On round elimination for special-sound multi-round identification and the generality of the hypercube for MPCitH

Andreas Hülsing[1,2⋆], David Joseph[1],
Christian Majenz[3⋆⋆], Anand Kumar Narayanan[1]

[1] SandboxAQ, Palo Alto, USA, david.joseph,anand.kumar@sandboxaq.com
[2] Eindhoven University of Technology, The Netherlands, andreas@huelsing.net
[3] Technical University of Denmark, Kgs. Lyngby, Denmark, chmaj@dtu.dk

Abstract. A popular way to build post-quantum signature schemes is by first constructing an identification scheme (IDS) and applying the Fiat-Shamir transform to it. In this work we tackle two open questions related to the general applicability of techniques around this approach that together allow for efficient post-quantum signatures with optimal security bounds in the QROM.
First we consider a recent work by Aguilar-Melchor, Hülsing, Joseph, Majenz, Ronen, and Yue (Asiacrypt'23) that showed that an optimal bound for three-round commit & open IDS by Don, Fehr, Majenz, and Schaffner (Crypto'22) can be applied to the five-round Syndrome-Decoding in the Head (SDitH) IDS. For this, they first applied a transform that replaced the first three rounds by one. They left it as an open problem if the same approach applies to other schemes beyond SDitH. We answer this question in the affirmative, generalizing their round-elimination technique and giving a generic security proof for it. Our result applies to any IDS with $2n+1$ rounds for $n>1$. However, a scheme has to be suitable for the resulting bound to not be trivial. We find that IDS are suitable when they have a certain form of special-soundness which many commit & open IDS have.
Second, we consider the hypercube technique by Aguilar-Melchor, Gama, Howe, Hülsing, Joseph, and Yue (Eurocrypt'23). An optimization that was proposed in the context of SDitH and is now used by several of the contenders in the NIST signature on-ramp. It was conjectured that the technique applies generically for the MPC-in-the-Head (MPCitH) technique that is used in the design of many post-quantum IDS if they use an additive secret sharing scheme but this was never proven. In this work we show that the technique generalizes to MPCitH IDS that use an additively homomorphic MPC protocol, and we prove that security is preserved.
We demonstrate the application of our results to the identification scheme of RYDE, a contender in the recent NIST signature on-ramp. While RYDE was already specified with the hypercube technique applied, this gives the first QROM proof for RYDE with an optimally tight bound.

1 Introduction

Digital signatures are one of the two main primitives in public key cryptography. Digital signatures have numerous applications. Among the most important ones are likely guaranteeing the authenticity of software updates, authenticating communication partners, and non-repudiation for electronic documents. The relevance of digital signatures is also demonstrated by the National Institute of Standards and Technology (NIST) picking signatures as one of the two first primitives for which to run a competition for post-quantum secure systems [NIS16].

One of the most common ways to construct digital signatures is to design an (interactive) identification scheme (IDS) and turn this into a signature scheme via the Fiat-Shamir (FS) transform [FS87]. This is not only the way DSA and ECDSA are designed, but also a widespread approach to design post-quantum signature schemes, including Dilithium [Duc+18], the scheme that NIST picked as general purpose winner of the competition, as well as at least eight of the proposals [Bet+23; Ara+23c; Adj+23; FR23; Aar+23; Ara+23a; Kim+23; Agu+23a]) in the recent NIST signature on-ramp [NIS22]. While we are blessed with beautiful three-round

[⋆] Andreas Hülsing is supported by an NWO VIDI grant (Project No. VI.Vidi.193.066).
[⋆⋆] Christian Majenz is supported by a DFF Sapere Aude grant grant (IM-3PQC, grant ID 10.46540/2064-00034B).

IDS with nice security properties in the case of (EC)DSA, the situation for post-quantum IDS is not as great and poses significant challenges to the cryptographic community.

The first challenge is the tightness of security proofs which determines the size of parameters for which formal security guarantees can be given. In general, the requirement of considering quantum adversaries does not simplify the situation around tightness. Given that the security of FS requires a random oracle model argument, the analysis of the necessary variations of FS has to be done in the quantum-accessible random oracle model (QROM). This is already challenging for plain three-round IDS, with loose bounds for the generic case being the result.

The IDS underlying Dilithium is three round, but makes use of rejection sampling which significantly complicates the analysis of the required FS variant [Bar+23; Dev+23]. However, because the IDS can be made lossy, there exists a relatively tight security bound. Many other post-quantum IDS are five or more round IDS. For these IDS the known generic Fiat-Shamir transforms in the QROM lead to extremely non-tight security bounds. However, many of these schemes are so-called commit & open schemes. For the Fiat-Shamir transform of this class of schemes, a recent work [Don+22a] gave an optimal security bound even in the QROM, although limited to three-round IDS. A follow-up work [AM+23] demonstrated that for the SDitH IDS [FJR22], a five-round commit & open IDS, one can apply the result of [Don+22a] by first merging the first three rounds into one. While this round-elimination works like FS (replace the challenge with the hash of the prover message), the security analysis is a lot simpler than for FS as no extraction is required. This leads to an optimal bound matching a straight-forward brute-force search attack. Thereby, this approach bears the potential of providing optimal security bounds for IDS-based signature schemes. The authors of [AM+23] left it as an open problem to analyze the applicability of this round-reduction step to other IDS.

Another problem is general performance. Most of the multi-round IDS have significantly worse performance than the classical three round schemes in terms of size as well as speed. While this is partially inherent as it is due to the bigger description size of the used hard problems, some of it seems to be caused by the specific design. Therefore, optimization techniques frequently pop-up. A recent optimization technique is the hypercube technique [Agu+23b] for MPC-in-the-head (MPCitH) based IDS. This technique has been used in [Agu+23b] to increase the speed of SDitH signing and verification by factors ranging from 4 to 12 while preserving security and sizes. For practical applications this is a massive improvement. For comparison, the difference between a Dilithium signature and a SPHINCS+ signature [Hül+22] is about a factor 5, and for many scenarios, people consider SPHINCS+ too big, but Dilithium acceptable. Again, while the applicability of the technique to a certain class of schemes was conjectured by the authors, a formal analysis was left as an open problem.

Our contribution. In this work we are tackling these two open problems: The applicability of round-elimination to general IDS, and the applicability of the hypercube technique to other MPCitH-based IDS. We apply our results to the IDS underlying RYDE [Ara+23a], a contender in the recent NIST signature on-ramp [NIS22] to demonstrate the application.

For the round-elimination we demonstrate that the technique of [AM+23] can be applied to a wide range of $(2n + 1)$-round IDS for $n > 1$. We actually give a security bound for a computational version of $\mathfrak{S}$-soundness [Don+22a] which can be viewed as a generalization of query-bounded special soundness [AM+23] to arbitrary challenge patterns, for any such transformed protocol. However, the bound becomes trivial for schemes that are not suitable (e.g., MQ-DSS [Sam+19]). Intuitively, we can merge the first three rounds of an IDS as long as the remaining interactive protocol has some form of special soundness with overwhelming probability when the first message is adversarially chosen and the first challenge is sampled afterwards from the uniform distribution. This process can be iterated until the above condition does not apply anymore. We also prove honest-verifier zero-knowledge of the resulting scheme by an application of the adaptive reprogramming lemma from [Gri+21]. While we only demonstrate the application of our result to RYDE, a high level analysis suggests that all other MPCitH-based signature schemes in the ongoing NIST signature on-ramp are suitable (i.e., [Bet+23; Ara+23c; Adj+23; FR23; Aar+23; Kim+23]).

With regard to the hypercube technique, we first introduce an abstraction for MPCitH-based identification schemes. We prove that for a scheme which fits the abstraction and is additively homomorphic, the hypercube technique can be applied and security with regard to soundness and honest-verifier zero-knowledge is preserved. More precisely, we require that the MPC computation is additively homomorphic with regards to all party-specific inputs and all communications it generates. Moreover, we define a final predicate that takes all communications and decides about acceptance of the computation. This predicate also has to be constant when replacing any two input communications by their sum. Again, we demonstrate the application of our approach for RYDE (RYDE already applied the hypercube technique, but we discuss how one would start from the flat scheme). However, the technique should be applicable to any MPCitH IDS that builds on the BN approach [BN20] or follow-up work [KZ22].

We demonstrate the full sequence of steps on the example of the flat version of the RYDE IDS. We first apply the round-elimination to the parallel-composed version of the RYDE IDS, providing the resulting security bounds. Next, we discuss how a single instance of the resulting three-round IDS fits the required abstraction of MPCitH-based IDS and argue that it is additively homomorphic in the above sense. Finally, we apply the result of [Don+22a] in a slightly adapted version of [AM+23] to obtain a security bound for the UF-CMA-security of the signature scheme that results from FS-transforming the three-round hypercube IDS. This blueprint can be followed by designers of similar schemes to easily get a security proof in the QROM for their signature scheme. Especially, we conjecture that our technique applies to all remaining MPCitH schemes in the NIST signature on-ramp. The main steps are to prove a fine-grained soundness statement for the plain IDS (something that is required for the security analysis of the IDS in any case), and matching the steps in the round-eliminated IDS to the MPCitH abstraction. We note here that the hypercube technique can also be skipped, e.g., in case it is not applicable. This is relevant as the round-elimination has much wider applicability than the hypercube technique.

Organization. We discuss round-elimination in Sec. 2. In Sec. 3, we discuss the hypercube technique. We finally present an application of our results to RYDE in Sec. 4. To be self-contained, we provide the required results on the FS-transform for three-round commit & open IDS in Appendix A.

2 Round Elimination

In this section, we devise a round elimination technique in the QROM for special-sound multi-round identification schemes. In fact, our results hold in the more demanding eQROM [HHM22, Sec. 4], which we briefly discuss in the Appendix A. In essence, the eQROM is a strong model that gives the adversary adaptive access to an additional extraction interface during the interaction. Looking ahead to the next sections, we will reduce a 5-round protocol to a 3-round protocol and then apply the hypercube aggregation formalism to construct a secure, efficient signature scheme. The round elimination results from this section will justify this 5 to 3 round reduction, by quantifying the security of the resulting 3-round protocol. The hypercube aggregation then preserves this security. But our round elimination applies in much greater generality than needed for optimized MPCitH signatures. In particular, it applies to identification schemes with many rounds that fulfill a certain variant of special soundness and is likely to find use in various settings. To this end, we identify and characterize properties of multi-round identification schemes that suffice for our round elimination in wide generality, accompanied by security proofs in the eQROM.

We begin with preliminaries recalling definitions used in [AM+23] where round elimination is already applied to a specific case. Then we generalize the notation to identification schemes of an arbitrary odd number of rounds. Later, we generalize the notion of special soundness to $\mathfrak{S}$-soundness, closely following [Don+22b]. Finally, in Sec. 2.2, we analyze eliminating one round of verifier interaction. We define fine-grained round-by-round notions of soundness to aid our analysis and characterize eQROM security guarantees.

2.1 Preliminaries

In the following we provide the definitions for commitments, and identification schemes. We closely follow [AM+23] in these as we later make use of results from that work when transforming identification schemes to signature schemes. In the below, we require that all honest algorithms are efficiently computable.

Com. In this work we consider only hash-based commitments. Hence, we define commitment scheme as an algorithm Com that given an input x and randomness $\rho \in \{0,1\}^r$ produces a commitment $\mathsf{com} = \mathsf{Com}(x;\rho) \in \{0,1\}^c$. We make the randomness explicit as given (com, x, ρ) everybody can check that indeed $\mathsf{com} = \mathsf{Com}(x;\rho)$. From our commitment schemes we require two properties: We want them to be *binding* and *hiding*.

We define the advantage of a possibly quantum adversary A against the computational bindingness of Com as

$$\mathrm{Adv}^{\mathsf{bind}}_{\mathsf{Com}}(\mathsf{A}) := \Pr[((x_1,\rho_1),(x_2,\rho_2)) \leftarrow \mathsf{A} : x_1 \neq x_2 \wedge \mathsf{Com}(x_1;\rho_1) = \mathsf{Com}(x_2;\rho_2)].$$

We define the advantage of a possibly quantum adversary A against the computational hidingness of Com as

$$\begin{aligned}\mathrm{Adv}^{\mathsf{hide}}_{\mathsf{Com}}(\mathsf{A}) := \Big| &\Pr[(x_1,x_2) \leftarrow \mathsf{A}; \rho \leftarrow \{0,1\}^k : 1 \leftarrow \mathsf{A}(\mathsf{Com}(x_1;\rho))] \\ &- \Pr[(x_1,x_2) \leftarrow \mathsf{A}; \rho \leftarrow \{0,1\}^k : 1 \leftarrow \mathsf{A}(\mathsf{Com}(x_2;\rho))]\Big| .\end{aligned}$$

In the analysis, Com may be recast with a random oracle, to prove security in the eQROM.

2.1.1 Identification Schemes, zero-knowledge, and soundness

We next discuss identification schemes (IDS), a variant of honest-verifier zero-knowledge (HVZK) and several (successively refined) variants of soundness. We start with canonical three-round IDS.

Three round identification schemes. Three round, public coin, commit and open identification schemes, which we abbreviate as IDS, will be rudimentary building blocks upon which our longer more intricate schemes are built and analyzed. An IDS is an interactive protocol between a prover P and a verifier V. It is defined by a tuple of algorithms $\mathsf{IDS} = (\mathsf{Keygen}, \mathsf{Commit}, \mathsf{Resp}, \mathsf{Vrf})$ and a challenge space $\mathcal{C}$. Prior to any interaction, Keygen is run and outputs a key pair $(\mathsf{pk}, \mathsf{sk})$. A protocol run starts with P running $(\mathsf{st}, \mathsf{w}) \leftarrow \mathsf{Commit}(\mathsf{sk})$. The commitment message w is sent to V which samples a challenge c from the uniform distribution over $\mathcal{C}$ and sends it to P. Upon receiving c, the prover P runs $\mathsf{z} \leftarrow \mathsf{Resp}(\mathsf{st}, \mathsf{c})$ and sends z back to V. The verifier accepts if $\mathsf{Vrf}(\mathsf{pk}, \mathsf{w}, \mathsf{c}, \mathsf{z}) = 1$ and rejects otherwise.

The *transcript* of a run of IDS is the tuple $(\mathsf{w}, \mathsf{c}, \mathsf{z})$ of messages exchanged. We are only interested in IDS that are *correct*, i.e., for any key pair output by Keygen, we want that the execution of IDS between honest P and V always accepts. A property that can be handy when turning IDS into signatures is that of *commitment-recoverable* IDS. An IDS is commitment recoverable if there exists an algorithm Rcvr, such that for any valid transcript $(\mathsf{w}, \mathsf{c}, \mathsf{z})$, $\mathsf{w} \leftarrow \mathsf{Rcvr}(\mathsf{c}, \mathsf{z})$.

We expect IDS to provide two security properties which are defined below.

Honest-verifier zero-knowledge (HVZK). The most commonly used variant of HVZK is the statistical one which has the advantage that it nicely composes and directly leads bounds for parallel repetition. This does not hold for the computational variant as previously pointed out in [Gri+21]. Hence, when considering computational HVZK, one directly has to prove multi-transcript HVZK to cover parallel composition. This approach was taken in [AM+23] as for hash-based commitments, statistical HVZK requires 2.5 times more commitment randomness [Lei18], which, in turn, impacts signature size. The below definition makes use of an honest transcript generator Trans and an HVZK simulator Sim. It closely follows [Gri+21].

Definition 1 (HVZK simulator and honest transcript generator). *An* HVZK simulator *for* IDS *is an algorithm* Sim *that takes as input the public key* pk *and outputs a transcript* $(\mathsf{w}, \mathsf{c}, \mathsf{z})$*. An* honest transcript generator *for* IDS *is an algorithm* Trans *that takes as input the secret key* sk *and outputs a transcript* $(\mathsf{w}, \mathsf{c}, \mathsf{z})$ *by means of an honest execution of* IDS*.*

Based on this we define computational t-HVZK of an IDS as follows:

Definition 2 (Computational t-HVZK). *We define the advantage of a possibly quantum adversary* A *against the computational* t-HVZK *of* IDS *with simulator* Sim*, making no more than* t *queries to its (transcript-)oracle as*

$$\begin{aligned}\mathrm{Adv}^{t\text{-}\mathsf{HVZK}}_{\mathsf{IDS},\mathsf{Sim}}(\mathsf{A}) := \Big| &\Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}() : 1 \leftarrow \mathsf{A}^{\mathsf{Sim}(\mathsf{pk})}(\mathsf{pk})] \\ &- \Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}() : 1 \leftarrow \mathsf{A}^{\mathsf{Trans}(\mathsf{sk})}(\mathsf{pk})]\Big| .\end{aligned}$$

Special Soundness. Conventionally, special soundness is a notion that requires the existence of an extractor Ext, which, given a set of transcripts that fulfills some requirements, will always output a secret key for the public key associated with these transcripts. In [AM+23], the authors only consider a notion of soundness for the three round IDS and give a direct proof for that. They use a computational version of special soundness (spS) which takes into account the computational effort of the algorithm A that produces the transcript set and consider extractors Ext which succeed with a probability that can be less than 1. Their definition links the success probability of Ext to the runtime of A. Moreover, their definition is tailored to τ-fold parallel-composition of some basic IDS'. As an abstraction of this parallel composition, they say IDS has a *splittable challenge* if the challenge of IDS can be split into τ challenges of IDS'. They introduce the notion of distance between two IDS challenges $\mathsf{Dist}(\mathsf{c}, \hat{\mathsf{c}})$ as the number of IDS' challenges on which they disagree, i.e., the number of indices $1 \leq j \leq \tau$ for which $(\mathsf{c})^{(j)} \neq (\hat{\mathsf{c}})^{(j)}$, where $(\mathsf{c})^{(j)}$ (resp. $(\hat{\mathsf{c}})^{(j)}$) is the j-th IDS' challenge in c (resp. $\hat{\mathsf{c}}$).

Definition 3 ((Query-bounded) distance-d special soundness for IDS with splittable challenge). *We define the advantage of a possibly quantum adversary* A *against the query* *bounded special soundness of a composed* IDS *with respect to extractor* Ext *in the (quantum-accessible) random oracle model as follows*

$$\begin{aligned}\mathrm{Adv}^{d\text{–}\mathsf{spS}}_{\mathsf{IDS},\mathsf{Ext}}(\mathsf{A}) := \Pr[&(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Keygen}(); ((\mathsf{w},\mathsf{c},\mathsf{z})),(\hat{\mathsf{w}},\hat{\mathsf{c}},\hat{\mathsf{z}})) \leftarrow \mathsf{A}^{\mathsf{RO}}(\mathsf{pk});\\ &\mathsf{sk}' \leftarrow \mathsf{Ext}^{\mathsf{RO}}((\mathsf{w},\mathsf{c},\mathsf{z})),(\hat{\mathsf{w}},\hat{\mathsf{c}},\hat{\mathsf{z}})) :\\ &(\mathsf{Vrf}(\mathsf{pk},(\mathsf{w},\mathsf{c},\mathsf{z}))) = \mathsf{Vrf}(\mathsf{pk},(\hat{\mathsf{w}},\hat{\mathsf{c}},\hat{\mathsf{z}})) = 1)\\ &\wedge (\mathsf{w} = \hat{\mathsf{w}}) \wedge (d = \mathsf{Dist}(\mathsf{c},\hat{\mathsf{c}})) \wedge \big((\mathsf{sk}',\mathsf{pk}) \notin \mathsf{Keygen}()\big)],\end{aligned}$$

where q is the maximum number of queries that A *makes to* RO *and we consider it understood that in this case all* IDS *algorithms may depend on* RO*.*

In contexts where it helps to clarify the number of splits τ underlying the protocol, we will write the adversary advantage $\mathrm{Adv}^{d\text{–}\mathsf{spS}}_{\mathsf{IDS},\mathsf{Ext}}(\mathsf{A})$ as $\mathrm{Adv}^{d\text{–}\mathsf{spS}}_{\mathsf{IDS}_\tau,\mathsf{Ext}}(\mathsf{A})$.

$(2\ell+1)$-round identification schemes. Our round elimination technique applies in much greater generality than required by our MPCitH example. In particular, it works for identification schemes with an arbitrary odd number $2\ell+1$ of rounds,[4] as long as there is a soundness structure that splits or factorizes in a certain way. We describe our results in this greater generality, since they are interesting beyond MPCitH constructions.

