Disincentivize Collusion in Verifiable Secret Sharing

Tiantian Gong[1], Aniket Kate[1,2], Hemanta K. Maji[1], and Hai H. Nguyen[3]

[1] Purdue University, West Lafayette IN 47906, USA
{tg,aniket,hmaji}@purdue.edu
[2] Supra Research
[3] ETH Zurich, Zürich, Switzerland
nhhai196@gmail.com

Abstract. In verifiable secret sharing (VSS), a *dealer* shares a secret input among several *parties*, ensuring each share is verifiable. Motivated by its applications in the blockchain space, we focus on a VSS where parties holding shares are *not* allowed to reconstruct the dealer's secret (even partially) on their own terms, which we address as *privacy-targeted collusion* if attempted.

In this context, our work investigates mechanisms deterring such collusion in VSS among rational and malicious parties. For this problem, we make both algorithmic and combinatorial contributions:

1. We provide two collusion-deterrent mechanisms to discourage parties from colluding and recovering the dealer's secret. Notably, when it is desired to achieve *fairness*—where non-colluding parties are not at a loss—while allowing for the best achievable malicious fault tolerance, we define "**trackable access structures**" (TAS) and design a deterrence mechanism tailored for VSS on these structures.
2. We estimate the size of the optimal TAS, construct them from Steiner systems, provide highly robust TAS using partial Steiner systems, and present efficient secret sharing schemes for the latter close-to-optimal TAS for various parameter regimes.
3. We demonstrate that **trackability** in access structures is connected to combinatorial objects like (partial) Steiner systems, uniform subsets with restricted intersections, and appropriate binary codes. The **robustness** of access structures is equivalent to the minimum vertex cover of hypergraphs.

We believe these connections between cryptography, game theory, and discrete mathematics will be of broader interest.

1 Introduction

Consider a threshold multi-device cryptocurrency wallet [2, 41]. A user shares her transaction signing key among many servers using a secret sharing scheme, and a threshold number of servers are expected to endorse transactions on the user's behalf using aggregatable partial signatures. However, nothing stops these servers from performing unauthorized transactions on the user's behalf by generating unauthorized partial signatures or simply reconstructing her signing key. This is not an isolated risk; *in most secret-sharing applications, colluding actors may surreptitiously recover cryptographic secrets before they are allowed to.* This concern of collusion to break privacy is commonplace in secret-sharing applications such as secure multi-party computations (MPC) [8, 15, 24, 46, 56], time-release encryption [47], distributed key generation [23, 36], distributed randomness beacons [1] and electronic voting [51]: universally, all bets are off when more than a threshold number of servers get compromised.

Considering the potentially catastrophic loss, this work *disincentivizes collusion by relying on rational behavior*—the threat of snitching will keep bad (yet sensible) actors in line. Here, rationality means taking actions that maximize one's utility. Previously, traceable secret sharing [12, 26] traced colluder(s) only when given access to a *pirate reconstruction program* created by colluding parties. Dziembowski et al. [21] allowed a colluder to generate fraud proofs against a target party when they collude via MPC (assuming the hardness of computing many hashes quickly with MPC). Rational parties are then discouraged from such collusion. However, as in the above applications, colluding parties need not necessarily construct a reconstruction box for a detector to query or run MPC. Instead, they may collude over alternative (even unforeseen) channels, e.g., outsourced cloud computing based on homomorphic encryption.

As such, our work investigates mechanisms to disincentivize collusion in verifiable secret sharing (VSS) [18] in a significantly harsher setting. We make *no assumptions* on how parties collude; unlike the aforementioned works, there is no pirate reconstruction program or MPC transcripts. We *only* rely on some parties being rational; others are malicious and can behave arbitrarily—even engaging in self-harm. Note that we only deter *successful collusion* where some party learns some non-trivial secret information (defined in Section 7).

Gong et al. [25] recently studied collusion deterrence in multiserver private information retrieval with a singleton access structure.[4] Their mechanism created a race among colluding parties to report and prove some non-trivial secret information; then nobody wanted others to learn about the secret. Incorporating their mechanism into secret sharing encounters two challenges. First, their mechanism fails even when one party is missing during reconstruction, for example, due to a benign failure. To address this, we measure and increase the *robustness* of access structures—the smallest number of absences that stall reconstruction.[5]

Finally, more worryingly, when used in general access structures, a colluding group can frame innocents. Consider the k-out-of-n threshold access structure: k parties can collude and frame any other party as a colluder because the parties are interchangeable, and the mechanism cannot tell the colluder set. We then define *trackable access structures* (TAS) and design a collusion deterrence mechanism $\mathbf{W}_1$ (described on page 5) for TAS.

1.1 Contributions

In summary, first, we design a collusion deterrence mechanism for VSS on arbitrary access structures. Second, we identify more structured homogeneous access structures – TAS – and create a deterrence mechanism with stronger guarantees for VSS over a TAS. We investigate the structural properties of TAS constructions by connecting them to various combinatorial objects.

Collusion deterrence mechanisms. Consider n parties where a strict subset of them are malicious, and the rest are rational. For $2 \leqslant k \leqslant n$, we consider (monotone) k-*homogeneous* access structures $\mathcal{A} \subseteq 2^{[n]}$ where $[n] = \{1, 2, \ldots, n\}$: All minimal sets in it have size k. An access structure $\mathcal{A}$ is ω-*trackable* when any size-ω subset is contained in at most one minimal set. For example, an (n, k, ω)-design, as defined in Section 6.1, is ω-trackable; every size-ω subset appears in a unique minimal set in this design.

We characterize and compare the two mechanisms along the following metrics in Table 1.

t_e-Effective: The mechanism induces the non-collusion outcome when there are $\leqslant t_e$ malicious parties.
t_f-Fair: Non-colluding parties are not at a loss in this mechanism with $\leqslant t_f$ malicious parties.
φ-Fairness hazard: With $(k-1)$ malicious parties, the maximum meaningful number to consider, the mechanism mislabels $\leqslant \varphi$ non-colluders as colluders.
t_r-Robust: The protocol-initiated reconstruction can be stalled only by $\geqslant t_r$ absentees. That is, reconstruction goes through with $< t_r$ absentees.

Note that $t_r \leqslant (n-k+1)$ because any minimal set has size $\geqslant k$. Naturally, it is only meaningful to consider $t_e, t_f < k$; otherwise, the situation is hopeless since k malicious parties can reconstruct the secret at will. For these metrics, increasing t_e, t_f, t_r and decreasing φ is desirable.

Our first mechanism. For an arbitrary k-homogeneous access structure, we provide a mechanism $(\mathbf{W}_0, \mathbf{P})$ on page 4. Here, $\mathbf{W}_0$ is the winner selection rule, and $\mathbf{P}$ is the payment rule. The mechanism $(\mathbf{W}_0, \mathbf{P})$ is a public algorithm specified by the two rules, and any party can observe and interact with it. It is $(k-2)$-effective, $(k-2)$-fair, has $(n-k)$-fairness hazard, and has optimal $(n-k+1)$-robustness.

The mechanism is similar to the one in [25]. It encourages colluders to call attention to collusion by proving their knowledge of the secret. It discourages collusion by inducing a race among rational colluders to be the first to submit proofs, resulting in parties who are not faster than others unwilling to collude.

[4] All queried parties need to provide inputs to reconstruct the client's queried index.

[5] In robust secret sharing [10,17], robustness is the ability to tolerate wrong shares in reconstruction. In VSS, shares are already verifiable, so the threat of wrong shares is mitigated. In our context, robustness then only concerns absentees in reconstruction.

Table 1: Evaluation of our two mechanisms. For fixed n, k, the function $f^{(n,k)}(\omega)$ is monotonically increasing in ω; summarized in Table 2.

Mechanisms	Malicious fault bounds		Fairness hazard	Applicable access structures	Robustness
	Fairness	Effectiveness			
$(\mathbf{W}_0, \mathbf{P})$	$k-2$	$k-2$	$n-k$	Any	$n-k+1$
$(\mathbf{W}_1, \mathbf{P})$	$k-1$	$k-1-\omega$	0	ω-trackable	$f^{(n,k)}(\omega)$

Our second mechanism. For ω-trackable access structures ($1 \leqslant \omega < k$), we propose $(\mathbf{W}_1, \mathbf{P})$ on page 5. It is $(k-1)$-fair, $(k-1-\omega)$-effective and has optimal 0 fairness hazard. Its robustness if $f^{(n,k)}(\omega)$, an appropriate increasing function defined in Table 2, indicating a *trade-off between effectiveness and robustness* when setting ω. In Section 5.1, we prove that trackability is necessary to achieve zero fairness hazard: zero fairness hazard is impossible to achieve in untrackable access structures.

