Round-Robin is Optimal: Lower Bounds for Group Action Based Protocols

Daniele Cozzo¹, Emanuele Giunta^{1,2}

¹ IMDEA Software Institute, Spain
 {daniele.cozzo, emanuele.giunta}@imdea.org
 ² Universidad Politecnica de Madrid, Spain.

Abstract. An hard homogeneous space (HHS) is a finite group acting on a set with the group action being hard to invert and the set lacking any algebraic structure. As such HHS could potentially replace finite groups where the discrete logarithm is hard for building cryptographic primitives and protocols in a post-quantum world.

Threshold HHS-based primitives typically require parties to compute the group action of a secret-shared input on a public set element. On one hand this could be done through generic MPC techniques, although they incur in prohibitive costs due to the high complexity of circuits evaluating group actions known to date. On the other hand round-robin protocols only require black box usage of the HHS. However these are highly sequential procedures, taking as many rounds as parties involved. The high round complexity appears to be inherent due the lack of homomorphic properties in HHS, yet no lower bounds were known so far.

In this work we formally show that round-robin protocols are optimal. In other words, any *at least passively secure* distributed computation of a group action making black-box use of an HHS must take a number of rounds greater or equal to the threshold parameter. We furthermore study *fair* protocols in which all users receive the output in the same round (unlike plain round-robin), and prove communication and computation lower bounds of $\Omega(n \log_2 n)$ for *n* parties. Our results are proven in Shoup's Generic Action Model (GAM), and hold regardless of the underlying computational assumptions.

Table of Contents

1	Introduction	3		
	1.1 Our Contributions	4		
	1.2 Our Techniques	6		
	1.3 Related Work	7		
2	Preliminaries			
	2.1 Notation	8		
	2.2 Hard Homogeneous Space	8		
	2.3 Shoup's Generic Action Model	9		
3	Technical Lemmas			
	3.1 Sequentiality Lemma	9		
	3.2 Interactive Protocols	10		
	3.3 Interactive Sequentiality Lemma	11		
4	Round Lower Bound			
5	Fair Protocols Lower Bounds	13		
	5.1 Fair Protocols	13		
	5.2 Refined Interactive Sequentiality Lemma	14		
	5.3 Tall Sub-tree Property	15		
	5.4 Main Result	16		
6	Conclusions			
A	CSIDH Group Action	28		
в	Optimal Fair Protocols			
\mathbf{C}	Postponed Proofs	32		
	C.1 Sequentiality Lemma	32		
	C.2 Properties	35		
	C.3 Interactive Sequentiality Lemma	35		
	C.4 Refined Interactive Sequentiality Lemma	38		
	C.5 Tall Sub-tree Properties	40		

1 Introduction

It is known from the '80s that Shor's algorithms [Sho94] on a powerful enough quantum computer solves in polynomial time the hidden subgroup problem for abelian groups of which Factoring and Discrete Logarithm (DL) are particular instances. This poses a menace to existing public-key cryptography and has motivated the exploration and study of post-quantum cryptographic problems.

One potential candidate are hard homogeneous spaces (HHS), where a finite group \mathbb{G} acts on a set \mathcal{E} through an action $\star : \mathbb{G} \times \mathcal{E} \to \mathcal{E}$ which is hard to invert. These resemble prime order groups where the group action corresponds to the exponentiation. However, unlike groups, in HHS the set \mathcal{E} lacks a group structure, making it immune to Schor's algorithms. So far the only known practical and versatile instantiations come from the Commutative Supersingular Diffie-Hellman (CSIDH) key-exchange, based on isogenies between supersingular elliptic curves. Remarkably, recent attacks [CD23, MMP⁺23, Rob23] directed at SIDH, a different family of isogeny-based key exchange, did not affect the security of CSIDH.

Over the last years CSIDH has been used to build a variety of primitives. Examples include cryptosystems [Sto12, MOT20, FP21], signatures [DG19, BKV19, EKP20, ABCP23a] and identification schemes [BCP21] as well as more advanced primitives such as ID-based signatures [SD21], adaptor signatures [TMM21], oblivious transfer [LGD21], linkable ring signatures [BKP20], group signatures [BDK⁺22] and importantly threshold schemes [DM20, CS20, BDPV21, CM22, ABCP23a, ABCP23b, ABCP23c].

HHS-based threshold schemes in particular have gained a lot of attention as post-quantum alternatives to replace currently deployed solutions. Typically protocols based on group action involve the evaluation of $s \star E$ for a secret shared $s \in \mathbb{G}$ and a public set element E. However, as opposed to the DL group setting, parties with shares s_i cannot simply compute $s_i \star E$ locally and aggregate them to get $s \star E$ in a single round due to the lack of group structure in \mathcal{E} . Because of such a limitation, only two orthogonal approaches are know to date.

One is through generic multi-party computation (MPC) techniques. These require to express the group action as an explicit arithmetic circuit and then compute it gate-by-gate, theoretically achieving constant round complexity in the number of parties. The downside though is that either round or communication complexity highly depends on the multiplicative depth of the circuit, for instance when using protocol based respectively on linear secret sharing schemes or garbled circuits. Unfortunately, currently known circuits for CSIDH (Algorithm 2 in [CLM⁺18]) are not MPC-friendly, as they involve extensive looping over secret shared values. Hence the MPC approach does not appear to be practical yet.

On the other hand there are *round-robin* protocols. These require users to sequentially apply (a function of) their secret share on a given set element. An example to compute $s = s_1 + \ldots + s_4$ acting on E_0 among 4 parties is given in Figure 1. The main advantage of such procedures is that they only make blackbox usage of the HHS, avoiding the overheads of generic MPC. However, they require as many rounds as users involved in the computation, which in a t out

of n secret sharing would amount to t rounds. This solution then appears hard to scale.

$$E_0 \underbrace{\xrightarrow{s_1}}_{P_1} E_1 \underbrace{\xrightarrow{s_2}}_{P_2} E_{21} \underbrace{\xrightarrow{s_3}}_{P_3} E_{321} \underbrace{\xrightarrow{s_4}}_{P_4} E_{4321} = s \star E_0$$

Fig. 1. Round-robin protocol for four parties. Here $E_{i_1i_2...i_t} = (s_{i_1} + s_{i_2} + \cdots + s_{i_t}) \star E_0$

Given the current unsatisfactory state of the art, we ask whether best of both worlds constructions exist. In other words:

Is it possible to securely compute the group action of a t out of n secret shared group element through black-box usage of the HHS with less than t rounds of communication?

1.1 Our Contributions

We answer the above question in the negative proving the following lower bounds:

Round Lower Bound. We show that any protocol which, given a t out of n secret shared s, attempts to compute $f(s) \star E_0$ making only black-box usage of the group action either:

- Requires t or more rounds of communication³
- Is insecure against t-1 passively corrupted users, which can recover f(s).

The notion of black-box group action is formalized through Shoup's Generic Action Model (GAM) recently proposed in [DHK⁺23]. Here set elements are represented by random labels, and actions are computed through oracle calls to \mathcal{O}_{act} . We prove this by describing for any k round protocol an explicit PPT adversary which, provided a transcript and the secret inputs of k users, recovers f(s). Our result cannot therefore be circumvented by

- Using external hardness assumptions. Our adversary is PPT, not only polynomially bounded in the number of queries to \mathcal{O}_{act} , and it could be used by any reduction to efficiently break the underlying assumption.
- Using element's representation, because this is already allowed in the GAM.

The only way we see to avoid our lower bound is through explicit usage of a circuit for the group action. In this case generic MPC techniques can be applied to improve asymptotic round complexity, however incurring in the overheads discussed previously.

³ Including one round to broadcast the result to other users.

Round-optimal Fair Protocols Lower Bounds. Although we prove round-robin protocols to be optimal in the GAM, in concrete applications they present some vulnerabilities. The most notable one is that only the last user receives the result. That party may then refuse to share it with the others or an adversary could cause him to crash, forcing the protocol to restart⁴. For this reason variants have been proposed [DM20] in which all parties get the output in the same round (we call such protocols *fair*).

A trivial solution would be to run n parallel round robin protocols (see Figure 2 for an example with n = 4), but this would require $O(n^2)$ communication and computation.

$$E_{0} \xrightarrow{s_{1}} E_{1} \xrightarrow{s_{2}} E_{21} \xrightarrow{s_{3}} E_{321} \xrightarrow{s_{4}} E_{4321} = s \star E_{0}$$

$$E_{0} \xrightarrow{s_{2}} E_{2} \xrightarrow{s_{3}} E_{32} \xrightarrow{s_{4}} E_{432} \xrightarrow{s_{1}} E_{1432} = s \star E_{0}$$

$$E_{0} \xrightarrow{s_{3}} E_{3} \xrightarrow{s_{4}} E_{43} \xrightarrow{s_{1}} E_{143} \xrightarrow{s_{2}} E_{2143} = s \star E_{0}$$

$$E_{0} \xrightarrow{s_{4}} E_{4} \xrightarrow{s_{1}} E_{14} \xrightarrow{s_{2}} E_{214} \xrightarrow{s_{3}} E_{3214} = s \star E_{0}$$

Fig. 2. Fair Round-robin protocol for four parties. Naive approach.

A better one, the binary splitting strategy [DM20], allows computing only $n(1 + \log_2 n)$ group actions and communicating $n(1 + \log_2 n) - 2$ set elements, at least when the number of parties is a power of two. An example for n = 4 is shown in Figure 3.

$$E_{0} \xrightarrow{s_{1}} E_{3} \xrightarrow{s_{2}} E_{21} \xrightarrow{s_{3}} E_{321} \xrightarrow{s_{4}} E_{4321} = s \star E_{0}$$

$$E_{0} \xrightarrow{s_{3}} E_{3} \xrightarrow{s_{4}} E_{43} \xrightarrow{s_{1}} E_{421} \xrightarrow{s_{3}} E_{3421} = s \star E_{0}$$

$$E_{143} \xrightarrow{s_{2}} E_{2143} = s \star E_{0}$$

$$E_{243} \xrightarrow{s_{1}} E_{1243} = s \star E_{0}$$

Fig. 3. Fair round-robin protocol for four parties. Optimal strategy.

We show that in the GAM this strategy is optimal for n a power of two, assuming the protocol does not allow n-1 honest-but-curious parties to recover f(s). More precisely, we prove that any n rounds protocol⁵ in the GAM computing $s \star E_0$ among n parties, with s being a function of user's private input, satisfying

- No set of n-1 can recover s

⁴ Or, in honest majority, to publicly reconstruct this user's secret share.

⁵ Due to our first result, this is also the best round complexity in this case.

- All users obtain the output in the last round

involves at least $n(1 + \log_2 n)$ action evaluations and the communication of at least $n(1 + \log_2 n) - 2$ set elements. As for the previous result, this lower bound holds in spite of the computational assumptions and the usage of element's representation.

1.2 Our Techniques

Our results are both based on the fact that in the GAM every computed set element is either randomly sampled or returned by the oracle call $\mathcal{O}_{act}(a, D)$ with $a \in \mathbb{G}$ and a previously computed $D \in \mathcal{E}$. This allows us to construct a graph among set elements⁶ where there is an edge from D to E if D was computed before E and $E = \mathcal{O}_{act}(a, D)$. In particular if there exists a path from D to E then, knowing the group elements used in all queries associated to the edges of this path, one can compute $a \in \mathbb{G}$ such that $a \star D = E$ by taking their sum.

Round-Robin Lower Bound. Our proof consists of two steps. First, we show that a path from E_0 to $f(s) \star E_0$ exists with high probability. Second, we prove that among all paths from E_0 to $f(s) \star E_0$ there exists w.h.p. one involving for each round queries performed by at most one user. Because the protocol has k rounds, this path can involve queries of at most k users. Our result then follows because these k users can then jointly compute f(s) as described above.

The first step, which we call Sequentiality Lemma⁷, is proved observing that the output is either connected to E_0 or to a randomly sampled element E'. In the first case, we get the thesis. In the second case, all parties can jointly compute $a \in \mathbb{G}$ such that $a \star E' = E_{out}$ as observed. However $E_{out} = f(s) \star E_0$ implies that $E' = (-a + f(s)) \star E_0$, meaning that parties successfully inverted the group action for a random element E', which is supposed to be hard. Hence this case only happens with negligible probability.

The main idea in the second step instead is that the order in which parties are executed within a round does not affect the protocol. A path containing queries of at most one user for each round can then be constructed by induction. Given F a set element computed by P_i at round r, we can assume P_i was executed before any other user (so its queries occurs *before* any other user's ones in the same round). In this way, given a path from E_0 to F, all the queries in the path at round r can only come from P_i . The remaining queries at previous rounds can then be handled through the inductive hypothesis.

Optimality of Fair Round-Robin. Regarding fair protocols, we exploit the fact that all parties eventually return the output E_{out_i} . A first approach to bound computation would be to find a path from E_0 to each E_{out_i} , and count the total

⁶ More precisely we later introduce a relation " \rightarrow " among query indices.

⁷ Because it implies that computing $a \star E_0$ can only be done through sequential applications of the group action starting from E_0 .

number of queries involved. If we only had one path this would work: Since we assume the protocol to have n rounds and that no subset of n-1 users can recover f(s), all users must contribute to this path by performing at least a query as discussed above. As a consequence each path contains $\Omega(n)$ queries. The issue however is that when many paths are considered, they may potentially share a large fraction of associated queries.

We address through the following critical abstraction: To each path connecting E_0 with E_{out_i} we associate the sequence $\pi_i(1), \ldots, \pi_i(n)$ of users performing queries in this path. Up to carefully choosing the paths, $\pi_i(r)$ is then the index of the only user performing queries at round r which contributes to the chosen path from E_0 to E_{out_i} . Our key observation is then that⁸ two paths can share the same queries at round r only if π_i and π_j share the same prefix of length r. In other words, shared queries only occurs if for the first r rounds the two paths involve the same users in the same order.

With this crucial result we can then link the total number of queries to properties of the *prefix tree* of π_1, \ldots, π_n . We then shift our focus to the study of this graph. Because π_1, \ldots, π_n can be proved to be permutations, their prefix tree must satisfy what we call the *tall sub-tree* (TS) property. Namely a tree is *tall* if all leaves have the same distance from the root and its height is greater than the number of leaves. A tree where all its sub-trees are tall is TS. We eventually translate our communication and computation lower bounds into bounds for TS trees and prove them with elementary graph-theoretic techniques.

Interestingly, the connection between TS trees and optimal fair protocols turned out to be more than a mere proof artifact: in the Appendix, Section B, we informally show how from any optimal TS tree one can derive optimal fair protocols to reconstruct an additive secret sharing over commutative group actions (such as CSIDH).

1.3 Related Work

The study of idealized models of computation dates back to the formalization of Shoup's [Sho97] and Maurer's [Mau05] Generic Group Models, where in the former group elements are labeled with random strings while in the latter no label is ever provided. In these frameworks many primitives were shown to be impossible including VDFs [RSS20], efficient accumulators [SGS20], signature schemes [DHH⁺21, CFGG22], vector commitments [CFGG22] and NIZKs [Giu23]. All these results are limited to Maurer's GGM either because the primitive is known to exists in Shoup's GGM, or because negative results in Shoup's GGM are typically harder, such as the IBE impossibility [PRV12, Zha22]. Moreover all these negative results are proved providing an *unbounded* adversary restricted to perform at most polynomially many GGM queries. This means that using non group theoretic assumptions could be enough to bypass them. We remark that our results instead do not suffer from such a limitation.

⁸ up to choosing paths satisfying a rather technical minimality condition.