A $(2\ell+1)$-round identification scheme Π is an interactive protocol between a prover P and a verifier V. It is defined by a tuple of algorithms $(\mathsf{Keygen}_\Pi,\ \{\mathsf{Commit}_{\Pi,i}\}_{i=0}^{\ell},\ \mathsf{Resp}_\Pi, \mathsf{Vrf}_\Pi)$ and a finite challenge space $\mathcal{C}_\Pi$. When the Π being referred to is clear from context, we will drop the subscript Π and write $(\mathsf{Keygen}, \{\mathsf{Commit}_i\}_{i=0}^{\ell}, \mathsf{Resp}, \mathsf{Vrf})$ and $\mathcal{C}$. The challenge space is a product $\mathcal{C} = \mathcal{C}_1 \times \mathcal{C}_2 \times \ldots \times \mathcal{C}_\ell$. Prior to any interaction, Keygen is run and outputs a key pair $(\mathsf{pk},\mathsf{sk})$. A protocol run starts with P running $(\mathsf{w}_1) \leftarrow \mathsf{Commit}_1(\mathsf{sk})$. The commitment message w_1 is sent to V which samples a challenge c_1 from the uniform distribution over $\mathcal{C}_1$ and sends it to P. Upon receiving c_1, the prover P runs $\mathsf{w}_2 \leftarrow \mathsf{Commit}_2(\mathsf{sk},\mathsf{c}_1)$ and sends w_2 to V. Then V independently samples a challenge c_2 from the uniform distribution over $\mathcal{C}_2$ and sends it to P. Then P sends $\mathsf{w}_3 \leftarrow \mathsf{Commit}_3(\mathsf{sk},\mathsf{c}_1,\mathsf{c}_2)$, and so on. On receiving the final challenge c_ℓ, P runs $\mathsf{z} \leftarrow \mathsf{Resp}(\mathsf{sk},\mathsf{c}_1,\ldots,\mathsf{c}_\ell)$ and sends z to V. The commitment functions can also take a state as part of the input, with each commitment function updating the state after a round of interaction, to be fed as the input of the successive commitment function. We suppress the state updates from the notation for readability. At the end of the interaction, V accepts if $\mathsf{Vrf}(\mathsf{pk},\mathsf{w}_1,\mathsf{c}_1,\ldots,\mathsf{w}_\ell,\mathsf{c}_\ell,\mathsf{z}) = 1$ and rejects otherwise. The prover's commitment/response sequence $(\mathsf{w}_1,\mathsf{w}_2,\ldots,\mathsf{w}_\ell,\mathsf{z})$ is constrained to a fixed finite product space $W_1 \times W_2 \times \ldots \times W_\ell \times Z$. The transcript of the interaction is denoted by

$$\mathbf{t} := (\mathsf{w}_1,\mathsf{c}_1,\mathsf{w}_2,\mathsf{c}_2,\ldots,\mathsf{w}_\ell,\mathsf{c}_\ell,\mathsf{z})$$

and its projection to the commitment and challenge sequences respectively by

$$\mathbf{w}(\mathbf{t}) := (\mathsf{w}_1,\mathsf{w}_2,\ldots,\mathsf{w}_\ell) \text{ and } \mathbf{c}(\mathbf{t}) := (\mathsf{c}_1,\mathsf{c}_2,\ldots,\mathsf{c}_\ell) \in \mathcal{C}.$$

For $m < \ell$, let $\mathbf{t}_{<m} := (\mathsf{w}_1,\mathsf{c}_1,\mathsf{w}_2,\mathsf{c}_2,\ldots,\mathsf{w}_m)$, $\mathbf{t}_{\leq m} := (\mathsf{w}_1,\mathsf{c}_1,\mathsf{w}_2,\mathsf{c}_2,\ldots,\mathsf{w}_m,\mathsf{c}_m)$, $\mathbf{t}_{>m} := (\mathsf{w}_{m+1},\mathsf{c}_{m+1},\mathsf{w}_{m+2},\mathsf{c}_{m+2},\ldots,\mathsf{c}_\ell,z)$, $T_{\leq m} := W_1 \times C_1 \times W_2 \times C_2 \ldots \times C_m$, $\mathbf{w}(\mathbf{t})_{\leq m} := (\mathsf{w}_1,\mathsf{w}_2,\ldots,\mathsf{w}_m)$, $\mathcal{C}_{>m} := \mathcal{C}_{m+1} \times \mathcal{C}_{m+2} \times \ldots \times \mathcal{C}_\ell$, and $\mathbf{c}(\mathbf{t})_{>m} := (\mathsf{c}_{m+1},\mathsf{c}_{m+2},\ldots,\mathsf{c}_\ell)$.

[4] For public-coin protocols, a first or last message from the verifier is useless. Thus, only odd numbers of rounds make sense.

Parallel repetition. The r-fold parallel repetition $\Pi^{\vee r}$ of Π is defined as follows. An index (j) in the superscript will indicate the (j)-th component in the repetition. The key generation algorithms are identical, $\mathsf{Keygen}_{\Pi^{\vee r}} = \mathsf{Keygen}$. The challenge space $\mathcal{C}_{\Pi^{\vee r}} := \mathcal{C}_1^r \times \mathcal{C}_2^r \times \ldots \times \mathcal{C}_\ell^r$. The commitment and response algorithms are merely r-fold copies, $\mathsf{Commit}_{\Pi^{\vee r},i} := \mathsf{Commit}_i^r$ and $\mathsf{Resp}_{\Pi^{\vee r},i} := \mathsf{Resp}^r$. That is, for $i \leq \ell$,

$$\mathsf{Commit}_{\Pi^{\vee r},i}\left(\mathsf{sk}, (\mathsf{c}_1^{(j)}, \mathsf{c}_2^{(j)}, \ldots, \mathsf{c}_{i-1}^{(j)})_{j=1}^r\right) = \left(\mathsf{Commit}_i(\mathsf{sk}, \mathsf{c}_1^{(j)}, \mathsf{c}_2^{(j)}, \ldots, \mathsf{c}_{i-1}^{(j)})\right)_{j=1}^r,$$

$$\mathsf{Resp}_{\Pi^{\vee r}}\left(\mathsf{sk}, (\mathsf{c}_1^{(j)}, \mathsf{c}_2^{(j)}, \ldots, \mathsf{c}_{\ell}^{(j)})_{j=1}^r\right) = \left(\mathsf{Resp}(\mathsf{sk}, \mathsf{c}_1^{(j)}, \mathsf{c}_2^{(j)}, \ldots, \mathsf{c}_{\ell}^{(j)})\right)_{j=1}^r.$$

The verifier predicate is the conjunction $\mathsf{Vrf}_{\Pi^{\vee r}}(\mathbf{t}^{(1)}, \mathbf{t}^{(2)}, \ldots, \mathbf{t}^{(r)}) := \bigwedge_{j=1}^r \mathsf{Vrf}(\mathbf{t}^{(j)})$, where $\mathbf{t}^{(j)}$ is the transcript of the j-th repetition.

$\mathfrak{S}$-soundness and naive cheating probabilities. Following [Don+22b], we define a notion of soundness (and corresponding extractors) that generalizes special soundness. Sets of transcripts whose acceptance by the verifier suffice for extracting a witness will qualify the soundness structure of the protocol, through the structure of their challenge sequences. If an extractor is presented with more accepting transcripts than necessary, this does not obstruct witness extraction. Therefore, the following notion of an increasing set of challenge sequence sets aids us in defining the soundness structure. Call a (possibly empty) set $\mathfrak{S} \subseteq 2^{\mathcal{C}}$ of subsets of $\mathcal{C}$ "increasing" if and only if $(S \in \mathfrak{S}) \wedge (S \subseteq S' \subseteq \mathcal{C}) \Rightarrow S' \in \mathfrak{S}$. An increasing set $\mathfrak{S}$ is a subset of the power set $2^{\mathcal{C}}$ of challenge sequences $\mathcal{C}$. An element $S \in \mathfrak{S}$ is hence a subset $S \subset \mathcal{C}$ of challenge sequences.

Definition 4 ($\mathfrak{S}$-soundness ([Don+22b](Def. 5.2))). *Let $\mathfrak{S} \subseteq 2^{\mathcal{C}}$ be increasing. For a non empty $\mathfrak{S}$, a $(2\ell+1)$-round identification protocol Π is called* $\mathfrak{S}$-sound *if there exists a probabilistic polynomial time algorithm $\mathsf{Ext}_{\mathfrak{S}}$ that takes as input*

- *a public key pk generated by Keygen, and*
- *a set $\mathcal{T}$ of transcripts whose*
  - *first messages are the same, that is, $\forall \mathbf{t}, \mathbf{t}' \in \mathcal{T}, \mathbf{t}_{<1} = \mathbf{t}'_{<1}$,*
  - *challenge sequences $\mathbf{c}(\mathbf{t}), \mathbf{t} \in \mathcal{T}$ form a set $\{\mathbf{c}(\mathbf{t}), \mathbf{t} \in \mathcal{T}\} \in \mathfrak{S}$,*
  - *transcripts pass verification, that is, $\forall \hat{\mathbf{t}} \in \mathcal{T}$, $\mathsf{Vrf}_\Pi(\mathsf{pk}, \hat{\mathbf{t}}) = 1$,*

and outputs a secret key sk such that $(\mathsf{sk}, \mathsf{pk}) \in \mathsf{Keygen}$. We say $\mathfrak{S}$ is an extraction structure *for Π.*

Remark 1. When demanding that $\mathsf{Ext}_{\mathfrak{S}}$ be probabilistic polynomial time, we follow the usual convention that the expected run time is bounded by a polynomial in the size of the input. The set $\mathcal{T}$ of transcripts is part of the input, and in our contexts will be presented to $\mathsf{Ext}_{\mathfrak{S}}$ by a polynomial time classical/quantum adversary. Therefore, in the cryptographic contexts we consider, $\mathsf{Ext}_{\mathfrak{S}}$ will be probabilistic polynomial time in the security parameter. In particular, we do not have to worry about a set $\{\mathbf{c}(\mathbf{t}), \mathbf{t} \in \mathcal{T}\}$ of challenge tuples whose description is of exponential size in the security parameter.

Remark 2. To clarify, the empty set case $\mathfrak{S} = \emptyset$ corresponds to there being no extraction guarantee. By default, every $(2\ell + 1)$-round identification protocol Π is deemed to be $\emptyset$-*sound* and the corresponding extraction algorithm $\mathsf{Ext}_\emptyset$ is not required to output a valid secret key sk.

Example: Special soundness of Σ-protocols. As an illustrative first example, let us express the familiar notion of special soundness of Σ-protocols in the language of Def. 4. The special case $\ell = 1$ corresponds to Σ-protocols. Special soundness of Σ-protocols is realized as a special case of Def. 4 by setting $\ell = 1$ and $\mathfrak{S} = \{S \subseteq \mathcal{C}_1 \mid |S| \geq 2\}$. In essence, two transcripts with the same first message and distinct challenges suffice for extraction. That is, a Σ-protocol $\mathit{\Pi}$ with key generation Keygen is $\{S \subseteq \mathcal{C}_1 \mid |S| \geq 2\}$*-sound* if there exists an efficient algorithm that

- given a public key pk generated by Keygen,
- and a set $\{(\mathsf{w}, \mathsf{c}, \mathsf{z}_\mathsf{c})_\mathsf{c}\}$ of at least two transcripts, indexed by distinct challenges c, but with the same first message w,
- such that for all input transcripts $\mathsf{Vrf}(\mathsf{pk}, (\mathsf{w}, \mathsf{c}, \mathsf{z}_\mathsf{c})) = 1$,

outputs a sk such that $(\mathsf{sk}, \mathsf{pk}) \in \mathsf{Keygen}$.

Example: Parallel repetition. The soundness of an r-fold parallel repetition of a ($\mathfrak{S}$-sound $2\ell + 1$-round) protocol (with challenge space $\mathcal{C}$) is also captured within the scope of Def. 4. To illustrate this structure, define

$$\mathfrak{S}^{\vee r} := \{S \subseteq \mathcal{C}^r \mid \exists\, j \in \{1, 2, \ldots, r\} : S^{(j)} \in \mathfrak{S}\},$$

where $S^{(j)} := \{\mathbf{c} \in \mathcal{C} \mid \exists\, (\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \ldots, \mathbf{c}^{(r)}) \in S : \mathbf{c}^{(j)} = \mathbf{c}\}$ is the projection onto the j-th repetition. The r-fold parallel repetition of a $\mathfrak{S}$-sound protocol is $\mathfrak{S}^{\vee r}$-sound, as seen by the extractor that searches for a repetition with an extractable set of transcripts and applies the extractor of the atomic protocol.

$\mathfrak{S}$-soundness implies soundness for an IDS $\mathit{\Pi}$ if key recovery for that scheme is hard. Indeed, any classical (non-quantum) adversary that will be successful for all challenges from a set $S \in \mathfrak{S}$ can be converted into a key recovery algorithm via rewinding and the $\mathfrak{S}$-soundness extractor. In the quantum setting, a similar reduction is possible via Unruh rewinding [Unr12]. We thus define a number we call the naive cheating bound, as the maximal success probability of any classical adversary that cannot be used for key recovery in this way,

Definition 5. *For an increasing $\mathfrak{S}$, define $p^{\mathfrak{S}}_{naive} := \frac{1}{|\mathcal{C}|} \max_{\hat{S} \subseteq \mathcal{C}, \hat{S} \notin \mathfrak{S}} |\hat{S}|$.*

In particular, $p^{\emptyset}_{naive} = 1$. By definition, $p^{\mathfrak{S}}_{naive} \in [0, 1]$. This naive cheating probability is multiplicative with respect to parallel repetition [Don+19, Lem. 3.5]. That is,

$$p^{\mathfrak{S}^{\vee r}}_{naive} = \left(p^{\mathfrak{S}}_{naive}\right)^r . \tag{1}$$

The naive cheating bound $p^{\mathfrak{S}}_{naive}$ has additional significance in the post-quantum context due to the extraction technique developed in [Don+22a], where a bound on the extraction error as a function of $p^{\mathfrak{S}}_{naive}$ is obtained.

Example. We next discuss distance-d special soundness of an IDS (call $\mathit{\Pi}$) that is a τ-parallel repetition of another IDS (call $\mathit{\Pi}'$) that has special soundness. Let $\mathcal{C}'$ be the challenge space of $\mathit{\Pi}'$, which implies $(\mathcal{C}')^\tau$ is the challenge space of $\mathit{\Pi}$. The notion of distance-d special soundness of $\mathit{\Pi}$ is captured by

$$\mathfrak{S}_{\mathsf{IDS}_{\tau,d}} := \left\{S \subseteq (\mathcal{C}')^r \,||\{j \in \{1, 2, \ldots, \tau\} : |S^{(j)}| \geq 2\}| \geq d\right\},$$

where, as before, $S^{(j)} = \{\mathbf{c}' \in \mathcal{C}' \mid \exists\, (\mathbf{c}'^{(1)}, \mathbf{c}'^{(2)}, \dots, \mathbf{c}'^{(\tau)}) \in S : \mathbf{c}'^{(j)} = \mathbf{c}'\}$ is the projection on to the j-th repetition. To compute the naive cheating probability, observe that $\max_{\hat{S} \subseteq \mathcal{C}, \hat{S} \notin \mathfrak{S}} |\hat{S}|$ is attained by

$$\left\{(\mathbf{c}'^{(1)}, \mathbf{c}'^{(2)}, \dots, \mathbf{c}'^{(\tau)}) \in (\mathcal{C}')^{\tau} \big| |\{j \in \{1, 2, \dots, \tau\} : \mathbf{c}'^{(j)} \neq \hat{\mathbf{c}}'^{(j)}\}| < d\right\},$$

for every choice of $(\hat{\mathbf{c}}'^{(1)}, \hat{\mathbf{c}}'^{(2)}, \dots, \hat{\mathbf{c}}'^{(\tau)}) \in (\mathcal{C}')^{\tau}$. The reader may recognize the set attaining the maximum as a Hamming ball of radius $d-1$. The centre $(\hat{\mathbf{c}}'^{(1)}, \hat{\mathbf{c}}'^{(2)}, \dots, \hat{\mathbf{c}}'^{(\tau)}) \in (\mathcal{C}')^{\tau}$ of the Hamming ball chosen does not matter, as every radius $d-1$ Hamming ball attains the maximum. The naive cheating probability is therefore

$$p_{naive}^{\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}} = \frac{\sum_{j=1}^{d-1} \binom{\tau}{j} (|\mathcal{C}'| - 1)^{j-1}}{|\mathcal{C}'|^{\tau}}, \tag{2}$$

where the numerator is the volume of the radius $d-1$ Hamming ball.

Fine-grained soundness. In the following Def. 6 to 8, we define fine grained soundness and extraction structures as well as a new notion of computational special soundness. These notions will later help analyze the protocols with early rounds eliminated. We begin with the notion of $(\mathbf{t}_{\leq m}, \mathfrak{S}_{>m})$-soundness and a corresponding extractor $\mathsf{Ext}_{(\mathbf{t}_{\leq m}, \mathfrak{S}_{>m})}$, parameterized by a transcript prefix $\mathbf{t}_{\leq m}$ (up to the m-th challenge) and an increasing set $\mathfrak{S}_{>m}$ of subsets of challenge sequence suffixes starting from the $(m+1)$-th challenge. Informally, this extractor can extract from a set of transcripts satisfying the following. Each transcript has prefix $\mathbf{t}_{\leq m}$ and the collection of transcript suffixes falls into the prescribed soundness family $\mathfrak{S}_{>m}$.

Definition 6 ($(\mathbf{t}_{\leq m}, \mathfrak{S}_{>m})$-soundness and $\mathsf{Ext}_{(\mathbf{t}_{\leq m}, \mathfrak{S}_{>m})}$ extractor). *Let Π be a $(2\ell+1)$-round identification protocol with challenge space $\mathcal{C} = \prod_{i=1}^{\ell} \mathcal{C}_i$. Fix an $m \in \{1, 2, \dots, \ell - 1\}$. Let $\mathbf{t}_{\leq m} \in T_{\leq m}$ be a transcript prefix and $\mathfrak{S}_{>m} \subseteq 2^{\mathcal{C}_{>m}}$ be increasing. An extractor $\mathsf{Ext}_{(\mathbf{t}_{\leq m}, \mathfrak{S}_{>m})}$ is a probabilistic polynomial time algorithm that*

- *given a public key* pk *generated by* Keygen*, and*
- *a set $\mathcal{T}$ of transcripts whose*
  - *prefixes up to the m-th challenge match $\mathbf{t}_{\leq m}$, i.e., $\forall \hat{\mathbf{t}} \in \mathcal{T},\ \hat{\mathbf{t}}_{\leq m} = \mathbf{t}_{\leq m}$,*
  - *challenge sequence suffixes starting with the $(m+1)$-th challenge satisfy*

$$\{\mathbf{c}(\hat{\mathbf{t}})_{>m} \mid \hat{\mathbf{t}} \in \mathcal{T}\} \in \mathfrak{S}_{>m},$$

- *such that all transcripts pass verification, that is, $\forall \hat{\mathbf{t}} \in \mathcal{T}$, $\mathsf{Vrf}_{\Pi}(\mathsf{pk}, \hat{\mathbf{t}}) = 1$,*

outputs a secret key sk *such that* $(\mathsf{sk}, \mathsf{pk}) \in \mathsf{Keygen}$*. Define Π to be $(\mathbf{t}_{\leq m}, \mathfrak{S}_{>m})$-*sound *if there exists such an extractor $\mathsf{Ext}_{(\mathbf{t}_{\leq m}, \mathfrak{S}_{>m})}$.*

Our next soundness notion is parameterized by a function mapping transcript prefixes up to the m-th challenge to extraction structures for the remainder of the protocol. This parameterization function encodes all the information about the identification protocol that we exploit in the round elimination soundness proofs. In particular, it is allowed to map certain prefix sets to the empty set, modelling scenarios where there is no extraction guarantee.

Definition 7 (prefix-conditioned $\mathfrak{s}$-soundness). *Let Π be a $(2\ell+1)$-round identification protocol and $m < \ell$. Define Π to be $\mathfrak{s}$-*sound *with prefix-conditioned-soundness function $\mathfrak{s} : T_{\leq m} \to 2^{(2^{\mathcal{C}_{>m}})}$, if for all transcript prefixes $\mathbf{t}_{\leq m} \in T_{\leq m}$, the protocol Π is $(\mathbf{t}_{\leq m}, \mathfrak{s}(\mathbf{t}_{\leq m}))$-sound.*

To such a function $\mathfrak{s}$, we associate an extractor $\mathsf{Ext}_{\mathfrak{s}}$ defined as follows. It takes as input a public key pk generated by Keygen, and a set $\mathcal{T}$ of transcripts whose prefixes up to the m-th challenge agree. Then $\mathsf{Ext}_{\mathfrak{s}}(\mathsf{pk}, \mathcal{T}) := \mathsf{Ext}_{(\mathbf{t}_{\leq m}, \mathfrak{s}(\mathbf{t}_{\leq m}))}(\mathsf{pk}, \mathcal{T})$, where $\mathbf{t}_{\leq m}$ is the common prefix of $\mathcal{T}$ and $\mathsf{Ext}_{(\mathbf{t}_{\leq m}, \mathfrak{s}(\mathbf{t}_{\leq m}))}$ is as in Def. 6.

Def. 4 to 7 are statistical in nature. Just like in [AM+23], our general round reduction technique will yield protocols that only enjoy a computational (or query-) bounded flavor of special soundness. To formalize computational security, we use the the *expected naive cheating bound* that is achievable by an adversary. The motivation for this is Theorem 4.2 in [Don+22a]. It provides an extractor for protocols with a particular structure (so-called commit-and-open IDS), and proves an extraction error bound that is linear in the naive cheating bound.