Informally, in $\mathbf{W}_1$, while parties from multiple minimal sets may report, the mechanism penalizes the last *free rider* (i.e., a party who does not make up a complete minimal set along with any other reporters) or the last reporter if there are no free riders. Otherwise, it locates a unique minimal set in the TAS and penalizes the remaining parties in it.

TAS and VSS. Given parameters n, k, ω, our objective is to construct efficient secret-sharing schemes on (n, k, ω)-TAS with high robustness. We systematically examine secret sharing on TAS, uncovering several connections with combinatorics: We establish links between TAS and well-known concepts in combinatorial design, coding theory, extremal graph theory, and additive combinatorics.

Specifically, TAS is equivalent to the partial Steiner system, binary constant weight code with high distance, and uniform subsets with restricted intersections. Additionally, there is a natural equivalence between the robustness of an access structure and the minimum vertex cover in hypergraphs, with its dual notion being the independence number. Leveraging these connections, we employ techniques and results from these fields to derive the following results:

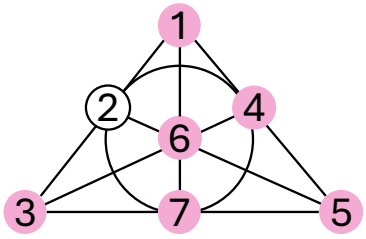
1) A tight upper bound on the size of any TAS (Theorem 3).
2) Constructions of TAS with optimal size (Theorem 5 and 6) and near-optimal size (Theorem 8).
3) Constructions of TAS with (asymptotically) optimal robustness (Corollary 3).

We present an efficient construction of secret sharing for $(n, k, 2)$-TAS with (asymptotically) optimal robustness—the ratio between the number of parties to corrupt and the number of parties tends to one. In $(n, k, 2)$-TAS, the number of minimal authourized sets is $O(n^2)$ (see Theorem 3). Naively applying generic constructions [9,30] results in an information ratio of $O(n^2)$, while the information ratio of our construction is $O(n)$, demonstrating a factor of n improvement. For $\omega \geqslant 3$, we apply the generic constructions for any access structures, resulting in an information ratio of roughly $O((n/k)^\omega)$. More efficient constructions remain open and are left for future work. Finally, the verifiable version of the secret sharing scheme can be constructed by applying generic transformations [5].

Lower bound. The family of TAS is a subclass of k-homogeneous access structures. Recently, Beimel [6] proves a lower bound of $n^{2-1/(k-1)/k}$ for some explicit k-homogeneous access structures—a simple variant of the ones considered by Csirmaz [20]. Interestingly, these structures are also $(k-2)$-trackable. As a result, the lower bound extends to TAS.

An example. Fig. 1a presents a Steiner system $S(n = 7, k = 3, \omega = 2)$. Construct an $\omega = 2$-trackable access structure $\mathcal{A}$ whose minimal sets are the hyperedges in the Steiner system; there are $\binom{n}{\omega} \cdot \binom{k}{\omega}^{-1} = 7$ such sets. This system has a minimum vertex cover of size $t_r = 3$. Our $(\mathbf{W}_1, \mathbf{P})$ mechanism is $(k-1) = 2$-fair, $(k-1-\omega) = 0$-effective, has 0 fairness hazard, and $t_r = 3$-robustness.

Likewise, consider the Steiner system $S(13, 4, 2)$, the projective plane of order 3, is 2-trackable. In a projective plane, the vertices of any hyperedge form a minimum vertex cover; so, here $t_r = 4$. Our $(\mathbf{W}_1, \mathbf{P})$ mechanism is 3-fair, 1-effective, has 0 fairness hazard, and 4-robustness.



(a) Consider a 2-trackble access structure for 7 parties with minimal sets $\{1,2,3\}$, $\{1,4,5\}$, $\{1,6,7\}$, $\{2,5,6\}$, $\{3,4,6\}$, $\{3,5,7\}$, $\{2,4,7\}$ where any 2 parties pinpoint one minimal set. Suppose $1,3,4,5,6,7$ (marked in pink) have colluded.

(b) Suppose parties 1, 3, 4, 5, and 6 have submitted reports. If party 7 reports, 7 is considered the last reporter, and there will be no free riders by then. If 7 is rational, it does not report due to the penalty on the last reporter (induced by rule 1.B) in $\mathbf{W}_1$. The same reasoning applies to party 6 given that 7 will not report.

(c) Given that 6 and 7 will not report, party 3 is considered a free rider (and unfortunately, the last one). If party 3 is rational, it does not report due to the penalty on the last free rider (induced by rule 1.A) in $\mathbf{W}_1$.

(d) Given that party 3 will not report, party 5 is considered the last reporter. If 5 reports, it is penalized as the last reporter. Otherwise, 5 is penalized as the remaining colluder of the group $\{1,4,5\}$ (due to rule 2) in $\mathbf{W}_1$. This in turn discourages 5 from colluding with $1,4$.

Fig. 1: An example illustrating that the slowest parties in minimal sets are disincentivized from collusion under $(\mathbf{W}_1, \mathbf{P})$. Colluding parties are colored in pink.

2 Technical overview

We overview our techniques and analysis in this section.

2.1 Collusion-deterrent mechanisms

Starting point. In privacy-targeted collusion in secret sharing, colluding parties can reconstruct the secret using external unobserved communication channels. Our goal is to design a mechanism that *indirectly* deters privacy-targeted collusion, assuming a blend of rational and malicious parties. Recall that we consider homogeneous access structures with size-k minimal sets. We eventually aim to tolerate up to $(k-1)$ malicious parties—which is optimal when allowing for adaptive malicious corruptions—with the remaining parties being rational.

First, consider the following simple mechanism:

Winner selection rule ($\mathbf{W}_0$): The first party to prove non-trivial knowledge of the secretis selected as the *winner*; All other parties are marked as *colluders*.
Payment rule ($\mathbf{P}$): Reward the (selected) *winner* and penalize the (marked) *colluders* with appropriate amounts.

We define the proof of knowledge and parameterize payment amounts explicitly in Section 4 and 5. With proper proof verification and payment parameterization, $(\mathbf{W}_0, \mathbf{P})$ discourages collusion when there are at least two rational parties in each access group in $\mathcal{A}$. The simple scheme $(\mathbf{W}_0, \mathbf{P})$ tolerates $(k-2)$ malicious parties as we make use of the race between the two remaining rational parties. Informally, the rationale is that first, rational colluders are incentivized to submit a proof to escape the penalty. Second, since at most one rational colluding party can become the winner, at least one rational party without a network advantage in submitting reports is restrained from collusion.

However, when one more malicious party is present, collusion can take place since the only remaining rational party in the access group may collude, report, become the winner, and avoid penalty. In this case of $(k-1)$ malicious parties, $(\mathbf{W}_0, \mathbf{P})$ becomes ineffective, and $(n-k)$ non-colluding parties bear penalties.

Consequently, we desire to reduce the fairness hazard, preferably to 0, where non-colluding parties are never at a loss when faced with up to $(k-1)$ malicious parties. An immediate attempt is to select a general number of winners, say $\omega < k$ winners. However, this does not boost the mechanism's fault tolerance because it still only treats the collusion reports as the signal of collusion *existence*, not its *participants*. Besides, it reduces the malicious fault tolerance to $(k-1-\omega)$.

Tolerate $(k-1)$ malicious parties. Intuitively, the ability to locate remaining colluders given a strict subset of colluding parties can potentially help improve malicious fault tolerance for achieving fairness. To this end, we are interested in one trait of a k-homogeneous access structure: given any set of ω $(1 \leqslant \omega < k)$ parties, is there either a *unique minimal set* in the access structure that contains them or none? We address this trait as ω-trackability: any ω parties belong to at most one minimal group. Given an (n, k, ω)-TAS, a naïve extension to the previous mechanism is as follows with the same payment rule $\mathbf{P}$:

Winner selection rule ($\tilde{\mathbf{W}}_1$): The first ω parties to prove non-trivial knowledge of the secret are selected as *winners*. If there is a unique minimal access group containing the winners, mark the remaining parties therein as *colluders*.