Regarding group actions, a Shoup-like model was proposed in [DHK⁺23] capturing the concrete case of CSIDH, in which quadratic twists are efficiently computable, through an external oracle. Our result is based on their model, but we will not use for simplicity their twisting oracle. Instead we capture twists by allowing non-commutative and non-free actions as proposed in [BGZ23]. We discuss this further in the Appendix, Section A.

Conversely a Maurer-like model for group actions was proposed in [BGZ23] and used to prove lower bounds on identification schemes. These apply to signature and zero-knowledge proofs obtained via the Fiat-Shamir transform. We stress that like any negative result in Maurer-like models, their bounds on interactive identification schemes only hold if set elements representation is never used. We again remark that this is not the case in our results.

2 Preliminaries

2.1 Notation

 λ is the security parameter and a function of λ is *negligible* if it approaches zero faster than the inverse of any polynomial. For a Turing machine $\mathcal{A}, y \leftarrow \mathcal{A}(x)$ means y was *deterministically* computed on input x, whereas $y \leftarrow^{\$} \mathcal{A}(x)$ means y was computed *probabilistically*. These must not be confused with " \rightarrow " and " \rightarrow ", two relations we define in Sections 3.1 and 3.3. $x \sim \mathcal{X}$ is a random variable with support \mathcal{X} . H(x) is the Shannon's entropy of x and I(x; y) is the mutual information between x and y, defined as H(x) - H(x|y). $x \leftarrow^{\$} \mathcal{X}$ means x is uniformly sampled from \mathcal{X} . For a finite group, denoted by ($\mathbb{G}, +$), we always use additive notation and denote by 0 the identity element. We will not implicitly assume groups to be commutative or prime order.

2.2 Hard Homogeneous Space

Here we recall the notion of possibly non-commutative group actions.

Definition 1. An action of a finite group $(\mathbb{G}, +)$ on a set \mathcal{E} is given by a map $\star : \mathbb{G} \times \mathcal{E} \to \mathcal{E}$ satisfying the following properties:

- 1. Identity: $0 \star E = E$ for all $E \in \mathcal{E}$.
- 2. Associativity: $(h+g) \star E = h \star (g \star E)$ for every $h, g \in \mathbb{G}$ and $E \in \mathcal{E}$.

The action is transitive if for all $E_1, E_2 \in \mathcal{E}$ there exists $g \in \mathbb{G}$ such that $E_2 = g \star E_1$. If g is unique the action is called free.

For a group action to be *effectively computable* we further assume that deciding membership in \mathbb{G} , \mathcal{E} and equality between elements in \mathbb{G} and \mathcal{E} is efficient. Moreover sampling uniformly from both sets, computing the operations and inverses in \mathbb{G} , and the group action \star is also efficient.

Typically, the set \mathcal{E} equipped with an effective group action \star is called *Homo*geneous Space if \star is transitive and free. However we will extend this notion to actions that are not necessarily free. This, in line with the approach of [BGZ23], allows for instance encoding quadratic twists for CSIDH as the action of specific group elements, as discussed in the Appendix, Section A.

For a PPT \mathcal{A} adversary we define the advantage for the *vectorization* and *parallelization* problems as

$$\begin{aligned} \mathsf{Adv}^{\mathsf{Vec}}\left(\mathcal{A}\right) &= \Pr\left[g = g' \mid \begin{array}{c} g \leftarrow^{\$} \mathbb{G}, \ E \leftarrow^{\$} \mathcal{E}, \ E' = g \star E \\ g' \leftarrow \mathcal{A}(E, E') \end{array}\right], \\ \mathsf{Adv}^{\mathsf{Par}}\left(\mathcal{A}\right) &= \Pr\left[E_t = (g + h) \star E \mid \begin{array}{c} g, h \leftarrow^{\$} \mathbb{G}, \ E \leftarrow^{\$} \mathcal{E} \\ E_g \leftarrow g \star E, \ E_h \leftarrow h \star E \\ E_t \leftarrow \mathcal{A}(E, E_g, E_h) \end{array}\right]. \end{aligned}$$

Definition 2. An homogeneous space is called hard if the vectorization and parallelization problems are hard. That is, if for any PPT adversary \mathcal{A} there exists negligible functions ε_{vec} , ε_{par} such that

$$\operatorname{Adv}^{\operatorname{Vec}}(\mathcal{A}) \leq \varepsilon_{\operatorname{vec}}(\lambda), \qquad \operatorname{Adv}^{\operatorname{Par}}(\mathcal{A}) \leq \varepsilon_{\operatorname{par}}(\lambda).$$

2.3 Shoup's Generic Action Model

In order to capture generic usage of the group action, we use an adaptation of Shoup's Generic Group Model [Sho97]. In this setting the group action \star : $\mathbb{G} \times \mathcal{E} \to \mathcal{E}$ is modeled through an oracle \mathcal{O}_{act} . Initially a random injective labeling function $\sigma : \mathcal{E} \to \{0,1\}^{\mu}$ is sampled and users receive $E_0 = \sigma(E'_0)$ the encoding of an element in \mathcal{E} . Action queries are then replied to with

$$\mathcal{O}_{\mathsf{act}}(a, E) = \begin{cases} \sigma(a \star E') & \text{If } E = \sigma(E') \\ \bot & \text{If } E \notin \operatorname{Im} \sigma \end{cases}$$

We do not provide an oracle to test membership in $\sigma(\mathcal{E})$ as this can be checked querying $\mathcal{O}_{act}(0, E)$, which returns E if $E \in \sigma(\mathcal{E})$ and \perp if $E \notin \sigma(\mathcal{E})$. We further remark that if $\mu = \log_2 |\mathcal{E}| + O(\log \lambda)$ the model allows sampling random elements of unknown "discrete logarithm" in base E_0 . Conversely if $\mu = \log_2 |\mathcal{E}| + \Omega(\lambda)$ sampling random elements is computationally hard.

In relation to previously proposed models [DHK⁺23, BGZ23] for generic group actions, ours allows parties to have an explicit representation for set elements as done in [DHK⁺23] and as opposed to [BGZ23]. However, as in [BGZ23] we allow non-free and non-commutative group actions in order to encode external operations such as quadratic twists as particular action evaluations. In this sense we do not need a separate oracle to capture twists as done in [DHK⁺23].

3 Technical Lemmas

3.1 Sequentiality Lemma

In this section we present our starting point, the Sequentiality Lemma, stating that any procedure computing $s \star E_0$ can obtain the right result only through

sequential applications of the group action to E_0 . A full proof of this Lemma can be found in the Appendix, Section C.1.

First let us introduce some notation. Given \mathcal{A} a PPT algorithm with oracle access to $\mathcal{O}_{\mathsf{act}}$ initially receiving E_0 and performing q queries, we denote these with $E_k \leftarrow \mathcal{O}_{\mathsf{act}}(a_k, D_k)$ for $k \in \{1, \ldots, q\}$. Next we introduce a relation \rightarrow among indices $\{0, \ldots, q\}$.

$$k_1 \rightarrow k_2 \quad \Leftrightarrow \quad k_1 < k_2, \quad E_{k_1} = D_{k_2}.$$

Intuitively this means that the k_1 -th set element was used to compute the k_2 -th. Note this relation is not yet a (strict) partial order as it is not transitive, but its transitive closure is. This is explicitly defined as

$$i \rightarrow j \Leftrightarrow \exists k_1, \dots, k_m : k_1 \rightarrow k_2 \rightarrow \dots \rightarrow k_m, \quad k_1 = i, \quad k_m = j$$

The lemma then ensures that if an algorithm computes $s \star E_0$ for a known s in the GAM, with high probability $s \star E_0$ is the output of some query, say the k-th (meaning $s \star E_0 = E_k$), and $0 \to k$.

Lemma 1. For any \mathcal{A} PPT algorithm with oracle access to $\mathcal{O}_{\mathsf{act}}$ making at most q queries and any $s \in \mathbb{G}$, such that $(s, E_{\mathsf{out}}) \leftarrow \mathcal{A}(E_0)$ then

$$\Pr\left[E_{\mathsf{out}} = s \star E_0, \ \nexists k \left(E_{\mathsf{out}} = E_k, \ 0 \to^+ k\right)\right] \leq \varepsilon_{\mathsf{vec}}(2q) + \frac{1}{|\mathcal{E}| - (q+1)} \coloneqq \varepsilon_{\mathsf{seq}}(q)$$

where $\varepsilon_{\text{vec}}(q)$ is the advantage of breaking the vectorization problem in q queries, see Definition 2.

3.2 Interactive Protocols

The main limitation of the Sequentiality Lemma is that it only applies to a single machine. Here we introduce notation for interactive protocols in the GAM in order to extend in the next section this result to the interactive case.

An interactive protocol is defined by n PPT machines P_1, \ldots, P_n with access to point-to-point (i.e. non-broadcast⁹) communication channels. We assume them to be synchronous and simultaneous i.e. such that messages from all parties are atomically sent at the beginning of each round and delivered at the end. To formally describe this model we initially call $trs_0 = \bot$ and inductively define for initial inputs x_1, \ldots, x_n the messages sent and transcript at round r as

$$M_{r,i} \leftarrow^{\$} P_i(x_i, \mathsf{trs}_{r-1}), \quad \mathsf{trs}_r = (\mathsf{trs}_{r-1}, M_{r,1}, \dots, M_{r,n})$$

where $M_{r,i} = (M_{r,i}^{(1)}, \ldots, M_{r,i}^{(n)})$ is the tuple of messages sent by P_i , and in particular $M_{r,i}^{(j)}$ is the message P_i sends to P_j . To ensure parties only use message

⁹ Even though broadcast could be achieved simply sending the same message to all parties, as we only focus on semi-honest adversaries.

delivered to them, we assume P_j can only read entries of the form $M_{r,i}^{(j)}$ in trs for any r and i.

Regarding the interaction with the GAM we denote $E_{r,i,j} \leftarrow \mathcal{O}_{act}(a_{r,i,j}, D_{r,i,j})$ the *j*-th query made by P_i to \mathcal{O}_{act} during the *r*-th round. As done in Section 3.1 we then define a relation \rightarrow among the indices (r, i, j) and (r', i', j') which indicates that the result from the former was used as input in the latter. Formally

$$(r,i,j) \rightarrow (r',i',j') \quad \Leftrightarrow \quad (r < r' \lor (r = r', i = i', j < j')) \land E_{r,i,j} = D_{r',i',j'}$$

This condition on the indices says that one query precedes another one only if it was performed on a previous round, or if both were asked in the same round by the same party P_i one before the other. Like \rightarrow , the relation \rightarrow is not yet a (strict) partial order. However its transitive closure \rightarrow is. This is explicitly defined as $(r, i, j) \rightarrow^+ (r', i', j') \Leftrightarrow$

$$\Rightarrow \exists (r_1, i_1, j_1), \dots, (r_t, i_t, j_t) : (r_1, i_1, j_1) \to \dots \to (r_t, i_t, j_t), \quad (r_1, i_1, j_1) = (r, i, j), \quad (r_t, i_t, j_t) = (r', i', j').$$

Finally, to further include E_0 in this relation, which might be technically never queried, we could either assume that parties initially query $E_0 \leftarrow \mathcal{O}_{\mathsf{act}}(0, E_0)$, or more simply say that

$$0 \rightarrow (r, i, j) \Leftrightarrow D_{r, i, j} = E_0.$$

We conclude this section with two elementary properties of \rightarrow . The proofs appears for completeness in the Appendix, Section C.2.

Lemma 2. Let $(r_1, i_1, j_1) \rightarrow \ldots \rightarrow (r_t, i_t, j_t)$. Then

 $E_{r_t, i_t, j_t} = (a_{r_t, i_t, j_t} + \ldots + a_{r_1, i_1, j_1}) \star D_{r_1, i_1, j_1}.$

Lemma 3. Let $0 \Rightarrow (r_1, i_1, j_1) \Rightarrow \ldots \Rightarrow (r_t, i_t, j_t)$. Then

$$|\{i_1,\ldots,i_t\}| \le |\{r_1,\ldots,r_t\}|.$$

3.3 Interactive Sequentiality Lemma

In this section we state the Interactive Sequentiality Lemma, which extends Lemma 1. This applies to a set of parties P_1, \ldots, P_n each holding an input $s_i \in \{0, 1\}^{\mathsf{poly}(\lambda)}$ and wishing to compute $f(s_1, \ldots, s_n) \star E_0$ for a given function f. Informally it states that the result must, up to negligible probability, come from the sequential application of $\mathcal{O}_{\mathsf{act}}$ to E_0 and that such sequence of queries involves at most one player for each round.

Lemma 4. Let P_1, \ldots, P_n be a k round protocol in the GAM and f a function such that given inputs $s_1, \ldots, s_n \sim \{0, 1\}^{\operatorname{poly}(\lambda)}$ there exists P_i which at round k returns $E_{\operatorname{out}} = f(s_1, \ldots, s_n) \star E_0$. Then

$$\Pr\left[\exists (r', i', j') : E_{\mathsf{out}} = E_{r', i', j'}, \quad 0 \to^+ (r', i', j')\right] \ge 1 - (k+1) \cdot \varepsilon_{\mathsf{seq}}(q)$$

where q is an upper bound on the total number of queries performed.

A full proof of this lemma appears in the Appendix, Section C.3.

4 Round Lower Bound

We now present our first result for distributed computation over black box HHS. We assume that parties P_1, \ldots, P_n initially receive a secret input $s_i \in \{0, 1\}^{\mathsf{poly}(\lambda)}$ and a public function f and execute a protocol compute $f(s_1, \ldots, s_n) \star E_0$ in Shoup's GAM. In this setting then we will prove that if such computation only requires k rounds, then a subset of k users can passively collude, and recover $f(s_1, \ldots, s_n)$.

Evaluating the group action of a t out of n secret shared group element $s \in \mathbb{G}$ is then a specific case of interest, as it affects threshold signatures and cryptosystems. This is captured by our result setting f as the reconstruction function. More concretely, for n out of n additive secret sharing, $f(s_1, \ldots, s_n)$ is simply the sum of all shares. Similarly, for t out of n Shamir secret sharing f is

$$f(s_1,\ldots,s_n) = \sum_{i\in R} \lambda_{i,R} s_i$$

with R being a reconstruction set of size t and $\lambda_{i,R} \in \mathbb{Z}$ the Lagrange coefficients¹⁰. In all these cases, since any set with less than t users is not entitled to recover s, our result implies that passive security cannot be achieved with less than t rounds. Finally, this will further imply the optimality of round-robin protocols in such cases.

Theorem 1. Let P_1, \ldots, P_n be a k-round protocol in the GAM and f a function such that on input $s_1, \ldots, s_n \sim \{0, 1\}^{\mathsf{poly}(\lambda)}$ there exists P_i returning at round k the element $E_{\mathsf{out}} = f(s_1, \ldots, s_n) \star E_0$.

Then up to probability $(k + 1)\varepsilon_{seq}$ there exists $S \subseteq \{1, \ldots, n\}$ and a PPT machine \mathcal{A} such that, calling ρ_i the random coins of P_i

1.
$$|S| \leq k$$

2. $s' \leftarrow \mathcal{A}(\mathsf{trs}, \{s_i, \rho_i\}_{i \in S})$ with $s' \star E_0 = E_{\mathsf{out}}$.