Definition 8 (Computational and query-bounded prefix-conditioned $\mathfrak{s}$-soundness). *Let Π be a $(2\ell + 1)$-round IDS and let $\mathfrak{s}' : T_{<m+1} \to 2^{(2^{\mathcal{C}_{\geq m+1}})}$ a soundness function.*[5] *For an adversary A, define the $\mathfrak{s}$-*shaping advantage

$$\mathrm{Adv}^{\Pi}_{\mathfrak{s}-shaping}(\mathsf{A}) = \mathbb{E}_{\mathbf{t} \leftarrow \langle \mathsf{A}, \mathsf{Vrf} \rangle}[p_{naive}^{\mathfrak{s}(\mathbf{t}_{<m+1})}].$$

Here, we slightly abuse notation by writing $\mathbf{t} \leftarrow \mathsf{A}$ to mean that A produces the transcript $\mathbf{t}$ by interacting with the (honest) verifier. We say Π is computationally (query-bounded) $\mathfrak{s}$-sound if the shaping advantage $\mathrm{Adv}^{\Pi}_{\mathfrak{s}-shaping}(\mathsf{A})$ is negligible for all polynomial-time (polynomial-query) adversaries A.

We note that a computationally bounded A is query-bounded by definition. Looking ahead, we will use this notion for partially Fiat-Shamir-transformed, or *round-eliminated*, protocols. Here, the challenges for a number of initial rounds have already been replaced by random oracle outputs (and are therefore absorbed in the first commitment message. The remainder of the protocol is still interactive. In this case, the adversary can *in principle* choose a first commitment such that the hashes yield unlikely challenges which map to much lower levels of special soundness under $\mathfrak{s}$. The extreme case is where $\mathfrak{s}$ maps to $\emptyset$ and extraction becomes impossible. Such first commitments can, however, only be found by completing an *infeasible* search problem with respect to the random oracle. It follows that query-bounded adversaries can only produce such a bad first commitment with small probability.

We next define a function $p_{\mathfrak{s}} : T_{<m} \longrightarrow [0, 1]$ to capture this expected naive cheating probability associated with $\mathfrak{s}$. It will later allow us to conveniently collect all prefixes $\mathbf{t}_{<m}$ that map to the same expected naive cheating probability. For a $(2\ell + 1)$-round identification protocol Π with prefix-conditioned $\mathfrak{s}$-soundness function $\mathfrak{s} : T_{\leq m} \to 2^{(2^{\mathcal{C}_{>m}})}$, define

$$\begin{aligned} p_{\mathfrak{s}} : T_{<m} &\longrightarrow [0, 1] \\ \mathbf{t}_{<m} &\longmapsto \mathbb{E}_{\mathsf{c}_m \in \mathcal{C}_m}\left[p_{naive}^{\mathfrak{s}(\mathbf{t}_{<m}, \mathsf{c}_m)}\right]. \end{aligned} \tag{3}$$

We also define the corresponding variance

$$\begin{aligned} v_{\mathfrak{s}} : T_{<m} &\longrightarrow [0, 1] \\ \mathbf{t}_{<m} &\longmapsto \mathrm{Var}_{\mathsf{c}_m \in \mathcal{C}_m}\left[p_{naive}^{\mathfrak{s}(\mathbf{t}_{<m}, \mathsf{c}_m)}\right]. \end{aligned} \tag{4}$$

Like the soundness function $\mathfrak{s}'$ in Def. 8, this function maps transcripts ending with the m-th commitment, unlike $\mathfrak{s}$ which maps transcripts ending with the m-th challenge. In Lemma 1, they

[5] As we regard the $(m+1)-th$ commitment as a form of set-up for this definition, the soundness function reflects a slightly different partition of the protocol rounds compared to the soundness function in Def. 7

play a role in bounding the adversary success probability for protocols with the first m-challenge rounds eliminated.

Our techniques apply to MPCitH protocols, eliminating the first challenge round to reduce a 5-round MPCitH protocol to a 3-round Σ-protocol. In this context, the transcript prefix consists of only the initial commitment which can be chosen freely by the adversary. The expected naive cheating bound for the optimal choice is therefore

$$\max_{\mathsf{w}_1 \in W_1} \mathbb{E}_{\mathsf{c}_1 \in \mathcal{C}_1} \left[p_{naive}^{s(\mathsf{w}_1,\mathsf{c}_1)} \right]. \tag{5}$$

This quantity is estimated carefully in many MPCitH protocols, sometimes under the name ”false positive probability”, to facilitate ad-hoc security proofs. These estimates of false positive probability from MPCitH design are readily translated by our technique to quantify soundness of the round-eliminated 3-round protocol.

2.2 Eliminating verifier challenge interactions in the eQROM.

We next describe the syntax of recursively eliminating one round of verifier interaction at a time, where one of the challenges is determined by a random oracle (instead of the verifier drawing it). The most important case in our context is when the first challenge is determined by a random oracle, as follows.

Definition 9 (First round elimination Π_1). *Let Π be a $(2\ell+1)$-round protocol with commitment and challenge spaces $W_1 \times W_2 \times \ldots \times W_\ell$ and $\mathcal{C}_1 \times \mathcal{C}_2 \times \ldots \times \mathcal{C}_\ell$. The first round elimination Π_1 of Π is the $(2\ell-1)$-round protocol whose commit, respond and verify functions are defined as follows. Let $\mathsf{RO} : W_1 \longrightarrow \mathcal{C}_1$ be a random oracle. The first commitment $\mathsf{Commit}_{\Pi_1,1}$ of Π_1 is defined as:*

$\mathsf{Commit}_{\Pi_1,1}(\mathsf{sk})$:

1. $\mathsf{w}'_{1,1} \leftarrow \mathsf{Commit}_{\Pi,1}(\mathsf{sk})$
2. $\tilde{\mathsf{c}} = \mathsf{RO}(\mathsf{w}'_{1,1})$
3. $\mathsf{w}'_{1,2} \leftarrow \mathsf{Commit}_{\Pi,2}(\mathsf{sk}, \tilde{\mathsf{c}}))$
4. *Output* $\mathsf{w}'_1 := (\mathsf{w}'_{1,1}, \mathsf{w}'_{1,2})$.

The remainder (that is, $i > 2$) of the commitment functions are advanced as

$$\mathsf{Commit}_{\Pi_1,i-1}(\mathsf{sk}, \mathsf{c}'_1, \mathsf{c}'_2, \ldots, \mathsf{c}'_{i-2}) := \mathsf{Commit}_{\Pi,i}(\mathsf{sk}, \tilde{\mathsf{c}}, \mathsf{c}'_1, \mathsf{c}'_2, \ldots, \mathsf{c}'_{i-2}),$$

$$\mathsf{Resp}_{\Pi_1}(\mathsf{sk}, \mathsf{c}'_1, \mathsf{c}'_2, \ldots, \mathsf{c}'_{\ell-1}) := \mathsf{Resp}_{\Pi}(\mathsf{sk}, \tilde{\mathsf{c}}, \mathsf{c}'_1, \mathsf{c}'_2, \ldots, \mathsf{c}'_{\ell-1}).$$

To verify a transcript $\mathbf{t}' = (\mathsf{w}'_1, \mathsf{c}'_1, \mathsf{w}'_2, \ldots, \mathsf{w}'_{\ell-1}, \mathsf{c}'_{\ell-1}, \mathsf{z}')$ of Π_1, Vrf_{Π_1} checks (using Vrf_{Π} and access to RO) the predicate

$$\mathsf{Vrf}_{\Pi}(\mathsf{pk}, \mathsf{w}'_{1,1}, \mathsf{RO}(\mathsf{w}'_{1,1}), \mathsf{w}'_{1,2}, \mathsf{c}'_1, \mathsf{w}'_2, \mathsf{c}'_2, \ldots, \mathsf{w}'_{\ell-1}, \mathsf{c}'_{\ell-1}, \mathsf{z}').$$

Here $\mathsf{w}'_1 = (\mathsf{w}'_{1,1}, \mathsf{w}'_{1,2}) \in W_1 \times W_2$ is the partitioning of the first commitment, as apparent from the definition of the first commitment function.

We can recurse this process to eliminate up to a chosen m-th challenge round. We next define notation to describe the resulting protocol. When eliminating more than one round, the security proofs require the use of a new random oracle (denoted RO_i in Def. 10) per elimination. In practice, these can be instantiated with a single hash function and domain separation.

From here on, it helps to distinguish the transcripts of Π from those of its round-eliminated version Π_m. To this end, as in Def. 10, the transcripts of Π_m will be denoted (using a superscript $'$) as $\mathbf{t}' = (\mathsf{w}'_1, \mathsf{c}'_1, \mathsf{w}'_2, \mathsf{c}'_2, \ldots, \mathsf{w}'_{\ell-m}, \mathsf{c}'_{\ell-m}, \mathsf{z}')$. We remind the reader that we assume all prover functions to implicitly share state which is necessary for the following definition to be meaningful.

Definition 10 (Multiple round elimination $\varPi_m$). *Let $\varPi$ be a $(2\ell+1)$-round protocol with commitment and challenge spaces $W_1 \times W_2 \times \ldots \times W_\ell$ and $\mathcal{C}_1 \times \mathcal{C}_2 \times \ldots \times \mathcal{C}_\ell$. For an $m \in \{1, 2, \ldots, \ell\}$, the m-th challenge round elimination $\varPi_m$ of $\varPi$ is the $(2(\ell - m)+1)$-round protocol with commit, respond and verify functions defined as follows. Let $\mathsf{RO} = (\mathsf{RO}_i : W_1 \times W_2 \times \ldots \times W_i \longrightarrow \mathcal{C}_i)_{i=1}^m$ be a sequence of independent random oracles. The first commitment of $\varPi_m$ is defined as*

```
$\underline{\mathsf{Commit}_{\varPi_m,1}(\mathsf{sk}) :}$
      1.  $\mathsf{w}'_{1,1} \leftarrow \mathsf{Commit}_{\varPi,1}(\mathsf{sk})$
      2.  $\tilde{\mathsf{c}}_1 = \mathsf{RO}_1(\mathsf{w}'_{1,1})$
      3.  $\mathsf{w}_{1,2} \leftarrow \mathsf{Commit}_{\varPi,2}(\mathsf{sk}, \tilde{\mathsf{c}}_1))$
      4.  $\tilde{\mathsf{c}}_2 = \mathsf{RO}_2(\mathsf{w}'_{1,1}, \mathsf{w}'_{1,2})$
          $\vdots$
   2m-1.  $\tilde{\mathsf{c}}_m = \mathsf{RO}_m(\mathsf{w}'_{1,1}, \mathsf{w}'_{1,2}, \ldots, \mathsf{w}'_{1,m})$
     2m.  $\mathsf{w}'_{1,m} \leftarrow \mathsf{Commit}_{\varPi_m}(\mathsf{sk}, \tilde{\mathsf{c}}_1, \tilde{\mathsf{c}}_2, \ldots, \tilde{\mathsf{c}}_m)$
   2m+1.  Output $\mathsf{w}'_1 := (\mathsf{w}'_{1,1}, \mathsf{w}'_{1,2}, \ldots, \mathsf{w}'_{1,m+1})$
```

This defines the expanded transcript prefix

$$\overline{\mathsf{w}'_{1_{<m}}} := (\mathsf{w}'_{1,1}, \mathsf{RO}_1(\mathsf{w}'_{1,1}), \mathsf{w}'_{1,2}, \mathsf{RO}_2(\mathsf{w}'_{1,1}, \mathsf{w}'_{1,2}), \ldots,$$
$$\mathsf{RO}_m(\mathsf{w}'_{1,1}, \mathsf{w}'_{1,2}, \ldots, \mathsf{w}'_{1,m})).$$

The remainder of the protocol $\varPi^m$ is the same as $\varPi$, except that the expanded transcript is used, i.e.,

$$\mathsf{Commit}_{\varPi_m, i-m}(\mathsf{sk}, \mathsf{c}'_1, \mathsf{c}'_2, \ldots, \mathsf{c}'_{i-1}) := \mathsf{Commit}_{\varPi,i}(\mathsf{sk}, \mathbf{c}\Big(\overline{\mathsf{w}'_{1_{<m}}}\Big), \mathsf{c}'_1, \mathsf{c}'_2, \ldots, \mathsf{c}'_{i-1}),$$
$$\mathsf{Resp}_{\varPi_m}(\mathsf{sk}, \mathsf{c}'_1, \mathsf{c}'_2, \ldots, \mathsf{c}'_{\ell-m}) := \mathsf{Resp}_{\varPi}(\mathsf{sk}, \mathbf{c}\Big(\overline{\mathsf{w}'_{1_{<m}}}\Big), \mathsf{c}'_1, \ldots, \mathsf{c}'_{\ell-m}), \textit{ and}$$
$$\mathsf{Vrf}_{\varPi^m}(\mathsf{pk}, \mathbf{c}\Big(\overline{\mathsf{w}'_{1_{<m}}}\Big), \mathsf{w}'_{1,m+1}, \mathsf{c}'_1, \mathsf{w}'_2, \mathsf{c}'_2, \ldots, \mathsf{c}'_{\ell-m}, \mathsf{z}').$$

Here $\mathsf{w}'_1 = (\mathsf{w}'_{1,1}, \mathsf{w}'_{1,2}, \ldots, \mathsf{w}'_{1,m+1}) \in W_1 \times W_2 \times \ldots \times W_{m+1}$ is the partitioning of the first commitment, as apparent in the first commitment function definition, and we recall that

$$\mathbf{c}\Big(\overline{\mathsf{w}'_{1_{<m}}}\Big) = (\mathsf{RO}_1(\mathsf{w}'_{1,1}), \mathsf{RO}_2(\mathsf{w}'_{1,1}, \mathsf{w}'_{1,2}), \ldots, \mathsf{RO}_m(\mathsf{w}'_{1,1}, \mathsf{w}'_{1,2}, \ldots, \mathsf{w}'_{1,m})).$$

We are now ready to prove a bound on the shaping advantage of any q-query adversary $\mathsf{A}_q^{\mathsf{RO}}$ against $\varPi_m$, proving that $\varPi_m$ is query-bounded prefix-conditioned $\mathfrak{s}$-sound.

Lemma 1 (Shaping advantage bound for $\varPi_m$). *Let $\varPi$ be a $(2\ell+1)$-round identification protocol with prefix-conditioned $\mathfrak{s}$-soundness function $\mathfrak{s} : T_{\leq m} \to 2^{(2^{\mathcal{C}_{>m}})}$ and the corresponding $p_{\mathfrak{s}}$-probability function $p_{\mathfrak{s}} : T_{<m} \longrightarrow [0,1]$. Let $\varPi_m$ be the round-eliminated version of $\varPi$ with the first m-challenges eliminated using a random oracle $\mathsf{RO} = (\mathsf{RO}_1, \mathsf{RO}_2, \ldots, \mathsf{RO}_m)$, as in Def. 10. Let $C := 304$, $\mu_{\mathfrak{s}} := \max_{\mathbf{t}_{<m} \in T_{<m}} p_{\mathfrak{s}}(\mathbf{t}_{<m})$ and $\sigma_{\mathfrak{s}} := \max_{\mathbf{t}_{<m} \in T_{<m}} \sqrt{v_{\mathfrak{s}}(\mathbf{t}_{<m})}$. The $\mathfrak{s}'$-shaping advantage of a quantum adversary $\mathsf{A}_q^{\mathsf{RO}}$ against $\varPi_m$, making at most q queries to RO in the eQROM is bounded as*

$$\mathrm{Adv}^{\varPi_m}_{\mathfrak{s}'-shaping}\left(\mathsf{A}_q^{\mathsf{RO}}\right) \leq \mu_{\mathfrak{s}} + 3\sqrt{C} q \sigma_{\mathfrak{s}} + 2Cq^2\sigma_{\mathfrak{s}}^2 \mu_{\mathfrak{s}} \log\left(\frac{1}{\sqrt{C} q \sigma_{\mathfrak{s}}}\right),$$

where $\mathfrak{s}'^{\mathsf{RO}} : W_1 \times W_2 \times \ldots \times W_m \to 2^{(2^{\mathcal{C}_{>m}})}$ is defined as $\mathfrak{s}'^{\mathsf{RO}}(\mathsf{w}'_1) = \mathfrak{s}(\overline{\mathsf{w}'_1})$.

Proof. Let A^{RO} be a quantum adversary against Π_m. We wish to bound the expectation value

$$\mathrm{Adv}^{\Pi^m}_{\mathfrak{s}-shaping}(\mathsf{A}) = \mathbb{E}_{\mathbf{t}\leftarrow\mathsf{A}^{\mathsf{RO}}}[p_{naive}^{\mathfrak{s}'(\mathbf{t}_{\leq m})}]$$

We can bound this maximum expectation by applying [HHM22, Cor. 4], which builds on the online extractability in the QROM results of [Don$^+$22b]. We begin by briefly paraphrasing [HHM22, Cor. 4]. Let $f : X \times Y \to [0,1]$ be a function where X and Y are finite non-empty sets. Let $\widehat{\mathsf{RO}} : X \to Y$ be a random oracle. Consider the maximization problem for an adversary with eQROM access, making at most q queries, to find an input $x \in X$ such that $f(x, \widehat{\mathsf{RO}}(x))$ is large. The expected value of $f(x, \widehat{\mathsf{RO}}(x))$ that such an adversary can achieve is at most

$$\mathbb{E}_x f(x, \widehat{\mathsf{RO}}(x)) \leq \mu + 3\sqrt{C}q\sigma + 2Cq^2\sigma^2\mu \log\left(\frac{1}{\sqrt{C}q\sigma}\right),$$

where (μ, σ^2) upper bound the mean $\mathbb{E}_y(f(x,y)) \leq \mu$ and variance $\mathrm{Var}_y(f(x,y)) \leq \sigma^2$ of the function values taken over the uniform distribution on Y. Applying [HHM22, Cor. 4] to the function

$$\begin{aligned} \tilde{p}_{\mathfrak{s}} : T_{<m} \times \mathcal{C}_m &\longrightarrow [0,1] \\ (\mathbf{t}_{<m}, \mathsf{c}_m) &\longmapsto p_{\mathrm{naive}}^{\mathfrak{s}((\mathbf{t}_{<m}, \mathsf{c}_m))} \end{aligned}$$

(from Eq. (3)) implies

$$\begin{aligned} \mathbb{E}_{\mathbf{t}\leftarrow\mathsf{A}^{\mathsf{RO}}}[p_{naive}^{\mathfrak{s}'(\mathbf{t}_{\leq m})}] =& \mathbb{E}_{\mathbf{t}\leftarrow\mathsf{A}^{\mathsf{RO}}}[p_{naive}^{\mathfrak{s}(\overline{\mathsf{w}_1})}] \\ =& \mathbb{E}_{\mathbf{t}\leftarrow\mathsf{A}^{\mathsf{RO}}}[p_{\mathfrak{s}}(\overline{\mathsf{w}_1}_{<m}, \mathsf{RO}_m(w_{1,m}))] \\ \leq& \mu_{\mathfrak{s}} + 3\sqrt{C}q\sigma_{\mathfrak{s}} + 2Cq^2\sigma_{\mathfrak{s}}^2\mu_{\mathfrak{s}} \log\left(\frac{1}{\sqrt{C}q\sigma_{\mathfrak{s}}}\right) \end{aligned}$$

proving the lemma. □

We now provide two corollaries for the special case of parallel repetition of 5-round IDS. The corollaries do not exploit the full power of the framework, but are sufficient for our example application to RYDE.

Corollary 1. *Let Π be a 5-round identification protocol with τ-splittable challenges and soundness function*

$$\mathfrak{s}(\mathbf{t}_{\leq 1}) = \begin{cases} \mathfrak{S}_{\mathsf{IDS}_{\tau,d}}, & \mathbf{t}_{\leq 1} \in G \\ \emptyset, & \mathbf{t}_{\leq 1} \notin G \end{cases}$$

for some subset $G \subseteq W_1 \times \mathcal{C}_1$. Let $c_{\bar{G}} := \max_{\mathsf{w}_1 \in W_1} |\{\mathsf{c}_1 \mid (\mathsf{w}_1, \mathsf{c}_1) \notin G\}|/|\mathcal{C}_1|$. Let Π_1 be the (Σ-protocol with τ-splittable challenges resulting from the) first challenge round elimination of Π. The $\mathfrak{s}'$-shaping advantage of a quantum adversary A^{RO} against the first challenge round elimination Π_1 of Π, making at most q queries to RO in the eQROM is bounded as

$$\mathrm{Adv}_{\mathfrak{s}'-shaping}\left(\mathsf{A}^{\mathsf{RO}}\right) \leq \mu_{\mathfrak{s}} + 3\sqrt{C}q\sigma_{\mathfrak{s}} + 2Cq^2\sigma_{\mathfrak{s}}^2\mu_{\mathfrak{s}} \log\left(\frac{1}{\sqrt{C}q\sigma_{\mathfrak{s}}}\right) \tag{6}$$

where $C := 304$,

$$\sigma_{\mathfrak{s}} = \sqrt{\left(p_{naive}^{\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}}\right)^2 (1 - c_{\bar{G}}) + c_{\bar{G}}} \leq p_{naive}^{\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}} + \sqrt{c_{\bar{G}}},$$

$$\mu_{\mathfrak{s}} = p_{naive}^{\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}}(1 - c_{\bar{G}}) + c_{\bar{G}} \leq p_{naive}^{\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}} + c_{\bar{G}},$$

$p_{naive}^{\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}}$ was defined in Eq. (2), and $\mathfrak{s}'$ is defined in terms of $\mathfrak{s}$ and RO as in Lemma 1.

