle ab \rangle$ and then open $\langle ab - c \rangle$ to test if it is equal to zero. If exactly one multiplication triple is incorrect, then the result cannot possibly be zero. However, it is possible that this result is zero but *both* triples are incorrect. To prove this is not the case, we incorporate a challenge e_1, e_2 from the verifier, *i.e.*, we sacrifice $\langle p \rangle, \langle q \rangle, \langle r \rangle$ to prove that $e_2\langle a \rangle, e_1\langle b \rangle, e_1e_2\langle c \rangle$ is a correct triple.

Some wires will eventually be opened, but their value must remain hidden until the very end of the protocol, when the MAC values are verified. The prover accomplishes this by committing to the polynomials of all the wires that should be opened before receiving the challenges from the verifier. At the end of the protocol, the prover opens the wires and allows the verifier to verify the commitments.

3.2 Protocol

We present the generic zero-knowledge protocol in two parts: $\mathcal{P}_{\mathsf{ACE}}$ a zero-knowledge proof of correct evaluation of a *linear* arithmetic circuit $\mathcal{H}$ in secret inputs; and $\mathcal{P}_{\mathsf{CMT}}$, a zero-knowledge protocol for proving the correctness of multiplication triples which is necessary for extension to *non-linear* arithmetic circuits. While conceptually distinct, there is little reason to use either protocol in a standalone manner: without the multiplication triples the circuit is linear and cannot hide secrets; and without the linear MAC check it is impossible to make the correct multiplication triple proof complete. We detail the proper construction of the overall protocol in the third part of this section.

Proof of Correct Arithmetic Circuit Evaluation. The protocol $\mathcal{P}_{\mathsf{ACE}}$, depicted visually in Fig. 1, proceeds as follows. The verifier knows the circuit $\mathcal{H} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ and the output $\mathbf{o} \in \mathbb{F}_q^m$, whereas the prover knows the secret input $\mathbf{i} \in \mathbb{F}_q^n$ such that $\mathcal{H}(\mathbf{i}) = \mathbf{o}$. Let I and O denote the sets of indices identifying the input and output wires to the circuit, respectively. The prover starts by choosing n random polynomials $w_i(x)$ with degree at most d to represent $\mathbf{i}$, *i.e.* such that $\mathbf{i} = (w_i(0))_{i \in I}$, and then computes all wires $w_i(x) \, \forall i \in I \cup O$ in the linear circuit $\mathcal{H}$ from these input polynomials. Next, the prover commits to u MACs $w_i(j) \, \forall i \in I \, \forall j \in U$ as well as to the polynomials of all output wires $w_i(x) \, \forall i \in O$, by sending these commitments to the verifier. The verifier chooses a random size-d subset $J \subset U$ indicating the MAC keys of his random audit. The prover responds to this challenge by revealing the requested MAC values and also by revealing the polynomials of all output wires. In the final step, the verifier verifies that the MAC values of the output wires satisfy the linear relations implied by $\mathcal{H}$, and that the list of output polynomials $(w_i(x))_{i \in O}$ evaluate to $\mathbf{o}$ in 0.

In order to make the protocol work for non-linear circuits in addition to linear ones, we must include k multiplication triples in the input $\mathbf{i}$, where k is the number of multiplication gates in $\mathcal{H}$. In order for the verifier to obtain the subset of weights a_k for non-linear gates, the wires $f(x) - a(x)$ and $g(x) - b(x)$ must be included in the output; by evaluating these in $x = 0$, the verifier obtains the weights and constants required to compute the MAC values of $f(x)g(x) = c(x) - (f(0) - a(0))b(x) - (g(0) - b(0))a(x) - (f(0) - a(0))(g(0) - b(0))$.

Proof of Correct Multiplication Triples. The interactive proof of correct multiplication triples is intended as a subprotocol of the overall proof. This allows us to assume that the verifier knows the MAC values of all wires corresponding to a set of d MAC keys J, which is a subset of the set of all possible MAC keys U. The verifier knows J but the prover is ignorant of these values. In order to succeed with high probability in convincing the verifier that the additive relations between the wires hold, the prover must guarantee the additive relations for all possible MAC keys — which is only possible if the wires themselves satisfy the linear relations.

The protocol $\mathcal{P}_{\mathsf{CMT}}$ proceeds as follows. The prover holds k sets of triples of polynomials $\mathbf{a}(x), \mathbf{b}(x), \mathbf{c}(x)$, denoted in bold to indicate that these are vectors of k elements. These represent k multiplication triples, *i.e.*, the element-wise product $\mathbf{a}(0) \cdot \mathbf{b}(0)$ is equal to the vector $\mathbf{c}(0)$. Similarly, the prover possesses another k size set of multiplication triples $\mathbf{f}(x), \mathbf{g}(x), \mathbf{h}(x)$ which will be sacrificed to prove that the first set are correctly formed. The verifier holds the evaluations of all these polynomials in the d points $J \subset U$, where J is kept secret from the prover. The protocol is started by the verifier who chooses a pair of k-size vectors over $\mathbb{F}_q$ to challenge with, denoted by $\mathbf{e_1}$ and $\mathbf{e_2}$. Upon receiving this challenge, the prover computes the polynomial vectors $\boldsymbol{\delta}(x) = \mathbf{e_1} \cdot \mathbf{b}(x) - \mathbf{g}(x)$, $\boldsymbol{\epsilon}(x) = \mathbf{e_2} \cdot \mathbf{a}(x) - \mathbf{f}(x)$, and $\mathbf{k}(x) = \mathbf{e_1} \cdot \mathbf{e_2} \cdot \mathbf{c}(x) - \mathbf{h}(x) - \boldsymbol{\delta}(0) \cdot \boldsymbol{\epsilon}(0) - \boldsymbol{\delta}(0) \cdot \mathbf{f}(x) - \boldsymbol{\epsilon}(0) \cdot \mathbf{g}(x)$. These polynomial vectors are then opened up by the prover, who sends them

Interactive Zero-Knowledge Proof $\mathcal{P}_{\mathsf{ACE}}$

$\mathcal{P}$

secret knowledge:
$\mathbf{i} \in \mathbb{F}_q^n$
s.t. $\mathbf{o} = \mathcal{H}(\mathbf{i})$

for all $k \in I$ do:
  $w_k(x) \xleftarrow{\$} \left(\mathbb{F}_q[x^{\leq d}]\right)^n$
s.t. $(w_k(0))_{k \in I} = \mathbf{i}$
for all gates $(+, h, i, j) \in \mathcal{H}$ do:
  $w_j \leftarrow w_i(x) + w_h(x)$

$\mathcal{V}$

public knowledge:
$\mathbf{o} \in \mathbb{F}_q^m$
$\mathcal{H} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$

$$\xrightarrow[\mathsf{com}(\{w_i(x)\}_{i \in O})]{\{\mathsf{com}(\{w_i(j)\}_{i \in I})\}_{j \in U},}$$

$J \xleftarrow{\$} \{V \subset U \,|\, \#V = d\}$

$$\xleftarrow{\quad J \quad}$$

$$\xrightarrow[\mathsf{rev}(\{w_i(x)\}_{i \in O})]{\{\mathsf{rev}(\{w_i(j)\}_{i \in I})\}_{j \in J}}$$

for all $i \in O$ do:
  $\mathsf{deg}(w_i(x)) \overset{?}{\leq} d$
  $w_i(0) \overset{?}{=} o_i$
  for all $j \in J$ do:
    $w_i(j) \overset{?}{=} \sum_{k \in I} a_k w_k(j) + b_k$

Fig. 1. Interactive proof of correct evaluation of an arithmetic circuit $\mathcal{H}$.

to the verifier. In the last step, the verifier checks that these polynomials are correctly formed, *i.e.*, that their MACs satisfy the claimed linear relations, that the wires related to $\mathbf{k}$ represent zero, *i.e.*, $\mathbf{k}(0) = \mathbf{0}$, and that all polynomials have degree at most d. The protocol is visually depicted in Fig. 2.

It is important to note that the challenge e_1, e_2 associated with a single triple-proof contains $2\log_2(q-1)$ bits, where q is the size of the finite field. The protocol's soundness depends on enough entropy coming from the verifier's challenge, and the finite field might not be large enough to contain this information. This problem is solved by adding more multiplication triples in cascade. The first triple is sacrificed to prove the correctness of the second, which is then sacrificed to prove the correctness of the third. After being validated by a cascade of several sacrifices, the non-sacrificial multiplication triple is used in the linearized circuit to compute a multiplication.

Interactive Zero-Knowledge Proof $\mathcal{P}_{\mathsf{CMT}}$

$\mathcal{P}$

secret knowledge:

$\mathbf{a}, \mathbf{b}, \mathbf{c} \in \left(\mathbb{F}_q[x^{\leq d}]\right)^k$

s.t. $\mathbf{a}(0) \cdot \mathbf{b}(0) = \mathbf{c}(0)$

$\mathbf{f}, \mathbf{g}, \mathbf{h} \in \left(\mathbb{F}_q[x^{\leq d}]\right)^k$

s.t. $\mathbf{f}(0) \cdot \mathbf{g}(0) = \mathbf{h}(0)$

$\mathcal{V}$

secret knowledge:

$\{\mathbf{a}(j), \mathbf{b}(j), \mathbf{c}(j)\} \, \forall j \in J$

$\{\mathbf{f}(j), \mathbf{g}(j), \mathbf{h}(j)\} \, \forall j \in J$

$J \subset U, \quad \#J = d$

$\mathbf{e_1}, \mathbf{e_2} \xleftarrow{\$} (\mathbb{F}_q \backslash \{0\})^k$

$$\xleftarrow{\quad \mathbf{e_1}, \mathbf{e_2} \quad}$$

$$\begin{aligned}
\boldsymbol{\delta} &\leftarrow \mathbf{e_1} \cdot \mathbf{b}(x) - \mathbf{g}(x) \\
\boldsymbol{\epsilon} &\leftarrow \mathbf{e_2} \cdot \mathbf{a}(x) - \mathbf{f}(x) \\
\mathbf{k} &\leftarrow \mathbf{e_1} \cdot \mathbf{e_2} \cdot \mathbf{c}(x) - \boldsymbol{\delta}(0) \cdot \boldsymbol{\epsilon}(0) \\
&\quad - \boldsymbol{\delta}(0) \cdot \mathbf{f}(x) - \boldsymbol{\epsilon}(0) \cdot \mathbf{g}(x) \\
&\quad - \mathbf{h}(x)
\end{aligned}$$

$$\xrightarrow{\quad \boldsymbol{\delta}(x), \boldsymbol{\epsilon}(x), \mathbf{k}(x) \quad}$$

for all $j \in J$ do:

$$\begin{aligned}
\boldsymbol{\delta}(j) &\overset{?}{=} \mathbf{e_1} \cdot \mathbf{b}(j) - \mathbf{g}(j) \\
\boldsymbol{\epsilon}(j) &\overset{?}{=} \mathbf{e_2} \cdot \mathbf{a}(j) - \mathbf{f}(j) \\
\mathbf{k}(j) &\overset{?}{=} \mathbf{e_1} \cdot \mathbf{e_2} \cdot \mathbf{c}(j) - \mathbf{h}(j) \\
&\qquad - \boldsymbol{\delta}(0) \cdot \boldsymbol{\epsilon}(0) \\
&\qquad - \boldsymbol{\delta}(0) \cdot \mathbf{f}(j) \\
&\qquad - \boldsymbol{\epsilon}(0) \cdot \mathbf{g}(j)
\end{aligned}$$

$\mathbf{k}(0) \overset{?}{=} \mathbf{0}$

$\mathsf{max}\{\mathsf{deg}(\boldsymbol{\delta}(x)), \mathsf{deg}(\boldsymbol{\epsilon}(x)),$

$\mathsf{deg}(\mathbf{k}(x))\} \overset{?}{\leq} d$

Fig. 2. Interactive proof of correct multiplication triples.