Proof. We begin by applying the *Interactive Sequentiality Lemma* 4, stating that up to probability $(k+1)\varepsilon_{seq}$ there exists (r, i, j) such that $E_{out} = E_{r,i,j}$ and $0 \rightarrow^+ (r, i, j)$. Next let $(r_1, i_1, j_1), \ldots, (r_t, i_t, j_t) = (r, i, j)$ be a chain for the above relation, i.e.

$$0 \rightarrow (r_1, i_1, j_1) \rightarrow \ldots \rightarrow (r_t, i_t, j_t).$$

We define $S = \{i_1, \ldots, i_t\}$ the set of users involved in the chain¹¹. To upper bound the size of S we use Lemma 3: because the protocol has k round, $|S| \leq |\{r_1, \ldots, r_t\}| \leq k$. Next we provide an explicit description of \mathcal{A} computing s'from $\{s_i, \rho_i\}_{i \in S}$ and trs. Initially \mathcal{A} executes P_i with $i \in S$ for all rounds. In this way it performs all queries $(r_1, i_1, j_1), \ldots, (r_t, i_t, j_t)$ and in particular knows the group elements used in those queries. The sum s' of all these group elements then is such that $s' \star E_0 = E_{\text{out}}$. A formal description is provided in Figure 4.

¹⁰ These can be defined for \mathbb{G} if \mathbb{Z}_N has an *exceptional set* of size at least n, with N being the order of \mathbb{G} .

¹¹ Note |S| may be smaller than t if there are repetitions among i_1, \ldots, i_t .

- 1: Compute $trs_0, trs_1 \dots, trs_r$ from trs
- 2: // Execute all users in S. Note that at round r' users get $\operatorname{trs}_{r'-1}$
- 3: For all $i' \in S$ and $r \in \{1, \ldots, k\}$:
- 4: Run $P_{i'}^{\mathcal{O}_{\mathsf{act}}}(s_{i'}, \mathsf{trs}_{r'-1}; \rho_{i'})$
- 5: // Compute s' from the users' queries
- 6: Find $(r_1, i_1, j_1), \ldots, (r_t, i_t, j_t)$ such that $0 \rightarrow (r_1, i_1, j_1) \rightarrow \ldots \rightarrow (r_t, i_t, j_t)$
- 7: Retrieve queried group elements $a_{r_1,i_1,j_1}, \ldots, a_{r_t,i_t,j_t}$
- 8: **Return** $s' \leftarrow a_{r_t, i_t, j_t} + \ldots + a_{r_1, i_1, j_1}$

Fig. 4. Adversary \mathcal{A} computing s' such that $s' \star E_0 = E_{out}$

Since the query $(r_{\alpha}, i_{\alpha}, j_{\alpha})$ for $\alpha \in \{1, \ldots, t\}$ is performed by $P_{i_{\alpha}}$ at round r_{α} , then \mathcal{A} also performs this query as by construction $i_{\alpha} \in S$ and \mathcal{A} runs $P_{i_{\alpha}}^{\mathcal{O}_{act}}(s_{i_{\alpha}}, \operatorname{trs}_{r_{\alpha}-1}; \rho_{i_{\alpha}})$ in line 4. Finally, by Lemma 2

$$s' \star E_0 = (a_{r_t, i_t, j_t} + \ldots + a_{r_1, i_1, j_1}) \star E_0 = E_{r_t, i_t, j_t} = E_{out}$$

This completes the proof.

5 Fair Protocols Lower Bounds

5.1 Fair Protocols

As proven in the previous section, round-robin protocols achieve the best round complexity in the GAM. These however do not achieve *fairness* even against weak adversaries. Indeed, since only the last user gets the result, it can simply halt instead of communicating it to others. Remarkably, in order to carry out this attack, an adversary only needs to be able to

- corrupt only 1 user.
- deviate from the protocol only through crashes.

Moreover, it does not even have to be *rushing*, i.e. able to receive for each round honest users' messages before computing and sending its own.

In this section we will study protocols that address this issue, and eventually provide communication and computation lower bounds for them in the GAM. More specifically we focus on protocols among n users to compute a function of all parties' private inputs acting on a given set elements such that:

- 1. it prevents n-1 honest-but-curious users from reconstructing the secret,
- 2. in honest executions, all parties obtain their output in the last round,
- 3. it requires exactly n rounds of communication (i.e. it is round optimal according to Theorem 1).

We immediately observe that these simple restrictions, the second one being necessary for fairness in general, imply fairness against the weak class of attacks described above. The idea is that if an adversary obtains the output before the n-th round, and then crashes the corrupted party, then using Theorem 1, one could find a subset of n-1 or less users who are able to recover the secret group element. Conversely, if it crashes at round n, as we assumed it not to be rushing, it can only halt after sending its own messages. Hence all parties eventually get their output as well.

Noticeably, the above argument does not imply that fair protocols have to be round-optimal. Even more so, as our lower bounds will only apply to roundoptimal protocols, this leaves open the possibility that fair solutions with suboptimal round complexity but better communication and computational costs exist. Since our techniques do not seem to easily generalize in such case, we leave this as an interesting open question.

Finally, it may appear as uninteresting in practice to only study security against such a weak class of attacks. We remark that, as we will prove lower bounds for these protocols, our results applies to stronger models of corruption as well.

5.2 Refined Interactive Sequentiality Lemma

In order to provide our second lower bound we will need an improved version of the Interactive Sequentiality Lemma, Section 3.3. First let us recall its statement. Given n parties with inputs s_1, \ldots, s_n jointly computing $E_{\mathsf{out}} = f(s_1, \ldots, s_n) \star E_0$, Lemma 4 states that up to negligible probability, $E_{\mathsf{out}} = E_{r,i,j}$ and $0 \rightarrow^+ (r, i, j)$. Let $(r_1, i_1, j_1), \ldots, (r_t, i_t, j_t)$ be a chain of queries for $0 \rightarrow^+ (r, i, j)$, i.e. such that

$$0 \rightarrow (r_1, i_1, j_1) \rightarrow (r_2, i_2, j_2) \rightarrow \ldots \rightarrow (r_t, i_t, j_t) = (r, i, j).$$

In our improved lemma we will show that this chain can be chosen so that the first query occurring at round r_{α} is *minimal* among all queries performed in the same round r_{α} with respect to the relation \rightarrow^+ . This means that the set elements used to perform this minimal query was not computed in the same round by $P_{i_{\alpha}}$. This property will prove useful when studying communication lower bounds, as it roughly implies that the set element used in minimal queries highly depends on messages previously received. More formally we give the following definition:

Definition 3. Given P_1, \ldots, P_n PPT defining a k rounds protocol, and a sequence of queries $(r_1, i_1, j_1), \ldots, (r_t, i_t, j_t)$ such that

$$0 \rightarrow (r_1, i_1, j_1) \rightarrow \ldots \rightarrow (r_t, i_t, j_t)$$

we call this a refined chain if for all $r \in \{r_1, \ldots, r_t\}$ there exists an index α such that $(r_{\alpha}, i_{\alpha}, j_{\alpha})$ is minimal among all queries of the form $(r_{\alpha}, \cdot, \cdot)$ with respect to \rightarrow^+ .

Lemma 5. Let P_1, \ldots, P_n be a k round protocol in the GAM and f a function such that on inputs $s_1, \ldots, s_n \sim \{0, 1\}^{\mathsf{poly}(\lambda)}$, there exists P_i which at round k returns $E_{\mathsf{out}} = f(s_1, \ldots, s_n) \star E_0$. Up to probability $(k+1)\varepsilon_{\mathsf{seq}}$ then $E_{\mathsf{out}} = E_{r,i,j}$ and there exists a refined chain such that $0 \to^+ (r, i, j)$.

The proof appears in the Appendix, Section C.4.

5.3 Tall Sub-tree Property

Our technique to study fair protocols will be to associate a tree with special properties to the protocol, and translate bounds for the tree size to communication and computation lower bounds. In this section we therefore introduce the *tall sub-tree* (TS for short) property for tree graphs, and lower bound their size.

Informally a tree is *tall* if all leaves have the same distance from the root, and its height is higher than the number of leaves. A tree then satisfies the TS property if all its (non-trivial) sub-trees are tall. To be more formal we introduce some notation. Given T = (V, E) a tree, height(T) is its height (the longest path's length) and leaves(T) its number of leaves. T_v for $v \in V$ is the sub-tree rooted in v.

Definition 4. A tree T = (V, E) is tall if all leaves have the same distance from the root and either |V| = 1 or $\text{height}(T) \ge \text{leaves}(T)$. T satisfies the tall sub-tree (TS) property if T_v is tall for all $v \in V$.

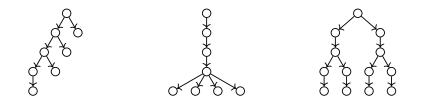


Fig. 5. Examples of non-TS (left), tall but non-TS (center) and TS (right) trees.

Proposition 1. Let T = (V, E) be a TS tree with height(T) = m and leaves(T) = n. Then

$$E| \geq m + n \log_2 n.$$

We quickly observe that because in any tree |V| = |E| + 1, the Proposition above could be restated as $|V| \ge m + 1 + n \log_2 n$. Next we prove a bound for the number of nodes of distance at least two from the root.

Proposition 2. Let T = (E, V) a TS tree with height(T) = m and leaves(T) = n. Furtherfore let $V_{\geq 2}$ the set of nodes with distance at least two from the root. Then

$$|V_{>2}| \ge m + n \log_2 n - 2.$$

Proofs for Proposition 1 and 2 appears in the Appendix, section C.5.

5.4 Main Result

We are finally ready to state and prove our second lower bound for fair protocols with optimal round complexity. Regarding our notation, we remind that μ denotes the GAM label size, i.e. the number of bits used to represent set elements, and that trs denotes the tuple of messages exchanged throughout the protocol's execution. In order to give meaningful lower bound on the communication complexity we define for the tuple of messages $M_r^{(i)}$ received by P_i at round r

$$\ell(M_r^{(i)}) \ := \ \mathrm{H}(M_r^{(i)} \,|\, \mathrm{trs}_{r-1}) \qquad \ell_{\mathrm{tot}} \ = \ \sum\nolimits_{r=1}^k \sum\nolimits_{i=1}^n \ell_{\mathrm{tot}}(M_r^{(i)}).$$

Roughly $\ell(\cdot)$ represent the amount of information contained in $M_r^{(i)}$ given all previous messages, and lower bound the information P_i receives conditioned only to messages it previously saw. Hence ℓ_{tot} lower bounds the total information sent throughout the protocol.

Theorem 2. Let P_1, \ldots, P_n be an n-round protocol in the GAM and f a function such that on input $s_1, \ldots, s_n \sim \{0, 1\}^{\mathsf{poly}(\lambda)}$, every P_i returns at last round the element $E_{\mathsf{out}_i} = f(s_1, \ldots, s_n) \star E_0$. If there exists no set $S \subseteq \{1, \ldots, n\}$ and adversary \mathcal{A} satisfying the conditions of Theorem 1 then up to probability $(n+1)\varepsilon_{\mathsf{seq}}$, calling q the total number of $\mathcal{O}_{\mathsf{act}}$ queries

$$q \ge n(1 + \log n),$$
 $\ell_{tot} \ge (n(1 + \log n) - 2) \cdot \left(\mu - \frac{q}{2^{\mu - 1} - q}\right).$

As for Theorem 1, this result readily generalizes to protocol reconstructing the action of a t out of n secret shared value, which requires exactly t rounds and at least a subset of t users get the output. In such case the protocol must involve at least $t(1 + \log_2 t)$ queries and no less than $\approx t(1 + \log_2)\mu$ bits of communication.

Proof. The proof consists of four steps:

- 1. Observing that any chain for $0 \rightarrow^+ \text{out}_i$ contains at least a query from each user. In particular we can associate to each chain a permutation π_i assigning to round r the (only) user whose round r queries appears in the chain.
- 2. Given π_1, \ldots, π_n permutations we build their *prefix tree* and show it is a TS tree, see Section 5.3. In particular it contains at least $n(1 + \log_1 n)$ nodes, excluding the root.
- 3. Using Lemma 5, we find refined chains for $0 \rightarrow^+ \text{out}_i$ so that the prefix tree of the associated permutations π_1, \ldots, π_n satisfies a certain minimality condition. Then we build an injective function f from the nodes (root excluded) to the set of query indexes. By Proposition 1 this yields $q \ge n(1 + \log n)$.
- 4. Proving each $M_r^{(i)}$ must have enough information about set element which figures as input in queries in Im f performed by P_i at round r+1. The bound on ℓ_{tot} is then a consequence of Proposition 2.

Regarding the first step, we begin with the following claim, stating that each chain for $0 \rightarrow \mathsf{out}_i$ must contain queries from all users and cannot skip any round.

Claim 1 For all $(r_1, i_1, j_1), \ldots, (r_t, i_t, j_t)$ such that

$$0 \rightarrow (r_1, i_1, j_1) \rightarrow \ldots \rightarrow (r_t, i_t, j_t) = \mathsf{out}_i$$

then $\{i_1, \ldots, i_t\} = \{1, \ldots, n\} = \{r_1, \ldots, r_t\}.$

Using this, for all chains $(r_1, i_1, j_1), \ldots, (r_t, i_t, j_t)$ for $0 \rightarrow^+ \text{out}_i$ we define a function π_i associating to each round r_α the user i_α who performed at least one query in the chain at that round

$$\pi_i: \{1, \dots, n\} \to \{1, \dots, n\} \quad : \quad \pi_i(r_\alpha) = i_\alpha \quad \forall \alpha \in \{1, \dots, t\}.$$

This is a function because \rightarrow implies there is at most one user performing queries for each round and, by Claim 1, π_i is defined for all r. In fact this is a permutation as stated in the next claim, which completes the first step.

Claim 2 For all $(r_1, i_1, j_1), \ldots, (r_t, i_t, j_t)$ chain for $0 \rightarrow^+ \text{out}_i, \pi_i$ is a bijection and $\pi_i(n) = i$.

Regarding the second step, assuming $0 \rightarrow \mathsf{out}_i$ for all i, we can define π_1, \ldots, π_n for any choice of chains realizing these relations. We then construct their prefix tree. This is done by defining for each $r \in \{0, \ldots, n\}$ an equivalence relation where π_i is equivalent to π_j if the two functions agree on the first r evaluations. Note that for r = 0 all permutations are equivalent and, due to Claim 2, for r = n no two distinct permutations are. Then the equivalence classes are

$$[\pi_i]_r = \{\pi_j : \pi_j(1) = \pi_i(1), \dots, \pi_i(r) = \pi_j(r)\}.$$

For the sake of clarity we notice that $[\pi_i]_0 = {\pi_1, \ldots, \pi_n}$ and, by Claim 2, $[\pi_i]_n = {\pi_i}$. With this notation their prefix tree T = (V, E) is defined as

$$\begin{aligned} v_{i,r} &\coloneqq ([\pi_i]_r, r) \\ V &= \{ v_{i,r} : i \in \{1, \dots, n\}, \ r \in \{0, \dots, n\} \} \\ E &= \{ (v_{i,r}, v_{i,r+1}) : i \in \{1, \dots, n\}, \ r \in \{1, \dots, n-1\} \} . \end{aligned}$$

i.e. the class $[\pi_i]_{r+1}$ is connected to the class $[\pi_i]_r$ it refines. We conclude the second step with the next claim.

Claim 3 If $0 \rightarrow t$ out_i for all $i \in \{1, ..., n\}$, then for all chains realizing them and associated permutations $\pi_1, ..., \pi_n$, their prefix tree T is a TS tree with height(T) = n and leaves(T) = n.