Proof. Specializing Lemma 1 to the $(\ell = 2, m = 1)$ case, the corollary holds with

$$\mu_{\mathfrak{s}} := \max_{\mathsf{w}_1 \in W_1} \mathbb{E}_{\mathsf{c}_1 \in \mathcal{C}_1}\left[p_{\text{naive}}^{\mathfrak{s}((\mathsf{w}_1,\mathsf{c}_1))}\right] \quad \text{and} \quad \sigma_{\mathfrak{s}} := \max_{\mathsf{w}_1 \in W_1} \sqrt{\mathrm{Var}_{\mathsf{c}_1 \in \mathcal{C}_1}\left[p_{\text{naive}}^{\mathfrak{s}((\mathsf{w}_1,\mathsf{c}_1))}\right]}.$$

The expressions for $\mu_{\mathfrak{s}}$ and $\sigma_{\mathfrak{s}}$ are derived using Eq. (2) and a straightforward optimization. □

We can reverse-engineer the query-bounded distance-d special soundness defined in Def. 3.

Corollary 2. *Let Π and Π_1 be as in Cor. 1. Then Π_1 has query-bounded distance-d special soundness. More precisely, the advantage of any distance-d special soudness adversary* A *is bounded as*

$$\mathrm{Adv}_{\mathsf{IDS},\mathsf{Ext}}^{d-\mathsf{spS}}(\mathsf{A}) \leq \frac{\Gamma - p_{naive}^{\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}}}{1 - p_{naive}^{\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}}} \overset{(*)}{\leq} \Gamma,$$

where Γ is the right hand side of Eq. (6) and the inequality $(*)$ holds if $\Gamma < 1$.

Proof. From any distance-d special soundness adversary A we can construct a shaping adversary A' as follows. A' runs $(\mathbf{t}, \hat{\mathbf{t}}) \leftarrow \mathsf{A}$ and outputs $\mathbf{t} = (\mathsf{w}, \mathsf{c}, \mathsf{z})$. When A succeeds, the extractor fails on two transcripts that should be extractable for the distance-d special soundness extractor, so we can conclude that in this case $\mathfrak{s}'(\mathsf{w}) = \emptyset$. It follows that

$$\begin{aligned}\mathrm{Adv}_{\mathfrak{s}'-shaping}\left(\mathsf{A}'^{\mathsf{RO}}\right) =& \mathrm{Adv}_{\mathsf{IDS},\mathsf{Ext}}^{d-\mathsf{spS}}(\mathsf{A}) + (1 - \mathrm{Adv}_{\mathsf{IDS},\mathsf{Ext}}^{d-\mathsf{spS}}(\mathsf{A}))p_{naive}^{\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}} \\ =& p_{naive}^{\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}} + \mathrm{Adv}_{\mathsf{IDS},\mathsf{Ext}}^{d-\mathsf{spS}}(\mathsf{A})\left(1 - p_{naive}^{\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}}\right)\end{aligned}$$

Rearranging the equation and applying Cor. 1 yields the desired bound. □

It remains to be shown that the HVZK property is preserved under round-elimination. The argument for this is straightforward, following the blueprint of the UF-NMA to UF-CMA reduction for Fiat-Shamir transformed identification schemes. The idea is that a transcript for the fully interactive IDS can be turned into a valid transcript of IDS where some challenges are taken as the output of a random oracle, by reprogramming the random oracle. Given that the proof contains not much novelty, it can be found in Appendix B.

Lemma 2 (R-HVZK of round elimination). *Let IDS_{-1} be the IDS that is obtained by applying round elimination to* IDS *using random oracle* RO*. If* IDS *has first message entropy $\gamma_{\mathsf{w}} := \mathbb{E}\max_{\mathsf{w}_1} \Pr[\mathsf{w}_1]$ Then it holds for any adversary* A *against the $R-\mathsf{HVZK}$ property of IDS_{-1} that makes q_{H} queries to* RO*, there exists an adversary* B *against $R-\mathsf{HVZK}$ of* IDS *with*

$$\mathrm{Adv}_{\mathsf{IDS}_{-1}}^{R-\mathsf{hvzk}}(\mathsf{A}) \leq \mathrm{Adv}_{\mathsf{IDS}}^{R-\mathsf{hvzk}}(\mathsf{B}) + \frac{3R}{2}\sqrt{(q_{\mathsf{H}} + R) \cdot \gamma_{\mathsf{w}}}.$$

3 The Hypercube Technique

In [Agu+23b], the hypercube technique was introduced as an optimization for the SDitH code-based signature scheme. It allowed to reduce the computational cost and communication size of the MPCitH-based IDS of SDitH while preserving its security and especially its soundness error. In this section we show that the hypercube technique can be applied to a more general class of MPC in the Head (MPCitH) based identification schemes (and consequently the signature schemes derived of them) as long as the used MPC protocol fits our abstraction, and is additively homomorphic, as defined below. We start with preliminaries and abstractions that we use. Afterwards, we describe the hypercube technique, and finally we prove when it applies. We note that in this section we present the results for single instance IDS (in contrast to parallel repetition IDS) for the sake of readability. However, all results can easily be extended to the parallel repetition of IDS equivalently, applying the techniques per instance.

3.1 Preliminaries

We now provide the necessary background on MPC in the head (MPCitH), and the MPC computation it uses. We start with the definition of pseudorandom generators and $\mathsf{TreePRG}$ which are used in optimizations of MPCitH-based identification schemes also discussed below. For the latter we focus on the functional properties as we do not make use of them in our security proofs. Afterwards, we provide definitions and abstractions for MPC and MPCitH that we use in this section. Many functions in this section work over sets $\{X_i\}_{i\in[N^D]}$ with index set $[N^D]$. Due to space constraints, we often omit the index set information for the case $[N^D]$ and write $\{X_i\}$ (or $\{X_i\}_{i\neq c}$ for $\{X_i\}_{i\in[N^D]}^{i\neq c}$).

PRG. A pseudorandom generator (PRG) is an efficiently computable function $\mathsf{PRG} : \{0,1\}^n \to \{0,1\}^{en}$ where e is the expansion factor. Security of a PRG is defined in terms of a real-or-random game. The advantage of a possibly quantum adversary A is defined as

$$\mathrm{Adv}^{\mathsf{ror}}_{\mathsf{PRG}}(\mathsf{A}) := \left|\Pr[x \leftarrow \{0,1\}^{en} : 1 \leftarrow \mathsf{A}(x)] - \Pr[x \leftarrow \{0,1\}^n : 1 \leftarrow \mathsf{A}(\mathsf{PRG}(x))]\right|.$$

TreePRG. In this work we make use of the $\mathsf{TreePRG}$ construction initially proposed by Goldreich, Goldwasser, and Micali [GGM84]. $\mathsf{TreePRG}$ makes use of a length-doubling PRG ($e = 2$) to reach an expansion factor of $e = 2^\lambda$ building a binary tree of height λ. The root of the tree is the input and the leaves are the outputs Out. To build the tree, every inner node Node is fed to PRG to generate its two child nodes. One strength of $\mathsf{TreePRG}$ is that λ inner nodes are enough to generate all but one leaf. Let $\mathsf{Out_i}$ denote the i^{th} leaf / output block of $\mathsf{TreePRG}$. Then the sibling path from the root to that leaf suffices to generate all Out but $\mathsf{Out_i}$. The interface of $\mathsf{TreePRG}$ is captured by the following three routines:

$\mathsf{GenLeaves}(\rho) \to \{\mathsf{Out_i}\}_{i\in[2^\lambda]}$: Given root ρ generates all leaves.
$\mathsf{GenPath}(\rho, c) \to \mathsf{path}$: Given the root, generates the sibling path for the cth leaf.
$\mathsf{LeavesFromPath}(\mathsf{path}) \to \{\mathsf{Out_i}\}_{i\in[2^\lambda-1]}$: Given a path for leaf c, generates the leaves for all but the cth leaf.

It fulfills the correctness condition

$$\mathsf{LeavesFromPath}(\mathsf{GenPath}(\rho, c)) = \mathsf{GenLeaves}(\rho) \setminus \{\mathsf{Out}_c\}$$

In general, we require that any $\mathsf{Out_i}$ is pseudorandom even when given the output of $\mathsf{GenLeaves}(\rho, i)$. In [AM+23] an even stronger security definition is given and proven: For a possibly quantum adversary A we define the advantage against $\mathsf{TreePRG}$ as

$$\begin{aligned}\mathrm{Adv}^{\mathsf{ror}}_{\mathsf{TreePRG}}(\mathsf{A}) := \Big|&\Pr[\{x_j\}_{j\in[\lambda]} \leftarrow (\{0,1\}^n)^{\lambda+1} : 1 \leftarrow \mathsf{A}(\{x_j\}_{j\in[\lambda]})] \\ &- \Pr[x, \mathsf{Out_i} \leftarrow \{0,1\}^n : 1 \leftarrow \mathsf{A}(\mathsf{GenPath}(x, i), \mathsf{Out_i}))]\Big|.\end{aligned}$$

MPC. In this work we are working on tooling that builds an IDS from a MPC protocol. We assume an N-party MPC protocol that is $N-1$ private in the semi-honest model, i.e., if any $N-1$ parties collude, they cannot learn anything about the input of the last remaining party.

Moreover, we assume perfect correctness: If all parties behave honestly, the correct result is computed.

The MPC protocol is used to privately evaluate a function F on some secret input x. For that, x is secret shared using an additively homomorphic secret sharing scheme which produces N shares $[\![x]\!]_i$ such that $x = \sum_{i=0}^{N-1}[\![x]\!]_i$. In addition to $[\![x]\!]_i$, the initial state of a party may contain further inputs. All this is subsumed in the party input $\mathtt{in}_i$. Moreover, an MPC protocol might take global auxiliary inputs like the random values used for multiplication triple verification following [LN17].

To compute non-linear functions F, the MPC protocol requires communication. We consider MPC with broadcast communication, i.e., every party broadcasts their messages to every other party. Upon receipt the functions continue their computation. This can go in multiple rounds. In the end, every party broadcasts their output share. As we are simulating the MPC computation, we capture all this by a function $\mathtt{MPC}_F(\{\mathtt{in}_i\}_{i=0}^{N-1}, \mathsf{xtra}) = \{\mathtt{bc}_i\}_{i=0}^{N-1}$ that takes as input the initial state of every party $\mathtt{in}_i$, as well as potential auxiliary inputs xtra (e.g., masking values for product verification), and returns the set of communications of all parties, i.e., $\mathtt{bc}_i$ is all communication broadcasted by party i. We omit the subscript F indicating the function implemented by the protocol where it is clear from the context. We overload notation, introducing as an alternative interface $\mathtt{MPC}(j, \{\mathtt{in}_i\}_{i\neq j,i=0}^{N-1}, \mathtt{bc}_j, \mathsf{xtra}) = \{\mathtt{bc}_i\}_{i=0}^{N-1}$ where the state of a party is replaced by its communications such that

$$\mathtt{MPC}\left(j, \{\mathtt{in}_i\}_{i=0;i\neq j}^{N-1}, \left(\mathtt{MPC}\left(\{\mathtt{in}_i\}_{i=0}^{N-1}\right), \mathsf{xtra}\right)_j, \mathsf{xtra}\right) = \mathtt{MPC}\left(\{\mathtt{in}_i\}_{i=0}^{N-1}, \mathsf{xtra}\right).$$

In this work we are interested in a setting where x is a secret value such that $F(x)$ fulfills some conditions. (E.g., the common case is $F(x) = 0$.) We model this via a predicate $\mathtt{Pred}(\{\mathtt{bc}_i\}_{i=0}^{N-1}) = b$ which takes as input all communication of the MPC computation and checks for the condition (in the example case that the outputs of all parties form a secret sharing of 0 and therefore sum up to 0).

Additively homomorphic MPC. We require the MPC protocol as well as the predicate to be additively homomorphic on all its inputs. We capture this by the following definition.

Definition 11 (Additively homomorphic MPC). *Let* $\mathtt{MPC}$ *denote the execution of an MPC protocol as described above, with communications checking predicate* $\mathtt{Pred}$*. We call* $\mathtt{MPC}$ *and* $\mathtt{Pred}$ *additively homomorphic if they fulfill the following two properties.*

1. *Order independent. For any permutation π of $[0, \ldots, N-1]$ we have*
$$\mathtt{MPC}\left(\left\{\mathtt{in}_{\pi(i)}\right\}_{i=0}^{N-1}, \mathit{xtra}\right) = \pi\left(\mathtt{MPC}\left(\{\mathtt{in}_i\}_{i=0}^{N-1}, \mathit{xtra}\right)\right), \tag{7}$$
where we use the notation $\pi(\{\mathtt{bc}_i\}_{i=0}^{N-1}) = \{\mathtt{bc}_{\pi(i)}\}_{i=0}^{N-1}$.
2. *Additively homomorphic. Let $\mathtt{in} = \{\mathtt{in}_i\}_{i=0}^{N}$ and $\mathtt{in}' = \{\mathtt{in}'_i\}_{i=0}^{N-1}$ such that $\mathtt{in}_i = \mathtt{in}'_i$ for $0 \leq i < N-1$ and $\mathtt{in}_{N-1} + \mathtt{in}_N = \mathtt{in}'_{N-1}$ then we have*
$$\begin{aligned}\left(\mathtt{MPC}\left(\left\{\mathtt{in}'_i\right\}_{i=0}^{N-1}, \mathit{xtra}\right)\right)_{N-1} &= \\ \left(\mathtt{MPC}\left(\{\mathtt{in}_i\}_{i=0}^{N}, \mathit{xtra}\right)\right)_{N-1} &+ \left(\mathtt{MPC}\left(\{\mathtt{in}_i\}_{i=0}^{N}, \mathit{xtra}\right)\right)_{N}, \textit{ and}\end{aligned} \tag{8}$$
$$\mathtt{Pred}\left(\mathtt{MPC}\left(\left\{\mathtt{in}'_i\right\}_{i=0}^{N-1}, \mathit{xtra}\right)\right) = \mathtt{Pred}\left(\mathtt{MPC}\left(\{\mathtt{in}_i\}_{i=0}^{N}, \mathit{xtra}\right)\right). \tag{9}$$

We note that for additively homomorphic schemes, there exists a variant of $\mathtt{MPC}(j, \{\mathtt{in}_i\}_{i\neq j,i=0}^{N-1}, \mathtt{bc}_j, \mathsf{xtra})$ that takes $\mathtt{bc}_\Sigma = \sum_{i\in N}\mathtt{bc}_i$ in place of $\mathtt{bc}_j$. This works, for the following reason. The communications of a party $\mathtt{bc}_i$ are a vector of the communications send in different rounds of the MPC protocol. Accordingly, $\mathtt{bc}$ is the vector of the sums of the communications of all parties for these different rounds. Consequently, the implementation of $\mathtt{MPC}$ can, round-by-round, compute the communications of all parties but party j and then compute that of party j as the difference of the total sum and the sum of all the parties but j. To avoid additional notation, we distinguish this variant only by the given input writing $\mathtt{MPC}(j, \{\mathtt{in}_i\}_{i \neq j, i=0}^{N-1}, \mathtt{bc}_\Sigma, \mathtt{xtra})$.

MPC-in-the-Head. First proposed in [Ish+07], the idea of MPC-in-the-Head (MPCitH) is to build a zero-knowledge proof for an arbitrary NP statement via the simulation of an N-party-MPC protocol to compute the evaluation circuit for the relation. In our case we aim for a slightly weaker goal of an identification scheme that provides HVZK and special soundness. The approach remains mostly the same but we prove knowledge of an input to a one-way function that produces a predetermined output. Here, the secret input becomes the secret key $\mathsf{sk} = x$, while the one-way function is defined by the public key pk. We first describe the high level idea, before we discuss generic optimizations and implementation details considered in this work by default. In MPCitH, the Prover P prepares the inputs for and simulates the execution of all parties of the MPC protocol given secret x and randomness ρ. This follows the description of the MPC protocol above. The prover generates the inputs $\{\mathtt{in}\}_{i=0}^{N-1}$ (including the secret sharing of x) for all parties and commits to each of them individually. In addition, the prover generates all auxiliary inputs xtra necessary. We abstract this by a function $\mathtt{GenXtra}(x, \rho)$ that gets all inputs of the P and thereby implicitly all values computed by the P so far. Then P runs the MPC protocol $\mathtt{MPC}_{\mathsf{pk}}(\{\mathtt{in}_i\}_{i=0}^{N-1}, \mathsf{xtra}) = \{\mathtt{bc}_i\}_{i=0}^{N-1}$ to obtain the communications of all parties. The prover sends the commitments together with all communications $\{\mathtt{bc}_i\}_{i=0}^{N-1}$ to the verifier V. In response, V samples a random value $\mathsf{c} \overset{\$}{\leftarrow} [0, \ldots, N-1]$ and sends it as challenge to P. To complete the protocol, P sends the opening information for all commitments except that for party c as response z to V. This allows V to compute the inputs $\{\mathtt{in}'_i\}_{i=0; i \neq \mathsf{c}}^{N-1}$ for all parties but party c as part of verify. Then V can run $\mathtt{MPC}_{\mathsf{pk}}(\mathsf{c}, \{\mathtt{in}'_i\}_{i \neq \mathsf{c}, i=0}^{N-1}, \mathtt{bc}_{\mathsf{c}}, \mathsf{xtra}) = \{\mathtt{bc}'_i\}_{i=0}^{N-1}$ and check consistency of all communications with the ones sent earlier, i.e., check that $(\forall 0 \leq i < N; i \neq \mathsf{c})\mathtt{bc}'_i = \mathtt{bc}_i$. Then V can evaluate $\mathtt{Pred}\left(\{\mathtt{bc}'_i\}_{i=0}^{N-1}\right)$ and accept whenever the consistency checks out and the predicate is fulfilled.

We do not formally prove security here as we start with the assumption of a secure MPCitH identification scheme but we provide a sketch. HVZK follows from the MPC protocol being $N-1$ private in the semi-honest model: If the commitment is hiding, the verifier does not learn anything about the inputs of party c. In a proof this allows the simulator to pick the challenge before everything else and then choose the communications of party c such that $\mathtt{Pred}$ is fulfilled.

Special soundness follows from the correctness of the MPC protocol (and the bindingness of the commitments). If there are two valid transcripts for the same commitments and communications, this means that all commitments get opened (to the same values in the different transcripts due to bindingness). This means that all communication and computation gets verified and therefore the MPC protocol was honestly simulated if both transcripts accept. Moreover, acceptance also implies that the result of the computation fulfills the required predicate. Hence, the extractor can simply recombine the shares of x found in the inputs to recover an x such that $F(x)$ fulfills $\mathtt{Pred}$.

Optimizations / Abstractions. We consider one optimization that we generally assume to be used and hide a second one behind an abstraction. First, our proof requires that the commitment is implemented via a hash function so that we can later model it as a random oracle. To commit to a value y_i, P computes $\mathsf{com} = H(y_i \| r)$ using hash function H and a fresh

Abstract functions

$\mathtt{GenSds}(x, N, \rho) \to (\{(\mathsf{seed}_i, r_i)\}_{i\in[N]}, \mathsf{aux})$ // Seed generation

$\mathtt{ExpSd}(\{\mathsf{seed}_i\}_{i\in[N]}, \mathsf{aux}) \to \{(\mathtt{in}_i)\}_{i\in[N]}$ // Seed expansion

$\mathtt{GenOpn}(x, N, \rho, \mathsf{c}) \to \mathsf{opnng}$ // Generate opening information

$\mathtt{Opn}(\mathsf{opnng}) \to (\{(\mathsf{seed}_i, r_i)\}_{i\neq\mathsf{c}, i\in[N]})$ // Open from opening information

$\mathsf{Com}(x, \rho) \to \mathsf{com}$ // Commitment function

$\mathtt{GenXtra}(x, \rho) \to \mathsf{xtra}$ // Generate auxiliary MPC input

$\mathtt{MPC}(\{\mathtt{in}_i\}_{i\in[N]}, \mathsf{xtra}) \to \{\mathsf{bc}_i\}_{i\in[N]}$ // Run MPC protocol

$\mathtt{MPC}(j, \{\mathtt{in}_i\}_{i\neq j, i\in[N]}, \mathsf{bc}_j, \mathsf{xtra}) \to \{\mathsf{bc}_i\}_{i\in[N]}$ // Run MPC protocol

$\mathtt{MPC}(j, \{\mathtt{in}_i\}_{i\neq j, i\in[N]}, \mathsf{bc}_\Sigma, \mathsf{xtra}) \to \{\mathsf{bc}_i\}_{i\in[N]}$ // Run MPC protocol

$\mathtt{Pred}(\{\mathsf{bc}_i\}_{i\in[N]}) \to b$ // Output predicate

Fig. 1: List of used abstractions

random value r. To open the commitment, P simply reveals y_i, r. Note that opening in this case requires V to actually recompute $H(y_i \| r)$ using the given values and comparing the result to com to ensure that this was indeed the correct opening.