Sequential Composition. Of course, the verifier should not merely assume that the wires indicated by the prover as such are indeed correctly formed multiplication triples; he should verify this. To this end, $\mathcal{P}_{\mathsf{CMT}}$ must be combined with $\mathcal{P}_{\mathsf{ACE}}$. The changes to the shape of the circuit $\mathcal{H}$ are obvious: the input must contain an additional triple of vectors of k elements representing $\mathbf{f}, \mathbf{g}$ and $\mathbf{h}$, the multiplication triples that are sacrificed to prove the correctness of $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$. Similarly, the output of $\mathcal{H}$ must contain k additional wires matching $\boldsymbol{\delta}, \boldsymbol{\epsilon}$ and $\mathbf{k}$, *i.e.*, the response values proving the correct formation of the multiplication triples. Thankfully, the verification of these last three sets of wires consists of validating the MACs against linear relations and evaluating the polynomials $x = 0$ — essentially the same as the last step of $\mathcal{P}_{\mathsf{ACE}}$ and hence easily combined.

Naïve parallel composition allows the prover to choose $\boldsymbol{\epsilon}(x)$, $\boldsymbol{\delta}(x)$ and $\mathbf{k}(x)$ as a function of J, guaranteeing valid MAC values but getting away with a different wire value. Since $\mathcal{P}_{\mathsf{CMT}}$ assumes that the prover is ignorant of J at the time he computes $\boldsymbol{\epsilon}(x)$, $\boldsymbol{\delta}(x)$ and $\mathbf{k}(x)$, we must somehow demand that this computation takes place before J is chosen. We propose one way of combining $\mathcal{P}_{\mathsf{ACE}}$ with $\mathcal{P}_{\mathsf{CMT}}$ which guarantees the proper order of computation, but note that other strategies may exist also. A high-level schematic overview of the composite protocol is shown in Fig. 3.

In this sequential composition structure, five messages are exchanged rather than three. After committing to the polynomials of the inputs and all their MAC values, the verifier responds with the $\mathcal{F}_{\mathsf{CMT}}$-challenge $\mathbf{e_1}, \mathbf{e_2}$. Only after the prover responds with commitments to the polynomials $\boldsymbol{\epsilon}(x), \boldsymbol{\delta}(x)$ and $\mathbf{k}(x)$, does the verifier challenge him with the random audit J and only after that does the prover respond by revealing the requested MAC values and output wire polynomials. While five messages is not optimal in terms of interactivity, both challenges are public coin challenges and hence either one step or both can be made non-interactive via application of the Fiat-Shamir transform [13].

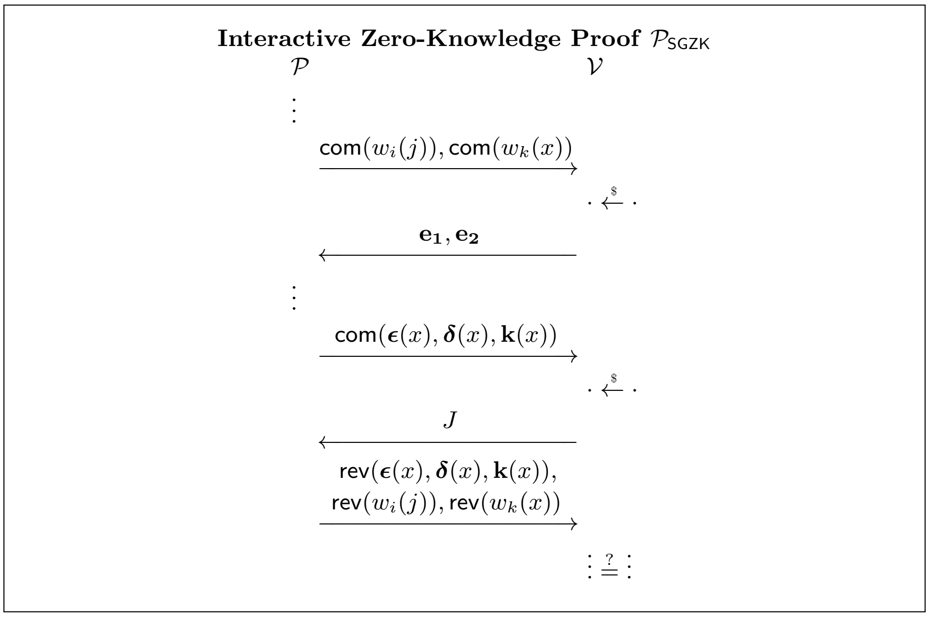
Interactive Zero-Knowledge Proof $\mathcal{P}_{\mathsf{SGZK}}$
$\mathcal{P}$
$\mathcal{V}$
$\mathsf{com}(w_i(j)), \mathsf{com}(w_k(x))$
$\cdot \overset{\$}{\leftarrow} \cdot$
$\mathbf{e_1}, \mathbf{e_2}$
$\mathsf{com}(\boldsymbol{\epsilon}(x), \boldsymbol{\delta}(x), \mathbf{k}(x))$
$\cdot \overset{\$}{\leftarrow} \cdot$
J
$\mathsf{rev}(\boldsymbol{\epsilon}(x), \boldsymbol{\delta}(x), \mathbf{k}(x)),$
$\mathsf{rev}(w_i(j)), \mathsf{rev}(w_k(x))$
$\vdots \overset{?}{=} \vdots$

Fig. 3. Sequential composition of $\mathcal{P}_{\mathsf{ACE}}$ with $\mathcal{P}_{\mathsf{CMT}}$ for generic zero-knowledge proofs. The ellipses leave out the identical computations by the prover and verifier found in protocols $\mathcal{P}_{\mathsf{ACE}}$ and $\mathcal{P}_{\mathsf{CMT}}$.

3.3 Interactive Security

The protocols $\mathcal{P}_{\mathsf{ACE}}$ and $\mathcal{P}_{\mathsf{CMT}}$ are presented as separate entities. It is tempting to prove their security separately as well. However, this strategy will fail: as the circuit for $\mathcal{P}_{\mathsf{ACE}}$ is linear, the notions knowledge and knowledge-soundness are meaningless. The extractor can compute the secret from the public information without ever interacting with the prover. Consequently, to prove security we must take into account the sequential composition with $\mathcal{P}_{\mathsf{CMT}}$ which allows for nonlinear and thus potentially one-way circuits.

Theorem 1. *Let* $(\mathsf{com}, \mathsf{rev})$ *be a correct, hiding and collapse-binding commitment scheme. Let* $\mathcal{H}$ *be a circuit for a one-way function defined over some finite field* $\mathbb{F}_q$*, and where every input is involved in at least one nonlinear gate. Then* $\mathcal{P}_{\mathsf{SGZK}}$ *is a complete, honest-verifier zero-knowledge quantum proof of knowledge of the relation*

$$\mathsf{ZKQPoK}\{(\mathbf{i}) : \mathcal{H}(\mathbf{i}) = \mathbf{o}\}$$

satisfying special 2*-soundness with knowledge error* $\sigma \leq \mathsf{max}\left(1/\binom{u}{d}, 1/(q-1)^D\right)$*, where* D *is the depth of the multiplication cascade and* $u = \#U$*.*

Proof. Completeness. Since $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ represent correct multiplication triples, the completeness follows from induction on the gates of $\mathcal{H}$. If the output of each gate is computed correctly from that gate's inputs, then the MACs of its outputs can successfully by verified all the way down to the circuit's output wires. The prover chooses polynomials of degree at most d and as the polynomials are only linearly combined this degree never increases, thus guaranteeing that the degree-check of the verifier succeeds as well. The verifier's test for multiplication triple validity is guaranteed to succeed as well because $\mathbf{f}, \mathbf{g}, \mathbf{h}$ also represent correct multiplication triples and

$$\begin{aligned}\mathbf{k} &= \mathbf{e_1 e_2 c} - \mathbf{h} - \boldsymbol{\delta\epsilon} - \boldsymbol{\delta}\mathbf{f} - \boldsymbol{\epsilon}\mathbf{g} \\ &= \mathbf{e_1 e_2 c} - \mathbf{h} - (\mathbf{e_1 b} - \mathbf{g})(\mathbf{e_2 a} - \mathbf{f}) - (\mathbf{e_1 b} - \mathbf{g})\mathbf{f} - (\mathbf{e_2 a} - \mathbf{f})\mathbf{g} \\ &= \mathbf{e_1 e_2 c} - \mathbf{h} - \mathbf{e_1 e_2 ab} + \mathbf{e_2 ag} + \mathbf{e_1 fb} - \mathbf{fg} - \mathbf{e_2 ag} + \mathbf{fg} - \mathbf{e_1 bf} + \mathbf{gf} \\ &= \mathbf{0} \ .\end{aligned}$$

Soundness. If $\mathcal{H}((w_i(0))_{i \in I}) \neq \mathbf{o}$ then at some point the circuit was computed incorrectly and then either a) not all linear relations between the polynomials hold; or b) the linear relations do hold but the multiplication triples are incorrect. We treat these cases separately.

In case (a), the linear mismatch will be exposed with high probability due to the verifier's random audit in d points indicated by the second challenge J. Assume that all involved polynomials are of degree at most d. (We justify this assumption later.) Then the linear mismatch means that for some i, $w_i(x) \neq \sum_{k \in I} a_k w_k(x) + b_k$. The difference between left and right hand sides must be nonzero and of degree at most d, and consequently it must evaluate to zero in at most d points. By the binding property of the commitment scheme, these polynomials are fixed before J is chosen. Consequently, the probability that these

d points where the difference polynomial evaluates to zero are exactly those in J is $1/\binom{u}{d}$ with $u = \#U$.