For the third step we use Lemma 5. Since each P_i returns $E_{\mathsf{out}_i} = f(s_1, \ldots, s_n) \star E_0$, up to probability $(n+1)\varepsilon_{\mathsf{seq}}$, for all *i* there exists a refined chain for $0 \to^+ \mathsf{out}_i$, see Definition 3. Conditioning on this event, we can chose n refined chains with associated permutations π_1, \ldots, π_n so that, calling $V_t = \{v_{i,r} : i \in \{1, \ldots, t\}, r \in \{0, \ldots, n\}\}$, the tuple

$$(|V_1|, |V_2|, \dots, |V_n|)$$

is minimal w.r.t. the lexicographic order. This means that for any other choice of refined chains, the associated permutations π'_1, \ldots, π'_n defines a prefix tree T' = (V', E') so that either $|V_t| = |V'_t|$ for all t or there exists a t such that

$$|V_1| = |V_1'| \land \ldots \land |V_t| = |V_t'| \land |V_{t+1}| < |V_{t+1}'|.$$

Using this we will construct an injective function f from the tree nodes (excluding the root) to the set of query indices. Each node $v = ([\pi_i]_r, r)$ for some π_i will be mapped to a query that is:

- in the chain for $0 \rightarrow \mathsf{out}_i$ used to construct π ,
- minimal among round r queries, with respect to \rightarrow^+ .

Claim 4 There exists refined chains for $0 \rightarrow \mathsf{out}_i$ so that, calling T = (V, E) the resulting prefix tree and V^* the set of nodes excluding the root, there exists $f: V^* \rightarrow \mathbb{N}^3$ such that

- 1. For each $v \in V$ there exist r, i, j so that $v = v_{i,r}$ and $f(v) = (r, \pi_i(r), j)$ is a query in the chain used to construct π_i . Moreover f(v) is minimal w.r.t. \rightarrow^+ among all queries of the form (r, \cdot, \cdot) .
- 2. f is injective.
- 3. Calling $a_{r,i,j}, D_{r,i,j}$ the input of $\mathcal{O}_{\mathsf{act}}$ in the (r, i, j)-th query, then

$$D_{f(u)} = D_{f(v)} \quad \Rightarrow \quad \exists w : (w, u), (w, v) \in E.$$

The first two properties combined implies that the set of queries contains at least $|V^*| = |E| \ge n(1 + \log_2 n)$ elements, where the last bound follows from Claim 3 and Proposition 1.

Finally we go through the last step. In order to bound communication we first bound the number of minimal queries using different set elements as input performed by P_i at round r. Next we will prove $M_{r-1}^{(i)}$ has to contain enough information about these elements. Toward this goal we define $U_{r,i}$ as the set of nodes to which f associate a query performed by P_i at round r, and $\Delta_{r,i}$ the collection of set elements used in those queries.

$$U_{r,i} := \{ v \in V : \exists j : f(v) = (r, i, j) \} \qquad \Delta_{r,i} := \{ D_{f(v)} : v \in U_{r,i} \}$$

First we give a bound on the size of $\Delta_{r,i}$.

Claim 5 $\sum_{r=1}^{n-1} \sum_{i=1}^{n} |\Delta_{r+1,i}| \ge n(1 + \log_2 n) - 2.$

Then, we relate the size of $\Delta_{r+1,i}$ with the entropy in $M_r^{(i)}$

Claim 6 With the previous notation,

$$\mathrm{H}\left(M_{r}^{(i)} \left| \mathsf{trs}_{r-1}\right) \right) \geq \left| \varDelta_{r+1,i} \right| \cdot \left(\mu - \frac{q}{2^{\mu-1} - q} \right)$$

This eventually concludes the proof of Theorem 2 because

$$\ell_{\text{tot}} = \sum_{r=1}^{n-1} \sum_{i=1}^{n} \ell(M_r^{(i)})$$

$$\geq \sum_{r=1}^{n-1} \sum_{i=1}^{n} |\Delta_{r+1,i}| \cdot \left(\mu - \frac{q}{2^{\mu-1} - q}\right)$$

$$\geq (n(1 + n \log_2 n) - 2) \cdot \left(\mu - \frac{q}{2^{\mu-1} - q}\right).$$

Proof of Claim 1. Assume by contradiction that there exists a chain for $0 \rightarrow^+$ out_i such that $S := \{i_1, \ldots, i_t\} \subseteq \{1, \ldots, n\}$. Then, as shown in the proof of Theorem 1, the adversary \mathcal{A} described in Figure 4 on input trs and $(s_i, \rho_i)_{i \in S}$, with ρ_i being the random coins of P_i , recovers s' such that $s' \star E_0 = E_{\text{out}_i}$. This contradicts the assumption that such a pair (S, \mathcal{A}) does not exist.

Next, using Lemma 3 and the fact that the protocol has n rounds,

$$n = |\{i_1, \dots, i_t\}| \le |\{r_1, \dots, r_t\}| = n \quad \Rightarrow \quad \{r_1, \dots, r_t\} = \{1, \dots, n\}.$$

Proof of Claim 2. We begin showing that π_i is a total function from $\{1, \ldots, n\}$. Let $\alpha < \beta$ be two indexes such that $r_\alpha = r_\beta$. Because $(r_\alpha, i_\alpha, j_\alpha) \rightarrow^+ (r_\beta, i_\beta, j_\beta)$, by the definition of \rightarrow^+ , $r_\alpha = r_\beta$ implies $i_\alpha = i_\beta$. Hence π_i associate the same value to r_α and r_β . Moreover, by Claim 1, for each $r \in \{1, \ldots, n\}$ there exists an α such that $r = r_\alpha$. As a consequence π_i is well defined function with domain $\{1, \ldots, n\}$.

Next we observe that $\operatorname{Im} \pi_i = \{i_1, \ldots, i_t\} = \{1, \ldots, n\}$ where we used again Claim 1, implying that $\pi_i : \{1, \ldots, n\} \to \{1, \ldots, n\}$ is a surjective function between finite sets of the same size, and therefore also a bijection.

Finally, since the query out_i is performed by P_i , it has the form (r_t, i, j_t) . If $r_t < n$ then we would find a chain with $\{r_1, \ldots, r_t\}$ of size strictly smaller than n, contradicting Claim 1. Therefore $r_t = n$ and $\pi_i(n) = \pi_i(r_t) = i$.

Proof of Claim 3. T is a tree because each node $v_{i,r}$ is connected to the root by the path

$$((v_{i,0}, v_{i,1}), \ldots, (v_{i,r-1}, v_{i,r})),$$

with $v_{i,0} = (\{\pi_1, \ldots, \pi_n\}, 0)$ being equal for all *i*, and each node has in-degree 1 because

$$\begin{array}{ll} (v_{j,r-1},\,v_{i,r})\,,\;(v_{k,r-1},\,v_{i,r})\in E &: & \begin{cases} v_{i,r}=([\pi_i]_r,\,r)\\ v_{j,r-1}=([\pi_j]_{r-1},\,r-1)\\ v_{k,r-1}=([\pi_k]_{r-1},\,r-1) \end{cases} \\ \Rightarrow & \pi_j(x)=\pi_i(x)=\pi_k(x) \quad \forall x\in\{1,\ldots,r-1\}\\ \Rightarrow & [\pi_j]_{r-1}=[\pi_k]_{r-1} \Rightarrow \quad v_{j,r-1}=v_{k,r-1}. \end{array}$$

By construction the leaves of T are $v_{i,n}$ with i ranging from 1 to n, and these are all distinct. Indeed for $i \neq j$ we have $\pi_i(n) = i \neq j = \pi_j(n)$ implying $v_{i,n} \neq v_{j,n}$. Thus leaves(T) = n. Moreover each leaf has distance n from the root, so height(T) = n.

Next we show that the sub-tree of $v_{i,r} = ([\pi_i]_r, r)$ is a tall tree. By previous observations its height is n - r. If r = n the node is a leaf and it is trivially tall. Conversely let v_{j_1}, \ldots, v_{j_n} be all the leaves of this sub-tree, so that $v_{j_\alpha} = ([\pi_{j_\alpha}]_n, n)$. Then for all α we have that $v_{i,r} = v_{j_\alpha,r}$ because from both nodes there exists a path to $v_{j_\alpha,n}$, and there exists no path connecting the two nodes. As a consequence

$$[\pi_i]_r = [\pi_{j_1}]_r = \dots = [\pi_{j_m}]_r.$$

Hence $\pi_{j_1}, \ldots, \pi_{j_m}$ have the same values when evaluated on the indexes from 1 to r. Moreover, since these are all permutations, their value on n (as we assumed n > r) must differ from their value on previous points. Thus

$$\{\pi_{j_1}(n), \dots, \pi_{j_m}(n)\} \cap \{\pi_i(1), \dots, \pi_i(r)\} = \emptyset$$

$$\Rightarrow \quad \{j_1, \dots, j_m\} \cap \{\pi_i(1), \dots, \pi_i(r)\} = \emptyset$$

where the implication uses Claim 2. Since j_1, \ldots, j_m are all distinct by construction, $\pi_i(1), \ldots, \pi_i(r)$ are all distinct as π_i is a bijection, and all these indexes lies in the range $\{1, \ldots, n\}$ we conclude

$$|\{j_1,\ldots,j_m\}| + |\{\pi_i(1),\ldots,\pi_i(r)\}| \le n \quad \Rightarrow \quad m+r \le n \quad \Rightarrow \quad m \le n-r.$$

The sub-tree of $v_{i,r}$ is therefore tall, concluding the claim's proof.

Proof of Claim 4. We recall $V_t = \{v_{i,r} : i \in \{1, \ldots, t\}, r \in \{0, \ldots, n\}\}$ and further define $V_t^* = V_t \setminus \{v_{1,0}\}$, i.e. the set of vertices in V_t without the root. To prove the claim we proceed by induction on t showing the existence of a function $f: V_t^* \to \mathbb{N}^3$ satisfying the required properties.

If t = 1, we set $f(v_{1,r}) = (r, \pi_1(r), j)$ to be the query in the refined chain, see Definition 3, minimal w.r.t. \rightarrow^+ among queries performed at round r. Then f is injective over V_1^* and satisfies the first condition by construction. Regarding the third property if $D_{f(v_{1,r})} = D_{f(v_{1,r'})}$ with r < r' then we could create a shorter chain skipping round r, thus violating Claim 1. Hence r = r', and in particular $v_{1,r}, v_{1,r'}$ are the same, meaning that $v \mapsto D_{f(v)}$ is injective over V_1^* and thus satisfies the third property.

Assuming the statement to be true for t-1, i.e. that we have $f: V_{t-1}^* \to \mathbb{N}^3$ satisfying the three properties, we show f can be extended to V_t^* . To so, let j_r be such that $(r, \pi_t(r), j_r)$ is the minimal query at round r of the chain for $0 \to^+ \text{out}_t$. We then define for all $v_{i,r} \in V_t^* \setminus V_{t-1}^*$ the function to be $f(v_{i,r}) = (r, \pi_t(r), j_r)$. The first property is thus satisfied by construction.

Next we show f is injective. If $f(v_{i,r}) = f(v_{i',r'})$ then r = r' from the first property¹². By inductive hypothesis f is injective over V_{t-1}^* . Without loss of

¹² r and r' are the first component of respectively the LHS and the RHS.

generality we can then assume $v_{i',r'} \in V_t^* \setminus V_{t-1}^*$, in which case t = i'. In order to prove $v_{i,r} = v_{t,r}$ we proceed by contradiction assuming the two nodes to be different.

This implies that $v_{i,r} \in V_{t-1}^*$. With loss of generality we can further assume because of the first property that $f(v_{i,r}) = (r, \pi_i(r), j^*)$ is a minimal query at round r for $0 \rightarrow^+ \text{out}_i$. This implies that there exist refined chains for the relations

$$0 \twoheadrightarrow^+ (r, \pi_t(r), j_r) \twoheadrightarrow^+ \mathsf{out}_t \qquad 0 \twoheadrightarrow^+ (r, \pi_i(r), j^*) \twoheadrightarrow^+ \mathsf{out}_i.$$

Since $(r, \pi_t(r), j_r) = (r, \pi_i(r), j^*)$ we can combine the first half of the second chain with the second half of the first one to obtain a new refined chain for $0 \rightarrow^+ \text{out}_t$. Let $\hat{\pi}_t$ be the associated permutations. By construction $\hat{\pi}_t(x) = \pi_i(x)$ for all $x \in \{1, \ldots, t\}$, and in particular $[\hat{\pi}_t]_r = [\pi_i]_r$. Hence, calling $\hat{T} = (\hat{V}, \hat{E})$ the prefix tree for $\pi_1, \ldots, \hat{\pi}_t, \ldots, \pi_n$, we will show it violates our minimality condition on T. Indeed

- $|V_{t-1}^*| = |\widehat{V}_{t-1}^*|,$ because the first t-1 permutations used to build T,\widehat{T} are the same.
- $|V_t^*| \ge |V_{t-1}^*| + (n-r+1)$, because $v_{t,r} \in V_{t-1}^* \setminus V_t^*$ and in particular, $v_{t,r'} \notin V_{t-1}^*$ for r' > r, or else $v_{t,r'} = v_{i,r'}$ for some i < t which implies $v_{t,r} = v_{i,r} \in V_{t-1}^*$. Hence

$$\{v_{i,r},\ldots,v_{i,n}\} \subseteq V_t^* \setminus V_{t-1}^* \quad \Rightarrow \quad |V_t^* \setminus V_{t-1}^*| \ge (n-r+1)$$

 $|\widehat{V}_t^*| \leq |\widehat{V}_{t-1}^*| + (n-r)$, because $[\widehat{\pi}_t]_r = [\pi_i]_r$ implies $[\widehat{\pi}_t]_{r'} = [\pi_i]_{r'}$ for all $r' \leq r$. Hence, again for all $r' \leq r$, $\widehat{v}_{t,r'} = \widehat{v}_{i,r'} \in \widehat{V}_{t-1}^*$ and in particular

$$\widehat{V}_t^* \setminus \widehat{V}_{t-1}^* \subseteq \{\widehat{v}_{t,r+1}, \dots, \widehat{v}_{t,n}\} \quad \Rightarrow \quad |\widehat{V}_t^* \setminus \widehat{V}_{t-1}^*| \le n - t.$$

Combining the three relations we conclude that our minimality assumption on T is violated because

$$|\widehat{V}_t^*| \le |\widehat{V}_{t-1}^*| + (n-r) < |V_{t-1}^*| + (n-r+1) \le |V_t^*|.$$

This means that the assumption $v_{i,r} \neq v_{t,r}$ leads to a contradiction, implying that $v_{i,r} = v_{t,r}$ and that f is injective.

To conclude we need to prove the third property holds for f. Let $D_{f(v_{i,r})} = D_{f(v_{i',r'})}$. We study two cases:

1. $r \neq r'$. Then without loss of generality r < r' and by the first property $v_{i,r}$, $v_{i',r'}$ are such that $f(v_{i,r}) = (r, \pi_i(r), j)$ and $f(v_{i',r'}) = (r', \pi_{i'}(r'), j')$ are minimal queries in their respective rounds with respect to \rightarrow ⁺. Among the queries appearing in the chain for $0 \rightarrow$ ⁺ out_i let (r'', i'', j'') be the predecessor of $(r, \pi_i(r), j)$, i.e. such that

$$0 \to^+ (r'', i'', j'') \to (r, \pi_i(r), j) \to^+ \mathsf{out}_i.$$

Note that since $(r, \pi_i(r), j)$ is minimal among the queries at round r, we must have r'' < r. Then if we call $E_{r'',i'',j''}$ the output of $\mathcal{O}_{\mathsf{act}}$ for query (r'',i'',j''), by the definition of \rightarrow we have $E_{r'',i'',j''} = D_{(r,\pi_r(i),j)}$. Therefore, again by the definition of \rightarrow^+

$$\begin{split} E_{r^{\prime\prime},i^{\prime\prime},j^{\prime\prime}} &= D_{(r^{\prime},\pi_{r^{\prime}}(i^{\prime}),j^{\prime})}, \ \wedge r^{\prime\prime} < r^{\prime} \quad \Rightarrow \\ &\Rightarrow \quad 0 \twoheadrightarrow^{+}(r^{\prime\prime},i^{\prime\prime},j^{\prime\prime}) \twoheadrightarrow (r^{\prime},i^{\prime},j^{\prime}) \twoheadrightarrow^{+} \operatorname{out}_{i^{\prime}}. \end{split}$$

Since r'' < r < r' the resulting chain would not include any query from round r, contradicting Claim 1. Therefore $r \neq r'$ is impossible.