The second optimization is about compressing the transcript replacing random values by the outcome of a PRG and only sending the used seeds. This considers the following sort of optimizations: Given that commonly P generates the party inputs $\mathtt{in}_i$ for the first $N-1$ parties at random and only picks the last party's input such that the inputs form a valid secret sharing, one can generate the first $N-1$ $\mathtt{in}$ from one seed each. Because P always has to open all but one party in the setting we consider, we can get down to logarithmic size opening information using a structure like $\mathsf{TreePRG}$ (which can be viewed as punctured PRF). More precisely, we use $\mathsf{TreePRG}$ to generate N seeds seed'_i from each of which we generate a pair (seed_i, r_i) such that seed_i gets later expanded into $\mathtt{in}_i$ and r_i is used as commitment randomness to commit to seed_i (which has the same binding properties as committing to $\mathtt{in}_i$ directly, while security of the PRG guarantees hiding). Then we can open all but one commitment, using the sibling path of the unopened seed. Given that our proofs bootstrap from the security of a basescheme that already uses these routines, they are independent of the precise realization. Only for the functional description do we need the following routines:

The function $\{(\mathtt{in}_i)\}_{i\in[N-1]} \leftarrow \mathtt{ExpSd}(\{\mathsf{seed}_i\}_{i\in[N-1]}, \mathsf{aux})$ is a deterministic function that takes a set of values seed_i as well as some optional auxiliary data (e.g., the delta of the last party inputs from a random value) and maps each to the corresponding input value $\mathtt{in}_i$. (This can be the identity function or a PRG which finally adds aux to $\mathtt{in}_{N-1}$.)

The function $(\{(\mathsf{seed}_i, r_i)\}_{i\in[N-1]}, \mathsf{aux}) \leftarrow \mathtt{GenSds}(x, N, \rho)$ takes as input the secret x, the number of parties N, and randomness ρ, and generates a set of inputs to $\mathtt{ExpSd}$ together with the corresponding commitment randomness per input. The function is deterministic as it takes the required randomness as explicit input.

The pair of functions $\mathsf{opnng} \leftarrow \mathtt{GenOpn}(x, N, \rho, \mathsf{c})$ and $(\{(\mathsf{seed}_i, r_i)\}^{i\neq\mathsf{c}}_{i\in[N-1]}) \leftarrow \mathtt{Opn}(\mathsf{opnng})$ such that

$$\mathtt{Opn}(\mathtt{GenOpn}(x, N, \rho, \mathsf{c})) = \mathtt{GenSds}(x, N, \rho)_1 \setminus \{(\mathsf{seed}_\mathsf{c}, r_\mathsf{c})\},$$

i.e., given the same inputs as $\mathtt{GenSds}$, the sequence of $\mathtt{GenOpn}$ and $\mathtt{Opn}$ generates all seed, randomness pairs except the one of party c.

A summary of all the abstractions we use can be found in Fig. 1.

```
Agg (N, j, $\{X_i\}$)
1 : Write $i$ in radix $N$: $(i_0, \dots, i_{D-1})$
2 : for $k \in [N]$ do
3 :     $X_k^j = \sum_{i \in [N^D], i_j = k} X_{i=(i_0,\dots,i_{D-1})}$
4 : return $\{X_k^j\}_{k \in [N]}$
```

```
Agg (N, j, $\{X_i\}$, c)
1 : Write $i$ in radix $N$: $(i_0, \dots, i_{D-1})$
2 : Write $\mathsf{c}$ in radix $N$: $(\mathsf{c}_0, \dots, \mathsf{c}_{D-1})$
3 : for $k \in [N], k \neq \mathsf{c}_j$ do
4 :     $X_k^j = \sum_{i \in [N^D], i_j = k} X_{i=(i_0,\dots,i_{D-1})}$
5 : return $\{X_k^j\}_{k \in [N]}^{k \neq \mathsf{c}_j}$
```

Fig. 2: Aggregation routine to produce the main party inputs or communications for the partitioning of parties according to dimension j. The right side gives the variant that aggregates all but one main party for which the necessary information is missing.

3.2 From flat to hypercube

We now discuss how to apply the hypercube technique to any MPCitH-based IDS that follows our abstraction above, and uses an additively homomorphic MPC protocol. From now on we use IDS to refer to MPCitH-based IDS. An IDS that uses N^D parties to achieve soundness error $(N^D)^{-1}$ needs to simulate the computation of N^D parties and send the communications of all of these. Using the hypercube technique, the computation of only DN parties is required, and consequently also only the communication of these. This is achieved by simulating several instances of the MPC protocol with what [Agu+23b] called *main parties*. These main parties are obtained by accumulating parties of the *flat* IDS which [Agu+23b] called *leaf parties* after partitioning them.

This aggregation is at the heart of the hypercube technique and depicted in Fig. 2. The Agg function is used in the hypercube IDS to aggregate the inputs and communications of parties. One run of Agg generates the aggregated values for one run of the MPC protocol. The name *hypercube* comes from how the partitioning for the aggregation is obtained. Starting from an IDS that used N^D parties, one may think of them as being arranged on a D dimensional hypercube. The position of a party i can be obtained by writing its index i in radix N as $(i_0, \dots, i_{D-1})$ (see line 1 of Fig. 2), e.g., for $N = 2$ this would refer to writing it in binary. Then i_j is the coordinate of i in dimension j. The hypercube technique runs the MPC protocol D times. For the jth protocol run, parties are partitioned according to their coordinate in dimension j (line 3 of left side of Fig. 2). The necessary aggregation for the j run is obtained by running $\mathtt{Agg}(N, j, \cdot)$.

A flat IDS with N^D parties is turned into its hypercube version by applying the aggregation and running the MPC protocol D times with the respective main parties. A side-by-side comparison of the two versions of an IDS is given in Fig. 3. We note that the protocols are identical except for the hypercube running D MPC computations with aggregation before. Notably, this means that the commitments are still on representatives of the leaf parties and we have N^D commitments in place of ND. While this is a disadvantage for the plain IDS described here, it allows for later optimizations. Namely, in the signature scheme, all commitments that get opened do not have to be sent. At the same time, using TreePRG to implement GenSds, the opening information can be reduced to $D \log_2 N$ seeds, in place of $DN - 1$ when using random main parties. Moreover, one of the parties in a secret sharing is not random and therefore cannot be compressed. When using D independent runs with N parties each, D uncompressed parties are necessary, of which at least half have to be opened and thereby communicated (there are different ways how this last party is communicated but all require the transmition of the same amount of uncompressed opening information).

Flat MPCitH

```
P.Commit($x, \rho$)
  $(\{(\mathsf{seed}_i, r_i)\}, \mathsf{aux}) \leftarrow \mathtt{GenSds}(x, N^D, \rho)$
  $\{(\mathtt{in}_i)\} \leftarrow \mathtt{ExpSd}(\{\mathsf{seed}_i\}, \mathsf{aux})$
  for $i \in [N^D]$ do
    $\mathsf{com}_i \leftarrow \mathsf{Com}(\mathsf{seed}_i)$
  $\mathsf{w} \leftarrow \{\mathsf{com}_i, r_i\}$
  $\mathsf{xtra} \leftarrow \mathtt{GenXtra}(x, \rho)$
  $\{\mathsf{bc}_i\} \leftarrow \mathtt{MPC}_{\mathsf{pk}}(\{\mathtt{in}_i\}, \mathsf{xtra})$
  return $\mathsf{w}, \{\mathsf{bc}_i\}, \mathsf{xtra}, \mathsf{aux}$

V.Challenge($\mathsf{w}, \{\mathsf{bc}_i\}, \mathsf{xtra}, \mathsf{aux}$)
  $\mathsf{c} \xleftarrow{\$} [N^D]$
  return $\mathsf{c}$

P.Resp($x, \rho, \mathsf{c}$)
  $\mathsf{z} \leftarrow \mathtt{GenOpn}(x, N^D, \rho, \mathsf{c})$
  return $\mathsf{z}$

Trans = ($\mathsf{w}$, $\{\mathsf{bc}_i\}$, $\mathsf{xtra}$, $\mathsf{aux}$, $\mathsf{c}$, $\mathsf{z}$)
```

Hypercube MPCitH

```
P.Commit($x, \rho$)
  $(\{(\mathsf{seed}_i, r_i)\}, \mathsf{aux}) \leftarrow \mathtt{GenSds}(x, N^D, \rho)$
  $\{(\mathtt{in}_i)\} \leftarrow \mathtt{ExpSd}(\{\mathsf{seed}_i\}, \mathsf{aux})$
  for $i \in [N^D]$ do
    $\mathsf{com}_i \leftarrow \mathsf{Com}(\mathsf{seed}_i)$
  $\mathsf{w} \leftarrow \{\mathsf{com}_i, r_i\}$
  $\mathsf{xtra} \leftarrow \mathtt{GenXtra}(x, \rho)$
  for $j \in [D]$ do
    $\{\mathtt{in}_i^j\}_{i \in [N]} \leftarrow \mathsf{Agg}(N, j, \{\mathtt{in}_i\})$
    $\{\mathsf{bc}_i^j\}_{i \in [N]} \leftarrow \mathtt{MPC}_{\mathsf{pk}}(\{\mathtt{in}_i^j\}_{i \in [N]}, \mathsf{xtra})$
  return $\mathsf{w}, \{\mathsf{bc}_i^j\}_{i,j \in [N] \times [D]}, \mathsf{xtra}, \mathsf{aux}$

V.Challenge($\mathsf{w}, \{\mathsf{bc}_i^j\}_{i,j \in [N] \times [D]}, \mathsf{xtra}, \mathsf{aux}$)
  $\mathsf{c} \xleftarrow{\$} [N^D]$
  return $\mathsf{c}$

P.Resp($x, \rho, \mathsf{c}$)
  $\mathsf{z} \leftarrow \mathtt{GenOpn}(x, N^D, \rho, \mathsf{c})$
  return $\mathsf{z}$

Trans = ($\mathsf{w}$, $\{\mathsf{bc}_i^j\}_{i,j \in [N] \times [D]}$, $\mathsf{xtra}$, $\mathsf{aux}$, $\mathsf{c}$, $\mathsf{z}$)
```

Fig. 3: Abstract three-round (flat) MPCitH IDS and hypercube version of the same IDS.

Verification of both variants is depicted side-by-side in Fig. 4. Verification consists of computing the openings, verifying the commitments, and afterwards checking the broadcast communications, including a check that they fulfill the predicate Pred. We externalized the two verification steps into functions VerCom and VerBC. Given that the only difference between flat and hypercube version is in the MPC computations, the whole first part of verification, including opening, commitment verification, and seed expansion is identical. Only verification of the broadcast communication is different. While there is only communication of one MPC run in the flat case, verification for the hypercube scheme has to aggregate the information to assemble the right inputs for each of the D MPC runs and verify each individually.

Translation of a flat to a hypercube IDS is done by first matching the scheme to our abstraction and checking the necessary homomorphic properties. Afterwards, the hypercube version is given by assembling the abstract functions according to the right-hand-side of Fig. 3. We provide an example of the process applying to RYDE in Sec. 4.

3.3 Security of the Hypercube technique

In this section we prove that the security of the hypercube version H-IDS of a flat three-round IDS IDS follows directly from the security IDS. More precisely we show that HVZK and soundness are preserved. At the heart of the argument are two translation routines depicted in Fig. 5 which translate flat transcripts into hypercube transcripts and vice versa. Either translation only corrects the broadcast communications, everything else remains the same. The translation from flat to hypercube transcripts simply aggregates the communications according to the hypercube. The translation from hypercube to flat transcripts is slightly more involved. This is the place

```
$\mathsf{Vrf}(\mathsf{pk}, \mathsf{w}, \{\mathsf{bc}_i\}, \mathsf{xtra}, \mathsf{aux}, \mathsf{c}, \mathsf{z})$
Parse $\{\mathsf{com}_i\} \leftarrow \mathsf{w}$
$(\{(\mathsf{seed}_i, r_i)\}_{i \neq \mathsf{c}}) \leftarrow \mathtt{Opn}(\mathsf{z})$
$a = \mathtt{VerCom}(\mathsf{c}, \{\mathsf{seed}_i, r_i\}_{i \neq \mathsf{c}}, \{\mathsf{com}_i\})$
$\{\mathtt{in}_i\}_{i \neq \mathsf{c}} \leftarrow \mathtt{ExpSd}(\{\mathsf{seed}_i\}, \mathsf{aux})$
$b = \mathtt{VerBC}(\mathsf{pk}, \mathsf{c}, \{\mathtt{in}_i\}_{i \neq \mathsf{c}}, \{\mathsf{bc}_i\}, \mathsf{xtra})$
return $a \wedge b$
```

```
$\mathsf{VrfHyp}(\mathsf{pk}, \mathsf{w}, \{\mathsf{bc}_i^j\}_{i,j \in [N] \times [D]}, \mathsf{xtra}, \mathsf{aux}, \mathsf{c}, \mathsf{z})$
Parse $\{\mathsf{com}_i\} \leftarrow \mathsf{w}$
$(\{(\mathsf{seed}_i, r_i)\}_{i \neq \mathsf{c}}) \leftarrow \mathtt{Opn}(\mathsf{z})$
$a = \mathtt{VerCom}(\mathsf{c}, \{\mathsf{seed}_i, r_i\}_{i \neq \mathsf{c}}, \{\mathsf{com}_i\})$
$\{\mathtt{in}_i\}_{i \neq \mathsf{c}} \leftarrow \mathtt{ExpSd}(\{\mathsf{seed}_i\}, \mathsf{aux})$
Write $\mathsf{c}$ in radix $N$: $(\mathsf{c}_0, \ldots, \mathsf{c}_{D-1})$
for $j \in [D]$ do
    $\{\mathtt{in}_i^j\}_{i \in [N]}^{i \neq \mathsf{c}_j} \leftarrow \mathsf{Agg}(N, j, \{\mathtt{in}_i\}_{i \neq \mathsf{c}}, \mathsf{c})$
    $\{\mathsf{bc}_i^j\}_{i \in [N]} \leftarrow \mathsf{Agg}(N, j, \{\mathsf{bc}_i\})$
    $b_k = \mathtt{VerBC}(\mathsf{pk}, \mathsf{c}_j, \{\mathtt{in}_i\}_{i \in [N]}^{i \neq \mathsf{c}_j},$
        $\{\mathsf{bc}_i^j\}_{i \in [N]}, \mathsf{xtra})$
return $a \wedge b_0 \wedge \ldots \wedge b_{D-1}$
```

```
$\mathtt{VerCom}\ (\mathsf{c},\ \{\mathsf{seed}_i, r_i\}_{i \neq \mathsf{c}},\ \{\mathsf{com}_i\})$
for $i \neq \mathsf{c}$ do
    $\mathsf{com}_i' \leftarrow \mathsf{Com}(\mathsf{seed}_i, r_i)$
    if $\mathsf{com}_i' \neq \mathsf{com}_i$ then return $0$
return $1$
```

```
$\mathtt{VerBC}\ (\mathsf{pk},\ \mathsf{c},\ \{\mathtt{in}_i\}_{i \in [N]}^{i \neq \mathsf{c}},\ \{\mathsf{bc}_i\}_{i \in [N]},\ \mathsf{xtra})$
$\{\mathsf{bc}_i'\}_{i \in [N]} \leftarrow \mathtt{MPC}_{\mathsf{pk}}(\mathsf{c}, \{\mathtt{in}_i\}_{i \in [N]}^{i \neq \mathsf{c}}, \mathsf{bc}_{\mathsf{c}}, \mathsf{xtra})$
for $i \neq \mathsf{c}$ do
    if $\mathsf{bc}_i' \neq \mathsf{bc}_i$ then return $0$
return $\mathtt{Pred}(\{\mathsf{bc}\}_{i \in [N]})$
```

Fig. 4: Verification of an MPCitH IDS, and the hypercube version of it including shared subroutines.

```
$\mathtt{TransH2F}(\mathsf{w}, \{\mathsf{bc}_i^j\}_{i,j \in [N] \times [D]}, \mathsf{xtra}, \mathsf{aux}, \mathsf{c}, \mathsf{z})$
$(\{(\mathsf{seed}_i, r_i)\}_{i \in [N^D]}^{i \neq \mathsf{c}}) \leftarrow \mathtt{Opn}(\mathsf{z})$
$\{\mathtt{in}_i\}_{i \neq \mathsf{c}} \leftarrow \mathtt{ExpSd}(\{\mathsf{seed}_i\}, \mathsf{aux})$
Compute $\mathsf{bc}_\Sigma = \sum_{i \in N} \mathsf{bc}_i^0$
$\{\mathsf{bc}_i\} \leftarrow \mathtt{MPC}_{\mathsf{pk}}(\mathsf{c}, \{\mathtt{in}_i\}_{i \neq \mathsf{c}}, \mathsf{bc}_\Sigma, \mathsf{xtra})$
return $(\mathsf{w}, \{\mathsf{bc}_i\}, \mathsf{xtra}, \mathsf{aux}, \mathsf{c}, \mathsf{z})$
```

```
$\mathtt{TransF2H}(\mathsf{w}, \{\mathsf{bc}_i\}, \mathsf{xtra}, \mathsf{aux}, \mathsf{c}, \mathsf{z})$
for $j \in [D]$ do
    $\{\mathsf{bc}_i^j\}_{i \in [N]} \leftarrow \mathsf{Agg}(N, j, \{\mathsf{bc}_i\})$
return $(\mathsf{w}, \{\mathsf{bc}_i^j\}_{i,j \in [N] \times [D]}, \mathsf{xtra}, \mathsf{aux}, \mathsf{c}, \mathsf{z})$
```

Fig. 5: Given a hypercube transcript, TransH2F describes the construction of a flat transcript, and vice versa for TransF2H.

where we require that the partial MPC computation can be done with the sum of all communications in place of the communication of the missing party (c.f., Def. 11 and the note following it). From the opening, we get the inputs for all but one party and again by the homomorphic properties, we can take the communications of any of the D MPC runs to compute the sum of the communications also for the flat scheme. With these, we can run $\mathtt{MPC}_{\mathsf{pk}}$ to compute all communications for the flat transcript.

We begin with soundness. For soundness preservation, we show that we can use any extractor for IDS to extract for H-IDS. This is done translating any given H-IDS transcripts using TransH2F into IDS transcripts. For this work, TransH2F has to preserve validity of transcripts, which is the first thing that we prove before giving the actual soundness proof:

Lemma 3. *Let* TransH2F *be as in Fig. 5,* $t^H = (\mathsf{w}, \{\mathsf{bc}_i^j\}_{i,j \in [N] \times [D]}, \mathit{xtra}, \mathit{aux}, \mathsf{c}, \mathsf{z})$ *be a transcript for the hypercube IDS, and* $t = (\mathsf{w}, \{\mathsf{bc}_i\}, \mathit{xtra}, \mathit{aux}, \mathsf{c}, \mathsf{z})$ *be a flat transcript derived from it running*

$\texttt{TransH2F}(t^H)$. *Then*

$$\mathsf{V.Vrf}(t^H) = 1 \Rightarrow \mathsf{V.VrfHyp}(t) = 1.$$

Proof of Lemma 3. We give a proof by contradiction, showing that whenever verification of t fails, also verification of t^H will fail, and thereby the implication has to hold. Verification of the flat transcript $\mathsf{V.Vrf}(t)$ is displayed in the left hand side of Fig. 4. The event that $\mathsf{V.Vrf}(t) = 0$ only occurs if at least one of the two subroutines, `VerCom` and `VerBC`, returns 0. This can only be triggered in line 3 of `VerCom`, or in lines 3 or 4 of `VerBC`. We now step through each of these cases and argue that each will also cause $\mathsf{V.VrfHyp}(t^H) = 0$.

Failed commitment (`VerCom`*, line 3):* First the verification function checks the commitments via a call to `VerCom` in line 3 of either verification routine. This step is identical for the flat and the hypercube setting. Note that `TransH2F` leaves everything untouched except the communications. Moreover, the first four lines of verification are identical for both cases. In consequence, the inputs given to `VerCom`, prepared in lines 1 and 2 of either verification are identical. Hence the result of the call must be too. So if there is a mismatch in line 3 of `VerCom` in one case, the same mismatch occurs in the other case.

Failed communications - consistency (`VerBC`*, line 3):* Next the verification procedure expands the seeds to obtain the inputs for all but the challenge party and runs `VerBC`. First `VerBC` computes the communications $\mathtt{bc}'_i$ for all parties from the information given (line 1), and then checks consistency with the previously transmitted communication $\mathtt{bc}_i$ for $i \neq \mathsf{c}$. We now argue that by construction of t, this check never fails. The inputs to $\mathsf{MPC}_{\mathsf{pk}}$ used in `VerBC` are identical to those used in `TransH2F` with one exception: In `VerBC`, $\mathtt{bc}_{\mathsf{c}}$ is used while `TransH2F` used $\mathtt{bc}_{\Sigma}$. However, $\mathtt{bc}_{\mathsf{c}}$ is the output of that computation and therefore by definition consistent with it. Hence, this step simply cannot cause verification to fail.

Failed communications - failed predicate (`VerBC`*, line 4):* Finally, `VerBC` applies the predicate to the communication. It is crucial to note that by iterated application of Eq. (8), we get that $\mathtt{Pred}\{\mathtt{bc}_i\} = \mathtt{Pred}\mathtt{bc}_\sigma$, where $\mathtt{bc}_\sigma = \sum_{i\in[N^D]} \mathtt{bc}_i$. But by construction we also have $\mathtt{bc}_\sigma = \sum_{i\in[N]\mathtt{bc}_i^0}$ for the communications of the first set of main party communications of the hypercube scheme, as the flat scheme communications where computed that way by `TransH2F`. And finally, we may also iteratively apply Eq. (8) to conclude $\mathtt{Pred}\{\mathtt{bc}_i^0\} = \mathtt{Pred}\mathtt{bc}_\sigma$ and thereby $\mathtt{Pred}\{\mathtt{bc}_i\} = \mathtt{Pred}\{\mathtt{bc}_i^0\}$.