In case (b), the cheating prover must produce polynomials $\boldsymbol{\delta}(x), \boldsymbol{\epsilon}(x)$ and $\mathbf{k}(x)$ that pass the verifier's test. As their linear relations hold, these polynomials satisfy

$$\begin{gathered}\boldsymbol{\delta}(x) = \mathbf{e_1} \cdot \mathbf{b}(x) - \mathbf{g}(x) \ ; \quad \boldsymbol{\epsilon}(x) = \mathbf{e_2} \cdot \mathbf{a}(x) - \mathbf{f}(x) \ ; \\ \mathbf{k}(x) = \mathbf{e_1} \cdot \mathbf{e_2} \cdot \mathbf{c}(x) - \mathbf{h}(x) - \boldsymbol{\delta}(0) \cdot \mathbf{f}(x) - \boldsymbol{\epsilon}(0) \cdot \mathbf{g}(x) - \boldsymbol{\epsilon}(0)\boldsymbol{\delta}(0) \ .\end{gathered}$$

Consider one pair of triples. Evaluate in 0 and drop (0) from notation. Then:

$$\begin{aligned}0 = k &= e_1 e_2 c - h - \delta f - \epsilon g - \epsilon \delta \\ &= e_1 e_2 c - h - (e_1 b - g) f - (e_2 a - f) g - (e_1 b - g)(e_2 a - f)\end{aligned}$$

$$\begin{aligned}h + e_1 b f - f g - f g - e_1 b f + f g &= e_1 e_2 c - e_2 a g - e_1 e_2 a b + e_2 a g \\ &= h - f g = e_1 e_2 (c - ab) \ .\end{aligned}$$

If either (a, b, c) or (f, g, h) is an incorrect triple but not both, then no assignment to $e_1, e_2 \in \mathbb{F}_q \backslash \{0\}$ can be valid. However, if both are incorrect triples, then there is exactly one valid assignment. By the binding property of the commitment scheme, the polynomials are fixed before e_1 and e_2 are chosen. Consequently, the probability of a valid pair (e_1, e_2) being selected is $1/(q-1)$. For a multiplication triple cascade of depth D, the effect is compounded: the probability of a valid sequence of pairs (e_1, e_2) being selected is $1/(q-1)^D$.

If any one of the polynomials has degree larger than d, then this fraud will be exposed by the multiplication challenge with high probability as well, provided that the wire is involved in at least one multiplication. Consider one multiplication triple audit and suppose $a(x), b(x), c(x), f(x), g(x), h(x)$ or a combination of them have degree larger than d. We know that the linear MAC check guarantees with high probability that

$$\begin{gathered}\delta(x) = e_1 b(x) - g(x) \ ; \quad \epsilon(x) = e_2 a(x) - f(x) \ ; \\ k(x) = e_1 e_2 c(x) - h(x) - \delta(0) f(x) - \epsilon(0) g(x) - \epsilon(0)\delta(0) \ .\end{gathered}$$

If exactly one from $\{a(x), b(x), c(x), f(x), g(x), h(x)\}$ has degree larger than d, then one of the degree checks must fail. If $\mathsf{deg}(a(x)) = \mathsf{deg}(f(x)) > d$ and their quotients by x^d are linearly dependent, then there is exactly one choice of e_2 that guarantees $\mathsf{deg}(\epsilon(x)) \leq d$ and the probability that it will be selected is $1/(q-1)$. The same argument holds for $b(x), g(x), e_1$ and $\delta(x)$. If $\mathsf{deg}(c(x)) = \mathsf{deg}(h(x)) > d$ (with linearly dependent quotients by x^d) but $\mathsf{deg}(f(x)), \mathsf{deg}(g(x)) \leq d$, then the random choice of e_1 and e_2 will cause the degree check on $k(x)$ to fail with probability $1/(q-1)$. This covers all possibilities. The effect compounds for multiplication cascades of depth D because polynomials of degree larger than d need to come in pairs from different triples. The probability of a cheating prover getting away with a polynomial of degree larger than d is at most $1/(q-1)^D$.

Quantum-knowledge-soundness. The computation of the prover can be represented by three unitary transforms $\mathfrak{P}_0, \mathfrak{P}_1$ and $\mathfrak{P}_2$ sharing a state. The collapse-binding property of the commitments guarantee that after measuring the first and third messages, the state of the prover is not disturbed too much by also measuring the last message. Consequently, we can rewind the prover's last computation by applying $\mathfrak{P}_2^{-1}$ and re-run it with a different challenge $J' \neq J$. This generates a second transcript, different from the first in only the last two messages. Efficient extractability follows from special 2-soundness.
Special 2-soundness. There exists an efficient extractor who, given two transcripts T_1 and T_2 with the same first three messages but with a different fourth and fifth one, is able to compute the secret information $\mathbf{i}$. The first message of T_1 and T_2 is the same, indicating that the wires $\{w_i(x)\}_{i\in I}$ are the same. The fourth messages are different and consequently $J \cup J'$ contains more than d points. Consequently, the extractor obtains from the sets $\{\mathsf{rev}(\{w_i(j)\}_{i\in I})\}_{j\in J}$ more than d points on every polynomial $w_i(x), i \in I$. As their degrees are at most d, the extractor can interpolate and obtain $(w_i(0))_{i\in I} = \mathbf{i}$.
Honest-verifier zero-knowledge. There exists a simulator who, given only the public information, is capable of generating a transcript T that is valid and indistinguishable from the transcript between an honest prover with the secret $\mathbf{i}$ and an honest verifier. The simulator proceeds as follows: he chooses the input wires $\{w_i(x)\}_{i\in I}$ at random and computes the circuit honestly, *i.e.*, using honest multiplication triples with an honest audit $\mathbf{e_1}, \mathbf{e_2}$ and correct linear relations. Next, the simulator chooses at random $J \subset U$ and evaluates the output polynomials $\{w_i(x)\}_{i\in O}$ in $x \in J$. Together with the points $\mathbf{o}$, this makes $d+1$ points for each output polynomial. The simulator interpolates between $d+1$ points for each output to find a new polynomial $\{w_i'(x)\}_{i\in O}$ satisfying both $(w_i'(0))_{i\in O} = \mathbf{o}$ and the linear relations $w_i'(j) = \sum_{k\in I} a_k w_k(j) + b_k \; \forall i \in O, j \in J$. The full transcript is given by

$$T = \begin{pmatrix} \{\mathsf{com}(\{w_i(j)\}_{i\in I})\}_{j\in U}, \mathsf{com}(\{w_k(x)'\}_{k\in O}) \\ \mathbf{e_1}, \mathbf{e_2} \\ \mathsf{com}(\boldsymbol{\epsilon}(x), \boldsymbol{\delta}(x), \mathbf{k}(x)) \\ J \\ \mathsf{rev}(\boldsymbol{\epsilon}(x), \boldsymbol{\delta}(x), \mathbf{k}(x)), \{\mathsf{rev}(\{w_i(j)\}_{i\in I})\}_{j\in J}, \mathsf{rev}(\{w_k'(x)\}_{k\in O}) \end{pmatrix} .$$

This transcript is indistinguishable from authentic as the distinguisher would have to recover the values of the input $\mathbf{i}$ and test them against $\mathcal{H}(\mathbf{i}) \overset{?}{=} \mathbf{o}$ in order to tell authentic transcripts from counterfeit. However, any such value-recovering adversary would break the hiding property of the commitment scheme. □

3.4 Optimizations

Bit packing. There are many choices for the finite field underlying the circuit. On the one hand, the field must contain more elements than the degree d of the polynomials so that they are not reduced modulo the field ideal. On the other hand, a larger field means more data and larger messages, which is especially redundant if the multiplication triples are used for nonlinear bit gates. Fortunately, the input and output can be densely packed if an extension field is used, such as *e.g.* $\mathbb{F}_{2^8}$. After all, if $f(x)$ is the polynomial associated with a wire and $f(0)$ is that wire's value, then to obtain the ith bit of this value amounts to a linear operation: $(f(0))_i = L(f(0))$. The transformation L propagates to the MAC values because $L(f(x))$ evaluates to $L(f(j)) = \sum_k L(a_k j^k) = L(\sum_k a_k j^k)$ in the MAC keys $j \in J$.

Field inverse. While addition and multiplication over a finite field are together a universal set of operations, *i.e.*, capable of computing any computable function, for some applications it may be worthwhile to include field inversion in this set. This operation may be simulated by raising an element to the order of the field minus one, but this requires a substantial number of multiplication triples, especially if the field is large. However, it requires only one multiplication triple to prove that $\langle g \rangle$ is the multiplicative inverse of $\langle f \rangle$ as all the verifier needs to do is to verify that $\langle fg \rangle$ opens to 1.

One extra point. As a result of the protocol, the verifier already obtains d points of each output polynomial. Rather than committing to the entire output polynomials and revealing them later, the prover commits to the evaluations in $x = 1$ and reveals those points later. Consequently, $x = 1 \notin U$ should be off-limits as a potential MAC key.

Commitment tree. At the start of the protocol, the prover commits to $u = \#U$ sets of MAC values: $\{\{w_i(j)\}_{i \in I}\}_{j \in U}$. However, not all of these commitments are eventually opened. Depending on the size of U, it may be prudent to arrange the commitments to the sets $\{w_i(j)\}_{i \in I}$ as leafs in a Merkle tree, where each node in the next layer is the hash of two nodes in the previous layer such as in Fig. 4. The prover commits to the root of this tree and eventually opens only those branches that correspond to the verifier's challenge J. While this does not reduce the work done by the prover, it drastically reduces the communication cost from $O(u)$ to $O(d \lceil \mathsf{log}_2 u \rceil)$.

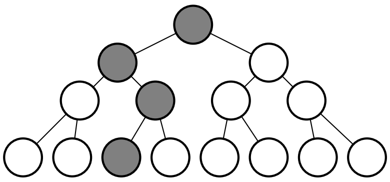

Fig. 4. Merkle commitment tree with one opened branch.

Specifically, the commitments $\mathsf{com}(\{w_i(2)\}_{i \in I}), \mathsf{com}(\{w_i(3)\}_{i \in I}), \ldots, \mathsf{com}(\{w_i(u+2)\}_{i \in I})$ are aligned horizontally on the bottommost layer (layer 0) with indices starting at 0, along with empty strings to pad this layer until the number of elements is a power of 2. Let $\nu_{i,j}$ denote the ith node on layer j, then except for layer 0 all elements are found as $\nu_{i,j} = \mathsf{RO}(\nu_{(i\|0),j-1}\|\nu_{(i\|1),j-1})$. The root of the tree is $\nu_{0,\lceil\log_2 u\rceil-1}$. The *path* to bottom node i is given by $\pi_i = \nu_{0,\lceil\log_2 u\rceil-1}\|\cdots\|\nu_{i,0}$, where the first index of the kth node in this expression represents the first $k-1$ most significant bits of i.