2. r = r'. By the inductive hypothesis if both vertices lie in V_{t-1}^* the property holds, so without loss of generality assume $v_{i',r} \in V_t^* \setminus V_{t-1}^*$ and $f(v_{i,r}) = (r, \pi_i(r), j_r)$, i.e. that the image of $v_{i,r}$ is a query on the chain for $0 \rightarrow^+ \text{out}_i$. We will denote p_i the predecessor of $f(v_{i,r})$ on the refined chains for $0 \rightarrow^+ \text{out}_i$. This means we have

$$0 \rightarrow p_i \to f(v_{i,r}) \to otic_i$$

Then by how \rightarrow was defined $E_{p_i} = D_{f(v_{i,r})} = D_{f(v_{t,r})}$ and by minimality of $f(v_{i,r})$ among the queries occurring at round r, p_i occurs at a round strictly smaller than r. Thus $p_i \rightarrow f(v_{t,r})$ and in particular we can find a chain for

$$0 \rightarrow p_i \to f(v_{t,r}) \to ot out_i$$

that is equal to the chain for $0 \to^+ \text{out}_i$ until query p_i . Calling $\hat{\pi}_t$ the associated permutation we would then have $[\hat{\pi}_t]_{r-1} = [\pi_i]_{r-1}$ since the chains are equal until round r-1 (we use Claim 1 to observe p_i occurs at round r-1). Finally assume that for the current chain chosen for $0 \to^+ \text{out}_t$ the nodes $v_{i,r}$ and $v_{t,r}$ are no siblings (otherwise the claim is proven), i.e. $v_{i,r-1} \neq v_{t,r-1}$, we distinguish two cases:

- $-v_{t,r-1} \notin V_{t-1}^*$. Then as done previously we can use $\hat{\pi}_t$ to build a prefix tree $\hat{T} = (\hat{V}, \hat{E})$ with $|\hat{V}_t| < |V_t|$, which contradicts our minimality assumption.
- $\ v_{t,r-1} \in V^*_{t-1}.$ Again using the new path we can build a prefix tree such that for all t' < t

$$|V_{t'}^*| = |\widehat{V}_{t'}^*| \qquad |V_t^*| = |V_{t-1}^*| + (n-r+1) \qquad |\widehat{V}_t^*| \le |\widehat{V}_{t-1}^*| + (n-r+1).$$

with the first equality holding as we are only replacing π_t with $\hat{\pi}_t$ and using the same π_1, \ldots, π_{t-1} in both prefix trees, the second one because $v_{t,r-1} \in V_t^* \setminus V_{t-1}^*$ while $v_{t,r-1} \in V_{t-1}^*$, and the third because $\hat{v}_{t,r-1} = \hat{v}_{i,r-1} \in \hat{V}_{t-1}^*$. We thus conclude $|\hat{V}_t| \leq |V_t|$ while preserving the size of smaller sub-trees.

Replacing the chain for $0 \to^+ \text{out}_t$, we finally have that for the new tree $\hat{v}_{i,r-1} = \hat{v}_{i,r-1}$. Notice that this change occurs only once since there can only be one node $v_{i,r} \in V_t \setminus V_{t-1}$ with $v_{i,r-1} \in V_{t-1}$. Furthermore after the change we still have $\hat{v}_{i,r} \in \hat{V}_t \setminus \hat{V}_{t-1}$ with $\hat{v}_{i,r-1} \in \hat{V}_{t-1}$ because

$$\widehat{v}_{t,r} \in \widehat{V}_{t-1}^* \quad \Rightarrow \quad |\widehat{V}_t^*| \le |\widehat{V}_{t-1}^*| + (n-r) < |V_{t-1}^*| + (n-r+1) = |V_t^*| \\ \Rightarrow \quad |\widehat{V}_t| < |V_t|$$

contradicting our minimality assumption. Therefore, all remaining nodes in $\hat{V}_t \setminus \hat{V}_{t-1}$ falls into the previous case.

This concludes the proof of the Claim.

Proof of Claim 5. Calling $V_{\geq 2}$ as in Section 5.3 the set of nodes of distance at least 2 from the root we observe that

$$V_{\geq 2} = \bigcup_{r=2}^{n} \bigcup_{i=1}^{n} U_{r,i}$$

Indeed given $v \in V_{\geq 2} \subseteq V^*$, by Claim 4 there exists r, i, j such that $v = v_{i,r}$ and f(v) = (r, i, j), implying $r \geq 2$ and $v \in U_{r,i}$. Next we show $|U_{r,i}| = |\Delta_{r,i}|$. To do so it suffices to show that the map $v \mapsto D_{f(v)}$ is injective over $U_{r,i}$. Let $u, v \in U_{r,i}$ such that $D_{f(u)} = D_{f(v)}$. By Claim 4 they must have the same parent. In particular if $u = ([\pi_{\ell}]_r, r)$ and $v = ([\pi_{\ell'}]_r, r)$ then having the same parent implies

$$[\pi_{\ell}]_{r-1} = [\pi_{\ell'}]_{r-1}, \quad \pi_{\ell}(r) = i = \pi_{\ell'}(r) \quad \Rightarrow \quad [\pi_{\ell}]_r = [\pi_{\ell'}]_r \quad \Rightarrow \quad u = v$$

where the second and third equality follows since $u, v \in U_{r,i}$. Finally, using Proposition 2 and Claim 3 stating that T is a TS tree we conclude

$$\sum_{r=1}^{n-1} \sum_{i=1}^{n} |\Delta_{r+1,i}| = \sum_{r=2}^{n} \sum_{i=1}^{n} |\Delta_{r,i}| \ge |V_{\ge 2}| \ge (n-2) + n \log_2 n.$$

Proof of Claim 6. In the following we denote input = $(s_1, \ldots, s_n, \rho_1, \ldots, \rho_n, E_0)$ where s_i, ρ_i are the private input and random coins of P_i . Furthermore, we will denote

$$\Gamma_{r+1,i,j} = \{ E_{r+1,i,j'} : j' < j \} \cup \{ D_{r+1,i,j'} : j' < j \}$$

We furthermore index $\Delta_{r+1,i} = \{D_{r+1,i,j_1}, \ldots, D_{r+1,i,j_m}\}$. Then

$$\begin{split} \operatorname{H}\left(M_{r}^{(i)} \left| \operatorname{trs}_{r-1}\right) &\geq \operatorname{H}\left(M_{r}^{(i)} \left| \operatorname{trs}_{r-1}, \operatorname{input}\right) \right. \\ &\geq \sum_{\alpha=1}^{m} \operatorname{I}\left(M_{r}^{(i)}; D_{r+1,i,j_{\alpha}} \left| \operatorname{trs}_{r-1}, \operatorname{input}, \left\{D_{r+1,i,j_{\beta}}\right\}_{\beta=1}^{\alpha-1}\right) \\ &\geq \sum_{\alpha=1}^{m} \operatorname{I}\left(M_{r}^{(i)}; D_{r+1,i,j_{\alpha}} \left| \operatorname{trs}_{r-1}, \operatorname{input}, \Gamma_{r+1,i,j_{\alpha}}\right). \end{split}$$

We will then lower bound each of these terms. The key observation is that, given $M_j^{(i)}$, trs_{r-1} , input and $\Gamma_{r+1,i,j}$, the execution of P_i becomes deterministic until the next query to $\mathcal{O}_{\operatorname{act}}$ is performed, meaning that $D_{r+1,i,j}$ is univocally determined. Therefore, if $\Delta_{r,i} = \{D_{r+1,i,j_1}, \ldots, D_{r+1,i,j_m}\}$, we have that

$$\mathrm{H}\left(D_{r+1,i,j_{\alpha}} \middle| M_{r}^{(i)}, \operatorname{trs}_{r-1}, \operatorname{input}, \Gamma_{r+1,i,j_{\alpha}}\right) = 0.$$

Conversely we observe that before round r, $D_{r+1,i,j_{\alpha}}$ was not returned as an output by $\mathcal{O}_{\mathsf{act}}$, or else we could build a chain for $D_{r+1,i,j_{\alpha}}$ skipping round r, which

violates Claim 1. Moreover by the minimality of $D_{r+1,i,j_{\alpha}}$ (see Claim 4), this set elements was not computed previously on the same round by P_i , meaning that is does not belong in $\Gamma_{r+1,i,j_{\alpha}}$. Moreover $D_{r+1,i,j_{\alpha}}$ is independent from the random coins and inputs of parties (which are sampled before any query is ever made to \mathcal{O}_{act}). Hence we conclude that $D_{r+1,i,j_{\alpha}}$ conditioned to trs_{r-1} , input, $\Gamma_{r+1,i,j_{\alpha}}$ is uniform in the set of not-yet queried labels, which has size $2^{\mu} - 2q$. Thus

$$\operatorname{H}\left(D_{r+1,i,j_{\alpha}} | \operatorname{trs}_{r-1}, \operatorname{input}, \Gamma_{r+1,i,j_{\alpha}}\right) \geq \log_{2}(2^{\mu} - 2q)$$

$$\geq \mu - \frac{2q}{2^{\mu} - 2q} = \mu - \frac{q}{2^{\mu-1} - q}.$$

Where second inequality follows since $\log(x)$ is concave and for all x > y > 0

$$\frac{1}{x} \leq \frac{\log(x) - \log(y)}{x - y} \leq \frac{1}{y}$$

replacing $x = 2^{\mu}$ and $y = 2^{\mu} - 2q$. As the mutual information is the difference of the above quantities, we have that $\mu - \frac{q}{2^{\mu-1}-q}$ lower bounds each term in the summation above. We can therefore conclude

$$\mathrm{H}\left(M_{r}^{(i)} \,\Big|\, \mathsf{trs}_{r-1}\right) \,\geq\, m \cdot \frac{q}{2^{\mu-1}-q} \,\geq\, |\varDelta_{r+1,i}| \cdot \frac{q}{2^{\mu-1}-q}.$$

6 Conclusions

In conclusion we proved two lower bounds for multi-party computation in the GAM, i.e. through black box usage of a group action. First, protocols that are at least passively secure cannot perform better than Round-Robin in terms of round complexity. Second, *fair* computation, in the sense of Section 5.1, requires $\Omega(n \log_2 n)$ computation and communication complexity. Remarkably, both results still hold under *any* computational assumptions, and even if parties make explicit use of the bit representation of elements in \mathcal{E} .

In the context of threshold protocols, including those for digital signatures and public-key encryptions schemes, these bounds hinder the scalability to large sets of parties. In these cases, if lower round complexity or communication is required, our result could be bypassed only through explicit usage of a circuit implementing the group action. General purpose MPC techniques for instance would apply in such case. Therefore future research in this direction should either focus on reducing the complexity for computing group actions (e.g. lowering the multiplicative depth), or on designing specialized *non-black box* protocols.

Finally, we leave two open questions related to fair protocols: First, as mentioned in Section 5.1, our lower communication and computation lower bounds only affect round-optimal constructions. We thus ask whether increasing the round complexity allows violating our bounds. Secondly, our result is tight only for n users with n being a power of 2. More generally we conjecture optimal solutions to require

 $n(1 + \log_2 n) \le q \le n(1 + \log_2 n) + c \cdot n$

many queries, for a tight constant c. We numerically estimate $c \approx 0.087$ for $n \leq 2^{24}$ and leave the question of whether such condition holds in general to future studies.

Acknowledgments

This work received funding from projects from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under project PICOCRYPT (grant agreement No. 101001283), from SE-CURING Project (ref. PID2019-110873RJ-I00), from the Spanish Government under projects PRODIGY (TED2021-132464B-I00) and ESPADA (PID2022-142290OB-I00). The last two projects are co-funded by European Union EIE, and NextGenerationEU/PRTR fund.

References

- ABCP23a. Shahla Atapoor, Karim Baghery, Daniele Cozzo, and Robi Pedersen. Csishark: Csi-fish with sharing-friendly keys. In Leonie Simpson and Mir Ali Rezazadeh Baee, editors, Information Security and Privacy - 28th Australasian Conference, ACISP 2023, Brisbane, QLD, Australia, July 5-7, 2023, Proceedings, volume 13915 of Lecture Notes in Computer Science, pages 471–502. Springer, 2023.
- ABCP23b. Shahla Atapoor, Karim Baghery, Daniele Cozzo, and Robi Pedersen. Practical robust dkg protocols for csidh. In Mehdi Tibouchi and XiaoFeng Wang, editors, Applied Cryptography and Network Security, pages 219–247, Cham, 2023. Springer Nature Switzerland.
- ABCP23c. Shahla Atapoor, Karim Baghery, Daniele Cozzo, and Robi Pedersen. VSS from distributed ZK proofs and applications. *IACR Cryptol. ePrint Arch.*, page 992, 2023.
- BCP21. Karim Baghery, Daniele Cozzo, and Robi Pedersen. An isogeny-based ID protocol using structured public keys. In Maura B. Paterson, editor, Cryptography and Coding - 18th IMA International Conference, IMACC 2021, Virtual Event, December 14-15, 2021, Proceedings, volume 13129 of Lecture Notes in Computer Science, pages 179–197. Springer, 2021.
- BDK⁺22. Ward Beullens, Samuel Dobson, Shuichi Katsumata, Yi-Fu Lai, and Federico Pintore. Group signatures and more from isogenies and lattices: Generic, simple, and efficient. In Orr Dunkelman and Stefan Dziembowski, editors, EUROCRYPT 2022, Part II, volume 13276 of LNCS, pages 95–126. Springer, Heidelberg, May / June 2022.
- BDPV21. Ward Beullens, Lucas Disson, Robi Pedersen, and Frederik Vercauteren. CSI-RAShi: Distributed key generation for CSIDH. In Jung Hee Cheon and Jean-Pierre Tillich, editors, Post-Quantum Cryptography - 12th International Workshop, PQCrypto 2021, pages 257–276. Springer, Heidelberg, 2021.
- BGZ23. Dan Boneh, Jiaxin Guan, and Mark Zhandry. A lower bound on the length of signatures based on group actions and generic isogenies. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 507–531. Springer, Heidelberg, April 2023.

- BKP20. Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In Shiho Moriai and Huaxiong Wang, editors, ASIACRYPT 2020, Part II, volume 12492 of LNCS, pages 464–492. Springer, Heidelberg, December 2020.
- BKV19. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, ASIACRYPT 2019, Part I, volume 11921 of LNCS, pages 227–247. Springer, Heidelberg, December 2019.
- CD23. Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V, volume 14008 of Lecture Notes in Computer Science, pages 423–447. Springer, 2023.
- CFGG22. Dario Catalano, Dario Fiore, Rosario Gennaro, and Emanuele Giunta. On the impossibility of algebraic vector commitments in pairing-free groups. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part II*, volume 13748 of *LNCS*, pages 274–299. Springer, Heidelberg, November 2022.
- CLM⁺18. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In Thomas Peyrin and Steven D. Galbraith, editors, Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III, volume 11274 of Lecture Notes in Computer Science, pages 395–427. Springer, 2018.
- CM22. Fabio Campos and Philipp Muth. On actively secure fine-grained access structures from isogeny assumptions. In Jung Hee Cheon and Thomas Johansson, editors, Post-Quantum Cryptography - 13th International Workshop, PQCrypto 2022, Virtual Event, September 28-30, 2022, Proceedings, volume 13512 of Lecture Notes in Computer Science, pages 375–398. Springer, 2022.
- CS20. Daniele Cozzo and Nigel P. Smart. Sashimi: Cutting up CSI-FiSh secret keys to produce an actively secure distributed signing protocol. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography* -11th International Conference, PQCrypto 2020, pages 169–186. Springer, Heidelberg, 2020.
- DG19. Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, EUROCRYPT 2019, Part III, volume 11478 of LNCS, pages 759–789. Springer, Heidelberg, May 2019.
- DHH⁺21. Nico Döttling, Dominik Hartmann, Dennis Hofheinz, Eike Kiltz, Sven Schäge, and Bogdan Ursu. On the impossibility of purely algebraic signatures. In Kobbi Nissim and Brent Waters, editors, TCC 2021, Part III, volume 13044 of LNCS, pages 317–349. Springer, Heidelberg, November 2021.
- DHK⁺23. Julien Duman, Dominik Hartmann, Eike Kiltz, Sabrina Kunzweiler, Jonas Lehmann, and Doreen Riepel. Generic models for group actions. In Alexan-

dra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 406–435. Springer, Heidelberg, May 2023.