Now the ω parties not only signal the existence but also potential participants of collusion. When exactly one minimal set colludes, the extended mechanism achieves effective collusion deterrence against $(k-1-\omega)$ malicious parties and achieves fairness against $(k-1)$ malicious parties. The latter bound is because $(k-\omega)$ malicious parties suffice to trigger collusion in their access group and $(\omega-1)$ remaining malicious party can help with falsely incriminating non-colluding parties in the worst case.

However, when more minimal sets collude in close time proximity (measured according to network delays), the above scheme loses its charm. This is because the first ω parties revealing collusion are now potentially from distinct groups and a third party (i.e., a public algorithm or an entity implementing the mechanism) cannot tell. Non-colluding parties can be falsely located, and fairness can no longer be guaranteed. Worse still, non-colluding parties can be the only ones marked as colluders, rendering the mechanism ineffective in deterring collusion. For example, consider four 2-trackable minimal groups $\{1,2,3\}$, $\{1,4,5\}$, $\{2,4,6\}$, $\{3,5,6\}$. Suppose $1,3,4,5,6$ collude, and $1,3$ report first. $\tilde{\mathbf{W}}_1$ identifies 2 as colluder. This is *neither effective nor fair even when no party is malicious*.

Hence, trackability alone does not guarantee fairness and effectiveness. We need to update the winner selection rule so that a sufficient number of colluding parties report to help spot colluders without noises that inculpate non-colluding parties. In this way, collusion among parties from more than one access groups does not neutralize the mechanism's effectiveness or jeopardize fairness. The updated rule $\mathbf{W}_1$ aims to encourage ω colluding parties in any one of the colluding groups to submit reports and others to stay silent to avoid noisy signals.

Winner selection rule ($\mathbf{W}_1$):

1. When there are more than ω but less than k reporters, jump to rule B. When there are $\geqslant k$ reporters, go to rule A.
   - A. For a party that does not make up a complete access group along with any other $(k-1)$ reporters, mark it as *free rider*. The last free rider is marked as *colluder* and all other reporters are marked as *winners*.
   - B. Otherwise, mark the last reporter as *colluder* and others as *winners*.
2. When there are ω reporters, and there exists a unique minimal group containing them, mark the ω reporters as *winners* and the remaining parties in that group as *colluders*.

In other cases, dismiss the reports.

We demonstrate that parties are disincentivized from colluding under $\mathbf{W}_1$ in Section 5 and give a visual proof in Fig. 1. Intuitively, this is because it filters out noises in reports: under $\mathbf{W}_1$, when a minimal group already exposes itself with ω reporters, rational colluding parties from other minimal groups are disincentivized to report as they risk being the last free rider or the last reporter; the slowest parties in the access group with the fastest ω parties are then discouraged from collusion, and so on.

2.2 TAS and VSS

Let $\mathcal{A}^\star$ denote the minimal sets of an (n, k, ω)-trackable access structure $\mathcal{A}$. As before, we also address $\mathcal{A}^\star$ as the minimal access structure.

Equivalence between TAS and partial Steiner system. A partial Steiner system is a combinatorial structure generalizing the Steiner system. In a Steiner system, the main idea is to find subsets (blocks) of a given ground set such that every subset of a certain size (called the "block size") is covered by exactly one block. In a partial Steiner system, this requirement is relaxed such that every subset of the given size is covered by at most one block. This allows for more flexibility in the construction of the system and can lead to partial solutions when full Steiner systems are not feasible or available. More formally, an $S_p(n, k, \omega)$ *partial Steiner system* is defined by a subset $\mathbb{K} \subseteq \binom{[n]}{k}$ such that for any subset $T \in \binom{[n]}{\omega}$, there is at most one subset $K \in \mathbb{K}$ satisfying $T \subseteq K$. Observe that an $S_p(n, k, \omega)$ partial Steiner system is an (n, k, ω)-TAS, and vice-versa.

TAS as a special case of uniform subsets with restricted intersections. Fix a set $L \subseteq \{0, 1, 2, \dots\}$. A family of k-uniform subsets $\mathcal{S} \subseteq \binom{[n]}{k}$ is L-intersecting if all distinct subsets $E, F \in \mathcal{S}$ satisfy $|E \cap F| \in L$. Observe that a (n, k, ω)-TAS is a k-uniform subsets with $L = \{1, 2, \dots, \omega - 1\}$.

Remark 1. In secret sharing context, recent works starting from [37] use (sparse) matching vectors and conditional disclosure of secret protocols to construct more efficient secret sharing for general access structures (see [7, page 5]).

Upper bound of the size of any (n, k, ω)-TAS. From a coding theory perspective, a subset of $[n]$ can be mapped to a bit string in $\{0, 1\}^n$, with each bit indicating the membership of each element. A minimal access structure $\mathcal{A}^\star$ is a set of weight-k bit strings with $(\omega - 1)$ pair-wise intersection. Thus, the set of bit strings in $\mathcal{A}^\star$ form a (non-linear) code of distance at least $2k - 2(\omega - 1)$. Applying the well-known Johnson bound [31] for binary constant weight code yields the desired bound $\binom{n}{\omega} \cdot \binom{k}{\omega}^{-1}$. The optimal ones can attain the bound if and only if a Steiner system $S(n, k, \omega)$ exists.

Optimal TAS. It follows from the upper bound that constructing optimal TAS reduces to constructing binary constant-weight codes, particularly Steiner systems for some parameter regime. We focus on specific parameters because in general, constructing the maximum size of constant weight codes and Steiner systems is a notoriously challenging problem. Constructing optimal binary constant weight codes and (n, k, ω)-Steiner systems for large values of n, k, ω is a long-standing open problem in coding theory and combinatorial design. We then construct TAS with the largest size for some parameter regimes based on existing constructions of Steiner systems and binary constant weight codes in Section 6.3.

Near-optimal TAS. Optimal TAS is theoretically intriguing. However, as mentioned above, their constructions pose significant challenges, particularly for large parameters. In practical applications, such as the mechanism under consideration in this work, efficient constructions and high robustness are paramount. Moreover, as previously observed, access structures with efficient constructions and concise descriptions are more likely to facilitate efficient secret-sharing constructions. We therefore turn to investigate near-optimal sized TAS with efficient constructions and high robustness.

The first construction, depicted in Fig. 2, relies on Reed-Solomon codes. This construction is a common technique in the literature of uniform subsets with restricted intersections (refer to Theorem 4.11 in [3]). It is effective when k is of order $\sqrt{n}$. We demonstrate that the fractional robustness of this access structure is $1/k$. To address the cases where k is larger, we naturally extend this construction to Fig. 3, based on algebraic geometry codes, albeit with a slight trade-off in other parameters. Notably, both constructions are efficient.

Parameters. n, k, ω satisfying $n > 2k^2$. Let p be a prime between n/k and $2n/k$.
Evaluation places. Let $P_1, P_2, \ldots, P_k$ be k distinct evaluation places in F_p.
Access Structure. Let the set of n parties contains the set $\{(P_i, Q) \colon i \in [k], Q \in F_p\}$. The access structure is defined as

$$\mathcal{A}^\star(n, k, \omega) = \left\{ \{(P_i,\ f(P_i)) \ :\ i \in [k]\} \ :\ f \in F_p[x],\ \deg(f) < \omega \right\}$$

Fig. 2: TAS construction with near-optimal size using Reed-Solomon codes.

Parameters. n, k, ω satisfying $k = \Theta(n)$. Let F/F_q be an algebraic function field.
Evaluation places. Let $D = P_1 + P_2 + \cdots + P_k$ be the sum of k distinct places of F/F_q of degree one.
Access structure. Let G be a divisor with disjoint support from D and $\mathcal{L}(G)$ be the Rieman-Roch space associated with G. Let the set of n parties contain the set $\{(P_i, Q) \colon i \in [k], Q \in F_q\}$. The access structure is defined as

$$\mathcal{A}^\star(n, k, \omega) = \left\{ \{(P_i,\ f(P_i)) \ :\ i \in [k]\} \ :\ f \in \mathcal{L}(G) \right\}$$

Fig. 3: TAS construction with near-optimal size using algebraic geometry codes.

Remark 2. There are randomized constructions for near-optimal partial Steiner Sytems and thus for TAS from combinatorial design [48]. We leave constructing efficient secret sharing for these access structures as an open problem.