Fiat-Shamir transform. The Fiat-Shamir transform [13] turns interactive proofs into non-interactive proofs by replacing the verifier's public coins with the random oracle's response. The query that generated this response equals the concatenation of all previous protocol messages. Let

$$\begin{aligned}\mathsf{ic} &= \nu_{0,\lceil\log_2 u\rceil-1}\|\mathsf{com}\big(\{w_i(1)\}_{i\in O}\big)\\ \mathsf{oc} &= \mathsf{com}\big(\{\boldsymbol{\delta}(1),\boldsymbol{\epsilon}(1),\mathbf{k}(1)\}\big)\end{aligned}$$

represent the first and second messages sent by the prover in $\mathcal{P}_{\mathsf{SGZK}}$. Then for $\mathcal{P}_{\mathsf{SGZK}}$ the Fiat-Shamir transform amounts to setting

$$\begin{aligned}(\mathbf{e_1},\mathbf{e_2}) &\leftarrow \mathsf{RO}\big(\mathsf{ic}\big)\\ J &\leftarrow \mathsf{RO}\big(\mathsf{ic}\|\mathbf{e_1}\|\mathbf{e_2}\|\mathsf{oc}\big)\ .\end{aligned}$$

The resulting non-interactive protocol retains security against classical adversaries in the random oracle model via the forking lemma as formalized by Pointcheval and Stern [26]. Unfortunately, no quantum analogue of the forking lemma is known.

3.5 Complexity

Let κ be a security parameter which is also the number of bits in the random oracle's responses. Let k be the number of multiplication gates in the circuit and let n be the number of inputs and m the number of outputs, not including the additional inputs and outputs that arise due to linearization.

There are several design choices to be made. For instance, while the choice of the finite field $\mathbb{F}_q$ does not influence security, it does impose restrictions on other parameters that do. Unless the circuit is specifically designed otherwise, it makes sense to choose a field with characteristic 2 as this simplifies the protocol's implementation. The number of elements q must be large enough to accommodate all potential MAC keys U as well as the values 0 and 1 that are off-limits as potential keys: $q \geq \#U + 2 = u + 2$. The maximum degree d of the wire polynomials is another parameter choice subject to restrictions. As the verifier's work is linear in d^2, and as the prover's work is linear in du, both numbers are preferably as small as possible. However, they must be large enough to ensure that the query $J \subset U$ of the verifier contains at least κ bits of information: $\binom{u}{d} \geq 2^\kappa$. Also, the size of the transcript is logarithmic in u but linear in d and consequently one may wish to minimize d at the expense of u to some degree. Table 1 shows some sets of design choices that satisfy the required criteria.

Table 1. Choices of parameters: security level κ, size of finite field q, number of MAC keys u, polynomial degree d, and the resulting scale of the communication cost $\lceil \mathsf{log}_2 u \rceil d$.

κ	80	80	80	128	128	256	256
$\mathsf{log}_2 q$	8	10	12	10	12	10	12
u	226	765	3859	855	3425	996	3920
d	16	11	8	19	14	44	31
$\lceil \mathsf{log}_2 u \rceil d$	128	110	96	190	168	440	372

The process of linearization exchanges every multiplication gate for three extra inputs (a, b, c) and two extra outputs $(a - f, b - g)$. So after linearization the number of inputs is $n' = n + 3k$ and the number of outputs is $m' = m + 2k$. Of course, every multiplication triple should be accompanied by a cascade of depth $\lceil \frac{\kappa}{\mathsf{log}_2(q-1)} \rceil$ of sacrificial multiplication triples to guarantee at least κ bits of information in the verifier's challenge e. Each sacrificial triple introduces three new inputs (f, g, h) and three new outputs (δ, ϵ, k). Thus, the total number of inputs and outputs are respectively

$$n'' = n + 3k \left(\left\lceil \frac{\kappa}{\mathsf{log}_2(q-1)} \right\rceil + 1 \right) \qquad m'' = m + 2k + 3k \left(\left\lceil \frac{\kappa}{\mathsf{log}_2(q-1)} \right\rceil \right) \quad .$$

The MAC values for the n'' inputs are stored in u vectors $(w_1(j), w_2(j), \ldots)$ of n'' elements. Every such vector is the leaf of a binary Merkle tree. In order to generate the full tree from all the u leafs, $2u$ hashes or oracle queries are necessary. The prover's first message (ic for *input commitment*) is the root of this Merkle tree and contains κ bits.

However, in order to open the n'' MAC values for key $x = j$ of all input wires, only $\lceil \mathsf{log}_2 u \rceil$ commitments must be opened. Only the bottom-most layer of the tree requires κ bits of randomness for semantically secure commitments. There are $\lceil \mathsf{log}_2 u \rceil$ mergers, each requiring the κ bits from the other branch to be verified. Lastly, the actual released MAC values constitute n'' finite field elements. Thus, the number of bits necessary to open one full branch of the tree is $\kappa + n'' \lceil \mathsf{log}_2 q \rceil + \kappa \lceil \mathsf{log}_2 u \rceil$. Not one but d branches must be opened.

There is no subset of the output wires that must remain hidden; all must be revealed. Consequently, a single commitment will do the trick and the prover's second message (oc for *output commitment*) consists of κ bits. This message is a commitment to the evaluations in 0 and 1 for each to-be-opened wire; opening it thus requires $\kappa + 2m'' \lceil \mathsf{log}_2 q \rceil$ bits.

Both challenges of the verifier consist of κ bits as any further information required can be deterministically generated from this seed. Alternatively, if the Fiat-Shamir transform is applied, both terms can be reduced to zero. The last message of the prover (r for *response*) opens all the necessary commitments. The size of this message is therefore $d(\kappa + n'' \lceil \mathsf{log}_2 q \rceil + \kappa \lceil \mathsf{log}_2 u \rceil) + \kappa + 2m'' \lceil \mathsf{log}_2 q \rceil$ bits. Asymptotically speaking, $u \gg d$ and consequently $2^\kappa \approx \binom{u}{d} \approx u^d$, making the asymptotic message size on the order of $O(\kappa^2)$.

The task of the prover is divisible into six parts: 1) evaluating the linearized circuit in n'' polynomials of $d+1$ coefficients each, resulting in m'' output polynomials; 2) evaluating n'' polynomials of degree at most d in u points; 3) arranging u leafs into a Merkle tree; 4) computing the effect of the verifier's multiplication challenge e; 5) committing to m'' output polynomials; and 6) revealing these polynomials along with d branches from the Merkle tree of MACs of depth $\lceil \mathsf{log}_2 u \rceil$. Asymptotically speaking, the computational cost of the prover is dominated by (1) and (2), *i.e.*, $O(d(m''n'' + u))$.

The task of the verifier consists of five parts: 1) generating entropy for the challenges; 2) verifying all the commitments associated with d branches in the tree of depth $\mathsf{log}_2 u$; 3) verifying the commitment associated with the output wires; 4) computing the d points of the m'' output polynomials from the n'' input polynomials; and 5) interpolating m'' polynomials of degree d in $d+1$ points in order to verify their values in $x = 1$. The computational load of the verifier is dominated by (4) and (5), *i.e.*, $O(d(m''n'' + d))$.

4 Zero-Knowledge for Multivariate Quadratic Systems

The protocol $\mathcal{P}_{\mathsf{SGZK}}$ as expounded so far applies to any arithmetic circuit. We now shift our attention to focus on systems of multivariate quadratic polynomials, a special case of arithmetic circuits that allows a specialized protocol and a significant performance enhancement.

The key insight which our tailored protocol exploits is the fact that the roughly n^3 field-multiplications in the base field $\mathbb{F}_q$ of the quadratic system $\mathcal{H} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ can be exchanged for a single non-linear operation in the extension field $\mathbb{F}_{q^n}$, along with several linear operations. As a consequence, we work over the large field $\mathbb{F}_{q^n}$, in contrast to the relatively small field of the last section.

Let x_1 represent one element of the input of $\mathcal{H}(\mathbf{x})$. Then $\mathbf{x} = (x_1, \ldots, x_n)^\mathsf{T}$ represents the vector of input elements. We employ bit packing and represent this vector in a single wire $\langle \mathbf{x} \rangle$ which can be seen as an element $\mathcal{X}$ of the extension field $\mathbb{F}_{q^n}$ and $\langle \mathcal{X} \rangle = \langle \mathbf{x} \rangle$. The Frobenius powers of $\mathcal{X}$ are $\mathcal{X}, \mathcal{X}^q, \mathcal{X}^{q^2}, \ldots, \mathcal{X}^{q^{n-1}}$ and are $\mathbb{F}_q$-linear functions of $\mathcal{X}$.

Let $\boldsymbol{\mathcal{X}} = (\mathcal{X}, \mathcal{X}^q, \ldots, \mathcal{X}^{q^{n-1}})^\mathsf{T}$ be the vector of Frobenius powers of $\mathcal{X}$ and let the use of boldface denote the application of the map $\mathcal{X} \mapsto \boldsymbol{\mathcal{X}}$. Note that the associated list of wires, $\langle \boldsymbol{\mathcal{X}} \rangle$ can be computed for free, *i.e.*, its computation requires only linear operations. Importantly, the homogeneous part of $\mathcal{H}$ can be regarded as a quadratic form over the vector space $\mathbb{F}_{q^n}^n$: $\hat{\mathcal{H}}(\mathcal{X}) = \boldsymbol{\mathcal{X}}^\mathsf{T} \mathfrak{H} \boldsymbol{\mathcal{X}}$, where $\mathfrak{H} \in \mathbb{F}_{q^n}^{n \times n}$ is the unique upper-triangular matrix associated with $\mathcal{H}$. Most importantly, $\mathfrak{H}$ is efficiently computable from $\mathcal{H}$, which is part of the public information.

To compute $\langle \boldsymbol{\mathcal{X}}^\mathsf{T} \mathfrak{H} \boldsymbol{\mathcal{X}} \rangle$ from $\langle \boldsymbol{\mathcal{X}} \rangle$ we require essentially one multiplication triple, albeit over the extension field $\mathbb{F}_{q^n}$ rather than over the base field $\mathbb{F}_q$. This multiplication triple is tailored to $\mathcal{H}$ and cannot be used for generic multiplications and hence a more accurate name is a *bilinear triple*. In particular, let

$\langle \mathcal{A} \rangle, \langle \mathcal{B} \rangle, \langle \mathcal{C} \rangle$ represent the wire vectors satisfying $\mathcal{C} = \boldsymbol{\mathcal{A}}^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{B}}$, then the computation of $\langle \boldsymbol{\mathcal{X}}^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{X}} \rangle$ from $\langle \boldsymbol{\mathcal{X}} \rangle$ proceeds straightforwardly from first opening $\langle \mathcal{X} - \mathcal{A} \rangle$ and $\langle \mathcal{X} - \mathcal{B} \rangle$ and then computing $\langle \boldsymbol{\mathcal{X}}^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{X}} \rangle = (\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{A}})^{\mathsf{T}} \mathfrak{H} (\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{B}}) + (\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{A}})^{\mathsf{T}} \mathfrak{H} \langle \boldsymbol{\mathcal{B}} \rangle + \langle \boldsymbol{\mathcal{A}} \rangle^{\mathsf{T}} \mathfrak{H} (\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{B}}) - \langle \mathcal{C} \rangle$.