- DM20. Luca De Feo and Michael Meyer. Threshold schemes from isogeny assumptions. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 187–212. Springer, Heidelberg, May 2020.
- EKP20. Ali El Kaafarani, Shuichi Katsumata, and Federico Pintore. Lossy CSI-FiSh: Efficient signature scheme with tight reduction to decisional CSIDH-512. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 157–186. Springer, Heidelberg, May 2020.
- FP21. Tako Boris Fouotsa and Christophe Petit. SHealS and HealS: Isogenybased PKEs from a key validation method for SIDH. In Mehdi Tibouchi and Huaxiong Wang, editors, ASIACRYPT 2021, Part IV, volume 13093 of LNCS, pages 279–307. Springer, Heidelberg, December 2021.
- Giu23. Emanuele Giunta. On the impossibility of algebraic nizk in pairing-free groups", booktitle="advances in cryptology crypto 2023. pages 702–730, Cham, 2023. Springer Nature Switzerland.
- LGD21. Yi-Fu Lai, Steven D. Galbraith, and Cyprien Delpech de Saint Guilhem. Compact, efficient and UC-secure isogeny-based oblivious transfer. In Anne Canteaut and François-Xavier Standaert, editors, EUROCRYPT 2021, Part I, volume 12696 of LNCS, pages 213–241. Springer, Heidelberg, October 2021.
- Mau05. Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, 10th IMA International Conference on Cryptography and Coding, volume 3796 of LNCS, pages 1–12. Springer, Heidelberg, December 2005.
- MMP⁺23. Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, Advances in Cryptology EUROCRYPT 2023 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V, volume 14008 of Lecture Notes in Computer Science, pages 448–471. Springer, 2023.
- MOT20. Tomoki Moriya, Hiroshi Onuki, and Tsuyoshi Takagi. SiGamal: A supersingular isogeny-based PKE and its application to a PRF. In Shiho Moriai and Huaxiong Wang, editors, ASIACRYPT 2020, Part II, volume 12492 of LNCS, pages 551–580. Springer, Heidelberg, December 2020.
- PRV12. Periklis A. Papakonstantinou, Charles Rackoff, and Yevgeniy Vahlis. How powerful are the DDH hard groups? *Electron. Colloquium Comput. Complex.*, page 167, 2012.
- Rob23. Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, Advances in Cryptology EUROCRYPT 2023 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V, volume 14008 of Lecture Notes in Computer Science, pages 472–503. Springer, 2023.
- RSS20. Lior Rotem, Gil Segev, and Ido Shahaf. Generic-group delay functions require hidden-order groups. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 155–180. Springer, Heidelberg, May 2020.

SD21.	Surbhi Shaw and Ratna Dutta. Identification scheme and forward-secure
	signature in identity-based setting from isogenies. In Qiong Huang and
	Yu Yu, editors, Provable and Practical Security - 15th International Con-
	ference, ProvSec 2021, Guangzhou, China, November 5-8, 2021, Proceed-
	ings, volume 13059 of Lecture Notes in Computer Science, pages 309–326.
	Springer, 2021.

- SGS20. Gili Schul-Ganz and Gil Segev. Accumulators in (and beyond) generic groups: Non-trivial batch verification requires interaction. In Rafael Pass and Krzysztof Pietrzak, editors, TCC 2020, Part II, volume 12551 of LNCS, pages 77–107. Springer, Heidelberg, November 2020.
- Sho94. Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, pages 124–134, 1994.
- Sho97. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997.
- Sto12. Anton Stolbunov. Cryptographic schemes based on isogenies. 2012.
- TMM21. Erkan Tairi, Pedro Moreno-Sanchez, and Matteo Maffei. Post-quantum adaptor signature for privacy-preserving off-chain payments. In Nikita Borisov and Claudia Díaz, editors, FC 2021, Part II, volume 12675 of LNCS, pages 131–150. Springer, Heidelberg, March 2021.
- Zha22. Mark Zhandry. To label, or not to label (in generic groups). In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 66–96. Springer, Heidelberg, August 2022.

A CSIDH Group Action

Here we recall the (only known) efficient instantiation of a HHS, namely CSIDH.

Let p be the prime $p = 4 \cdot f \prod_{i=1}^{r} \ell_i - 1$, where ℓ_1, \ldots, ℓ_r are the first r odd prime numbers and f is a cofactor chosen so that p is a prime. Let E_0 be the supersingular elliptic curve $E_0/\mathbb{F}_p: y^2 = x^3 + x$. Let $\operatorname{End}_{\mathbb{F}_p}(E_0)$ be the ring of endomorphisms of E_0 that are defined over \mathbb{F}_p . Since $p \equiv 3 \mod 4$, this ring is isomorphic to the order $\mathcal{O} = \mathbb{Z}[\sqrt{-p}]$ via the Frobenius element π . By abuse of notation, we will identify $\operatorname{End}_{\mathbb{F}_p}$ with \mathcal{O} .

From \mathcal{O} we can construct the ideal class group $\mathsf{cl}(\mathcal{O})$, which consists of the invertible ideals on \mathcal{O} modulo the non-zero principal ideals. Given an element $\mathfrak{a} \in \mathsf{cl}(\mathcal{O})$ one can construct the subgroup

$$E[\mathfrak{a}] := \bigcap_{\varphi \in \mathfrak{a}} \operatorname{Ker}(\varphi).$$

This gives an action of $cl(\mathcal{O})$ on the class of supersingular elliptic curves \mathcal{E} in the isogeny class of E_0 :

$$\star: \mathsf{cl}(\mathcal{O}) \times \mathcal{E} \longrightarrow \mathcal{E}$$
$$(\mathfrak{a}, E) \longmapsto E_0/E_0[\mathfrak{a}]$$

which is free and transitive $[CLM^+18]$.

The following problems are conjecturally hard to solve, even for a quantum computer: Group action inverse problem (GAIP): Given $E, E' \in \mathcal{E}$, find the ideal class $\mathfrak{a} \in \mathsf{cl}(\mathcal{O})$ such that $E' = \mathfrak{a} \star E$.

Group action computational Diffie-Hellman problem: Given $E, E', E'' \in \mathcal{E}$, find the elliptic curve $\mathfrak{a} \star \mathfrak{b} \star E$, where $\mathfrak{a}, \mathfrak{b} \in \mathsf{cl}(\mathcal{O})$ are such that $E' = \mathfrak{a} \star E, E'' = \mathfrak{b} \star E$.

The structure of the class group for the 511-bit prime $p = 4 \cdot 587 \cdot \prod_{i=1}^{73} \ell_i - 1$, was explicitly determined by [BKV19] to be the cyclic group $(\mathbb{Z}, +)$, with $N \approx \sqrt{p} \approx 2^{257}$, a generator being $\mathfrak{g} = (3, \pi - 1)$. This way $\mathsf{cl}(\mathcal{O})$ can be identified with the integers modulo N and the identification $\mathfrak{a} \mapsto \mathfrak{g}^a$ is efficiently computable. Thus we can write the action of an ideal $\mathfrak{a} \star E$ as $a \star E$, where $\mathfrak{a} = \mathfrak{g}^a$. This makes CSIDH-512 an HHS, according to definition in Section 2.2.

The choice of the curve E_0 , means that all the curves in the isogeny class of E_0 have the Montgomery form $y^2 = x^3 + Ax^2 + X$ for a $A \in \mathbb{F}_p$. This is very convenient in parctice, as it means that any elliptic curve can be represented with a single field element. Another important implication is that, in this family, twists are also efficiently computable. The quadratic twist of $E_A : y^2 = x^3 + Ax^2 + x$ being isomorphic to $E_{-A} : y^2 = x^3 - Ax^2 + x$. From a group action perspective, this means that computing $-a \star E$ from $a \star E$ can be done efficiently.

Modelling the twists From now on we will generally refer to quadratic twists as twists. Note that twists are not captured by the above group action. To do so, we take the semidirect product of \mathbb{Z}_N with \mathbb{Z}_2 :

$$\mathbb{G}' = \mathbb{Z}_2 \ltimes \mathbb{Z}_N.$$

Where one defines the group operation \circ via

$$(s_1, g_1) \circ (s_2, g_2) = (s_1 s_2, g_1 + (-1)^{s_2} g_2).$$

Note that (\mathbb{G}', \circ) is not commutative. The action of \mathbb{G}', \circ on the set \mathcal{E} is defined as

$$(s,g) \star (h \star E_0) := (g + (-1)^s h) \star E_0.$$

The action of (1,0) on E corresponds to the application of a twist on E. Note that this action is not free, as $(0,0) \star E_0 = (1,0) \star E_0 = E_0$.

B Optimal Fair Protocols

Our lower bound for fair computation "in the exponent" proved in Section 5.4 states that any such protocol has to perform at least $n(1 + \log_2 n)$ action evaluations and communicate roughly $n \log_2 n$ set elements. This bound is met by the binary splitting strategy in [DM20] for group of users that are a power of 2. When dealing with a number of users that is not a power of two, this strategy could be adapted. However this requires subgroups of different size, which will

not conclude their computation in the same number of rounds. So, to preserve round optimality, one must carefully interleave the computation done by the two groups.

In this section we explain how the proof of Theorem 1 could be used to provide a constructive and different way to find fair protocols with optimal communication and computation for any set of users. First, recall that the proof initially associate to each user's output a chain, which induce a permutation among users, describing in which order users acts on the chain. Next, given this permutations, we prove that their prefix tree is a TS tree, see Section 5.3, and use Proposition 1 to get computation and communication bounds.

Here we show that the process could be reversed when parties wish to compute $(s_1 + \ldots + s_n) \star E_0$ given $s_1, \ldots, s_n \in \mathbb{G}$. The idea is starting with an optimal TS tree with the lowest number of nodes possible, find a labeling of the tree with indexes in $\{1, \ldots, n\}$ (root excluded) so that each path is a permutation with leaves assuming different values, and use this to build our protocol. This guarantees the resulting procedure to be optimal since its associated prefix tree is the TS tree we started with, which has already the minimum number of nodes.

3 Parties Protocol. Instead of formalizing this, we provide an example for 3 parties. All TS trees with 3 leaves have at least $3 \cdot (1 + \log_2 3) + 1 \approx 8.8$ nodes. In Figure 6 (left graph) we present an example with 9 nodes, which is therefore optimal.

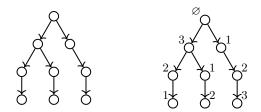


Fig. 6. Optimal TS-tree with 3 leaves (left), and a labeling (right).

Next we find for this tree a labeling with indexes in $\{1, \ldots, n\}$ such that each label is contained in each path exactly once, meaning in particular that each path is associated to a permutation. Finally we define a protocol assigning to each node a set element. The root is assigned to E_0 , and if a node has label *i* we assign to it $s_i \star (\cdot)$ applied on the set element associated to its parent. So in this example, the right and left children of the root will be assigned to $s_1 \star E_0$ and $s_3 \star E_0$.

Finally, this defines a protocol in which all the set elements associated to nodes at distance r from the root are computed at round r. We described the protocol in Figure 7.

$P_1(s_1)$	$P_2(s_2)$	$P_3(s_3)$
1: $E_1 \leftarrow s_1 \star E_0$	1:	$1: E_3 \leftarrow s_3 \star E_0$
2: Send E_1 to P_2	2:	2: Send E_3 to P_1 , P_2
$3: E_{1,3} \leftarrow s_1 \star E_3$	$3: E_{1,2} \leftarrow s_2 \star E_1$	3:
4: Send $E_{1,3}$ to P	$P_2 \qquad 4: E_{2,3} \leftarrow s_2 \star E_3$	4:
5:	5: Send $E_{1,2}$ to I	$D_3 5:$
6:	6: Send $E_{2,3}$ to I	$P_1 6:$
$7: E_{1,2,3} \leftarrow s_1 \star E_2$	$e_{,3} \qquad 7: E_{1,2,3} \leftarrow s_2 \star E$	$1_{1,3}$ 7: $E_{1,2,3} \leftarrow s_3 \star E_{1,2}$

Fig. 7. Parties P_1 , P_2 , P_3 computing $(s_1 + s_2 + s_3) \star E_0$.

We observe that for 3 players our bound requires that

$$q \geq 3(1 + \log_2 3) \approx 7.8$$
 $\ell_{\text{tot}} \gtrsim (1 + 3\log_2 3)\mu \approx 5.8\mu$

hence our protocol achieving q = 8 and $\ell_{tot} = 6\mu$ is essentially optimal.

5 Parties Protocol. For 5 parties we can repeat all steps performed in the previous case. Initially we find an optimal TS tree with 5 leaves, which should have at least $5 \cdot (1 + \log_2 5) + 1 \approx 17.6$ nodes. We describe a tree with 18 nodes in Figure 8 (left).

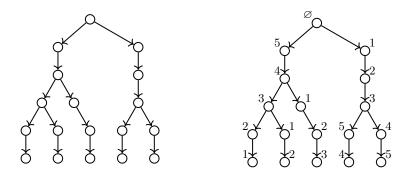


Fig. 8. Optimal TS-tree with 5 leaves (left), and its labeling (right).

As done before we can then label each node in such a way that each path contains all indices from 1 to 5, shown in Figure 8 (right). Note that it can be proven that such labeling exists for any TS tree with height(T) = leaves(T). Finally, we obtain a fair protocol among 5 users by assigning E_0 to the root, and to each node with label *i* the element $s_i \star (\cdot)$ applied to the set element

associated to its parent. The protocol is then defined by letting parties compute at round r the element associated to nodes with distance r from root.

Although we do not explicitly write the protocol this time, we observe it would require q = 17 group action evaluations and $\ell_{tot} = 15\mu$, which is consistent with our lower bounds:

$$q \ge 5 \cdot (1 + \log_2 5) \approx 16.6$$
 $\ell_{\text{tot}} \gtrsim (3 + 5 \log_2 5)\mu \approx 14.6\mu.$

C Postponed Proofs

C.1 Sequentiality Lemma

Proof of Lemma 1. In the following we will use $\mathcal{E} \subseteq \{0,1\}^{\mu}$ to denote the set of labels and \mathcal{E}' the set in our group action, so that $\star : \mathbb{G} \times \mathcal{E}' \to \mathcal{E}'$ and $\sigma : \mathcal{E}' \to \mathcal{E}$ is the labeling function. We further denote with abuse of notation $\star : \mathbb{G} \times \mathcal{E} \to \mathcal{E}$ the action defined by \mathcal{O}_{act} .