Hence, we have shown that in every case that makes $\mathsf{V.Vrf}(t)$ fail, also $\mathsf{V.VrfHyp}(t^H)$ will fail concluding the proof. □

We now use this result to argue that query-bounded distance d special-soundness (Def. 3) of the flat scheme implies query-bounded distance-d soundness of its hypercube version with the same adversarial advantage. We do this showing that we can use the extractor of the flat scheme to build an extractor for the hypercube version.

Lemma 4 (Soundness preservation). *Let* IDS *be a flat MPCitH-based IDS as depicted in Fig. 3 that uses an additively homomorphic MPC protocol as defined in Def. 11, and* $\mathsf{H\text{-}IDS}$ *the hypercube version of* IDS*. For any extractor* Ext *for* IDS*, and any adversary* A *against the query-bounded distance-d special soundness of* $\mathsf{H\text{-}IDS}$*, there exist an extractor* $\mathsf{H\text{-}IDS}$ *and an adversary* B *against the query-bounded distance-d special soundness of* IDS *such that*

$$\mathrm{Adv}^{d-\mathsf{spS}}_{\mathsf{H\text{-}IDS},\mathsf{H\text{-}Ext}}(\mathsf{A}) = \mathrm{Adv}^{d-\mathsf{spS}}_{\mathsf{IDS},\mathsf{Ext}}(\mathsf{B}),$$

with $\mathsf{TIME}(\mathsf{B}) = \mathsf{TIME}(\mathsf{A}) + \mathsf{TIME}(\texttt{TransH2F}) \approx \mathsf{TIME}(\mathsf{A})$ *and* $\mathsf{TIME}(\mathsf{H\text{-}Ext}) = \mathsf{TIME}(\mathsf{Ext}) + \mathsf{TIME}(\texttt{TransH2F}) \approx \mathsf{TIME}(\mathsf{Ext})$*. Moreover, the number of queries that* B *makes to the random oracle is that of* A *plus the number of queries made by an honest verification.*

Proof of Lemma 4. To prove the statement, we construct $\mathsf{H\text{-}Ext}$ using Ext. Our strategy is based on the previous result. Extractor $\mathsf{H\text{-}Ext}$ takes any hypercube transcripts t^H it is given, translates them to flat transcripts t using $\texttt{TransH2F}$ (Fig. 5), runs Ext on these, and outputs the result. For this to work, three conditions have to hold: First, the translated transcripts have to still fulfill the challenge pattern required for Ext to succeed. Second, the translated transcripts have to be valid. Third, the translation has to be efficient. If these hold, the part of $\mathsf{H\text{-}Ext}$ that translates the messages can be viewed as adversary B and therefore, the probability that Ext does not successfully extract is exactly the claimed advantage.

The first condition trivially holds as both schemes have special soundness with respect to the same challenge pattern and $\texttt{TransH2F}$ does not touch c. The second condition holds according to Lemma 3 which we just proved. The third condition can be verified by inspection. Extractor $\mathsf{H\text{-}Ext}$ takes two transcripts. Therefore, the only overhead compared to Ext are two calls to $\texttt{TransH2F}$. The translation has about the same complexity as Vrf without $\texttt{VerCom}$. After computing the MPC inputs from z, it computes a sum over N communications and then runs $\texttt{MPC}_{\mathsf{pk}}$. While it is a different variant of $\texttt{MPC}$ than the one used by Vrf, the overhead is limited to one sum per broadcast communication round of $\texttt{MPC}$ which is negligible compared to any cryptographically relevant attack effort. Given that the only difference to Vrf is that the role of the sum of communications and the challenge party communications are switched, the translation adds at most as many RO queries as Vrf makes.

In conclusion, whenever the inputs to $\mathsf{H\text{-}Ext}$ fulfill the requirements for extraction, also the inputs that it feeds to Ext do. Therefore, $\mathsf{H\text{-}Ext}$ succeeds with the same success probability as Ext.

□

Lemma 5 (HVZK preservation). *Let* IDS *be a flat MPCitH-based IDS as depicted in Fig. 3 that uses an additively homomorphic MPC protocol as defined in Def. 11, and* $\mathsf{H\text{-}IDS}$ *the hypercube version of* IDS. *Then it holds for any adversary* A *against the* HVZK *property of* $\mathsf{H\text{-}IDS}$ *that there exists an adversary* B *against* HVZK *of* IDS *with*

$$\mathrm{Adv}^{\mathsf{hvzk}}_{\mathsf{H\text{-}IDS}}(\mathsf{A}) = \mathrm{Adv}^{\mathsf{hvzk}}_{\mathsf{IDS}}(\mathsf{B}) .$$

Proof. Our proof strategy is similar to that for soundness. We show how to construct a simulator $\mathsf{H\text{-}Sim}$ for the hypercube scheme using the simulator for IDS that has to exist by assumption. For this we make use of $\texttt{TransF2H}$ to translate the flat transcripts t to hypercube transcripts t^H. I.e., we use $\mathsf{H\text{-}Sim} := \texttt{TransF2H} \circ \mathsf{Sim}$. For this to work, we have to show that any adversary that can distinguish the outputs of $\mathsf{H\text{-}Sim}$ from honest transcripts, can be used to distinguish the outputs of Sim from honest transcripts.

We give a proof by reduction. Given adversary A against HVZK of $\mathsf{H\text{-}IDS}$, we construct reduction B against HVZK of IDS as follows. On input the public key pk, B runs $\mathsf{A}^{\texttt{TransF2H}(t)}(\mathsf{pk})$, and outputs the result. Let $\mathsf{Trans}(\mathsf{sk})$, and $\mathsf{H\text{-}Trans}(\mathsf{sk})$ refer to algorithms that execute IDS and $\mathsf{H\text{-}IDS}$, respectively, and output the resulting transcript. By definition, we have

$$\mathrm{Adv}^{\mathsf{hvzk}}_{\mathsf{IDS}}(\mathsf{B}) = |\Pr[1 \leftarrow \mathsf{B}^{\mathsf{Sim}(\mathsf{pk})}(\mathsf{pk})] - \Pr[1 \leftarrow \mathsf{B}^{\mathsf{Trans}(\mathsf{sk})}(\mathsf{pk})]|,$$

where the probabilities are taken over the randomness involved in generating $(\mathsf{sk}, \mathsf{pk}) \overset{\$}{\leftarrow} \mathsf{Keygen}()$, as well as the coins of A, Sim, and Trans. Plugging in the definition of B, this becomes

$$\mathrm{Adv}^{\mathsf{hvzk}}_{\mathsf{IDS}}(\mathsf{B}) = |\Pr[1 \leftarrow \mathsf{A}^{\texttt{TransF2H}(\mathsf{Sim}(\mathsf{pk}))}(\mathsf{pk})] - \Pr[1 \leftarrow \mathsf{A}^{\texttt{TransF2H}(\mathsf{Trans}(\mathsf{sk}))}(\mathsf{pk})]|.$$

By construction, we have that $\texttt{TransF2H}(\mathsf{Sim}(\mathsf{pk})) = \mathsf{H\text{-}Sim}(\mathsf{pk})$. If we are able to show that $\texttt{TransF2H}(\mathsf{Trans}(\mathsf{sk})) \approx \mathsf{H\text{-}Trans}(\mathsf{sk})$, i.e., the output distribution of $\texttt{TransF2H}$, given honest

IDS transcripts, is identical to that of honest transcripts for $\mathsf{H\text{-}IDS}$, then the above becomes $\mathrm{Adv}^{\mathsf{hvzk}}_{\mathsf{H\text{-}IDS}}(\mathsf{A})$.

Note that when comparing the algorithms in Fig. 3, they are identical except for the way the broadcast communications are computed in line 7 of the flat Commit and lines 7-9 of the hypercube commit. At the same time, $\mathtt{TransF2H}$ only changes these broadcast communications. Now, this means that $\mathsf{H\text{-}Trans}$ and $\mathtt{TransF2H} \circ \mathsf{Trans}$ start from the same MPC inputs $\mathtt{in}_i$. For an honest hypercube run, $\mathsf{H\text{-}Trans}$ generates the $\mathtt{bc}_i^j$ by first aggregating the inputs and then executing the MPC protocol for each hypercube dimension. In contrast, $\mathtt{TransF2H} \circ \mathsf{Trans}$ first runs the full MPC protocol and then aggregates the generated communications for each hypercube dimension to obtain $\mathtt{bc}_i^j$. However, because $\mathtt{MPC}$ is additively homomorphic, these computations give the identical result (one only has to iteratively apply Eq. (8)). Thereby, the distribution of outputs in both cases is identical.

Applying this to the above equation, we get

$$\begin{aligned}\mathrm{Adv}^{\mathsf{hvzk}}_{\mathsf{IDS}}(\mathsf{B}) &= |\Pr[1 \leftarrow \mathsf{A}^{\mathsf{H\text{-}Sim(pk)}}(\mathsf{pk})] - \Pr[1 \leftarrow \mathsf{A}^{\mathsf{H\text{-}Trans(sk)}}(\mathsf{pk})]| \\ &= \mathrm{Adv}^{\mathsf{hvzk}}_{\mathsf{H\text{-}IDS}}(\mathsf{A})\,.\end{aligned}$$

The runtime of $\mathsf{H\text{-}Sim}$ is essentially that of Sim, only adding D calls to $\mathtt{Agg}$, each of which is running N sums with D terms. □

4 RYDE scheme

In this section we exemplify the application of our results on round reduction and the hypercube technique, applying them to a five-round IDS using flat MPCitH that is similar to the RYDE signature scheme [Ara+23b]. RYDE is a contender in the NIST on-ramp for signature schemes. At its core is a five round identification scheme that follows the BN-style MPCitH approach. The underlying one-way problem it is based on is Rank Syndrome Decoding. The signature scheme is obtained by applying the Fiat-Shamir transform for five round identification schemes. We note that in the NIST submission, the authors already introduce the hypercube optimization. Our example shows how easily one can obtain a QROM proof for an optimized version of this class of signature schemes.

Mapping the items in Algorithm 1 into the transcript items for the abstractions used in the previous section, we show how to transform a flat RYDE-like IDS into a hypercube one following Fig. 3. We present the basic five-round flat IDS that RYDE is built around, even though in the specification [Ara+23b] only the Fiat Shamir-transformed hypercube signature scheme running τ protocols in parallel is presented. Previously we used N as the side length of the hypercube, but here we fix the side length as 2, which is optimal in almost all cases, and use notation that is similar to the original description of RYDE for the purpose of the example, in particular emulating the interactive version of Algorithm 1 in the RYDE specification. Firstly, we let N, the number of parties in the flat scheme shown in Algorithm 1, be a power of 2, so $N = 2^D$.

Lines 1-6 generate the seeds and inputs for all N parties, and these are captured in our abstractions by the functions $\mathtt{ExpSd}, \mathtt{GenSds}$ as demonstrated in Fig. 3. Included in the setup is the commitment to each party's seeds after line 2 using commitment scheme Com. The first message σ_1 in line 7 of Algorithm 1 should also be the same as for the flat scheme.

4.1 Round collapse

Next comes the first challenge in the five-round setting, where V generates and sends to P a challenge that defines masking points for Beaver triplets (ϵ) and evaluation points for a polyno-

Algorithm 1 RYDE: Five round IDS from Rank Syndrome Decoding with flat MPCitH

```
Input: $\mathbf{H}, \mathbf{x}, \mathbf{y}, \boldsymbol{\beta}$: $\mathbf{y} = \mathbf{H}(\mathbf{x}) \in \mathbb{F}_{q^m}^{n-k}$. Define $\mathbf{x}_B$ as the last $k$ coordinates of $\mathbf{x}$.

  Round 1 (Setup for MPC protocol):
  1. $\rho \xleftarrow{\$} \{0,1\}^{2\lambda}, \mathsf{root} \xleftarrow{\$} \{0,1\}^{\lambda}$
  2. $\mathsf{seed}_1, r_1, \ldots, \mathsf{seed}_N, r_N \leftarrow \mathsf{GenLeaves}(\mathsf{root})$
  for $i \in [N]$ do
      $\llbracket \mathbf{x}_B \rrbracket_i, \llbracket \boldsymbol{\beta} \rrbracket_i, \llbracket \mathbf{a} \rrbracket_i, \llbracket c \rrbracket_i \leftarrow \mathsf{PRG}(\mathsf{seed}_i, r_i)$
      $\mathsf{com}_i \leftarrow \mathsf{Com}(\mathsf{seed}_i, r_i)$
  4. $\mathit{\Delta}\mathbf{a} \leftarrow \mathsf{SampleFqmVector}(\mathsf{seed}_{N-1})$, $\mathbf{a} \leftarrow \sum_i \llbracket \mathbf{a}_i \rrbracket_{i \in [N]}$, $\mathit{\Delta}c \leftarrow -\langle \mathbf{a}, \boldsymbol{\beta} \rangle - \sum_{i=0}^{N-2} \llbracket c \rrbracket_i$
  5. $\mathit{\Delta}\mathbf{x}_B \leftarrow \mathbf{x}_B - \sum_{i=0}^{N-2} \mathbf{x}_B, \mathit{\Delta}\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} - \sum_{i=0}^{N-2} \llbracket \boldsymbol{\beta} \rrbracket_i$
  6. $\llbracket \mathbf{x}_B \rrbracket_{N-1} \leftarrow \mathit{\Delta}\mathbf{x}_B$, $\llbracket \boldsymbol{\beta} \rrbracket_{N-1} \leftarrow \mathit{\Delta}\boldsymbol{\beta}$, $\llbracket c \rrbracket_{N-1} \leftarrow \mathit{\Delta}c$        ▷ $\mathsf{aux} = (\mathit{\Delta}\mathbf{x}_B, \mathit{\Delta}\boldsymbol{\beta}, \mathit{\Delta}c)$
  7. $\sigma_1 \leftarrow (\{\mathsf{com}_i\}_{i \in [N]}, (\mathit{\Delta}\mathbf{x}_B, \mathit{\Delta}\boldsymbol{\beta}, \mathit{\Delta}c))$
  8. $\mathsf{P}\colon \sigma_1 \rightarrow \mathsf{V}$        ▷ $\{\mathtt{in}_i\} = (\llbracket \mathbf{x}_B \rrbracket_i, \llbracket \boldsymbol{\beta} \rrbracket_i, \llbracket \mathbf{a} \rrbracket_i, \llbracket c \rrbracket_i)_{i \in [N]}$
  Round 2 (First Challenge):
  9. $\mathsf{P} \xleftarrow{\$} \mathsf{c}_1 = (\boldsymbol{\gamma}, \boldsymbol{\epsilon})\colon \mathsf{V}$        ▷ $\mathsf{xtra} = (\boldsymbol{\gamma}, \boldsymbol{\epsilon})$
  Round 3 (Simulation of checking protocols):
  10. $(\llbracket \boldsymbol{\alpha} \rrbracket_i, \llbracket v \rrbracket_i)_{i \in [N]} \leftarrow \mathtt{MPC}_N((\llbracket \mathbf{x}_B \rrbracket_i, \llbracket \boldsymbol{\beta} \rrbracket_i, \llbracket \mathbf{a} \rrbracket_i, \llbracket c \rrbracket_i)_{i \in [N]}, \mathsf{c}_1, \mathsf{pk})$
  11. $\sigma_2 \leftarrow (\llbracket \boldsymbol{\alpha} \rrbracket_i, \llbracket v \rrbracket_i)_{i \in [N]}$
  12. $\mathsf{P}\colon \sigma_2 \rightarrow \mathsf{V}$        ▷ $\{\mathsf{bc}_i\}_{i \in [N]} = (\llbracket \boldsymbol{\alpha} \rrbracket_i, \llbracket v \rrbracket_i)_{i \in [N]}$
  Round 4 (MPC party challenge):
  13. $\mathsf{P} \xleftarrow{\$} \mathsf{c}_2\colon \mathsf{V}$, where $\mathsf{c}_2 \in [N]$
  Round 5 (Openings):
  14. $\mathsf{path} \leftarrow \mathsf{GenPath}(\mathsf{root}, \mathsf{c}_2))$
  15. $\sigma_3 \leftarrow \mathsf{path}$
  16. $\mathsf{P}\colon \sigma_3 \rightarrow \mathsf{V}$

  Verification$(\sigma_1, \mathsf{c}_1, \sigma_2, \mathsf{c}_2, \sigma_3, \mathsf{pk} = (\mathbf{H}, \mathbf{y}))$
  1. $\{\mathsf{com}_i\}_{i \in [N]}, (\mathit{\Delta}\mathbf{x}_B, \mathit{\Delta}\boldsymbol{\beta}, \mathit{\Delta}c) \leftarrow \sigma_1, (\boldsymbol{\gamma}, \boldsymbol{\epsilon}) \leftarrow \mathsf{c}_1,$
     $(\llbracket \boldsymbol{\alpha} \rrbracket_i, \llbracket v \rrbracket_i)_{i \in [N]} \leftarrow \sigma_2, \mathsf{c}_2 \leftarrow \mathsf{c}_2$, $\mathsf{path} \leftarrow \sigma_3$
  2. $\{\mathsf{seed}_i, r_i\}_{i \neq \mathsf{c}_2} \leftarrow \mathsf{LeavesFromPath}(\mathsf{path})$
  3. $\{\mathsf{com}'_i\}_{i \neq \mathsf{c}_2} \leftarrow \{\mathsf{Com}(\mathsf{seed}_i, r_i)\}_{i \neq \mathsf{c}_2}$
  4. Check $\mathsf{com}'_i = \mathsf{com}_i$ for all $i \neq \mathsf{c}_2$
  5. $(\llbracket \mathbf{x}_B \rrbracket_i, \llbracket \boldsymbol{\beta} \rrbracket_i, \llbracket \mathbf{a} \rrbracket_i, \llbracket c \rrbracket_i) \leftarrow \mathsf{PRG}(\mathsf{seed}_i, r_i)$ for all $i \neq \mathsf{c}_2, i \neq N-1$
  6. $(\llbracket \mathbf{x}_B \rrbracket_{N-1}, \llbracket \boldsymbol{\beta} \rrbracket_{N-1}, \llbracket \mathbf{a} \rrbracket_{N-1}, \llbracket c \rrbracket_{N-1}) \leftarrow (\mathit{\Delta}\mathbf{x}_B, \mathit{\Delta}\boldsymbol{\beta}, \mathsf{SampleFqmVector}(\mathsf{seed}_{N-1}), \mathit{\Delta}c)$
  7. $(\llbracket \boldsymbol{\alpha} \rrbracket'_i, \llbracket v \rrbracket'_i)_{i \in [N]} \leftarrow \mathtt{MPC}_N(\mathsf{c}_2, (\llbracket \mathbf{x}_B \rrbracket_i, \llbracket \boldsymbol{\beta} \rrbracket_i, \llbracket \mathbf{a} \rrbracket_i, \llbracket c \rrbracket_i)_{i \in [N]}^{i \neq \mathsf{c}_2}, (\llbracket \boldsymbol{\alpha} \rrbracket_{\mathsf{c}_2}, \llbracket v \rrbracket_{\mathsf{c}_2}), \mathsf{c}_1, \mathsf{pk})$
  8. Check $(\llbracket \boldsymbol{\alpha} \rrbracket'_i, \llbracket v \rrbracket'_i) = (\llbracket \boldsymbol{\alpha} \rrbracket_i, \llbracket v \rrbracket_i)$ for $i \in [N]$, $i \neq \mathsf{c}_2$
  return $\mathtt{Pred}(\{(\llbracket \boldsymbol{\alpha} \rrbracket_i, \llbracket v \rrbracket_i)_{i \in [N]}\}, \mathsf{pk})$
```

mial $(\boldsymbol{\gamma})$. As the challenge is chosen with the uniform distribution from a given set, the results of Sec. 2 can be applied to collapse this round of interaction[6]. Thus the challenge is computed using a random oracle RO as $\mathsf{c} = \mathsf{RO}(\sigma_1)$. In the later abstraction for the hypercube, this is hidden in $\mathtt{GenXtra}(\mathbf{s}, \rho)$, and the challenge values are denoted as xtra.

The RYDE protocol is an IDS that proves knowledge of a solution to the Rank Syndrome Decoding problem, i.e. that P knows some solution $\mathbf{x} \in \mathbb{F}_{q^m}^n$ with low rank weight. We refer to [Bid+23] for details and reasoning and stick to the necessary funcitonal description here. This is done simulating an MPC computation that verifies the correct evaluation of a certain (secret-shared) polynomial over an extension field $\mathbb{F}_{q^{mn}}$. This check is done using the BN20 multiplication check protocol from [BN20]. In [Bid+23] it is argued that following [Fen22] a prover that does not know a valid solution can remain undetected in this step with probability $p_\eta = \frac{2}{q^{m\eta}} - \frac{1}{q^{2m\eta}}$, known as the false positive rate. As common for MPCitH schemes, in RYDE the MPC computation for all parties is simulated by the prover and the communications are

[6] This can indeed always be done, it may only lead to trivial security bounds. We show that in this case we get usable bounds.

sent to the verifier. Afterwards, the verifier asks to open the commitments on the internal state of all but one party.