High robustness. Recall that the robustness of an access structure is the minimum number of parties to corrupt to make reconstruction impossible. Observe that it is equivalent to the minimum cover of the hypergraph representing the access structure—Each party is a vertex, and each minimal set is a hyperedge. By the monotone property, the robustness equals the minimum cover of the hypergraph representing the minimal access structure.

It is well-known that the dual of the minimum cover problem is the maximum independence set problem. The sum of the minimum cover and the maximum independence set in a hypergraph equals the total number of vertices. Thus, a high robustness access structure is equivalent to a hypergraph with a small independence number—the size of the smallest independent set.

Our construction of highly robust TAS is based on the construction of partial Steiner systems with small independence numbers, which are well-studied in combinatorial design and extremal graph theory. Fix k and ω, using the randomized construction of partial Steiner systems [48], we show that there exists a randomized construction of (n, k, ω)-TAS such that the ratio between the robustness and the number of parties n, called *fractional robustness*, tends to 1 as n tends to infinity. Recently, in the context of randomness extractors, Chattopadhyay and Goodman [13] have presented a deterministic construction based on binary BCH codes and recent results in additive combinatorics. It is also worth mentioning that in an earlier work of [13], Chattopadhyay, Goodman, Goyal, and Li [14] constructed a deterministic construction for a special $S_p(n, 3, 2)$ using the connections with the well-known cap set problems. Table 2 summarizes the best-known lower bound for the optimal robustness value of (n, k, ω)-TAS.

Construct VSS. A straightforward method for constructing trackable secret sharing with generic approaches [9, 30] would entail each party's share size being proportionate to the minimal access structure's size, which is approximately $\binom{n}{\omega}$ for (n, k, ω)-trackable access structures with the largest size. This means that the information ratio can be huge for access structures of large sizes. A natural objective is to construct secret-sharing schemes more efficiently.

The Fano plane access structure—the $S(n, 3, 2)$ Steiner triple system in Fig. 1a admits an ideal secret-sharing scheme [42, 43]. More generally, Martí-Farré and Padró [42] provided a complete characterization of the ideal access structures with an intersection number equal to one, i.e., structures in which at most one participant is in the intersection of any two different minimal authorized subsets. Notice that [42] characterizes

Table 2: Robustness for different values of k and ω, where $\omega < k$. A question mark symbol "?" indicates that the respective setting is an open problem.

	$\omega = 1$	$\omega = 2$	$\omega = 3$	$\omega = 3k/4$	$\omega = k - 1$
$k = n$	1	1	1	1	1
$k = n/3$	?	?	?	?	?
$k = n^c, c < 1/8$	n^{1-c}	?	?	$n - k^4 n^{1/2}$	$n - k^4 n^{\frac{2}{n^c}}$
$k = (\log n)^d$	n/k	?	?	$n - k^4 n^{1/2}$	$n - k^4 n^{\frac{2}{k}}$
$k = 4$	$n/4$	$n - n^{0.973}$	$n(1 - o(1))$		
$k = 3$	$n/3$	$n - \frac{n}{(\log n)^{1.01}}$			
$k = 2$	$n/2$				

which $(n, k, 2)$-TAS admit an ideal secret sharing. Therefore, an $(n, k, 2)$-TAS is ideal if and only if each of its connected components is a complete bipartite graph, a star, the Fano plane access structure, or some specific small graphs.[6] Here, an access structure is a *star* if a party is contained in every authorized set. Using this characterization [42] and decomposition techniques [9, 55], we construct secret sharing on $(n, k, 2)$-TAS:

1) Decompose the (n, k, ω)-TAS into n stars, one for each party. Construct an ideal secret sharing for each star.
2) Applying the well-known decomposition techniques [9, 55] yields a linear secret sharing with information ratio n.

Instantiating the above construction for the TAS with high robustness yields a secret sharing for $(n, k, 2)$-TAS with asymptotically optimal robustness. We finally apply the generic compiler [4] that transforms any secret sharing to the verifiable one to obtain the verifiable secret sharing.

Overall, TAS is a mathematically interesting object, and interesting connections can be discovered in future work.

3 Definitions and model

This section introduces the core definitions. We add additional preliminaries including commitments (with committing function $\mathsf{Comm}(\cdot)$) and zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs) in Appendix A.

3.1 Secret sharing definitions

TAS. We consider monotone access structures: if a set is in an access structure, then its supersets are also in the access structure. We also consider k-homogeneous access structures where their minimal sets are of size k $(2 \leqslant k \leqslant n)$.

Definition 1 (Access structure). *Given a set $[n] = \{1, \ldots, n\}$ of n parties, an access structure $\mathcal{A}$ on $[n]$ is a collection of subsets of $[n]$, $\mathcal{A} \subseteq 2^{[n]}$.*

We address a set in $\mathcal{A}$ as an *access group* or an *authorized set*. We now define two key traits of the access structure of our interest, **robustness** and **trackability**.

Definition 2 (Robustness). *The robustness of an access structure $\mathcal{A}$, denoted as $r(\mathcal{A})$, is the minimum number of parties that need to be corrupted to make reconstruction impossible.*

Definition 3 (ω-trackability). *An access structure is ω-trackable if given any ω parties, there exists either none or a unique minimal set containing them.*

$\mathcal{F}_{\text{NI-VSS}}^{t,\mathcal{A}}$

The simulator $\mathcal{S}$ specifies a set $\mathcal{C} \subset [n]$ of t maliciously corrupted parties. $\mathcal{S}$ controls $\mathcal{C}$ by determining their inputs to and observing their received messages from $\mathcal{F}_{\text{NI-VSS}}^{t,\mathcal{A}}$. The simulator $\mathcal{S}$ controls the dealer in the same way if $\mathcal{S}$ corrupts it.

Dealing. (1) Upon receiving $(\mathsf{Deal}, s^\star)$ from the dealer, and it is not corrupted by $\mathcal{S}$:

(i) Compute shares $\{s_i\}_{i\in[n]}$ and commitments of $s^\star$ and $\{s_i\}_{i\in[n]}$, denoted as cmt.

(2) Upon receiving $(\mathsf{Deal}, s^\star)$ from $\mathcal{S}$, and the dealer is corrupted by $\mathcal{S}$:

(i) Wait for $\{s_i\}_{i\in[n]}$ from $\mathcal{S}$. If the shares violate $\mathcal{A}$, ignore the message.[a] Compute commitments cmt of the secret and each share.

After step (i):

(ii) Send cmt to all. For $i \in [n]\backslash\mathcal{C}$, send s_i to i. If the dealer is not corrupted, additionally send $\{s_j\}_{j\in\mathcal{C}}$ to $\mathcal{S}$.
(iii) Store $\mathsf{tab}(i) = (s^\star, s_i)$ for all $i \in [n]$.

Reconstruction. Upon receiving $(\mathsf{Reconstruct})$ from any party, send $(\mathsf{Reconstruct})$ to $\mathcal{S}$. If $\mathcal{S}$ responds with $(\mathsf{Reconstruct})$, send $(\mathsf{Reconstruct})$ to all parties and collect responses $\tilde{s}_i \in \{s_i, \mathsf{Silent}\}$ from each $i \in [n]\backslash\mathcal{C}$ and $\tilde{s}_j \in \{s_j, \perp, \mathsf{Silent}\}$ from $j \in \mathcal{C}$. For all $i \in [n]$, if $\mathsf{tab}(i) = (s^\star, \tilde{s}_i)$, add i to set I. Wait until $I \in \mathcal{A}$, send $s^\star$ to all.

[a] The shares do not comply with $\mathcal{A}$ if any set of shares corresponding to an access group in $\mathcal{A}$ does not allow the reconstruction of a unique secret, e.g., the degree of the polynomial is $\geqslant t$ in a (t, n)-threshold secret sharing.

Fig. 4: Ideal functionality for NI-VSS, parameterized by t and access structure $\mathcal{A}$. This functionality is adapted from [16, 33].

.

We later provide equivalent definitions in Section 6. For a k-homogeneous access structure defined on party set $[n]$ and with ω-trackability, we denote it as (n, k, ω)-TAS. In our context, a meaningful ω satisfies $\omega \leqslant k-1$.

Ideal functionality for non-interactive VSS $\mathcal{F}_{\text{NI-VSS}}$. The NI-VSS ideal functionality interacts with the dealer, an ideal adversary or the simulator $\mathcal{S}$, and n parties. The functionality is parameterized by an access structure $\mathcal{A}$ on party set $[n]$ and with the minimum size of its minimal sets being k and t ($t \leqslant k-1$). $\mathcal{S}$ can corrupt a set of t parties. $\mathcal{S}$ can also corrupt the dealer. Note that the access structure can be but does not have to be k-homogeneous or trackable.