4.1 Proof of Correct Bilinear Triples

Unfortunately, $\mathcal{P}_{\mathsf{CMT}}$, the earlier protocol for proving correctness of multiplication triples, cannot be translated to the case of bilinear triples. The reason is that the verifier's challenge $\mathcal{E}$ does not propagate properly in the bilinear form $(\boldsymbol{\mathcal{E}} \cdot \boldsymbol{\mathcal{F}})^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{G}} \neq \mathcal{E}\mathcal{H}$. However, the sequel describes another protocol, $\mathcal{P}_{\mathsf{CBT}}$, which addresses this problem at the cost of three more bilinear triples. Like $\mathcal{P}_{\mathsf{CMT}}$, $\mathcal{P}_{\mathsf{CBT}}$ is not designed for standalone use but for sequential composition with $\mathcal{P}_{\mathsf{ACE}}$, either instead of or alongside instances of $\mathcal{P}_{\mathsf{CMT}}$.

The protocol $\mathcal{P}_{\mathsf{CBT}}$ starts by allowing the verifier to randomize two of the prover's wires: $\langle \mathcal{M} \rangle = \langle \mathcal{I} \rangle + \mathcal{E} \langle \mathcal{J} \rangle$ and $\langle \mathcal{N} \rangle = \langle \mathcal{K} \rangle + \mathcal{D} \langle \mathcal{L} \rangle$, where $\mathcal{E}$ and $\mathcal{D}$ are chosen by the verifier. The prover uses four bilinear triples to compute each of the cross terms. Additionally, the verifier chooses two constants, $\mathcal{F}$ and $\mathcal{G}$ and the prover uses the fifth bilinear triple to compute $\langle (\boldsymbol{\mathcal{M}} + \boldsymbol{\mathcal{F}})^{\mathsf{T}} \mathfrak{H} (\boldsymbol{\mathcal{N}} + \boldsymbol{\mathcal{G}}) \rangle$. Next, the prover subtracts all cross-terms from the product of $\langle \mathcal{M} \rangle$ and $\langle \mathcal{N} \rangle$ and opens this wire to the prover as $\mathcal{Q}(\mathcal{X})$, in addition to all the opened wires required to compute products. Lastly, the prover opens the sum $\mathcal{O} = \mathcal{H}_0 + \mathcal{H}_1$ of two correction terms $\mathcal{H}_0 = (\boldsymbol{\mathcal{M}} + \boldsymbol{\mathcal{F}})^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{G}}$ and $\mathcal{H}_1 = \boldsymbol{\mathcal{F}}^{\mathsf{T}} \mathfrak{H} (\boldsymbol{\mathcal{N}} + \boldsymbol{\mathcal{G}})$ to the verifier, who can verify its MACs. At this point the verifier has enough information to verify that the product of $\mathcal{M} + \mathcal{F}$ with $\mathcal{N} + \mathcal{G}$ was computed correctly. Moreover, if any one of the bilinear triples was fraudulent, the verifier's challenge values will expose this fraud with overwhelming probability. As the verifier remains ignorant of $\langle \mathcal{M} + \mathcal{F} \rangle$ and $\langle \mathcal{N} + \mathcal{G} \rangle$, the triple that was used to compute their product is likewise kept secret, and can consequently be used elsewhere in a composed protocol for circuit evaluation. A schematic overview of protocol $\mathcal{P}_{\mathsf{CBT}}$ is given in Fig. 5, where we use $\otimes_i$ to denote consumption of the ith bilinear triple to compute the bilinear product, and δ_i and ϵ_i to denote the wires that were opened in the process.

4.2 Composed Proof for MQ Systems

In order to prove security in a meaningful sense we must once again consider the composition with $\mathcal{P}_{\mathsf{ACE}}$. Let $\mathcal{P}_{\mathsf{MQ}}$ be the protocol obtained by composing $\mathcal{P}_{\mathsf{ACE}}$ with one instance of $\mathcal{P}_{\mathsf{CBT}}$ sequentially, and having the structure depicted in Fig. 6. A given homogeneous multivariate quadratic function $\hat{\mathcal{H}} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ of the input $\mathbf{i} \in \mathbb{F}_q^n$ is computed, resulting in output $\mathbf{o} \in \mathbb{F}_q^n$. This is accomplished by working in the extension field $\mathbb{F}_{q^n}$, *i.e.*, by packing the input elements $\mathbf{i}$ into a single input $\mathcal{Y} = \varphi(\mathbf{i}) \in \mathbb{F}_{q^n}$; using one bilinear triple $(\mathcal{A}_0(\mathcal{X}), \mathcal{B}_0(\mathcal{X}), \mathcal{C}_0(\mathcal{X}))$ to compute $\mathcal{Z} = \boldsymbol{\mathcal{Y}}^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{Y}}$; and then unpacking $\mathbf{o} = \varphi^{-1}(\mathcal{Z})$. The packing and unpacking operations are for free. The next theorem shows that this composed

Interactive Zero-Knowledge Proof $\mathcal{P}_{\mathsf{CBT}}$

$\mathcal{P}$

secret knowledge:
$\mathcal{A}_i, \mathcal{B}_i, \mathcal{C}_i \in \mathbb{F}_{q^n}[\mathcal{X}^{\leq d}]$
s.t. $\mathcal{C}_i(0) = \boldsymbol{\mathcal{A}}_i(0)^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{B}}_i(0)$
$\mathcal{I}, \mathcal{J}, \mathcal{K}, \mathcal{L} \in \mathbb{F}_{q^n}[\mathcal{X}^{\leq d}]$

$\mathcal{V}$

secret knowledge:
$\{\mathcal{A}_i(j), \mathcal{B}_i(j), \mathcal{C}_i(j)\} \, \forall j \in J$
$\{\mathcal{I}(j), \mathcal{J}(j), \mathcal{K}(j), \mathcal{L}(j)\} \, \forall j \in J$
$J \subset U, \quad \#J = d$

$\mathcal{D}, \mathcal{E} \xleftarrow{\$} \mathbb{F}_{q^n}$
$\mathcal{F}, \mathcal{G} \xleftarrow{\$} \mathbb{F}_{q^n} \backslash \{0\}$

$$\xleftarrow{\quad \mathcal{D}, \mathcal{E}, \mathcal{F}, \mathcal{G} \quad}$$

$\mathcal{N} \leftarrow \mathcal{I}(\mathcal{X}) + \mathcal{E}\mathcal{J}(\mathcal{X})$
$\mathcal{M} \leftarrow \mathcal{K}(\mathcal{X}) + \mathcal{D}\mathcal{L}(\mathcal{X})$
$\mathcal{Q}_0 \leftarrow (\mathcal{N}(\mathcal{X}) + \mathcal{F}) \otimes_0 (\mathcal{M}(\mathcal{X}) + \mathcal{G})$
$\mathcal{Q}_1 \leftarrow \mathcal{I}(\mathcal{X}) \otimes_1 \mathcal{D}\mathcal{L}(\mathcal{X})$
$\mathcal{Q}_2 \leftarrow \mathcal{I}(\mathcal{X}) \otimes_2 \mathcal{K}(\mathcal{X})$
$\mathcal{Q}_3 \leftarrow \mathcal{E}\mathcal{J}(\mathcal{X}) \otimes_3 \mathcal{D}\mathcal{L}(\mathcal{X})$
$\mathcal{Q}_4 \leftarrow \mathcal{E}\mathcal{J}(\mathcal{X}) \otimes_4 \mathcal{K}(\mathcal{X})$
$\mathcal{Q} \leftarrow \mathcal{Q}_0(\mathcal{X}) - \sum_{i=1}^{4} \mathcal{Q}_i(\mathcal{X})$
$\mathcal{H}_0 \leftarrow \boldsymbol{\mathcal{F}}^{\mathsf{T}} \mathfrak{H} (\boldsymbol{\mathcal{M}}(\mathcal{X}) + \boldsymbol{\mathcal{G}})$
$\mathcal{H}_1 \leftarrow (\boldsymbol{\mathcal{N}}(\mathcal{X}) + \boldsymbol{\mathcal{F}})^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{G}}$
$\mathcal{O} \leftarrow \mathcal{H}_0(\mathcal{X}) + \mathcal{H}_1(\mathcal{X})$

$$\xrightarrow{\quad \delta_i, \epsilon_i, \mathcal{Q}, \mathcal{O} \quad}$$

for all $j \in J$ do:
  verify $\delta_i(j)$, $\epsilon_i(j)$, $\mathcal{Q}(j), \mathcal{O}(j)$
$\mathcal{Q}(0) - \mathcal{O}(0) \overset{?}{=} 0$
$\mathsf{max}\{\mathsf{deg}(\delta_i(\mathcal{X})), \mathsf{deg}(\epsilon_i(\mathcal{X})),$
  $\mathsf{deg}(\mathcal{Q}(\mathcal{X})), \mathsf{deg}(\mathcal{O}(\mathcal{X}))\} \overset{?}{\leq} d$

Unless indicated otherwise, all expressions with subscript i hold for $i \in \{0, 1, 2, 3, 4\}$. The symbols $\otimes_i$ indicate that the triple $(\mathcal{A}_i, \mathcal{B}_i, \mathcal{C}_i)$ is consumed to compute the bilinear product of the left- and right-hand-sides. As a result, the wires δ_i and ϵ_i are created.

Fig. 5. Interactive proof of correct bilinear triples.

protocol is secure, although we note that its proof generalizes to any number of MQ gates and any number of multiplication gates.