We begin proving that if E_{out} is not a label returned by \mathcal{O}_{act} , then

$$\Pr\left[E_{\mathsf{out}} = s \star E_0 \,|\, E_{\mathsf{out}} \notin \{E_0, \dots, E_q\}\right] \leq \frac{1}{|\mathcal{E}| - (q+1)}.$$

This holds because if $s \star E_0 \in \{E_0, \ldots, E_q\}$ or $E_{out} \notin \mathcal{E}$ then the probability of $E_{out} = s \star E_0$ is zero. Conversely, when these two events do not occur, \mathcal{A} correctly guessed the label $s \star E_0$ given only the image of σ (the labeling function) on q+1 different points. Since $\sigma : \mathcal{E}' \to \mathcal{E}$ is uniformly sampled

$$\Pr\left[E_{\mathsf{out}} = s \star E_0 \,|\, E_{\mathsf{out}}, s \star E_0 \in \mathcal{E} \setminus \{E_i\}_{i=0}^q\right] \leq \frac{1}{|\mathcal{E}| - (q+1)}$$

Next, we assume $E_{out} = E_k$ for some $k \in \{0, \ldots, q\}$. To conclude we need to prove that the probability of $E_{out} = k$ but not $0 \to^+ k$ is negligible. We do so by reducing to the discrete logarithm problem. The idea is that since \to^+ is a partial order on a finite set, there exists a minimal element $h \to^+ k$ that is non-zero. Since E_h was not obtained through the action oracle, the adversary must have randomly sampled its representation and in particular it has (almost) no information on its discrete logarithm in base E_0 . However because $h \to^+ k$ the adversary has to know $a \in \mathbb{G}$ for which $a \star E_h = E_k = s \star E_0$, implying that $E_h = (-a + s) \star E_0$, i.e. that it can find the discrete logarithm of E_0 .

More formally we provide a reduction \mathcal{B} to the vectorization problem in the GAM described in Figure 9. At a high level $\mathcal{B}(E_0, H)$ lazily simulate a GAM oracle through a labeling function $\tilde{\sigma}$.

Every time \mathcal{A} makes a query to the GAM on a previously seen set element the query is stored in \mathcal{Q} . Conversely, if a query is done on the representation of a point \widetilde{D} not yet obtained through other queries, \mathcal{B} either rejects with the same probability $\mathcal{O}_{\mathsf{act}}$ would, or internally maps \widetilde{D} to the set element $r \star H$ with r uniformly sampled in \mathbb{G} , storing the tuple (r, D) in a set \mathcal{C} , and then performs the requested query invoking $\mathcal{O}_{\mathsf{act}}$. Finally, if \mathcal{A} returns the right output $E_{out} = E_k$ but $0 \not\rightarrow^+ k$, then \mathcal{B} uses the information in \mathcal{Q} and \mathcal{C} to break the vectorization problem. To sum up the set \mathcal{Q} and \mathcal{C} respectively contain:

- \mathcal{Q} : GAM queries of the form (a, D, E), meaning that $E = a \star D$, as well as trivial relations of the form (0, E, E).
- C: Challenges (r, D) with $D = r \star H$, created when \mathcal{A} query a point not previously seen.

 $\mathcal{B}^{\mathcal{O}_{\mathsf{act}}}(E_0,H)$

Setup a partial function $\widetilde{\sigma}: \{0,1\}^{\mu} \to \{0,1\}^{\mu} \cup \{\bot\}$, initially $\widetilde{\sigma} = \emptyset$ 1:Sample $\widetilde{E}_0 \leftarrow^{\$} \{0,1\}^{\mu}$ and set $\widetilde{\sigma}(\widetilde{E}_0) = E_0$ 2:Initialize two sets $\mathcal{Q} = \{(0, E_0, E_0)\}, \mathcal{C} = \emptyset$ 3: Run $\mathcal{A}(\widetilde{E}_0)$ 4:When \mathcal{A} queries $\mathcal{O}_{\mathsf{act}}(a_i, \widetilde{D}_i)$: 5:If $\widetilde{D}_i \notin \mathsf{Dom}(\widetilde{\sigma})$: 6: Let p_i be the probability that \widetilde{D}_i is a valid set element 7:With probability $1 - p_i$: Set $\tilde{\sigma}(\tilde{D}_i) = \perp$ 8: 9: Else: Sample $r_i \leftarrow {}^{\$} \mathbb{G}$ such that $r_i \star H \notin \{E_0, \ldots, E_{i-1}\}$ 10:Set $D_i \leftarrow r_i \star H$ and $\tilde{\sigma}(\tilde{D}_i) = D_i$ 11: Store $\mathcal{C} \leftarrow \mathcal{C} \cup \{(r_i, D_i)\}$ and $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(0, D_i, D_i)\}$ 12: Get $D_i \leftarrow \widetilde{\sigma}(\widetilde{D}_i)$ and $E_i \leftarrow \mathcal{O}_{\mathsf{act}}(a_i, D_i)$ 13: For all $(b, D_j, D_i) \in \mathcal{Q}$ store $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(a_i + b, D_j, E_i)\}$ 14: Sample $\widetilde{E}_i \leftarrow {}^{\$} \{0,1\}^{\mu} \setminus \mathsf{Dom}(\widetilde{\sigma}) \text{ and set } \widetilde{\sigma}(\widetilde{E}_i) = E_i$ 15:Answer the query with $\mathcal{A} \leftarrow \widetilde{E}_i$ 16: When $(s, \widetilde{E}_{out}) \leftarrow \mathcal{A}$: 17:If $\widetilde{E}_{out} \notin \{\widetilde{E}_0, \ldots, \widetilde{E}_q\}$: Return \perp . 18: Find k such that $\widetilde{E}_{out} = \widetilde{E}_k$ 19: If $0 \to^+ k$: Return \perp 20:Let $E_k \leftarrow \widetilde{\sigma}(\widetilde{E}_k)$ and find $(a, D_j, E_k) \in \mathcal{Q}$ and $(r, D_j) \in \mathcal{C}$ 21: 22:If $E_k \neq \mathcal{O}_{\mathsf{act}}(s, E_0)$: Return \perp Return the group element -(a+r)+s23:

Fig. 9. Reduction \mathcal{B} using \mathcal{A} to break the vectorization problem.

Initially observe that \mathcal{B} perfectly simulates the \mathcal{O}_{act} oracle since for previously queried elements, for which a representation $\tilde{\sigma}$ was chosen, the operation is consistent with \mathcal{O}_{act} , while queries on set element not previously obtained from

the oracle do not gives errors with probability

$$p_i = \frac{|\mathrm{Im}\,(\widetilde{\sigma} \setminus \{\bot\})|}{2^{\mu} - |\mathsf{Dom}(\widetilde{\sigma})|}.$$

and in this case the obtained element is different from previously queried ones.

To continue, we break down the proof in the following sequence of claims:

Claim 1 For all $(a, D_j, E_i) \in \mathcal{Q}$ then $a \star D_j = E_i$.

Claim 2 For all $(r, D_i) \in \mathcal{C}$ then $D_i = r \star H$.

Claim 3 Given $h \in \{1, ..., q\}$ that is minimal with respect to \rightarrow^+ then after the *h*-th query $(0, D_h, D_h) \in \mathcal{Q}$ and $(r, D_h) \in \mathcal{C}$.

Claim 4 Given $h \in \{1, \ldots, q\}$ that is minimal with respect to \rightarrow^+ , then $h \rightarrow^+ k$ implies that $(a, D_h, E_k) \in \mathcal{Q}$.

To see why these Claims implies the thesis, let us assume that the three condition on steps 18, 20 and 22 are false. Then $\tilde{E}_{out} = E_k$ and $0 \not\rightarrow^+ k$. Since \rightarrow^+ is a partial order on a finite set, there exists a minimal $h \rightarrow^+ k$ and, importantly, $h \neq 0$. From Claim 4 we have that $(a, D_h, E_k) \in \mathcal{Q}$, which by Claim 1 means $E_k = a \star D_h$. Similarly by Claim 3, $(r, D_h) \in \mathcal{C}$ which by Claim 2 means $D_h = r \star H$. Since we assumed all three final checks to fail, we further have that $E_k = s \star E_0$, meaning that

$$s \star E_0 = E_k = a \star D_h = (a+r) \star H \quad \Rightarrow \quad H = (-(a+r)+s) \star E_0.$$

That is, \mathcal{B} returns the correct value. Finally observe that the three final conditions are false if

$$E_{\text{out}} = s \star E_0 \land E_{\text{out}} = E_k \land 0 \not\to^+ k.$$

Hence, switching to the original notation

$$\Pr\left[E_{\mathsf{out}} = s \star E_0, \ E_{\mathsf{out}} = E_k, \ 0 \not\to^+ k\right] \leq \mathsf{Adv}(\mathcal{B}) \leq \varepsilon(2q)$$

which completes the proof.

Proof of Claim 1. By induction, initially $(0, E_0, E_0)$ satisfies the hypothesis. Furthermore the tuple $(0, D_i, D_i)$ added in step 12 also satisfies the hypothesis. Assume this holds for all elements in \mathcal{Q} when $(a_i + b, D_j, E_i)$ is added. Then $(b, D_j, D_i) \in \mathcal{Q}$ implies $D_i = b \star D_j$ so $E_i = a_i \star D_i = (a_i + b) \star D_j$.

Proof of Claim 2. Follows by construction.

Proof of Claim 3. Since h is minimal, the h-th query $\tilde{E}_h = a_h \star \tilde{D}_h$ is such that \tilde{D}_h was never returned by \mathcal{B} through the action queries, meaning that before the h-th query $\tilde{D}_h \notin \mathsf{Dom}(\tilde{\sigma})$. By construction then $(0, D_h, D_h)$ and (r_h, D_h) are added to \mathcal{Q}, \mathcal{C} respectively.

Proof of Claim 4. Since h is minimal, when D_h is queried, $(0, D_h, D_h) \in \mathcal{Q}$ and after \mathcal{B} computes the correct query, (a_h, D_h, E_h) is added in \mathcal{Q} . Next, by definition of \rightarrow^+ there exists indices i_1, \ldots, i_m with $i_1 = h$, $i_m = k$ such that $\widetilde{D}_{i_j} = \widetilde{E}_{i_j}$. By induction we will prove that for all j there exists some $a \in \mathbb{G}$ such that $(a, D_h, E_{i_j}) \in \mathcal{Q}$. Assuming this holds for j, when $(a_{i_{j+1}}, \widetilde{D}_{i_{j+1}})$ we have that $\widetilde{D}_{i_{j+1}} = \widetilde{E}_{i_j}$ so $D_{i_{j+1}} = \widetilde{\sigma}(\widetilde{D}_{i_{j+1}}) = E_{i_j}$. Thus by construction, in step 14 \mathcal{B} will add $(a_{i_{j+1}}, D_h, E_{i_{j+1}})$.

C.2 Properties

Proof of Lemma 2. By induction on t. If t = 1 we only have one query and $D_{r_1,i_1,j_1} = E_{r_1,i_1,j_1}$ by definition. Assuming the thesis for t-1 we have that

$$E_{r_t,i_t,j_t} = a_{r_t,i_t,j_t} \star D_{r_t,i_t,j_t}$$

= $a_{r_t,i_t,j_t} \star E_{r_{t-1},i_{t-1},j_{t-1}}$
= $a_{r_t,i_t,j_t} \star \left(\left(\sum_{\alpha=t-1}^{1} a_{r_\alpha,i_\alpha,j_\alpha} \right) \star D_{r_1,i_1,j_1} \right)$
= $\left(\sum_{\alpha=t}^{1} a_{r_\alpha,i_\alpha,j_\alpha} \right) \star D_{r_1,i_1,j_1}.$

Proof of Lemma 3. We prove the statement by induction on t. For t = 1 it is trivially true as $1 = |\{i_1\}| = |\{r_1\}|$. Assuming the statement is true for t, then for t + 1 we have

$$(r_t, i_t, j_t) \to (r_{t+1}, i_{t+1}, j_{t+1}).$$

We prove the inductive case studying two cases:

 $-r_t < r_{t+1}$. Then r_{t+1} is larger than all r_0, \ldots, r_t , therefore

$$|\{r_1, \dots, r_{t+1}\}| = 1 + |\{r_1, \dots, r_t\}| \ge 1 + |\{i_1, \dots, i_t\}| \ge |\{i_1, \dots, i_{t+1}\}|.$$

 $-r_t = r_{t+1}$. Then $i_t = i_{t+1}$ meaning that both sets do not increase in size, therefore

 $|\{r_1, \dots, r_{t+1}\}| = |\{r_1, \dots, r_t\}| \ge |\{i_1, \dots, i_t\}| \ge |\{i_1, \dots, i_{t+1}\}|.$

C.3 Interactive Sequentiality Lemma

Proof of Lemma 4. In order to bridge the Sequentiality Lemma in this context we wrap the execution of the distributed protocol within an environment Ω . This will take as input all s_i , execute parties and manage messages delivery. A full description of Ω appears in Figure 10

Because Ω forwards parties' queries to $\mathcal{O}_{\mathsf{act}}$ we can define a partial function $\xi : \mathbb{N}^3 \to \mathbb{N}$ so that the *j*-th query of P_i at round *r* corresponds to the $\xi(r, i, j)$ -th query of Ω . From the correctness of the distributed protocol we have that $E_{\mathsf{out}} = f(s_1, \ldots, s_n) \star E_0$. By Lemma 1, up to probability $\varepsilon_{\mathsf{seq}}(q)$, E_{out} is the

 $\Omega^{\mathcal{O}_{\mathsf{act}}}(s_1,\ldots,s_n)$

1:Set $trs_0 \leftarrow \perp$ For all $r \in \{1, ..., k - 1\}$: 2:For all $i \in \{1, ..., n\}$: 3: $M_{r,i} \leftarrow P_i^{\mathcal{O}_{\mathsf{act}}}(s_i, \mathsf{trs}_{r-1})$ 4: // Update the view at the end of a round 5: $\mathsf{trs}_r \leftarrow \mathsf{trs}_{r-1} \cup \{M_{r,i}\}_{i=1}^n$ 6: 7:// For the last round, execute all parties until we get the output For $i \in \{1, ..., n\}$: 8: If $E_{\text{out}} \leftarrow P_i^{\mathcal{O}_{\text{act}}}(s_i, \text{trs}_{k-1})$: 9: **Return** $(f(s_1,\ldots,s_n), E_{out})$ 10 :

Fig. 10. Environment Ω executing P_1, \ldots, P_n to compute $E_{out} = f(s_1, \ldots, s_n) \star E_0$.

output of the $\xi(r', i', j')$ -th query and that $0 \to^+ \xi(r', i', j')$. This implies that E_{out} is the output of the j'-th query $P_{i'}$ performs at round r', but it is not enough to imply $0 \to^+ (r', i', j')$. Indeed, chains from E_0 to E_{out} may involve queries performed at the same round by different players, something \to^+ does not allow. This is addressed by the following claim.

Claim 1 If $0 \to^+ \xi(r, i, j)$, then up to probability $r \cdot \varepsilon_{seq}$ we have $0 \to^+ (r, i, j)$, *i.e.*

 $\Pr\left[0 \to^{+} \xi(r, i, j), \quad 0 \not\Rightarrow^{+} (r, i, j)\right] \leq r \cdot \varepsilon_{\mathsf{seq}}$

We immediately observe this claim implies the thesis as

$$\Pr\left[\nexists r, i, j : E_{\mathsf{out}} = E_{r,i,j}, \quad 0 \to^+ (r, i, j)\right]$$

$$\leq \Pr\left[\nexists r, i, j : E_{\mathsf{out}} = E_{r,i,j}, \quad 0 \to^+ \xi(r, i, j)\right] +$$

$$+ \Pr\left[E_{\mathsf{out}} = E_{r',i',j'}, \quad 0 \to^+ \xi(r', i', j'), \quad 0 \not\to^+ (r', i', j')\right]$$

$$\leq \varepsilon_{\mathsf{seq}} + r' \varepsilon_{\mathsf{seq}} \leq (r'+1)\varepsilon_{\mathsf{seq}} \leq (k+1)\varepsilon_{\mathsf{seq}}.$$

Proof of Claim 1. Proceeding by induction on r, the base case r = 0 is trivially true. Assume now the statement to be true for all r' < r. We then construct \mathcal{A} computing $E_{r,i,j}$ which behaves as Ω for the first r-1 rounds. At round r it initially executes P_i and then the remaining P_1, \ldots, P_n . If it finds a path $0 \rightarrow^+ \xi(r, i, j)$, it computes α such that $E_{\text{out}} = \alpha \star E_0$ and returns (α, E_{out}) , otherwise it aborts. A full description appears in Figure 11.