Definition 4 (Non-interactive VSS, adapted from [16, 33]). *A protocol $\Pi_{NI\text{-}VSS}$ is a secure NI-VSS if it securely achieves the NI-VSS ideal functionality defined in Fig. 4 assuming a public-key infrastructure.*

3.2 Game theory definitions

Game representation. In game theory, a normal-form game can be characterized by parties and their action space and utility functions. In our setting, let $N_m \subseteq [n]$ be the set of malicious parties where $|N_m| \leqslant k-1$, $N_r = [n] - N_m$ be the set of rational parties ($|N_r| \geqslant (n-k+1)$), $\times_{i\in N_r} A_i$ be the rational parties' joint action space with A_i being party i's individual action space, and $\{u_i\}_{i\in N_r}$ be their utility functions. We can then represent the game as a tuple $(N_r, \times_{i\in N_r} A_i, \{u_i\}_{i\in N_r})$. Considering that parties' interactions are sequential in our proposed mechanisms, we additionally need to capture parties' knowledge of past actions and beliefs of others' future actions when describing the game. In our design, the mechanisms are public knowledge, and parties act openly. This means that parties have complete information about the game structure and perfect information about historical moves.

[6] Connected components of an (n, k, ω)-TAS are the connected components of the hypergraph corresponding to it.

Solution concepts. A *strategy* s_i of party i is a probability distribution $\mathcal{D}$ over action space A_i, with all mass concentrated at one action for a pure strategy. We denote i's strategy space as $\mathcal{D}_i$. We denote malicious parties' joint strategy space as $\mathcal{D}_M$ (which is unknown). The utility function of i can then be described more explicitly as a function mapping all parties' strategies to real-valued utilities, $u_i : (\times_{j \in N_r} \mathcal{D}_j) \times \mathcal{D}_M \mapsto \mathbb{R}$. A *strategy profile* then records the strategies of all rational parties. Strategy profiles yielding certain desired properties are called *equilibria* or *solution concepts*. For example, in a Nash equilibrium (NE), no party can increase its utility by unilaterally deviating from the equilibrium strategy.

The solution concept we adopt is *Subgame Perfect Equilibrium* (SPE [52]), where the equilibrium strategy profile specifies the NE strategies for each party for every subgame. Here, a subgame at a step (where any party needs to make a move) is the continuation of the complete game from this step. SPE is a refinement of NE as NE is susceptible to empty (non-credible) threats.

Definition 5 (SPE [52]). *A Nash equilibrium (NE) is a strategy profile* $\mathbf{s}$ *where no party increases utility by unilaterally deviating from* $\mathbf{s}$*. A subgame perfect equilibrium (SPE) is a strategy profile* $\mathbf{s}$ *that forms an NE for any subgame.*

Fairness hazard. Given an equilibrium strategy profile $\mathbf{s}^*$, we now formally define "fairness". Denote the honest non-colluding strategy played by party i as s_i^h, the strategy profile of all other rational parties in equilibrium as $\mathbf{s}^*_{-i}$, and the arbitrary strategy profile of malicious parties as $\mathbf{s}_M$.

Definition 6 (Fairness). *For all* $i \in N_r, \forall s_i \in \mathcal{D}_i, \forall \mathbf{s}_M \in \mathcal{D}_M$, $u_i(s_i^h, \mathbf{s}^*_{-i}, \mathbf{s}_M) \geqslant u_i(s_i, \mathbf{s}^*_{-i}, \mathbf{s}_M)$.

This means that regardless of malicious parties' actions, the honest non-colluding strategy is the dominant strategy for rational parties in equilibrium. A collusion deterrence mechanism is fair if it ensures fairness for every non-colluding rational party in equilibrium. If a mechanism does not ensure fairness given $(k-1)$ malicious parties, we capture the risk of falsely implicating non-colluding parties with the *fairness hazard* notion.

Definition 7 (Fairness hazard). *Given* $(k-1)$ *malicious parties, fairness hazard counts the number of parties among* $(n-k+1)$ *rational parties whose utilities can be strictly increased by not playing the honest strategy.*

3.3 Model

System. A dealer shares private input $s^\star$ among a group of n parties with a secure VSS on access structure $\mathcal{A}$. The share-holding parties do not have private inputs and are only allowed to reconstruct after certain conditions are satisfied. This is controlled by an external function, and we focus on ensuring secrecy before the conditions are met. These parties can hold *arbitrary* prior knowledge about the secret input and can communicate over *any unobserved channels*. We aim to discourage parties from learning non-trivial information about the secret $s^\star$ for a privacy protection window $\Delta^\star$. We discuss setting $\Delta^\star$ in Section 4.3.

Assumptions. We assume up to $(k-1)$ of the n parties are malicious, and the malicious adversary corrupts parties *adaptively*. The remaining parties are rational. We assume rational parties are initially incentivized to participate in the VSS application. We also assume that they have quasi-linear utilities and that there are no unknown and unbounded externalities from unmodeled third parties, e.g., the government. When they collude, we assume they know which of *their inputs* are being used in the reconstruction.

4 The first collusion-deterrence mechanism

We first look at mechanism $(\mathbf{W}_0, \mathbf{P})$ in a single run of $\Pi_{\text{NI-VSS}}$. Recall that a mechanism is simply a public algorithm consisting of a winner selection rule and a payment rule. We start with an **honest host** that implements the mechanism and has access to private authenticated channels. We later employ a secure distributed system (without private channels) as the host in Section 4.2. We then extend the analysis to repeated runs of VSS in Section 4.3. We discuss the second mechanism providing the optimal fairness in Section 5.

4.1 Single-shot collusion deterrence with an honest host

Order of events. Since we allow colluding parties to use any collusion protocol, their action space can then be abstracted as follows: collude (C), not collude ($\bar{\mathsf{C}}$), report their knowledge (R), and not report ($\bar{\mathsf{R}}$). Consider the following secret-shared secret reconstruction game $\mathcal{G}$ with a host:

- (Stage 1) Rational parties decide whether to collude to learn about a secret (C) or not ($\bar{\mathsf{C}}$).
- (Stage 2) Rational parties decide whether to submit a report to the host about the learned information (R) or not ($\bar{\mathsf{R}}$).

Our goal is to make $\bar{\mathsf{C}}$ the equilibrium strategy for rational parties. We first assume an honest host H_h.

The collusion-deterrent NI-VSS protocol. As a first attempt, consider the following intuitive protocol. The dealer shares the secret among the n parties with $\Pi_{\text{NI-VSS}}$ and sends the secret to the honest host H_h. H_h then accepts reports of non-trivial knowledge about the secret through private authenticated channels. When there exist correct reports, H_h executes the mechanism. The first party that submits a correct report is picked as the winner and rewarded an amount λ_r. Others are all marked as colluders and penalized an amount λ_p.

There are three issues with the simple protocol. First, the parties can submit many guesses of non-trivial information about the secret even if collusion does not happen, treating the mechanism as an oracle for answering queries about the secret. Second, if the secret is not a random string, the parties may already have some non-trivial private knowledge about it (e.g., because they know the dealer), making framing others possible. Third, the dealer knows the secret and can collude with a share-holding party to frame others.

For the first issue, we penalize the sender of each wrong report the amount λ_p. Further, to discourage a random guess, we introduce the *non-triviality parameter* $\gamma \in [0, 1)$, which we provide details in Section 7. Roughly, if the information specified by the reporter can be guessed correctly in one shot with probability $\geqslant \gamma$, then we consider the information gain to be trivial and labels the report as wrong. Let $V > 0$ be the upper bound of the worth of secrets that the system sets out to protect. We then let λ_p satisfy the following

$$\gamma(V + \lambda_r) < (1 - \gamma)\lambda_p \tag{1}$$

where the left-hand side is what the party expects to receive from a correctly guessed report, and the right-hand side is what the party loses in expectation.