Theorem 2. *Let* $(\mathsf{com}, \mathsf{rev})$ *be a correct, hiding and collapse-binding commitment scheme. Let* $\hat{\mathcal{H}} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$ *be a list of homogeneous multivariate quadratic polynomials, and let* $\mathfrak{H} \in \mathbb{F}_{q^n}^{n \times n}$ *be a matrix of extension field elements satisfying* $\varphi \circ \hat{\mathcal{H}} \circ \varphi^{-1}(\mathcal{X}) = \boldsymbol{\mathcal{X}}^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{X}}$ *where* $\boldsymbol{\mathcal{X}} = (\mathcal{X}, \mathcal{X}^q, \ldots, \mathcal{X}^{q^{n-1}})^{\mathsf{T}}$*. Then* $\mathcal{P}_{\mathsf{MQ}}$ *is a complete, honest-verifier zero-knowledge quantum proof of knowledge of the relation*

$$\mathsf{ZKQPoK}\{(\mathbf{i}) : \mathcal{H}(\varphi(\mathbf{i})) = \varphi(\mathbf{o})\}$$

Interactive Zero-Knowledge Proof $\mathcal{P}_{\mathsf{MQ}}$

$\mathcal{P}$

secret knowledge:
$\mathbf{i} \in \mathbb{F}_q^n$
s.t. $\mathbf{o} = \mathcal{H}(\mathbf{i})$

$\mathcal{V}$

public knowledge:
$\mathbf{o} \in \mathbb{F}_q^m$
$\mathcal{H} : \mathbb{F}_q^n \to \mathbb{F}_q^n$

```
$\mathcal{Y} \xleftarrow{\$} \mathbb{F}_{q^n}[\mathcal{X}^{\leq d}]$
s.t. $\mathcal{Y}(0) = \varphi(\mathbf{i})$
for $i \in \{0, \ldots, 4\}$ do:
  $\mathcal{A}_i, \mathcal{B}_i, \mathcal{C}_i \xleftarrow{\$} \mathbb{F}_{q^n}[\mathcal{X}^{\leq d}]$
  s.t. $\mathcal{C}_i(0) = \boldsymbol{\mathcal{A}_i}(0)^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{B}_i}(0)$
$\mathcal{I}, \mathcal{J}, \mathcal{K}, \mathcal{L} \xleftarrow{\$} \mathbb{F}_{q^n}[\mathcal{X}^{\leq d}]$
$\mathcal{V} \leftarrow \mathcal{Y}(\mathcal{X}) - \mathcal{A}_0(\mathcal{X})$
$\mathcal{W} \leftarrow \mathcal{Y}(\mathcal{X}) - \mathcal{B}_0(\mathcal{X})$
$\mathcal{Z} \leftarrow \boldsymbol{\mathcal{V}}(0)^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{W}}(0) - \mathcal{C}_0(\mathcal{X})$
      $- \boldsymbol{\mathcal{V}}(0)^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{B}_0}(\mathcal{X})$
      $- \boldsymbol{\mathcal{A}_0}(\mathcal{X})^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{W}}(0)$
```

$$\xrightarrow{\{\mathsf{com}(\{w_\ell(j)\}_{\ell \in I})\}_{j \in U},\ \mathsf{com}(\{w_k(\mathcal{X})\}_{k \in O})}$$

$\cdot \xleftarrow{\$} \cdot$

$$\xleftarrow{\mathcal{D}, \mathcal{E}, \mathcal{F}, \mathcal{G}}$$

$\vdots$

$$\xrightarrow{\mathsf{com}(\epsilon_i(\mathcal{X}), \delta_i(\mathcal{X}), \mathcal{Q}(\mathcal{X}), \mathcal{O}(\mathcal{X}))}$$

$J \xleftarrow{\$} \{V \subset U | \#V = d\}$

$$\xleftarrow{J}$$

$$\xrightarrow{\mathsf{rev}(\epsilon_i(\mathcal{X}), \delta_i(\mathcal{X}), \mathcal{Q}(\mathcal{X}), \mathcal{O}(\mathcal{X})),\ \{\mathsf{rev}(\{w_\ell(j)\}_{\ell \in I})\}_{j \in J},\ \mathsf{rev}(\{w_k(\mathcal{X})\}_{k \in O})}$$

```
$\vdots \overset{?}{=} \vdots$
$\mathsf{max}\{\mathsf{deg}(\mathcal{V}(\mathcal{X})), \mathsf{deg}(\mathcal{W}(\mathcal{X})),$
    $\mathsf{deg}(\mathcal{Z}(\mathcal{X}))\} \overset{?}{\leq} d$
for $j \in J$ do:
  $\mathcal{V}(j) \overset{?}{=} \mathcal{Y}(j) - \mathcal{A}_0(j)$
  $\mathcal{W}(j) \overset{?}{=} \mathcal{Y}(j) - \mathcal{B}_0(j)$
  $\mathcal{Z}(j) \overset{?}{=} \boldsymbol{\mathcal{V}}(0)^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{W}}(0) - \mathcal{C}_0(j)$
        $- \boldsymbol{\mathcal{V}}(0)^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{B}_0}(j)$
        $- \boldsymbol{\mathcal{A}_0}(j)^{\mathsf{T}} \mathfrak{H} \boldsymbol{\mathcal{W}}(0)$
$\mathcal{Z}(0) \overset{?}{=} \varphi(\mathbf{o})$
```

Here $\{w_\ell\}_{\ell \in I}$ is shorthand for $\{\mathcal{Y}, \mathcal{A}_i, \mathcal{B}_i, \mathcal{C}_i, \mathcal{I}, \mathcal{J}, \mathcal{K}, \mathcal{L}\}$ and $\{w_k\}_{k \in O}$ for $\{\mathcal{Z}, \mathcal{V}, \mathcal{W}\}$.

Fig. 6. Sequential composition of $\mathcal{P}_{\mathsf{ACE}}$ with $\mathcal{P}_{\mathsf{CBT}}$ for a zero-knowledge proofs. The ellipses leave out the identical computations by the prover and verifier in $\mathcal{P}_{\mathsf{CBT}}$.

with soundness error at most $\mathsf{max}\left(1/\binom{u}{d}, 1/(q^n-1)\right)$ *and satisfying the special* 2*-soundness property.*

Proof. Completeness. Follows from construction. If $(\mathcal{A}_0, \mathcal{B}_0, \mathcal{C}_0)$ is a valid triple then, by evaluating in 0 and dropping (0), we have

$$\begin{aligned}\mathcal{Z} &= \boldsymbol{\mathcal{V}}^\mathsf{T}\mathfrak{H}\boldsymbol{\mathcal{W}} - \mathcal{C}_0 - \boldsymbol{\mathcal{V}}^\mathsf{T}\mathfrak{H}\boldsymbol{\mathcal{B}_0} - \boldsymbol{\mathcal{A}_0}^\mathsf{T}\mathfrak{H}\boldsymbol{\mathcal{W}}\\ &= (\boldsymbol{\mathcal{Y}}-\boldsymbol{\mathcal{A}_0})^\mathsf{T}\mathfrak{H}(\boldsymbol{\mathcal{Y}}-\boldsymbol{\mathcal{B}_0}) - \mathcal{C}_0 - (\boldsymbol{\mathcal{Y}}-\boldsymbol{\mathcal{A}_0})^\mathsf{T}\mathfrak{H}\boldsymbol{\mathcal{B}_0} - \boldsymbol{\mathcal{A}_0}^\mathsf{T}\mathfrak{H}(\boldsymbol{\mathcal{Y}}-\boldsymbol{\mathcal{B}_0})\\ &= \boldsymbol{\mathcal{Y}}^\mathsf{T}\mathfrak{H}\boldsymbol{\mathcal{Y}}\\ &= \varphi\circ\hat{\mathcal{H}}\circ\varphi^{-1}(\mathcal{X}) = \varphi(\hat{\mathcal{H}}(\mathbf{i}))\\ &= \varphi(\mathbf{o})\ .\end{aligned}$$

Moreover, since the polynomials are only ever weighted and added they must be of degree at most d.

As for the bilinear triples, if the honest prover possesses $\mathcal{A}_i, \mathcal{B}_i, \mathcal{C}_i, \mathcal{I}, \mathcal{J}, \mathcal{K}, \mathcal{L}$ satisfying the required criteria, then the linear relations of the MAC values in the verifier's secret points J are guaranteed to hold. Moreover, the degrees of $\delta_i, \epsilon_i, \mathcal{Q}, \mathcal{O}$ are at most d. Lastly, independently of the verifier's challenge, these polynomials satisfy

$$\begin{aligned}&\mathcal{Q}(0) - \mathcal{O}(0) + \boldsymbol{\mathcal{F}}^\mathsf{T}\mathfrak{H}\boldsymbol{\mathcal{G}}\\ &= \mathcal{Q}_0(0) - \mathcal{Q}_1(0) - \mathcal{Q}_2(0) - \mathcal{Q}_3(0) - \mathcal{Q}_4(0) - \mathcal{H}_0(0) - \mathcal{H}_1(0) + \boldsymbol{\mathcal{F}}^\mathsf{T}\mathfrak{H}\boldsymbol{\mathcal{G}}\\ &= (\boldsymbol{\mathcal{I}}(0) + \boldsymbol{\mathcal{E}}\cdot\boldsymbol{\mathcal{J}}(0) + \boldsymbol{\mathcal{F}})^\mathsf{T}\,\mathfrak{H}\,(\boldsymbol{\mathcal{K}}(0) + \boldsymbol{\mathcal{D}}\cdot\boldsymbol{\mathcal{L}}(0) + \boldsymbol{\mathcal{G}}) - \boldsymbol{\mathcal{I}}(0)^\mathsf{T}\mathfrak{H}(\boldsymbol{\mathcal{D}}\cdot\boldsymbol{\mathcal{L}}(0))\\ &\quad - \boldsymbol{\mathcal{I}}(0)^\mathsf{T}\mathfrak{H}\boldsymbol{\mathcal{K}}(0) - (\boldsymbol{\mathcal{E}}\cdot\boldsymbol{\mathcal{J}}(0))^\mathsf{T}\mathfrak{H}(\boldsymbol{\mathcal{D}}\cdot\boldsymbol{\mathcal{L}}(0)) - (\boldsymbol{\mathcal{E}}\cdot\boldsymbol{\mathcal{J}}(0))^\mathsf{T}\mathfrak{H}\boldsymbol{\mathcal{K}}(0)\\ &\quad - \boldsymbol{\mathcal{F}}^\mathsf{T}\mathfrak{H}(\boldsymbol{\mathcal{M}}(0) + \boldsymbol{\mathcal{G}}) - (\boldsymbol{\mathcal{N}}(0) + \boldsymbol{\mathcal{F}})^\mathsf{T}\mathfrak{H}\boldsymbol{\mathcal{G}} + \boldsymbol{\mathcal{F}}^\mathsf{T}\mathfrak{H}\boldsymbol{\mathcal{G}}\\ &= 0\ .\end{aligned}$$

Soundness. The case where the polynomials are of degree at most d but their linear relations do not hold has been covered in the proof of Thm. 1. In this case the success probability of a cheating prover is at most $1/\binom{u}{d}$, coinciding with the probability of choosing the right set of MAC keys at random. This leaves two strategies for cheating: a) invalid bilinear triples, and b) using polynomials of degree higher than d.