First, as for Ω , we can define an indexing partial function $\eta : \mathbb{N}^3 \to \mathbb{N}$ such that the *j*-th query performed by P_i at round *r* is also the $\eta(r, i, j)$ -th query of \mathcal{A} . By Sequentiality Lemma 1 we get

$$\Pr\left[0 \to^+ \xi(r, i, j), \ 0 \not\to^+ \eta(r, i, j)\right] \le \varepsilon_{\mathsf{seq}}$$

 $\mathcal{A}^{\mathcal{O}_{\mathsf{act}}}(s_1,\ldots,s_n)$

1: // Behave as Ω for the first r-1 rounds 2: Set $trs_0 \leftarrow \perp$ 3: **For** all $r' \in \{1, \ldots, r-1\}$: For all $i' \in \{1, ..., n\}$: 4: $M_{r,i'} \leftarrow P_{i'}^{\mathcal{O}_{\mathsf{act}}}(s_{i'}, \mathsf{trs}_{r'-1})$ 5: $\mathsf{trs}_{r'} \leftarrow \mathsf{trs}_{r'-1} \cup \{M_{r',i'}\}_{i'=1}^n$ 6: 7: // Execute P_i first at round r 8: Run $P_i^{\mathcal{O}_{\mathsf{act}}}(s_i, \mathsf{trs}_{r-1})$ 9: **For** $i' \in \{1, ..., n\} \setminus \{i\}$: Run $P_{i'}^{\mathcal{O}_{\mathsf{act}}}(s_{i'}, \mathsf{trs}_{r-1})$ // Find a chain according to Ω 's query order 10: 11: If $0 \not\rightarrow^+ \xi(r, i, j)$: Return \perp 12: Use a chain $0 \to^+ \xi(r, i, j)$ to find α such that $E_{r,i,j} = \alpha \star E_0$ 13 : **Return** $(\alpha, E_{r,i,j})$

Fig. 11. Program \mathcal{A} computing $E_{r,i,j}$.

where we used the fact that $0 \to \xi(r, i, j)$ implies that \mathcal{A} 's outputs satisfies $\alpha \star E_0 = E_{\text{out}}$ and $\eta(r, i, j)$ is known to be the query index in which \mathcal{O}_{act} returns $E_{r,i,j}$.

Assuming instead $0 \rightarrow^+ \eta(r, i, j)$, let (r', i', j') the maximal element in the chain such that r' < r. Then we will show that

$$0 \to^+ \eta(r, i, j) \quad \Rightarrow \quad 0 \to^+ \xi(r', i', j') \land (r', i', j') \to^+ (r, i, j).$$

The first part of the implication follows since for all rounds before the r-th, \mathcal{A} and Ω behaves identically. Therefore they make the same queries in the same order, meaning that $0 \rightarrow^+ \eta(r', i', j')$ if and only if $0 \rightarrow^+ \xi(r', i', j')$. For the second part, from the way we defined (r', i', j'), all queries in a chain from (r', i', j') to (r, i, j) occurs at round r. Since \mathcal{A} first executes P_i at round r, it means all these queries only involves P_i , therefore $(r', i', j') \rightarrow^+ (r, i, j)$.

As a consequence we can bound the studied probability in the case $0 \rightarrow^+ \eta(r, i, j)$ as follows:

$$\begin{aligned} &\Pr\left[0 \to^{+} \xi(r, i, j), \ 0 \not\Rightarrow^{+} (r, i, j), \ 0 \to^{+} \eta(r, i, j)\right] \\ &\leq &\Pr\left[0 \to^{+} \xi(r, i, j), \ 0 \not\Rightarrow^{+} (r, i, j), \ 0 \to^{+} \xi(r', i', j'), \ (r', i', j') \to^{+} (r, i, j)\right] \\ &\leq &\Pr\left[0 \to^{+} \xi(r', i', j'), \ 0 \not\Rightarrow^{+} (r', i', j')\right] \\ &\leq &r' \cdot \varepsilon_{\mathsf{seq}}. \end{aligned}$$

where the last step follows by induction. Note that the second inequality follows because on LHS, if $0 \rightarrow^+ (r', i', j')$ then by transitivity $0 \rightarrow^+ (r, i, j)$, contradicting the clause $0 \not \rightarrow^+ (r, i, j)$. The event on the LHS then implies $0 \not \rightarrow^+ (r', i', j')$. In

conclusion, since r' < r

$$\begin{aligned} &\Pr\left[0 \rightarrow^{+} \xi(r, i, j), \ 0 \not\Rightarrow^{+} (r, i, j)\right] \\ &\leq &\Pr\left[0 \rightarrow^{+} \xi(r, i, j), \ 0 \not\Rightarrow^{+} \eta(r, i, j)\right] + \\ &\quad + &\Pr\left[0 \rightarrow^{+} \xi(r, i, j), \ 0 \not\Rightarrow^{+} (r, i, j), \ 0 \rightarrow^{+} \eta(r, i, j)\right] \\ &\leq &\varepsilon_{\mathsf{seq}} + r' \varepsilon_{\mathsf{seq}} \leq r \varepsilon_{\mathsf{seq}}. \end{aligned}$$

C.4 Refined Interactive Sequentiality Lemma

Proof of Lemma 5. The proof is similar to Lemma 4. As in that case, we define Ω as in Figure 10 and $\xi(r, i, j)$ and indexing function so that the *j*-th query performed by P_i at round *r* corresponds to the $\xi(r, i, j)$ -th performed by Ω . Through the Sequentiality Lemma 1 we then have that up to probability ε_{seq}

$$E_{\mathsf{out}} = E_{r',i',j'} \qquad 0 \to^+ \xi(r',i',j').$$

For notational convenience we call $\operatorname{Ref}(r, i, j)$ the event $0 \rightarrow^+ (r, i, j)$ through a refined chain. To conclude we need to prove the following claim

Claim 1 $\Pr[0 \to^+ \xi(r, i, j), \neg \mathsf{Ref}(r, i, j)] \leq r \cdot \varepsilon_{\mathsf{seq}}.$

Proof of Claim 1. We proceed by induction on r. If r = 0 the statement is trivially true. Assuming it holds for all r' < r we will prove it for (r, i, j). Let j^* be a minimal element in the poset of queries performed by P_i at round r, ordered with \rightarrow^+ .

By definition of \rightarrow^+ then there exists a chain j_0, \ldots, j_t of queries such that

$$(r, i, j^*) = (r, i, j_0) \to (r, i, j_1) \to \ldots \to (r, i, j_t) = (r, i, j)$$

and in particular $(r, i, j^*) \rightarrow^+ (r, i, j)$. If $0 \rightarrow^+ \xi(r, i, j)$ then, summing the group elements appearing in a chain for this relation and for $(r, i, j^*) \rightarrow^+ (r, i, j)$ we would obtain two group elements α, β such that

$$\alpha \star E_0 \ = \ E_{r,i,j}, \quad \beta \star E_{r,i,j^*} \ = \ E_{r,i,j} \quad \Rightarrow \quad E_{r,i,j^*} \ = \ (-\beta + \alpha) \star E_0$$

Summing the group elements appearing in this chain of queries, as well as Next we define \mathcal{A} which computes E_{r,i,j^*} by executing at round r first P_i and then all other users. A full description of \mathcal{A} appears in Figure 12.

As done with Ω , we define an indexing function η so that the *j*-th query performed by P_i at round *r* is \mathcal{A} 's $\eta(r, i, j)$ -th query. Since \mathcal{A} and Ω executes parties in the same order until round *r*, ξ and η agrees for r' < r.

Next, applying the Sequentiality Lemma 1 we have that

$$\Pr\left[0 \to^{+} \xi(r, i, j), \quad 0 \not\to^{+} \eta(r, i, j^{*})\right] \leq \varepsilon_{\mathsf{seq}}$$

because \mathcal{A} computes the correct element if $0 \to^+ \xi(r, i, j)$ and $(r, i, j^*) \to^+ (r, i, j)$ which is true by construction.

 $\mathcal{A}^{\mathcal{O}_{\mathsf{act}}}(s_1,\ldots,s_n)$

1: // Behave as Ω for the first r-1 rounds 2: Set $trs_0 \leftarrow \perp$ 3: **For** all $r' \in \{1, \ldots, r-1\}$: For all $i' \in \{1, ..., n\}$: 4: $M_{r,i'} \leftarrow P_{i'}^{\mathcal{O}_{\mathsf{act}}}(s_{i'}, \mathsf{trs}_{r'-1})$ 5: $\mathsf{trs}_{r'} \leftarrow \mathsf{trs}_{r'-1} \cup \{M_{r',i'}\}_{i'=1}^n$ 6: 7: // Execute P_i first at round r 8: Run $P_i^{\mathcal{O}_{\mathsf{act}}}(s_i, \mathsf{trs}_{r-1})$ 9: For $i' \in \{1, \ldots, n\} \setminus \{i\}$: Run $P_{i'}^{\mathcal{O}_{\mathsf{act}}}(s_{i'}, \mathsf{trs}_{r-1})$ // Find a chain according to Ω 's query order 10 : 11: If $0 \not\rightarrow^+ \xi(r, i, j)$: Return \perp Use a chain $0 \to^+ \xi(r, i, j)$ to find α such that $E_{r,i,j} = \alpha \star E_0$ 12:Use a chain $(r, i, j^*) \rightarrow^+ (r, i, j)$ to find β such that $E_{r,i,j} = \beta \star E_{r,i,j^*}$ 13:14 : **Return** $(-\beta + \alpha, E_{r,i,j^*})$

Fig. 12. Program \mathcal{A} computing E_{r,i,j^*} .

Next, assuming $0 \to^+ \eta(r, i, j^*)$, let (r', i', j') the predecessor in a given chain for this relation. We will show the following implications:

$$\begin{bmatrix} 0 \to^{+} \xi(r, i, j) \\ \neg \mathsf{Ref}(r, i, j) \\ 0 \to^{+} \eta(r, i, j^{*}) \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} 0 \to^{+} \xi(r, i, j) \\ r' < r \\ \neg \mathsf{Ref}(r', i', j') \end{bmatrix}$$

The first is trivial. The second one follow since if r = r' then $\eta(r, i', j') \rightarrow \eta(r, i, j^*)$ implies i = i' since P_i is the first executed player in round r. In particular this implies $(r, i, j') \rightarrow (r, i, j^*)$ which contradicts the minimality of j^* . We also prove the third one by contradiction. If Ref(r', i', j') then we can extend any refined chain for $0 \rightarrow^+ (r', i', j')$ with

$$(r',i',j') \rightarrow (r,i,j^*) \rightarrow^+ (r,i,j)$$

where the first relation is true as r' < r and $\eta(r', i', j') \rightarrow \eta(r, i, j^*)$ and the second one follows by construction of j^* . By minimality of j^* the resulting chain would be refined, implying $\mathsf{Ref}(r, i, j)$, which is assumed to be false.

Using this implication we next bound the following probability:

$$\begin{split} &\Pr\left[0 \rightarrow^{+} \xi(r, i, j), \ \neg \mathsf{Ref}(r, i, j), \ 0 \rightarrow^{+} \eta(r, i, j^{*})\right] \\ &\leq &\Pr\left[0 \rightarrow^{+} \xi(r, i, j), \ \neg \mathsf{Ref}(r', i', j'), \ r' < r\right] \\ &\leq &r' \cdot \varepsilon_{\mathsf{seq}} \ \leq \ (r-1)\varepsilon_{\mathsf{seq}}. \end{split}$$

Finally, combining the two bounds we conclude that the claim is true.

$$\begin{aligned} &\Pr\left[0 \rightarrow^{+} \xi(r, i, j), \ \neg \mathsf{Ref}(r, i, j)\right] \\ &\leq &\Pr\left[0 \rightarrow^{+} \xi(r, i, j), \ 0 \not\rightarrow^{+} \eta(r, i, j^{*})\right] \\ &+ &\Pr\left[0 \rightarrow^{+} \xi(r, i, j), \ \neg \mathsf{Ref}(r, i, j), \ 0 \rightarrow^{+} \eta(r, i, j^{*})\right] \\ &\leq &\varepsilon_{\mathsf{seq}} + (r - 1)\varepsilon_{\mathsf{seq}} = r\varepsilon_{\mathsf{seq}}. \end{aligned}$$

C.5 Tall Sub-tree Properties

Proof of Proposition 1. We proceed by induction on $n \geq 1$. If n = 1 then $n \log_2 n = 0$ and T is a chain, implying |E| = m. Assuming the thesis holds for all n' < n, we prove it for n. Let $r \in V$ be the root of T and v_1, \ldots, v_d be the successors of r. Then $T_i \coloneqq T_{v_i} = (V_i, E_i)$ is a TS trees and, calling $m_i = \mathsf{height}(T_i)$ and $n_i = \mathsf{leaves}(T_i)$, we have that

$$|E| = d + \sum_{i=1}^{d} |E_i| \ge d + \sum_{i=1}^{d} m_i + \sum_{i=1}^{d} n_i \log_2 n_i$$

$$\ge d + d(m-1) + n \log_2(n/d)$$

$$= m + n \log_2 + ((d-1)m - n \log_2 d)$$

where the first inequality is the inductive hypothesis and the second one uses $m_i = m - 1$ and the convexity of $x \log_2 x$. To conclude we need to show that for all integer $d \ge 1$ the last term in brackets is non negative. This is true for d = 1 and, because $m \ge n$, for d = 2. Finally, for any d > 2 the function $f(d) = (d-1)m - n \log_2 d$ is increasing as its derivative is

$$f'(d) = m - \frac{n}{d \cdot \ln 2} \ge m - n \ge 0.$$

Therefore $f(d) \ge 0$ for all integer d and the proof is complete.

Proof of Proposition 2. If $|V_{\geq 2}| = 0$ then $m \leq 1$ and therefore $n \leq 1$, in which case the inequality holds. Conversely let v_1, \ldots, v_d be the root's children, and call $T_i \coloneqq T_{v_i} = (V_i, E_i)$. Let $m_i = \mathsf{height}(T_i), n_i = \mathsf{leaves}(T_i)$. Using the fact that T_1, \ldots, T_d are TS trees and the same inequalities used in the proof of Proposition 1

$$|V_{\geq 2}| = \sum_{i=1}^{d} |V_i \setminus v_i|$$

= $\sum_{i=1}^{d} m_i + n_i \log_2 n_i$
= $dm - d + n \log_2(n/d)$
= $m + n \log_2 n - 2 + ((d - 1)m - d - n \log_2 d).$

To conclude we have to show the last term in brackets is always non negative. For d = 1 is trivial and d = 2 is implied by $m \ge n$. For $d \ge 3$ we would have $n \ge 3$ in which case $(2 - \log_2 3)n - 1$ is positive. Finally for d > 3 the derivative of that term in d is always positive, using the fact that $n \ge 3$.