A cheating P has two ways of remaining undetected. They can hope to simply pass the MPC computation without being detected while using a wrong solution. In this case they will pass the opening phase with certainty. Otherwise, they can falsify their communications $\{\mathtt{bc}_i\}$ for one party i, and then hope that V selects that party as their challenge index $\mathsf{c}_2 = i$, leaving it unopened. In this case, they will get caught in the opening phase with probability $1/N$. Thus the soundness of the five-round proof of knowledge is $\frac{1}{N} + p_\eta(1 - \frac{1}{N})$. We now determine the query-bounded distance-d special soundness of the τ-times parallel repetition of the three-round IDS obtained by eliminating the first round. We can apply Cor. 2 which makes use of Cor. 1 where G is the set of pairs of commitments and challenge points such that a cheating P does not succeed at the first stage. Therefore, we get that $c_{\bar{G}} \leq (p_\eta)^\tau$, and $\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}$, the extraction structure for the d-special soundness of the last three rounds, is $p_{naive}^{\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}} = \frac{1}{N^\tau}\sum_{j=1}^{d-1}\binom{\tau}{j}(N-1)^{j-1}$ following Eq. (2). Therefore, $\sigma_{\mathfrak{s}} = \sqrt{\left(p_{naive}^{\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}}\right)^2(1 - c_{\bar{G}}) + c_{\bar{G}}}$, $\mu_{\mathfrak{s}} = p_{naive}^{\mathfrak{S}_{\mathsf{IDS}_{\tau,d}}}(1 - c_{\bar{G}}) + c_{\bar{G}}$, and

$$\mathrm{Adv}_{\mathsf{IDS},\mathsf{Ext}}^{d-\mathsf{spS}}(\mathsf{A}) \leq \mu_{\mathfrak{s}} + 3\sqrt{C}q\sigma_{\mathfrak{s}} + 2Cq^2\sigma_{\mathfrak{s}}^2\mu_{\mathfrak{s}}\log\left(\frac{1}{\sqrt{C}q\sigma_{\mathfrak{s}}}\right),$$

where $C = 304$, as long as the given bound is ≤ 1.

Next we apply our results on τ-HVZK. We consider the τ-fold parallel repetition. We see that $\mathrm{Adv}_{\mathsf{IDS}_{-1}}^{\tau-\mathsf{hvzk}}(\mathsf{A}) \leq \mathrm{Adv}_{\mathsf{IDS}}^{\tau-\mathsf{hvzk}}(\mathsf{B}) + \frac{3\tau}{2}\sqrt{(q_{\mathsf{H}} + \tau)\cdot 2^{3\lambda}}$, where A is an adversary against the round collapsed IDS, and B against the preceding five-round IDS, and γ_{w}, the commitment entropy, is $2^{3\lambda}$ as RYDE uses 3λ random bits for the first message, λ for root and 2λ for ρ. The soundness and HVZK properties are thus preserved tightly under the round collapse.

4.2 Hypercube transform

Only at this point does the protocol change under the hypercube transform. In the flat scheme, P performs $\mathtt{MPC}_{\mathsf{pk}}(\{\mathtt{in}_i\}, \mathsf{xtra})$ to obtain the per party broadcast communications which is captured by line 10 in Algorithm 1. Instead, the P, for each of the D dimensions, aggregates the input shares to form two main parties. Then the two-party protocol is run $\mathtt{MPC}_{\mathsf{pk}}(\{\mathtt{in}_i\}, \mathsf{xtra})$ for each of the D dimensions, to obtain the communications $\{\mathtt{bc}_j^k\}_{k,j\in[D]\times[2]}$.

In the round-compressed setting, this is the point when the first message containing w, $\{\mathtt{bc}_j^k\}$, $\mathsf{xtra}, \mathsf{aux}$ is sent to V, who responds with a challenge $\mathsf{c} \leftarrow \mathsf{V}$, as per Round 4 of Algorithm 1. P finally opens the views of all leaf parties except the challenge party. Where a $\mathsf{TreePRG}$ is used, this is done with the function $\mathsf{GenPath}(\mathsf{root}, \mathsf{c}_2)$. In the case of RYDE, this returns a sibling path allowing to reconstruct the seeds and party randomness for all $2^D - 1$ leaf parties except c.

We describe in Sec. 3.1 the abstractions $\mathtt{GenSds}, \mathtt{ExpSd}$ to reflect the routines used for generating the openings and inputs into algorithms. We note that the cost of clean abstractions is that in our descriptions, some work is duplicated. For example, $\mathtt{GenSds}$ returns aux in our abstraction, even though in reality this requires calling expanding seeds internally. The security of the new, hypercube-transformed scheme, follows from from straightforward application of Lemma 4 and Lemma 5, with no loss.

Other optimizations are applied when switching from an IDS formulation to a signature format. In signature schemes, for example, only the missing communications $\mathtt{bc}_{\mathsf{c}}$ are sent as the verifier recomputes the other $\mathtt{bc}_i$ from $\{\mathsf{seed}_i, r_i\}$ which are committed to at an early stage.

The elements of the hypercube transcript, derived from Algorithm 1, are described in Fig 6.

RYDE hypercube transcript components

$\mathsf{w} = \{\mathsf{com}_i\}_{i \in 2^D}$

$\{\mathsf{bc}_k^j\}_{(j,k)\in[2]\times[D]} = ([\![\boldsymbol{\alpha}]\!]_i^k, [\![v]\!]_i^k)_{i,k\in[2]\times[D]}$

$\mathsf{xtra} = (\boldsymbol{\gamma}, \boldsymbol{\epsilon})$

$\mathsf{aux} = (\Delta\mathbf{x}_B, \Delta\boldsymbol{\beta}, \Delta c)$

$\mathsf{c} \in [2^D]$

$\mathsf{z} = \{\mathsf{seed}_i, r_i\}_{i\in[2^D]}^{i\neq\mathsf{c}}$

Fig. 6: Description of (hypercube) RYDE in terms of abstraction in this work.

Fiat-Shamir transformation into DSS. The last step is to turn the resulting three-round IDS into a fully non-interactive digital signature scheme. This is canonically done by applying the Fiat-Shamir transform to all interactive rounds. Applying the result from [AM$^+$23] which we provide in Cor. 4 the final DSS that results after applying the FS transform, is UF-CMA secure in the eQROM with security bound:

$$\begin{aligned}\mathrm{Succ}^{\mathsf{UF\text{-}CMA}}_{\mathsf{FS}[\mathrm{RYDE},\mathsf{RO}]}(\mathsf{A}) \leq \epsilon_{RSD} + \mathrm{Adv}^{d-\mathsf{spS}}_{\mathsf{IDS},\mathsf{Ext}}(\mathsf{A}) + (22 \cdot 2^D + 60)q^3 2^{-2\lambda} + 20q^2 \frac{1}{|\mathcal{C}|^{\tau-d}} \\ + q_s(\mathrm{Adv}^{\tau-\mathsf{hvzk}}_{\mathsf{IDS}}(\mathsf{B}) + \frac{3\tau}{2}\sqrt{(q+\tau)\cdot 2^{3\lambda}}) \\ + \frac{3q_{\mathsf{S}}}{2}\sqrt{(q_{\mathsf{RO}} + q_{\mathsf{S}} + 1)\cdot 2^{3\lambda}}\ ,\end{aligned}$$

where ϵ_{RSD} is the hardness of solving rank-based syndrome decoding, $\mathrm{Adv}^{\tau-\mathsf{hvzk}}_{\mathsf{IDS}}(\mathsf{B})$ is the $\tau - \mathsf{hvzk}$ bound for the flat, five-round IDS of RYDE, $\mathrm{Adv}^{d-\mathsf{spS}}_{\mathsf{IDS},\mathsf{Ext}}(\mathsf{A})$ is the bound on distance-d special soundness of the τ-times parallel repetition of the three-round IDS given above and $\mathcal{C}$ is the second challenge space of the flat, five-round IDS of RYDE.

References

[Aar$^+$23] N. Aaraj, S. Bettaieb, L. Bidoux, A. Budroni, V. Dyseryn, A. Esser, P. Gaborit, M. Kulkarni, V. Mateu, M. Palumbi, L. Perin, and J. Tillich. *PERK*. Tech. rep. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures. National Institute of Standards and Technology, 2023.

[Adj$^+$23] G. Adj, L. Rivera-Zamarripa, J. Verbel, E. Bellini, S. Barbero, A. Esser, C. Sanna, and F. Zweydinger. *MiRitH — MinRank in the Head*. Tech. rep. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures. National Institute of Standards and Technology, 2023.

[Agu$^+$23a] C. Aguilar-Melchor, T. Feneuil, N. Gama, S. Gueron, J. Howe, D. Joseph, A. Joux, E. Persichetti, T. H. Randrianarisoa, M. Rivain, and D. Yue. *SDitH — Syndrome Decoding in the Head*. Tech. rep. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures. National Institute of Standards and Technology, 2023.

[Agu+23b] C. Aguilar-Melchor, N. Gama, J. Howe, A. Hülsing, D. Joseph, and D. Yue. "The Return of the SDitH". In: *EUROCRYPT 2023, Part V*. Ed. by C. Hazay and M. Stam. Vol. 14008. LNCS. Springer, Heidelberg, Apr. 2023, pp. 564–596. DOI: 10.1007/978-3-031-30589-4_20.

[AM+23] C. Aguilar-Melchor, A. Hülsing, D. Joseph, C. Majenz, E. Ronen, and D. Yue. "SDitH in the QROM". In: *ASIACRYPT 2023, Part VII*. Ed. by J. Guo and R. Steinfeld. Vol. 14444. LNCS. Singapore: Springer, 2023, pp. 317–350. DOI: 10.1007/978-981-99-8739-9_11.

[Ara+23a] N. Aragon, M. Bardet, L. Bidoux, J.-J. Chi-Domínguez, V. Dyseryn, T. Feneuil, P. Gaborit, A. Joux, M. Rivain, J. Tillich, and A. Vinçotte. *RYDE*. Tech. rep. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures. National Institute of Standards and Technology, 2023.

[Ara+23b] N. Aragon, M. Bardet, L. Bidoux, J.-J. Chi-Domínguez, V. Dyseryn, T. Feneuil, P. Gaborit, A. Joux, M. Rivain, J. Tillich, and A. Vinçotte. *RYDE*. Tech. rep. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures. National Institute of Standards and Technology, 2023.

[Ara+23c] N. Aragon, M. Bardet, L. Bidoux, J. Chi-Domínguez, V. Dyseryn, T. Feneuil, P. Gaborit, R. Neveu, M. Rivain, and J. Tillich. *MIRA*. Tech. rep. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures. National Institute of Standards and Technology, 2023.

[Bar+23] M. Barbosa, G. Barthe, C. Doczkal, J. Don, S. Fehr, B. Grégoire, Y.-H. Huang, A. Hülsing, Y. Lee, and X. Wu. "Fixing and Mechanizing the Security Proof of Fiat-Shamir with Aborts and Dilithium". In: *CRYPTO 2023, Part V*. Ed. by H. Handschuh and A. Lysyanskaya. Vol. 14085. LNCS. Springer, Heidelberg, Aug. 2023, pp. 358–389. DOI: 10.1007/978-3-031-38554-4_12.

[Bet+23] L. Bettale, D. Kahrobaei, L. Perret, and J. Verbel. *Biscuit*. Tech. rep. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures. National Institute of Standards and Technology, 2023.

[Bid+23] L. Bidoux, J.-J. Chi-Domínguez, T. Feneuil, P. Gaborit, A. Joux, M. Rivain, and A. Vinçotte. *RYDE: A Digital Signature Scheme based on Rank-Syndrome-Decoding Problem with MPCitH Paradigm*. 2023. arXiv: 2307.08726 [cs.CR].

[BN20] C. Baum and A. Nof. "Concretely-Efficient Zero-Knowledge Arguments for Arithmetic Circuits and Their Application to Lattice-Based Cryptography". In: *PKC 2020, Part I*. Ed. by A. Kiayias, M. Kohlweiss, P. Wallden, and V. Zikas. Vol. 12110. LNCS. Springer, Heidelberg, May 2020, pp. 495–526. DOI: 10.1007/978-3-030-45374-9_17.

[Dev+23] J. Devevey, P. Fallahpour, A. Passelègue, and D. Stehlé. "A Detailed Analysis of Fiat-Shamir with Aborts". In: *CRYPTO 2023, Part V*. Ed. by H. Handschuh and A. Lysyanskaya. Vol. 14085. LNCS. Springer, Heidelberg, Aug. 2023, pp. 327–357. DOI: 10.1007/978-3-031-38554-4_11.

[Don+19] J. Don, S. Fehr, C. Majenz, and C. Schaffner. "Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model". In: *CRYPTO 2019, Part II*. Ed. by A. Boldyreva and D. Micciancio. Vol. 11693. LNCS. Springer, Heidelberg, Aug. 2019, pp. 356–383. DOI: 10.1007/978-3-030-26951-7_13.

[Don+22a] J. Don, S. Fehr, C. Majenz, and C. Schaffner. "Efficient NIZKs and Signatures from Commit-and-Open Protocols in the QROM". In: *CRYPTO 2022, Part II*. Ed. by Y. Dodis and T. Shrimpton. Vol. 13508. LNCS. Springer, Heidelberg, Aug. 2022, pp. 729–757. DOI: 10.1007/978-3-031-15979-4_25.

[Don+22b] J. Don, S. Fehr, C. Majenz, and C. Schaffner. "Online-Extractability in the Quantum Random-Oracle Model". In: *EUROCRYPT 2022, Part III*. Ed. by O. Dunkelman and S. Dziembowski. Vol. 13277. LNCS. Springer, Heidelberg, 2022, pp. 677–706. DOI: 10.1007/978-3-031-07082-2_24.

[Duc+18] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. "CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme". In: *IACR TCHES* 2018.1 (2018). https://tches.iacr.org/index.php/TCHES/article/view/839, pp. 238–268. ISSN: 2569-2925. DOI: 10.13154/tches.v2018.i1.238-268.

[Fen22] T. Feneuil. *Building MPCitH-based Signatures from MQ, MinRank, Rank SD and PKP*. Cryptology ePrint Archive, Report 2022/1512. https://eprint.iacr.org/2022/1512. 2022.

[FJR22] T. Feneuil, A. Joux, and M. Rivain. "Syndrome Decoding in the Head: Shorter Signatures from Zero-Knowledge Proofs". In: *CRYPTO 2022, Part II*. Ed. by Y. Dodis and T. Shrimpton. Vol. 13508. LNCS. Springer, Heidelberg, Aug. 2022, pp. 541–572. DOI: 10.1007/978-3-031-15979-4_19.

[FR23] T. Feneuil and M. Rivain. *MQOM — MQ on my Mind*. Tech. rep. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures. National Institute of Standards and Technology, 2023.

[FS87] A. Fiat and A. Shamir. "How to Prove Yourself: Practical Solutions to Identification and Signature Problems". In: *CRYPTO'86*. Ed. by A. M. Odlyzko. Vol. 263. LNCS. Springer, Heidelberg, Aug. 1987, pp. 186–194. DOI: 10.1007/3-540-47721-7_12.

[GGM84] O. Goldreich, S. Goldwasser, and S. Micali. "How to Construct Random Functions (Extended Abstract)". In: *25th FOCS*. IEEE Computer Society Press, Oct. 1984, pp. 464–479. DOI: 10.1109/SFCS.1984.715949.

[Gri+21] A. B. Grilo, K. Hövelmanns, A. Hülsing, and C. Majenz. "Tight Adaptive Reprogramming in the QROM". In: *ASIACRYPT 2021, Part I*. Ed. by M. Tibouchi and H. Wang. Vol. 13090. LNCS. Springer, Heidelberg, Dec. 2021, pp. 637–667. DOI: 10.1007/978-3-030-92062-3_22.

[HHM22] K. Hövelmanns, A. Hülsing, and C. Majenz. "Failing Gracefully: Decryption Failures and the Fujisaki-Okamoto Transform". In: *ASIACRYPT 2022, Part IV*. Ed. by S. Agrawal and D. Lin. Vol. 13794. LNCS. Springer, Heidelberg, Dec. 2022, pp. 414–443. DOI: 10.1007/978-3-031-22972-5_15.

[Hül+22] A. Hülsing, D. J. Bernstein, C. Dobraunig, M. Eichlseder, S. Fluhrer, S.-L. Gazdag, P. Kampanakis, S. Kölbl, T. Lange, M. M. Lauridsen, F. Mendel, R. Niederhagen, C. Rechberger, J. Rijneveld, P. Schwabe, J.-P. Aumasson, B. Westerbaan, and W. Beullens. *SPHINCS$^+$*. Tech. rep. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. National Institute of Standards and Technology, 2022.

[Ish+07] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. "Zero-knowledge from secure multiparty computation". In: *39th ACM STOC*. Ed. by D. S. Johnson and U. Feige. ACM Press, June 2007, pp. 21–30. DOI: 10.1145/1250790.1250794.

[Kim+23] S. Kim, J. Cho, M. Cho, J. Ha, J. Kwon, B. Lee, J. Lee, J. Lee, S. Lee, D. Moon, M. Son, and H. Yoon. *AIMer*. Tech. rep. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures. National Institute of Standards and Technology, 2023.

[KZ22] D. Kales and G. Zaverucha. *Efficient Lifting for Shorter Zero-Knowledge Proofs and Post-Quantum Signatures*. Cryptology ePrint Archive, Report 2022/588. https://eprint.iacr.org/2022/588. 2022.

[Lei18] D. Leichtle. *Post-quantum signatures from identification schemes.* Master's thesis, Technische Universiteit Eindhoven. https://pure.tue.nl/ws/portalfiles/portal/125545339/Dominik_Leichtle_thesis_final_IAM_307.pdf. 2018.

[LN17] Y. Lindell and A. Nof. "A Framework for Constructing Fast MPC over Arithmetic Circuits with Malicious Adversaries and an Honest-Majority". In: *ACM CCS 2017*. Ed. by B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu. ACM Press, 2017, pp. 259–276. DOI: 10.1145/3133956.3133999.

[NIS16] NIST. *Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process.* https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf. 2016.

[NIS22] NIST. *Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process.* https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf. 2022.

[Sam+19] S. Samardjiska, M.-S. Chen, A. Hulsing, J. Rijneveld, and P. Schwabe. *MQDSS.* Tech. rep. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions. National Institute of Standards and Technology, 2019.

[Unr12] D. Unruh. "Quantum Proofs of Knowledge". In: *EUROCRYPT 2012*. Ed. by D. Pointcheval and T. Johansson. Vol. 7237. LNCS. Springer, Heidelberg, Apr. 2012, pp. 135–152. DOI: 10.1007/978-3-642-29011-4_10.

A From IDS to Signature

In [AM+23] the authors showed how to go from a distance-d special sound commit-and-open IDS like the IDS that we are constructing to a secure signature scheme. To be self-contained, we recall this result, closely following the text in [AM+23], with minor adoptions to the notation used in this work.

A.1 Preliminaries

We first define commitment recoverable IDS. Then, we recall definitions of syntax and security of digital signature schemes as well as the Fiat-Shamir transform. We closely follow [Gri+21] for these definitions as did [AM+23].

Definition 12 (Commitment-recoverable IDS). *A three-round IDS with transcripts* $(\mathsf{w}, \mathsf{c}, \mathsf{z})$ *is called commitment-recoverable if there exists a function* Rcvr *that given the challenge and response* (c, z) *for a transcript, outputs the commitment message.*

Note that for the MPCitH-based IDS discussed in Sec. 3 (which also works for most other commit & open IDS where the opening information for the commitment allows to recompute the inputs) there exists a trivial transform to turn them into a commitment recoverable IDS': We replace w by $\mathsf{w}' = H(\mathsf{w})$ using the same hash function that is used by the commitment function Com. We also add all information from the original w that cannot be recovered from the original z to the new z', i.e., $\mathsf{z}' = (\mathsf{z}, \mathsf{com}_{\mathsf{c}}, \mathsf{bc}_{\mathsf{c}}, \mathsf{xtra}, \mathsf{aux})$. This reduces the transcript size of the IDS significantly as it removes $N^D - 1$ commitments and at $DN - 1$ $(N^D - 1)$ communications in the hypercube (flat) setting. One can view this as using an N^D-ary tree commitment with N^D leaves, as considered by the security bounds below.