For the second issue, we ask the dealer to generate q random string(s) ($q \geqslant 1$) and generate shares for the n parties for both the actual secret and the random string. The shares are permuted uniformly at random and then sent to the recipients. We then discourage the informed fake reports by asking a reporter to specify the corresponding inputs used in learning the non-trivial information and ensure that they expect a reduced utility by submitting fake reports:

$$(\frac{1}{q+1} + \frac{q}{q+1}\gamma)\lambda_r < \frac{q}{q+1}(1 - \gamma)\lambda_p \Rightarrow (1 + q\gamma)\lambda_r < q(1 - \gamma)\lambda_p \tag{2}$$

For the last problem, one way is to charge a *proper* service fee λ_s from the dealer so that she expects to pay more than what she expects to earn from fake reports:

$$(n - 1)\lambda_s > \lambda_r \tag{3}$$

An alternative is to employ MPC in the sharing phase so that the dealer does not learn the shares or the permutation similar to [26].

Finally, we summarize the single-shot collusion-deterrent NI-VSS protocol in Fig. 5. Our remaining task is to determine the parameters.

Parameterize q and payment amounts for the non-collusion outcome. If the secret is a random string, e.g., a secret key, we can let $q = 0$. Otherwise, we can let $q = 1$. What remains is deciding the payment amounts λ_r and λ_p such that we achieve the non-collusion outcome.

Preprocessing
⓪ A dealer holds a secret $x^{(0)}$ with worth $\leqslant V$, and obtains $q \geqslant 0$ unbiased random secrets $x^{(1)}, \ldots, x^{(q)}$. The dealer permutes the secrets at random. Note that V is a system parameter.
Dealing
① The dealer secret-shares the $(q+1)$ permuted secrets $\tilde{x}^{(0)}, \ldots, \tilde{x}^{(q)}$ among the n parties via $\Pi_{\text{NI-VSS}}$.[a]
Mechanism setup
② The dealer pays the host H_h service fees $n\lambda_s$ and sends the permuted secrets and the parameterization of the mechanism $(\mathbf{W}_0, \mathbf{P})$ to H_h, including the secret protection time window $\Delta^\star$, triviality parameter γ, collusion penalty amount λ_p and the winner bonus amount λ_r. The service fees $(n\lambda_s)$ will eventually be distributed to each of the n parties that are not marked as colluders during the $\Delta^\star$ time window.
③ If the service fee amount is compatible with λ_r (Eq. (3)), and other parameters satisfy Eq. (1), (2) and Proposition 1, H_h accepts reports about non-trivial knowledge about any secret from all parties until $\Delta^\star$ time has passed. Otherwise, H_h aborts.
Mechanism implementation
④ Let there be m reporters ($m \geqslant 1$). H_h records the m senders and committed reports in a public FIFO queue,

$$\langle (p_1, \mathsf{Comm}(g_1), f_1, \tilde{x}_{p_1}^{(i_1)}), \ldots, (p_m, f_m, \mathsf{Comm}(g_m), \tilde{x}_{p_m}^{(i_m)}) \rangle$$

Here, $\tilde{x}_{p_j}^{(i_j)}$ $(j \in [m], p_j \in [n], i_j \in \{0, \ldots, q\})$ is the share that the j-th reporter p_j received for secret $\tilde{x}^{(i_j)}$, and $\mathsf{Comm}(g_j)$ is the commitment of the the non-trivial information gain g_j, which is the function f_j evaluated at secret $\tilde{x}^{(i_j)}$.
⑤ Parties privately reveal their reports to H_h. For each de-committed report, H_h checks if the function is γ-non-trivial (described in Section 7) and the correctness of the reported value. If verification passes, H_h considers i for winner selection.
⑥ H_h executes $\mathbf{W}_0$: Pick the first party with the correct report as the *winner* and mark all other parties as *colluders*.
H_h executes $\mathbf{P}$: The winner receives reward λ_r. Each colluder is fined with penalty λ_p. Each reporter submitting an incorrect report is penalized λ_p.

[a] Because of the q random strings, we let the sample space of the secret satisfy $\gg q+1$.

Fig. 5: $\Pi^H_{\text{NI-VSS}}$: Single-shot collusion-deterrent NI-VSS with an honest host H_h. $\Delta^\star$ is set by the dealer according to the needs of the application.

Under the winner selection rule $\mathbf{W}_0$, there is at most one winner. When there are up to $(k-2)$ malicious parties, there are at least 2 rational parties in any access group in the access structure. Then for rational colluding parties in any access group, at least one of them becomes the winner with probability $\leqslant 1/2$. Then we only need to ensure that the relatively "slower" party (who always exists) is disincentivized from collusion.

Now we are ready to state the equilibria of the game $\mathcal{G}$.

Proposition 1. *Consider the secret reconstruction game $\mathcal{G}$ with an honest host in protocol $\Pi^H_{NI\text{-}VSS}$ and a secret of worth at most V. Given $(k-2)$ malicious parties, the SPE under mechanism $(\mathbf{W}_0, \mathbf{P})$ is each rational party playing $\bar{\mathsf{C}}$ in Stage 1 (i.e., the non-collusion outcome) if $\lambda_p > 0$ and $\frac{1}{2}(\lambda_p + \lambda_s - \lambda_r) > V$. $(\mathbf{W}_0, \mathbf{P})$ is fair in the same setting. Its fairness hazard is $(n-k)$.*

Its proof utilizes backward induction. Intuitively, given $(k-2)$ malicious parties, any access group has at least 2 rational parties. If collusion has happened, the rational parties are incentivized to report collusion to escape the penalty λ_p: in any other strategy profile, one can improve its utility by changing their actions to R (if not already). Reasoning backward, the slower parties who cannot become the winner with a probability higher than $\frac{1}{2}$ are disincentivized from collusion. Note that we make black-box use of the NI-VSS protocol $\Pi_{\text{NI-VSS}}$ so we only need to prove the non-collusion outcome.

Proof. We first solve for the SPE of the sequential game under mechanism $(\mathbf{W}_0, \mathbf{P})$ with backward induction. We know that under $\mathbf{W}_0$, only a single winner is selected, and all others are marked as colluders. Consider

the worst case where an authorized set has $(k-2)$ malicious parties. In Table 3, we demonstrate the payoffs of the remaining two rational parties playing the game. If they do not collude in the first stage, the game terminates, and they both receive the service fees. If they collude, in the second stage, the NE is for both parties to report since $\lambda_p > 0$. This means at least one of the two parties receives $\leqslant V + \frac{1}{2}(\lambda_r + \lambda_s) - \frac{1}{2}\lambda_p < \lambda_s$. Then in the first stage, the dominant strategy for this party is to not collude.

Table 3: Payoffs of two parties playing $\mathcal{G}$. $p_1 \in [0,1]$ is the probability of the column party submitting the report first.

	$\bar{\mathsf{C}}$	$\mathsf{C}\ \bar{\mathsf{R}}$	$\mathsf{C}\ \mathsf{R}$
$\bar{\mathsf{C}}$	(λ_s, λ_s)	(λ_s, λ_s)	(λ_s, λ_s)
$\mathsf{C}\ \bar{\mathsf{R}}$	(λ_s, λ_s)	$(V + \lambda_s, V + \lambda_s)$	$(V - \lambda_p, V + \lambda_r + \lambda_s)$
$\mathsf{C}\ \mathsf{R}$	(λ_s, λ_s)	$(V + \lambda_r + \lambda_s, V - \lambda_p)$	$(V + p_1(\lambda_r + \lambda_s) - (1-p_1)\lambda_p,$ $V + (1-p_1)(\lambda_r + \lambda_s) - p_1\lambda_p)$

More generally, when there are $(k-u)$ $(u > 2)$ malicious parties in an authorized set, u rational parties play the reconstruction game. At least one rational party becomes the winner with probability $\leqslant 1/u$, which is $< 1/2$. In the second stage, the NE is still playing R due to $\lambda_p > 0$. Then at least one of the u rational parties receives negative utility. As such, this slowest party is not incentivized to participate in collusion and plays $\bar{\mathsf{C}}$ in Stage 1.

In summary, when up to $(k-2)$ parties are malicious, at least one party does not collude in equilibrium. As a result, no party learns the secret via collusion in equilibrium. The mechanism thus achieves fairness in this setting.

However, when there are $(k-1)$ malicious parties, they can appear in the same access group due to adaptive corruption. The remaining rational party in the access group can improve its utility by colluding and earning the privacy worth V and report to earn the reward λ_r. In this case, up to $(n-k)$ non-colluding parties are wrongly penalized. They can increase their utility by deviating from the non-colluding strategy and colluding to earn the privacy worth V. Therefore, the mechanism has fairness hazard of $(n-k)$.