In the case (a), the prover must produce polynomials $\delta_i, \epsilon_i, \mathcal{Q}, \mathcal{O}$ that pass the verifier's tests. Dropping the (0) from notation, the verifier's last equality test will only succeed if

$$\mathcal{Q} - \mathcal{O} = 0\ .$$

If $\mathcal{O} \neq \boldsymbol{\mathcal{F}}^\mathsf{T}\mathfrak{H}(\boldsymbol{\mathcal{M}}(\mathcal{X}) + \boldsymbol{\mathcal{G}}) + (\boldsymbol{\mathcal{N}}(\mathcal{X}) + \boldsymbol{\mathcal{F}})^\mathsf{T}\mathfrak{H}\boldsymbol{\mathcal{G}}$, then the MAC test will fail. So expand $\mathcal{Q} = \mathcal{O}$ into

$$\mathcal{Q}_0 - \mathcal{Q}_1 - \mathcal{Q}_2 - \mathcal{Q}_3 - \mathcal{Q}_4 = \boldsymbol{\mathcal{F}}^\mathsf{T}\mathfrak{H}(\boldsymbol{\mathcal{M}} + \boldsymbol{\mathcal{G}}) + (\boldsymbol{\mathcal{N}} + \boldsymbol{\mathcal{F}})^\mathsf{T}\mathfrak{H}\boldsymbol{\mathcal{G}}\ . \tag{1}$$

If Eqn. 1 is satisfied despite the prover cheating, this can only be the effect of a very favorable (for the prover) choice of $\mathcal{D}, \mathcal{E}, \mathcal{F}, \mathcal{G}$. By expanding the equation, we get

$$\begin{aligned}&\boldsymbol{\mathcal{F}}^{\mathsf{T}}\mathfrak{H}(\boldsymbol{\mathcal{K}}+\boldsymbol{\mathcal{D}}\cdot\boldsymbol{\mathcal{L}}+\boldsymbol{\mathcal{G}})+(\boldsymbol{\mathcal{I}}+\boldsymbol{\mathcal{E}}\cdot\boldsymbol{\mathcal{J}}+\boldsymbol{\mathcal{F}})^{\mathsf{T}}\mathfrak{H}\boldsymbol{\mathcal{G}}\\ &=(\boldsymbol{\mathcal{I}}+\boldsymbol{\mathcal{E}}\cdot\boldsymbol{\mathcal{J}}+\boldsymbol{\mathcal{F}})^{\mathsf{T}}\mathfrak{H}_0(\boldsymbol{\mathcal{K}}+\boldsymbol{\mathcal{D}}\cdot\boldsymbol{\mathcal{L}}+\boldsymbol{\mathcal{G}})-\boldsymbol{\mathcal{I}}^{\mathsf{T}}\mathfrak{H}_1\boldsymbol{\mathcal{K}}\\ &\quad-\boldsymbol{\mathcal{I}}^{\mathsf{T}}\mathfrak{H}_2(\boldsymbol{\mathcal{D}}\cdot\boldsymbol{\mathcal{L}})-(\boldsymbol{\mathcal{E}}\cdot\boldsymbol{\mathcal{J}})^{\mathsf{T}}\mathfrak{H}_3\boldsymbol{\mathcal{K}}-(\boldsymbol{\mathcal{E}}\cdot\boldsymbol{\mathcal{J}})^{\mathsf{T}}\mathfrak{H}_4(\boldsymbol{\mathcal{D}}\cdot\boldsymbol{\mathcal{L}})\ .\end{aligned}\tag{2}$$

In this expression, $\mathfrak{H}_i$ is the potentially fraudulent bilinear relation that satisfies $\boldsymbol{\mathcal{A}_i}(0)^{\mathsf{T}}\mathfrak{H}_i\boldsymbol{\mathcal{B}_i}(0)=\mathcal{C}_i(0)$ and which was used for computing $\mathcal{Q}_i(\mathcal{X})$.

If every $\mathfrak{H}_i=\mathfrak{H}$, then the equation is guaranteed to hold for all $\mathcal{E},\mathcal{D},\mathcal{F},\mathcal{G}$ — but in this case the prover is not cheating. However, if one or more of the $\mathfrak{H}_i$ are different from $\mathfrak{H}$, then the discrepancy must be compensated for with a favorable assignment to $\mathcal{E},\mathcal{D},\mathcal{F},\mathcal{G}$. In particular, if $\mathfrak{H}\neq\mathfrak{H}_0=\mathfrak{H}_1=\mathfrak{H}_2=\mathfrak{H}_3=\mathfrak{H}_4$, then the $\mathfrak{H}$-terms are not canceled unless by a suitable assignment to $\mathcal{D},\mathcal{E},\mathcal{F},\mathcal{G}$. If $\mathfrak{H}_1\neq\mathfrak{H}_0$ then the term $\boldsymbol{\mathcal{I}}^{\mathsf{T}}\mathfrak{H}_1\boldsymbol{\mathcal{K}}$ cannot be canceled unless by a suitable assignment to $\mathcal{D},\mathcal{E},\mathcal{F},\mathcal{G}$. Conversely, if $\mathfrak{H}_2\neq\mathfrak{H}_0$ then the term $\boldsymbol{\mathcal{I}}^{\mathsf{T}}\mathfrak{H}_2(\boldsymbol{\mathcal{D}}\cdot\boldsymbol{\mathcal{L}})$ cannot be canceled unless by a suitable assignment to $\mathcal{D},\mathcal{E},\mathcal{F},\mathcal{G}$. Similar arguments hold for $\mathfrak{H}_3$ and $\mathfrak{H}_4$. As one of these inequalities must hold (or else the prover is not cheating at all) the cheating prover's only hope for success is the suitable assignment to $\mathcal{D},\mathcal{E},\mathcal{F},\mathcal{G}$. By the binding property of the commitment scheme, the prover's polynomials are fixed before $\mathcal{D},\mathcal{E},\mathcal{F},\mathcal{G}$ are chosen. The probability of a suitable tuple being chosen is thus at most $1/(q^n-1)$.

In the case (b) where the prover wishes to get away with using polynomials of degree larger than d, his fraud is equally likely to be exposed by the random challenge $\mathcal{D},\mathcal{E},\mathcal{F},\mathcal{G}$. Since $\epsilon_i(\mathcal{X}),\delta_i(\mathcal{X}),\mathcal{Q}(\mathcal{X}),\mathcal{O}(\mathcal{X})$ have degree at most d, any terms in higher-than-d powers of $\mathcal{X}$ must cancel in Eqn. 2. However, any assignment of canceling terms in higher-than-d powers of $\mathcal{X}$ to the polynomials under the prover's control rely on a suitable assignment to $\mathcal{D},\mathcal{E},\mathcal{F},\mathcal{G}$ for success. The probability of this event is $1/(q^n-1)$.

Quantum-knowledge-soundness and special 2-soundness. Identical to the proof of Thm. 1: collapse-binding commitments allow rewinding and replaying the prover on a different last challenge $J'\neq J$, thus obtaining the values of the polynomials in more than d points. Interpolation yields the polynomials and evaluation in 0 yields the secret $\mathbf{i}$.

Honest-verifier zero-knowledge. Identical to the proof of Thm. 1: the simulator computes the circuit honestly but with random input polynomials, in order to obtain the MAC values. Interpolation yields the polynomials that should be comitted to. The resulting transcript is indistinguishable because to distinguish from authentic requires testing $\mathcal{H}(\mathbf{i})\overset{?}{=}\mathbf{o}$ and that requires breaking the hiding property of the commitment scheme. □

4.3 Signature Scheme

Protocol $\mathcal{P}_{\mathsf{MQ}}$ is in and of itself a valid zero-knowledge proof for multivariate quadratic systems. We now describe how to transform it into a provably secure signature scheme. From a high-level point of view, the public key consists of a random seed R from which a multivariate quadratic system $\mathcal{H}\in\mathbb{F}_2[\mathbf{x}]$ can be extracted using a cryptographically secure pseudo-random number generator

(PRG), along with one output image $\mathbf{o}$; the secret key consists of the matching input $\mathbf{i}$ such that $\mathbf{o} = \mathcal{H}(\mathbf{i})$. The zero-knowledge protocol is made non-interactive and dependent on the message via the Fiat-Shamir transform [13]; the result is a signature. We let $\mathsf{PoK}\{\cdots\}$ denote the generation of a zero-knowledge proof of knowledge using the protocol $\mathcal{P}_{\mathsf{MQ}}$ with the given challenges from the verifier, and where ic (input commitment) and oc (output commitment) represent the first two messages from the prover, respectively. We use a hash function H but for the purpose of proving security this is modeled as a random oracle. The three algorithms of the public key signature scheme are given in Fig. 7.

```
Gen(1^κ):
  $R \xleftarrow{\$} \{0,1\}^\kappa$
  $\mathcal{H} \leftarrow \mathsf{PRG}(R)$
  $\mathbf{i} \xleftarrow{\$} \{0,1\}^\kappa$
  $\mathbf{o} \leftarrow \mathcal{H}(\mathbf{i})$
   secret key: $\mathbf{i}$
   public key: $R, \mathbf{o}$
```

```
Sign(sk, $m \in \{0,1\}^*$):
  $T \leftarrow \mathsf{PoK}\{(\mathbf{i}) : \mathcal{H}(\mathbf{i}) = \mathbf{o}\}$
      with $e \leftarrow \mathsf{H}(\mathsf{ic}\|m)$
      and $J \leftarrow \mathsf{H}(\mathsf{ic}\|m\|\mathsf{oc})$
   signature: $T$
```

```
Verify(pk, $m$, $T$):
  $\mathcal{H} \leftarrow \mathsf{PRG}(R)$
  $\mathsf{verify}(T, \mathcal{H}, \mathbf{o})$
  $T.e \stackrel{?}{=} \mathsf{H}(\mathsf{ic}\|m)$
  $T.J \stackrel{?}{=} \mathsf{H}(\mathsf{ic}\|m\|\mathsf{oc})$
```

Fig. 7. Three algorithms of the signature scheme based on the generic zero-knowledge proof for multivariate quadratic systems.

The signature scheme uses the random oracle in three different ways: firstly, the quadratic system $\mathcal{H}$ is generated from the PRG. Secondly, the function H, which is used for the Fiat-Shamir transform, is modeled by the random oracle. Thirdly, the generation of the interactive part of the zero-knowledge proof requires many instances of the commitment function, which uses the random oracle as a building block. Consequently, in order for the signature scheme to be secure, we must assume that whatever function we instantiate the random oracle with is secure. The only additional assumption is the hardness of the MQ problem.

Theorem 3. *If the MQ Assumption is true, then the MQ signature scheme based on $\mathcal{P}_{\mathsf{MQ}}$ is a valid and secure signature scheme in the EUF-CMA and random oracle models.*

Proof. The validity of signatures generated with the knowledge of the secret key follows from the completeness of the underlying zero-knowledge protocol. Its knowledge-soundness, which is retained after application of the Fiat-Shamir transform, guarantees that without knowledge of the secret key the adversary cannot produce valid signatures. The zero-knowledge property of the protocol guarantees that the adversary does not learn more than a negligible amount of information on the secret key from the signatures, even after any polynomial number of querying messages and receiving their signature. Alternatively, the adversary can attack the public key and attempt to invert $\mathcal{H}$ in $\mathbf{o}$ but as this is essentially an instance of the MQ problem, this strategy is infeasible under the MQ Assumption. □

In order to target κ bits of classical security, we recommend generating random quadratic systems that map κ bits to κ bits over small-order base fields, *e.g.*, $\mathbb{F}_2^\kappa \rightarrow \mathbb{F}_2^\kappa$. This parameter choice is supported by the empirically observed exponential (in κ) complexity of an algebraic attack on the MQ problem, see for example type V and VI systems in Yasuda *et al.* treatment of the MQ challenge [35].