Definition 13 (Signature scheme). *A digital signature scheme* DSS *is defined as a triple of algorithms* $\mathsf{DSS} = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{Vrfy})$.

```
Game UF-NMA
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}()$
    $(m^*, \sigma^*) \leftarrow \mathsf{A}(\mathsf{pk})$
    return $\mathsf{Vrfy}(\mathsf{pk}, m^*, \sigma^*)$

Game UF-CMA
    $\mathcal{L} \leftarrow \{\}$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}()$
    $(m^*, \sigma^*) \leftarrow \mathsf{A}^{\mathsf{Sign}(\mathsf{sk}, \cdot)}(\mathsf{pk})$
    if $m^* \in \mathcal{L}$ return $0$
    return $\mathsf{Vrfy}(\mathsf{pk}, m^*, \sigma^*)$

Sign(sk, m)
    $\mathcal{L} := \mathcal{L} \cup \{m\}$
    $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$
    return $\sigma$
```

Fig. 7: Games UF-CMA and UF-NMA.

```
Sign(sk, m)
    $(\mathsf{w}, \mathsf{st}) \leftarrow \mathsf{Commit}(\mathsf{sk})$
    $\mathsf{c} \leftarrow \mathsf{RO}(\mathsf{w}, m)$
    $\mathsf{z} \leftarrow \mathsf{Resp}(\mathsf{sk}, \mathsf{w}, \mathsf{c}, \mathsf{st})$
    return $\sigma := (\mathsf{c}, \mathsf{z})$

Vrfy(pk, m, σ = (c, z))
    $\mathsf{w} \leftarrow \mathsf{Rcvr}(\mathsf{c}, \mathsf{z})$
    $a \leftarrow (\mathsf{c} = \mathsf{RO}(\mathsf{w}, m))$
    $b \leftarrow \mathsf{Vrf}(\mathsf{pk}, \mathsf{w}, \mathsf{c}, \mathsf{z})$
    return $a \wedge b$
```

Fig. 8: Signing and verification algorithms of $\mathsf{DSS} = \mathsf{FS}[\mathsf{IDS}, \mathsf{RO}]$.

- *The probabilistic key generation algorithm* $\mathsf{Keygen}()$ *returns a key pair* $(\mathsf{pk}, \mathsf{sk})$*. We assume that* pk *defines the message space* $\mathcal{M}$*.*
- *The possibly probabilistic signing algorithm* $\mathsf{Sign}(\mathsf{sk}, m)$ *returns a signature* σ*.*
- *The deterministic verification algorithm* $\mathsf{Vrfy}(\mathsf{pk}, m, \sigma)$ *returns 1 (accept) or 0 (reject).*

UF-CMA, and UF-NMA security. We define unforgeability under chosen message attacks (UF-CMA), and unforgeability under no message attacks, i.e., with no access to a signing oracle (also known as UF-KOA, or $\mathsf{UF\text{-}CMA}_0$) success functions of a possibly quantum adversary A against DSS as

$$\mathrm{Succ}^{\mathsf{UF\text{-}X}}_{\mathsf{DSS}}(\mathsf{A}) := \Pr[1 \leftarrow \mathsf{UF\text{-}X}^{\mathsf{A}}_{\mathsf{DSS}}] \ ,$$

where the games for $X \in \{\mathsf{CMA}, \mathsf{NMA}\}$ are given in Fig. 7.

The Fiat-Shamir transform. Here we describe the Fiat-Shamir transform for commitment recoverable IDS. From a security perspective it is equivalent to the standard Fiat-Shamir transform, which can be shown by a straight-forward reduction. To a commitment-recoverable identification scheme $\mathsf{IDS} = (\mathsf{Keygen}, \mathsf{Commit}, \mathsf{Resp}, \mathsf{Vrf}, \mathsf{Rcvr})$ with commitment space $\mathcal{COM}$, and random oracle $\mathsf{RO} : \mathcal{COM} \times \mathcal{M} \to \mathcal{C}$ for some message space $\mathcal{M}$, we associate

$$\mathsf{FS}[\mathsf{IDS}, \mathsf{RO}] := \mathsf{DSS} := (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{Vrfy}) \ ,$$

where algorithms Sign and Vrfy of DSS are defined in Fig. 8.

In [Gri+21] the following result was stated that relates the UF-NMA and UF-CMA security of a Fiat-Shamir transformed IDS in the QROM, and the HVZK property of the IDS. The bound makes use of what they call commitment entropy:

$$\gamma_{\mathsf{w}} := \mathbb{E} \max_{\mathsf{w}} \Pr[\mathsf{w}] \ ,$$

where the expectation is taken over $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}$, and the probability is taken over $(\mathsf{w}, \mathsf{st}) \leftarrow \mathsf{Commit}(\mathsf{sk})$.

Theorem 1. *[Gri+21, Theorem 3] For any (quantum)* $\mathsf{UF\text{-}CMA}$ *adversary* A *issuing at most* q_{S} *(classical) queries to the signing oracle* sign *and at most* q_{H} *quantum queries to* RO*, there exists a* $\mathsf{UF\text{-}NMA}$ *adversary* B *and a* q_{S}*-*HVZK *adversary* C *such that*

$$\begin{aligned}\mathrm{Succ}^{\mathsf{UF\text{-}CMA}}_{\mathsf{FS[IDS,RO]}}(\mathsf{A}) \leq \mathrm{Succ}^{\mathsf{UF\text{-}NMA}}_{\mathsf{FS[IDS,RO]}}(\mathsf{B}) + \mathrm{Adv}^{q_{\mathsf{S}}-\mathsf{HVZK}}_{\mathsf{IDS}}(\mathsf{C})\\ + \frac{3q_{\mathsf{S}}}{2}\sqrt{(q_{\mathsf{H}}+q_{\mathsf{S}}+1)\cdot\gamma_{\mathsf{w}}}\ ,\end{aligned} \tag{10}$$

and the running time of B *and* C *is about that of* A*, where* γ_{w} *is the maximum over the probability that* w *takes any given value. The bound given in Eq. (10) also holds for the modified Fiat-Shamir transform that defines challenges by letting* $\mathsf{c} := \mathsf{RO}(\mathsf{w}, m, \mathsf{pk})$ *instead of letting* $\mathsf{c} := \mathsf{RO}(\mathsf{w}, m)$*.*

In [AM+23], the following result is proven, which is a variant of Theorem 5.2 in [Don+22a]. It relates the $\mathsf{UF\text{-}NMA}$ security of the Fiat-Shamir transformed IDS to its distance-d special soundness in a relatively tight way.

Theorem 2 (Variant of Theorem 5.2 from [Don+22a]). *Let* $\mathsf{IDS}^{\mathsf{Com},G}$ *be a distance-d special-sound commit-and-open identification scheme with* ϕ*-ary tree commitment with* n_c *leaves using a random oracle* Com *with output length c, splittable challenge, challenge space* $\mathcal{C}^\tau$ *and an additional random oracle G. Let further* A *be a* $\mathsf{UF\text{-}NMA}$*-adversary against* $\mathsf{FS[IDS,RO]}$ *making* q_{RO}*,* q_{Com} *and* q_G *queries to* RO*,* Com *and G respectively. Then there exists a* (q_{Com}, q_G)*-query QROM+ adversary* B *against the query-bounded distance-d special soundness of* $\mathsf{IDS}^{\mathsf{Com},G}$ *with respect to the special soundness extractor* Ext_d *of* IDS *such that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{FS[IDS,RO]}}(\mathsf{A}) \leq &\Pr[\mathsf{sk}' \leftarrow \mathsf{Ext}_d \circ \mathsf{B} : (\mathsf{sk}', \mathsf{pk}) \in \mathsf{Keygen}()] + \mathrm{Adv}^{d-\mathsf{spS}}_{\mathsf{IDS,Ext}}(\mathsf{B})\\ &+ (22 n_c \log_\phi n_c + 60) q^3 2^{-c} + 20 q^2 \frac{1}{|\mathcal{C}|^{\tau-d}}\ ,\end{aligned}$$

where $q = q_{\mathsf{Com}} + q_{\mathsf{RO}}$*. The runtime of* B *is bounded as* $\mathsf{TIME}(\mathsf{B}) \leq \mathsf{TIME}(\mathsf{A}) + \xi(q + q_G)^2))$*, where* ξ *is polynomial in the input and output lengths of the random oracles.*

Combining the two theorems, we get as a corollary

Corollary 3 ($\mathsf{UF\text{-}CMA}$ security of FS (QROM+)). *Let* $\mathsf{IDS}^{\mathsf{Com},G}$ *be a distance-d special-sound commit-and-open identification scheme that is honest-verifier zero-knowledge with* ϕ*-ary tree commitment with* n_c *leaves using a random oracle* Com *with output length c, splittable challenge, challenge space* $\mathcal{C}^\tau$ *and an additional random oracle G. Let further* A *be a* $\mathsf{UF\text{-}CMA}$ *adversary against* $\mathsf{FS[IDS,RO]}$ *issuing at most* q_{S} *(classical) queries to the signing oracle* sign*, as well as making* q_{RO}*,* q_{Com}*, and* q_G *queries to* RO*,* Com*, and G respectively. Then for every* $d = 0, \ldots \tau$ *there exists a* (q_{Com}, q_G)*-query QROM+ adversary* B *against the query-bounded distance-d special soundness of* $\mathsf{IDS}^{\mathsf{Com},G}$ *with respect to the special soundness extractor* Ext_d *of* IDS*, and a* q_{S}*-*HVZK *adversary* C *such that*

$$\begin{aligned}\mathrm{Succ}^{\mathsf{UF\text{-}CMA}}_{\mathsf{FS[IDS,RO]}}(\mathsf{A}) \leq &\Pr[\mathsf{sk}' \leftarrow \mathsf{Ext}_d \circ \mathsf{B} : (\mathsf{sk}', \mathsf{pk}) \in \mathsf{Keygen}()] + \mathrm{Adv}^{d-\mathsf{spS}}_{\mathsf{IDS,Ext}}(\mathsf{B})\\ &+ (22 n_c \log_\phi n_c + 60) q^3 2^{-c} + 20 q^2 \frac{1}{|\mathcal{C}|^{\tau-d}}\\ &+ \mathrm{Adv}^{q_{\mathsf{S}}-\mathsf{HVZK}}_{\mathsf{IDS}}(\mathsf{C}) + \frac{3q_{\mathsf{S}}}{2}\sqrt{(q_{\mathsf{RO}}+q_{\mathsf{S}}+1)\cdot\gamma_{\mathsf{w}}}\ ,\end{aligned}$$

where $q = q_{\mathsf{Com}} + q_{\mathsf{RO}} + q_{\mathsf{S}}$*. The runtime of* B *is bounded as* $\mathsf{TIME}(\mathsf{B}) \leq \mathsf{TIME}(\mathsf{A}) + \xi(q + q_G)^2))$*, where* ξ *is polynomial in the input and output lengths of the random oracles.*

It should be noted that this result is based on a bound for the soundness in the QROM+ introduced in [AM+23]. In our case, the bound we obtain applying the round elimination is in the eQROM [Don+22b]. The eQROM is a model that is intuitively even stronger than the QROM+ as it gives the adversary adaptive access to an extraction interface during the interaction, while the QROM+ only provides access to the measured database after the interaction. Hence, the proof for Theorem 2 can be adapted to use the eQROM in place of the QROM+ by making the necessary extraction queries after all interaction. Thereby, we get the following corollary:

Corollary 4 (UF-CMA security of FS (eQROM)). *Let* $\mathsf{IDS}^{\mathsf{Com},G}$ *be a distance-d special-sound commit-and-open identification scheme that is honest-verifier zero-knowledge with* ϕ*-ary tree commitment with* n_c *leaves using a random oracle* Com *with output length c, splittable challenge, challenge space* $\mathcal{C}^\tau$ *and an additional random oracle* G*. Let further* A *be a* UF-CMA *adversary against* FS[IDS, RO] *issuing at most* q_{S} *(classical) queries to the signing oracle* sign*, as well as making* q_{RO}*,* q_{Com}*, and* q_G *queries to* RO*,* Com*, and* G *respectively. Then for every* $d = 0, \dots \tau$ *there exists a* (q_{Com}, q_G)*-query eQROM adversary* B *against the query-bounded distance-d special soundness of* $\mathsf{IDS}^{\mathsf{Com},G}$ *with respect to the special soundness extractor* Ext_d *of* IDS*, and a* q_{S}-HVZK *adversary* C *such that*

$$\begin{aligned}\mathrm{Succ}^{\mathsf{UF\text{-}CMA}}_{\mathsf{FS[IDS,RO]}}(\mathsf{A}) \leq& \Pr[\mathsf{sk}' \leftarrow \mathsf{Ext}_d \circ \mathsf{B} : (\mathsf{sk}', \mathsf{pk}) \in \mathsf{Keygen}()] + \mathrm{Adv}^{d-\mathsf{spS}}_{\mathsf{IDS},\mathsf{Ext}}(\mathsf{B}) \\ &+ (22 n_c \log_\phi n_c + 60) q^3 2^{-c} + 20 q^2 \frac{1}{|\mathcal{C}|^{\tau-d}} \\ &+ \mathrm{Adv}^{q_{\mathsf{S}}-\mathsf{HVZK}}_{\mathsf{IDS}}(\mathsf{C}) + \frac{3 q_{\mathsf{S}}}{2} \sqrt{(q_{\mathsf{RO}} + q_{\mathsf{S}} + 1) \cdot \gamma_{\mathsf{w}}} \ ,\end{aligned}$$

where $q = q_{\mathsf{Com}} + q_{\mathsf{RO}} + q_{\mathsf{S}}$*. The runtime of* B *is bounded as* $\mathsf{TIME}(\mathsf{B}) \leq \mathsf{TIME}(\mathsf{A}) + \xi(q + q_G)^2))$*, where* ξ *is polynomial in the input and output lengths of the random oracles.*

B Proof of Lemma 2

Below we give a short proof of Lemma 2 which is as follows

Lemma 2 (R-HVZK of round elimination). *Let* IDS_{-1} *be the IDS that is obtained by applying round elimination to* IDS *using random oracle* RO*. If* IDS *has first message entropy* $\gamma_{\mathsf{w}} := \mathbb{E} \max_{\mathsf{w}_1} \Pr[\mathsf{w}_1]$ *Then it holds for any adversary* A *against the* $R-$ HVZK *property of* IDS_{-1} *that makes* q_{H} *queries to* RO*, there exists an adversary* B *against* $R-$ HVZK *of* IDS *with*

$$\mathrm{Adv}^{R-\mathsf{hvzk}}_{\mathsf{IDS}_{-1}}(\mathsf{A}) \leq \mathrm{Adv}^{R-\mathsf{hvzk}}_{\mathsf{IDS}}(\mathsf{B}) + \frac{3R}{2}\sqrt{(q_{\mathsf{H}} + R) \cdot \gamma_{\mathsf{w}}}.$$

The proof makes use of the adaptive reprogramming technique of [Gri+21]. Specifically of the following proposition which makes use of the games defined in Fig. 9:

Proposition 1 ([Gri+21], Proposition 2). *Let* X_1*,* X_2*,* X' *and* Y *be some finite sets, and let p be a distribution on* $X_1 \times X'$*. Let* A *be any distinguisher, issuing q many (quantum) queries to* RO *and R many reprogramming instructions such that each instruction consists of a value* x_2*, together with the fixed distribution p. Then*

$$|\Pr[\mathsf{Repro}_1^{\mathsf{A}} \Rightarrow 1] - \Pr[\mathsf{Repro}_0^{\mathsf{A}} \Rightarrow 1]| \leq \frac{3R}{2}\sqrt{q p_{\max}} \ ,$$

where $p_{\max} := \max_{x_1} p(x_1)$*.*

However, we only need the case where x_2 is the empty string.

```
GAME Repro_b
    $\mathsf{RO}_0 \xleftarrow{\$} Y^X$
    $\mathsf{RO}_1 := \mathsf{RO}_0$
    $b' \leftarrow \mathsf{A}^{\mathsf{RO}_b, \mathsf{Reprogram}}$
    return $b'$

Reprogram(p)
    $(x, x') \leftarrow p$
    $y \xleftarrow{\$} Y$
    $\mathsf{RO}_1 := \mathsf{RO}_1^{x \mapsto y}$
    return $(x, x')$
```

Fig. 9: Adaptive reprogramming games Repro_b for bit $b \in \{0,1\}$. The adversary gets quantum access to RO_b but only classical access to $\mathsf{Reprogram}$

Proof. Our proof strategy is similar to that for the hypercube transform. We show how to construct a simulator Sim_{-1} for the round-eliminated scheme IDS_{-1} using the simulator for IDS. For this we make use of adaptive programming. To implement Sim_{-1}, we run $t \leftarrow \mathsf{Sim}(\mathsf{pk})$ to obtain a transcript. Then we simply program $\mathsf{c}_1 := \mathsf{RO}(\mathsf{w}_1)$ and output t.

We give a proof by reduction. Given adversary A against $R - \mathsf{HVZK}$ of IDS_{-1}, we construct reduction B against $R - \mathsf{HVZK}$ of IDS as follows. On input the public key pk, and access to transcript oracle $\mathcal{O}$, B runs $\mathsf{A}^{\mathsf{R}(\mathcal{O})}(\mathsf{pk})$, and outputs the result. The algorithm R, implements the reprogramming mentioned above, i.e., given a transcript $t = (\mathsf{w}_1, \mathsf{c}_1, \ldots)$ it programs $\mathsf{c}_1 := \mathsf{RO}(\mathsf{w}_1)$ and outputs t. Let $\mathsf{Trans}(\mathsf{sk})$, and $\mathsf{Trans}_{-1}(\mathsf{sk})$ refer to algorithms that execute IDS and IDS_{-1}, respectively, and output the resulting transcript. By definition, we have

$$\mathrm{Adv}_{\mathsf{IDS}}^{R-\mathsf{hvzk}}(\mathsf{B}) = |\Pr[1 \leftarrow \mathsf{B}^{\mathsf{Sim}(\mathsf{pk})}(\mathsf{pk})] - \Pr[1 \leftarrow \mathsf{B}^{\mathsf{Trans}(\mathsf{sk})}(\mathsf{pk})]|,$$

where the probabilities are taken over the randomness involved in generating $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{\$} \mathsf{Keygen}()$, as well as the coins of A, Sim, and Trans. Plugging in the definition of B, this becomes

$$\mathrm{Adv}_{\mathsf{IDS}}^{R-\mathsf{hvzk}}(\mathsf{B}) = |\Pr[1 \leftarrow \mathsf{A}^{\mathsf{R}(\mathsf{Sim}(\mathsf{pk}))}(\mathsf{pk})] - \Pr[1 \leftarrow \mathsf{A}^{\mathsf{R}(\mathsf{Trans}(\mathsf{sk}))}(\mathsf{pk})]|.$$

By construction, we have that $\mathsf{R}(\mathsf{Sim}(\mathsf{pk})) = \mathsf{Sim}_{-1}(\mathsf{pk})$. It remains to show that $\mathsf{R}(\mathsf{Trans}(\mathsf{sk})) \approx_c \mathsf{Trans}_{-1}(\mathsf{sk})$, i.e., the output distribution of `TransF2H`, given honest IDS transcripts, is indistinguishable from that of honest transcripts for IDS_{-1}, then the above becomes $\mathrm{Adv}_{\mathsf{IDS}_{-1}}^{R-\mathsf{hvzk}}(\mathsf{A})$.

Towards this end, we will show that any adversary A that can distinguish $\mathsf{R}(\mathsf{Trans}(\mathsf{sk}))$ from $\mathsf{Trans}_{-1}(\mathsf{sk})$ can be used to distinguish the reprogramming games for the distribution $p = (\mathsf{Commit}(\mathsf{sk}, X), X)$ which is the joint distribution of all the inputs to the Commit function except sk and the resulting first message.

For this, we use an algorithm Trans' which produces transcripts by querying $\mathsf{Reprogram}(p) \to (\mathsf{w}_1, x')$, using the distribution p outlined above, to obtain w_1 and the inputs x' used to compute it. Afterwards it continues as usual, i.e., it queries $\mathsf{RO}(\mathsf{w}_1)$ to obtain c_1, and so on. We build a reduction C that runs A with Trans' as oracle and the random oracle it is provided with by Repro as RO. The reduction C outputs whatever A outputs. Note that whenever C plays in Repro_0 the view of A is identical to that when interacting with Trans_{-1} and an independent RO. Whenever C plays in Repro_1, the view is identical to that when interacting with $\mathsf{R} \circ \mathsf{Trans}$ and the RO that R reprograms. The number of queries that Trans' makes on top of those by A are exactly R, one per transcript. Hence

$$\begin{aligned}
&|\Pr[1 \leftarrow \mathsf{A}^{\mathsf{R}(\mathsf{Trans}(\mathsf{sk}))}(\mathsf{pk})] - \Pr[1 \leftarrow \mathsf{A}^{\mathsf{Trans}_{-1}(\mathsf{sk})}(\mathsf{pk})]| \\
&= |\Pr[\mathsf{Repro}_1^{\mathsf{C}} \Rightarrow 1] - \Pr[\mathsf{Repro}_0^{\mathsf{C}} \Rightarrow 1]| \\
&\leq \frac{3R}{2}\sqrt{(q_{\mathsf{H}} + R) \cdot \gamma_{\mathsf{w}}}
\end{aligned}$$

Putting everything together, and using the triangle inequality, we get

$$\begin{aligned}
&\mathrm{Adv}^{\mathsf{hvzk}}_{\mathsf{IDS}}(\mathsf{B}) + \frac{3}{2}\sqrt{(q_{\mathsf{H}}+1)\cdot\gamma_{\mathsf{w}}} \\
&\qquad \geq |\Pr[1 \leftarrow \mathsf{A}^{\mathsf{Sim}_{-1}(\mathsf{pk})}(\mathsf{pk})] - \Pr[1 \leftarrow \mathsf{A}^{\mathsf{Trans}_{-1}(\mathsf{sk})}(\mathsf{pk})]| \\
&\qquad = \mathrm{Adv}^{\mathsf{hvzk}}_{\mathsf{IDS}_{-1}}(\mathsf{A}) .
\end{aligned}$$

The runtime of Sim_{-1} is essentially that of Sim, only adding the reprogramming. □