4.2 Distributed host and privacy-preserving report verification

Order of events. We now replace the honest host in $\Pi^{H}_{\text{NI-VSS}}$ (Fig. 5) with a secure distributed system and update details concerning report verification. The updated protocol $\Pi^{\mathcal{V},\mathbf{W}}_{\text{NI-VSS}}$ is summarized in Fig. 6. Specifically, the report verifier is an algorithm run by parties in the distributed system. We denote this public verifier as $\mathcal{V}$. All messages to and from $\mathcal{V}$ are *observed by all.*

In step ⑤, the potentially implicated parties prove their innocence with zkSNARKs. For $k = \max\{\omega, 1\} + 1$, only one party needs to generate the proof, and Groth16 [28] or Plonk [22] can be employed. Otherwise, collaborative zkSNARKs [45] can be utilized for proof generation. If there are non-responding parties during the proof generation, one can resort to publicly auditable MPC for generating the proof. The silent parties are marked as colluders.

We can now remove the trusted host and re-state the result.

Theorem 1. *Consider the secret reconstruction game $\mathcal{G}$ with a distributed host in protocol $\Pi^{\mathcal{V},\mathbf{W}_0}_{NI\text{-}VSS}$ and a secret of worth at most V. Given $(k-2)$ malicious parties, the SPE under mechanism $(\mathbf{W}_0, \mathbf{P})$ is each rational party playing $\bar{\mathsf{C}}$ in Stage 1 if $\lambda_p > 0$ and $\frac{1}{2}(\lambda_p + \lambda_s - \lambda_r) > V$. $(\mathbf{W}_0, \mathbf{P})$ is fair in the same setting. Its fairness hazard is $(n-k)$.*

Online dealer If the dealer is always online, we can adopt an alternative approach for report verification in step ⑤: the dealer can directly generate equality or inequality proofs for submitted reports.

Dealing
① The dealer generate shares for the $(q+1)$ permuted secrets $\tilde{x}^{(0)}, \ldots, \tilde{x}^{(q)}$ for the n parties via $\Pi_{\text{NI-VSS}}$. The dealer sends cmt to $\mathcal{V}$, including the commitments of the shares which we denote as

$$\langle \mathsf{Comm}(\tilde{x}_1^{(0)}), \ldots, \mathsf{Comm}(\tilde{x}_n^{(q)}) \rangle$$

The dealer then sends de-commit information to the corresponding servers.
Mechanism setup
② The dealer pays $\mathcal{V}$ service fees $n\lambda_s$ and sends the parameterization of the mechanism $(\mathbf{W}, \mathbf{P})$ to $\mathcal{V}$, including the secret protection time window $\Delta^\star$, triviality parameter γ, collusion penalty amount λ_p and the winner reward amount λ_r.
③ If the service fee amount is compatible with λ_r (Eq. (3)), and other parameters satisfy Eq. (1), (2) and Theorem 1, $\mathcal{V}$ accepts reports about non-trivial knowledge about any secret from all parties until $\Delta^\star$ time has passed. Otherwise, $\mathcal{V}$ aborts.
Mechanism implementation
④ Let there be m reporters ($m \geqslant 1$). $\mathcal{V}$ records the m senders and committed reports in a public FIFO queue,

$$\langle (p_1, \mathsf{Comm}(g_1), f_1, (\tilde{x}_{p_1}^{(i_1)}, \tilde{r}_{p_1}^{(i_1)})), \ldots, (p_m, f_m, \mathsf{Comm}(g_m), (\tilde{x}_{p_m}^{(i_m)}, \tilde{r}_{p_m}^{(i_m)})) \rangle$$

Here, $(\tilde{x}_{p_j}^{(i_j)}, \tilde{r}_{p_j}^{(i_j)})$ $(j \in [m], p_j \in N, i_j \in \{0, \ldots, t\})$ is the de-commit information for the commitment of share $\tilde{x}_{p_j}^{(i_j)}$ with $\tilde{r}_{p_1}^{(i_1)}$ being the randomness used in committing $\tilde{x}_{p_j}^{(i_j)}$. The rest are the same as Fig. 5.
⑤ If the de-commit information in a report is incorrect, $\mathcal{V}$ directly marks the sender as *colluder*. For each revealed report $(p_j, g_j, f_j, (\tilde{x}_{p_j}^{(i_j)}, \tilde{r}_{p_j}^{(i_j)}))$ with correctly de-committed share $\tilde{x}_{p_j}^{(i_j)}$, $\mathcal{V}$ waits for an evidence collection time window of Δ. During the period, any $(k-1)$ parties in any authorized group with p_j can submit a zero-knowledge proof π that proves the following: Either the function f_j is trivial (described in Section 7), or $f_j(\tilde{x}^{(j)}) \neq g_j$ and
- Their inputs are correct with respect to the commitment of shares.
- Function f_j is being evaluated at the reconstructed secret.

If no valid proof is provided in time, $\mathcal{V}$ considers p_j as a candidate for winner selection. Otherwise, $\mathcal{V}$ marks p_j as *colluder*.
⑥ $\mathcal{V}$ executes mechanism $(\mathbf{W}, \mathbf{P})$.

Fig. 6: $\Pi_{\text{NI-VSS}}^{\mathcal{V},\mathbf{W}}$: Single-shot collusion deterrence with distributed verifier $\mathcal{V}$. We omit the pre-processing routines that are the same as $\Pi_{\text{NI-VSS}}^{H}$ in Fig. 5. The winner selection rule $\mathbf{W}$ can be substituted with $\mathbf{W}_0$ or $\mathbf{W}_1$.

Setting proof collection window Δ When setting Δ, we take into account the offline time of participants δ_1, network delays δ_2 that can be induced by potential distributed denial-of-service (DDoS) attacks, and proof generation time δ_3. Specifically, we set $\Delta = \max\{\delta_1, \delta_2\} + \delta_3$. δ_1 depends on the application scenario. For typical blockchain applications, we can give a conservative estimate of multiple weeks. For δ_2, based on the DDoS attack report in 2024 Q2 released by Cloudflare [29], less than 1% of the network-layer DDoS attacks last over 3 hours. We can set it conservatively to multiple weeks as well. For δ_3, the relatively more expensive collaborative zkSNARKs [45] only takes hundreds of microseconds per constraint. Overall, we can set Δ to be multiple weeks.

4.3 Collusion deterrence in repeated VSS runs

We next discuss the reconstruction game with d secrets ($d > 1$) where each secret is of individual worth $\leqslant V$.
Order of events. We denote this d-secret game as $\mathcal{G}_d$, consisting of d instances of the original reconstruction game $\mathcal{G}$. We summarize the repeated collusion-deterrent NI-VSS in Fig. 7, which is slightly updated from Fig. 6. Overall, repetition essentially only changes the parameterization of the mechanism.

Given a finite and known d, we achieve the same results as in Theorem 1 as we can still apply backward induction from the end game. However, given an infinite or unknown d, as observed in prior works [25], there

Preprocessing
⓪ Each of the d dealers of each secret $x^{(0,i)}$ ($i \in [d]$) obtains $q_i \geqslant 0$ random secrets $x^{(1,i)}, \ldots, x^{(q_i,i)}$ uniformly at random. Each dealer permutes their $(q_i + 1)$ secrets at random.
Dealing
① Each dealer i generate shares for the $(q_i + 1)$ permuted secrets $\tilde{x}^{(0,i)}, \ldots, \tilde{x}^{(q_i,i)}$ for the n parties via $\Pi_{\text{NI-VSS}}$. The dealer sends cmt to $\mathcal{V}$, including the commitments of the shares

$$\langle \mathsf{Comm}(\tilde{x}_1^{(0,1)}), \ldots, \mathsf{Comm}(\tilde{x}_n^{(q_d,d)}) \rangle$$

The dealer then sends de-commit information to the corresponding servers.
③ If the service fee amount is compatible with λ_r (Eq. (3)), and other parameters satisfy Eq. (1), (2) and Corollary 1, $\mathcal{V}$ accepts reports about non-trivial knowledge about any secret from all parties until $\Delta^\star$ time has passed. Otherwise, $\mathcal{V}$ aborts.
Routines ② and ④-⑥ are the same as Fig. 6.

Fig. 7: Repeated collusion deterrence with a distributed verifier $\mathcal{V}$.

is no end game, and the outcome depends on how patient the parties are. Let $\delta \in [0, 1]$ be the discount factor for how the most patient party among the n parties discounts future returns. Higher δ means that they are more patient and value future returns closer to current returns. We then have the following corollary.