The number of (extension field) inputs to the linear circuit involved in the zero-knowledge proof is $n'' = 20$; the number of outputs is $m'' = 13$. For a degree-ℓ extension of the base field $\mathbb{F}_q$, the size of the signature is computed using the formula

$$|T| = |\mathsf{ic}| + |\mathsf{oc}| + |\mathsf{r}| = d(\kappa + n''\ell\lceil\log_2 q\rceil + \kappa\lceil\log_2 u\rceil) + 3\kappa + 2m''\ell\lceil\log_2 q\rceil \enspace ,$$

whose terms were derived in the previous section. Table 2 shows some signature sizes for various parameter choices. The purpose of the fourth column is to mimic the parameter choices of Chen *et al.* [9] and thus enable an apples-to-apples comparison.

Table 2. Size of signature T (and public key pk) for various parameter choices.

κ	80	128	256	256
u	765	855	996	996
d	11	19	44	44
q	2	2	2	31
ℓ	80	128	256	64
$\|T\|$ (bits)	29 600	79 104	356 608	414 592
$\|\mathrm{pk}\|$ (bits)	160	256	512	225

5 Conclusion

This paper presents a new construction for generic zero-knowledge proofs in the random oracle model. The only computational assumption necessary for a secure realization of our protocol is that of the existence of a one-way function to generate the hash function that instantiates the random oracle. Our protocol is remarkably efficient: for a fixed security parameter, the computational cost of the prover and of the verifier scales quadratically with the size of the circuit; the communication cost scales linearly with the circuit size but quadratically with the security level.

Our protocol is particularly well-suited to zero-knowledge proofs for multivariate quadratic systems. The communication cost in this case drops to a fraction of the size of the circuit, scaling linearly not with the number of multiplication gates but with the size of the input and output. As a result, we obtain reasonably compact provably secure signatures whose security is based on the MQ problem. While more work is required to fully understand the security of the Fiat-Shamir transform in the quantum random oracle model, we do not rely on any construction or computational problem whose hardness is known to fail under attack by quantum computers. We thus conjecture that the protocol and signature scheme we obtain from them resist attacks on quantum computers. A future quantum analogue of the forking lemma would lift the scheme's provable security to cover quantum adversaries.

While our construction is a generic construction, it surprisingly suitable for unexpected optimizations that enhance its efficiency. In particular, the linearity of the Frobenius transform allows bit packing and as a result cuts the size of the messages by a constant factor. Moreover, multiplication triples apply not just to multiplication gates but to *any* bilinear form, or even quadratic form over the extension field. Consequently, any quadratic function can be computed from just one bilinear triple. We expect this insight to be of independent interest.

A very natural direction of future research will aim to decrease the complexity of the generic zero-knowledge proof construction. The key cause of the still sizeable communication and proving cost is the commitment function, which by virtue of being based on the random oracle, forces the adoption of a split-and-choose strategy. If one is prepared to depart from the well-emphasized minimalistic assumptions and opt instead for a commitment function that is richer in structure, this cost may be reduced quite significantly. Alternatively, perhaps there is a more efficient and less verbose way of releasing d out of u polynomial points without first computing and then committing to each of those u points individually, and without sacrificing random oracles.

References

1. Alaoui, S.M.E.Y., Dagdelen, Ö., Véron, P., Galindo, D., Cayrel, P.: Extended security arguments for signature schemes. In: Mitrokotsa, A., Vaudenay, S. (eds.) Progress in Cryptology - AFRICACRYPT 2012 - 5th International Conference on Cryptology in Africa, Ifrance, Morocco, July 10-12, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7374, pp. 19–34. Springer (2012), http://dx.doi.org/10.1007/978-3-642-31410-0_2
2. Ambainis, A., Rosmanis, A., Unruh, D.: Quantum attacks on classical proof systems: The hardness of quantum rewinding. In: 55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014. pp. 474–483. IEEE Computer Society (2014), http://dx.doi.org/10.1109/FOCS.2014.57
3. Beaver, D.: Efficient multiparty protocols using circuit randomization. In: Feigenbaum, J. (ed.) Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings. Lecture Notes in Computer Science, vol. 576, pp. 420–432. Springer (1991), http://dx.doi.org/10.1007/3-540-46766-1_34
4. Bellare, M., Goldreich, O.: On defining proofs of knowledge. In: Brickell, E.F. (ed.) Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings.

Lecture Notes in Computer Science, vol. 740, pp. 390–420. Springer (1992), http://dx.doi.org/10.1007/3-540-48071-4_28

5. Bendlin, R., Damgård, I., Orlandi, C., Zakarias, S.: Semi-homomorphic encryption and multiparty computation. In: Paterson, K.G. (ed.) Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6632, pp. 169–188. Springer (2011), http://dx.doi.org/10.1007/978-3-642-20465-4_11
6. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings. Lecture Notes in Computer Science, vol. 7073, pp. 41–69. Springer (2011), http://dx.doi.org/10.1007/978-3-642-25385-0_3
7. Cayrel, P., Lindner, R., Rückert, M., Silva, R.: Improved zero-knowledge identification with lattices. In: Heng, S., Kurosawa, K. (eds.) Provable Security - 4th International Conference, ProvSec 2010, Malacca, Malaysia, October 13-15, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6402, pp. 1–17. Springer (2010), http://dx.doi.org/10.1007/978-3-642-16280-0_1
8. Cayrel, P., Véron, P., Alaoui, S.M.E.Y.: A zero-knowledge identification scheme based on the q-ary syndrome decoding problem. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) Selected Areas in Cryptography - 17th International Workshop, SAC 2010, Waterloo, Ontario, Canada, August 12-13, 2010, Revised Selected Papers. Lecture Notes in Computer Science, vol. 6544, pp. 171–186. Springer (2010), http://dx.doi.org/10.1007/978-3-642-19574-7_12
9. Chen, M., Hülsing, A., Rijneveld, J., Samardjiska, S., Schwabe, P.: From 5-pass MQ -based identification to MQ -based signatures. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10032, pp. 135–165 (2016), http://dx.doi.org/10.1007/978-3-662-53890-6_5
10. Dagdelen, Ö., Fischlin, M., Gagliardoni, T.: The fiat-shamir transformation in a quantum world. In: Sako, K., Sarkar, P. (eds.) Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II. Lecture Notes in Computer Science, vol. 8270, pp. 62–81. Springer (2013), http://dx.doi.org/10.1007/978-3-642-42045-0_4
11. Damgård, I., Pastro, V., Smart, N.P., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7417, pp. 643–662. Springer (2012), http://dx.doi.org/10.1007/978-3-642-32009-5_38
12. Ding, J., Gower, J.E., Schmidt, D.: Multivariate Public Key Cryptosystems, Advances in Information Security, vol. 25. Springer (2006), http://dx.doi.org/10.1007/978-0-387-36946-4
13. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings. Lecture Notes in Com-

puter Science, vol. 263, pp. 186–194. Springer (1986), http://dx.doi.org/10.1007/3-540-47721-7_12

14. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
15. Giacomelli, I., Madsen, J., Orlandi, C.: Zkboo: Faster zero-knowledge for boolean circuits. IACR Cryptology ePrint Archive 2016, 163 (2016), http://eprint.iacr.org/2016/163
16. Goldreich, O.: The Foundations of Cryptography - Volume 1, Basic Techniques. Cambridge University Press (2001)
17. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. J. ACM 38(3), 691–729 (1991), http://doi.acm.org/10.1145/116825.116852
18. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems (extended abstract). In: Sedgewick, R. (ed.) Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA. pp. 291–304. ACM (1985), http://doi.acm.org/10.1145/22145.22178
19. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. 17(2), 281–308 (1988), http://dx.doi.org/10.1137/0217017
20. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge proofs from secure multiparty computation. SIAM J. Comput. 39(3), 1121–1152 (2009), http://dx.doi.org/10.1137/080725398
21. Lyubashevsky, V.: Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In: Matsui, M. (ed.) Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5912, pp. 598–616. Springer (2009), http://dx.doi.org/10.1007/978-3-642-10366-7_35
22. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval and Johansson [25], pp. 738–755, http://dx.doi.org/10.1007/978-3-642-29011-4_43
23. Maurer, U.M. (ed.): Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding, Lecture Notes in Computer Science, vol. 1070. Springer (1996)
24. Melchor, C.A., Gaborit, P., Schrek, J.: A new zero-knowledge code based identification scheme with reduced communication. In: 2011 IEEE Information Theory Workshop, ITW 2011, Paraty, Brazil, October 16-20, 2011. pp. 648–652. IEEE (2011), http://dx.doi.org/10.1109/ITW.2011.6089577
25. Pointcheval, D., Johansson, T. (eds.): Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings, Lecture Notes in Computer Science, vol. 7237. Springer (2012), http://dx.doi.org/10.1007/978-3-642-29011-4
26. Pointcheval, D., Stern, J.: Security proofs for signature schemes. In: Maurer [23], pp. 387–398, http://dx.doi.org/10.1007/3-540-68339-9_33
27. Ranellucci, S., Tapp, A., Zakarias, R.W.: Efficient generic zero-knowledge proofs from commitments. Cryptology ePrint Archive, Report 2014/934 (2014), http://eprint.iacr.org/

28. Sakumoto, K.: Public-key identification schemes based on multivariate cubic polynomials. In: Fischlin, M., Buchmann, J.A., Manulis, M. (eds.) Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7293, pp. 172–189. Springer (2012), http://dx.doi.org/10.1007/978-3-642-30057-8_11
29. Sakumoto, K., Shirai, T., Hiwatari, H.: Public-key identification schemes based on multivariate quadratic polynomials. In: Rogaway, P. (ed.) Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6841, pp. 706–723. Springer (2011), http://dx.doi.org/10.1007/978-3-642-22792-9_40
30. Shor, P.W.: Polynominal time algorithms for discrete logarithms and factoring on a quantum computer. In: Adleman, L.M., Huang, M.A. (eds.) Algorithmic Number Theory, First International Symposium, ANTS-I, Ithaca, NY, USA, May 6-9, 1994, Proceedings. Lecture Notes in Computer Science, vol. 877, p. 289. Springer (1994), http://dx.doi.org/10.1007/3-540-58691-1_68
31. Unruh, D.: Quantum proofs of knowledge. In: Pointcheval and Johansson [25], pp. 135–152, http://dx.doi.org/10.1007/978-3-642-29011-4_10
32. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9057, pp. 755–784. Springer (2015), http://dx.doi.org/10.1007/978-3-662-46803-6_25
33. Unruh, D.: Collapse-binding quantum commitments without random oracles. IACR Cryptology ePrint Archive 2016, 508 (2016), http://eprint.iacr.org/2016/508
34. Unruh, D.: Computationally binding quantum commitments. In: Fischlin, M., Coron, J. (eds.) Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9666, pp. 497–527. Springer (2016), http://dx.doi.org/10.1007/978-3-662-49896-5_18
35. Yasuda, T., Dahan, X., Huang, Y., Takagi, T., Sakurai, K.: MQ challenge: Hardness evaluation of solving multivariate quadratic problems. IACR Cryptology ePrint Archive 2015, 275 (2015), http://eprint.iacr.org/2015/275