Corollary 1. *Consider the d-secret reconstruction game $\mathcal{G}_d$ with a distributed host in protocol $\Pi^{\mathcal{V},\mathbf{W}_0}_{NI\text{-}VSS}$, where each secret is of worth at most V, and the most patient party has discount factor $\delta \in [0, 1]$. If d is known and finite, Theorem 1 holds. Otherwise, given $(k - 2)$ malicious parties, the SPE under mechanism $(\mathbf{W}_0, \mathbf{P})$ is each rational party playing $\bar{\mathsf{C}}$ in Stage 1 if $\lambda_p > 0$, $\frac{1}{2}(\lambda_p + \lambda_s - \lambda_r) > V$ and Eq. (4) holds.*

$$\frac{\delta}{1-\delta} V < \frac{1}{2}(\lambda_r + \lambda_s - \lambda_p) - \frac{\lambda_s}{1-\delta} \tag{4}$$

$(\mathbf{W}_0, \mathbf{P})$ is fair in the same setting. Its fairness hazard is $(n - k)$.

Intuitively, we only need to make R appealing in Stage 2 by letting repeatedly colluding with each other without reporting undesirable compared with receiving reward λ_r from reporting. Then reasoning backwards, the slower parties are discouraged from colluding in Stage 1.

Proof (sketch). Consider the alternative strategy of always colluding and never reporting, and when someone reports collusion, its colluding partners never collude with this party again. This strategy $(\mathsf{C}\ \bar{\mathsf{R}})$ gives returns

$$(V + \lambda_s) \sum_{i=0}^{\infty} \delta^i = \frac{V + \lambda_s}{1 - \delta}$$

The alternative (our desired) strategy of $(\mathsf{C}\ \mathsf{R})$ yields returns at least

$$p(V + \lambda_r + \lambda_s) + (1 - p)(V - \lambda_p) = V + p(\lambda_r + \lambda_s) - (1 - p)\lambda_p$$

where p is the probability of a party becoming the winner. For the slowest rational party, $p \leqslant 1/2$. Letting the second quantity be greater than the previous quantity gives us Eq. (4). This means that rational parties are discouraged from playing strategy $(\mathsf{C}\ \bar{\mathsf{R}})$. The rest then follows from Theorem 1.

Setting privacy protection window $\Delta^\star$ for repeated games In known finite runs of VSS, the dealers determine $\Delta^\star$ according to the needs of the applications. Otherwise, if one implements the penalty by having each party make a deposit in the beginning and deprive a party of λ_p of its deposit if it is marked as a colluder, the deposit can be set according to the self-insurance in [25]. Overall, it needs to be large enough to account for possibly frequent collusion attempts in repeated VSS runs and in the meantime, be affordable to share-holding parties.

5 The second mechanism

We now introduce the second mechanism $(\mathbf{W}_1, \mathbf{P})$ utilizing trackability of access structures. We start with a single run of VSS with a distributed host and then extend the discussion to repeated VSS runs.

5.1 Impossibility

We first define the untrackability of access structures and establish that given untrackable access structures, one cannot achieve a zero fairness hazard while ensuring non-trivial effectiveness.

Definition 8 (Untrackability). *An access structure with the smallest minimal set size being k is untrackable if there exist $(k-1)$ parties so that any subset of them co-exist in at least two minimal sets that are non-overlapping except for at the subset itself.*

An example of an untrackable access structure is the threshold access structure $\mathcal{A}^{(k)} = \{A \subseteq [n] : |A| \geqslant k\}$ where parties can substitute each other. We now state the impossibility result.

Proposition 2. *For VSS defined on an untrackable access structure, there does not exist an effective collusion deterrence mechanism with 0 fairness hazard in the current model.*

Proof. Let $\mathcal{A}'$ be an untrackable access structure. Suppose for contradiction that we have a collusion deterrence mechanism with 0 fairness hazard for VSS defined on $\mathcal{A}'$. By untrackability definition, there exists a group of $(k-1)$ parties that belong to at least two access groups in $\mathcal{A}'$. Denote this set of $(k-1)$ parties as D and the two authorized groups that they are in as B, C. Let p_B, p_C be the two distinct parties of B, C, i.e., $B = D \cup p_B$ and $C = D \cup p_C$. Consider the following scenarios where p_B, p_C have a network disadvantage in submitting reports:

- World 1 Only parties in B collude. Because the mechanism has fairness hazard 0, p_C is never marked as a colluder.
- World 2 Only parties in C collude. Because the mechanism has fairness hazard 0, p_B is never marked as a colluder.

The mechanism cannot distinguish between World 1 and 2 when p_B, p_C do not submit reports. Then p_B, p_C cannot be penalized by the mechanism by staying silent. Then by the effectiveness of the mechanism, at least one of the colluding parties in D must suffer from loss imposed by the mechanism to be discouraged from collusion. Then this party is disincentivized from submitting reports. This is similar to the reason that p_B and p_C do not report, i.e., silence allows it to take advantage of the untrackability of the access structure. We can apply this reasoning until there is no reporters, which contradicts the effectiveness of the mechanism.

5.2 Optimally fair collusion deterrence with TAS

Optimally fair single-shot collusion deterrence. We adopt the same protocol in Fig. 6 but with two slight changes: First, the winner selection rule in the mechanism is now parameterized with $\mathbf{W}_1$, i.e., $\Pi^{\mathcal{V},\mathbf{W}_1}_{\text{NI-VSS}}$; second, the underlying NI-VSS protocol $\Pi_{\text{NI-VSS}}$ is now constructed on (n, k, ω)-TAS.

Recall that $\mathbf{W}_1$ (presented in Section 2.1) states three rules. Let there be m reporters. Rule 1.A applies when $m \geqslant k$, and it marks the last free rider (i.e., a party that does not make up a complete access group with any other $(k-1)$ reporters) as the colluder. Rule 1.B applies when $\omega < m < k$ or when there is no free rider, and it marks the last reporter as the colluder. Rule 2 applies when there are exactly ω reporters. It first locates the minimal set that contains these parties and marks the remaining members in the access group as colluders.

We now formally state the following theorem.

Theorem 2. *Consider the secret reconstruction game $\mathcal{G}$ with a distributed host in protocol $\Pi^{\mathcal{V},\mathbf{W}_1}_{NI\text{-}VSS}$ constructed on an (n, k, ω)-TAS and a secret of worth at most V. Given up to $(k-1-\omega)$ malicious parties, the SPE under mechanism $(\mathbf{W}_1, \mathbf{P})$ is each rational party playing $\bar{\mathsf{C}}$ in Stage 1 if $\lambda_p > 0$ and $\frac{1}{\omega+1}(\omega\lambda_p + \omega\lambda_s - \lambda_r) > V$. $(\mathbf{W}_1, \mathbf{P})$ is fair if there are up to $(k-1)$ malicious parties, yielding a fairness hazard of 0.*

We provide a visual proof in Fig. 1. The formal proof for the effectiveness of the mechanism given up to $(k-1-\omega)$ malicious parties shares a similar rationale to the proof in Proposition 1. So we slightly simplify the first part of the proof.

Proof (of Theorem 2). We first solve for the SPE of the sequential game under mechanism $(\mathbf{W}_1, \mathbf{P})$ with backward induction. Consider any authorized set. When there are up to $(k-1-\omega)$ malicious parties, this authorized set has at least $(\omega+1)$ rational parties. *If no group colludes*, every rational party receives λ_s. *If one group colludes*, in the second stage, reporting is the NE because $\lambda_p > 0$. Note that at least one rational party does not win with a probability higher than $1/(\omega+1)$. Then in the first stage, this slowest party selects action $\bar{\mathsf{C}}$ because its expected returns from C is $\leqslant V + \frac{1}{\omega+1}(\lambda_r + \lambda_s) - \frac{\omega}{\omega+1}\lambda_p < \lambda_s$. *If more than one access group collude*, then in the second stage, we consider the following cases:

(a) All other colluding parties have reported. Then the remaining party picks $\bar{\mathsf{R}}$ since R results in penalty λ_p according to rule 1.B in $\mathbf{W}_1$.
(b) All but one of the other colluding parties have reported. Considering that the last party picks $\bar{\mathsf{R}}$ in case (a), then the other remaining party picks $\bar{\mathsf{R}}$ because R results in penalty λ_p according to rule 1.A in $\mathbf{W}_1$.
(c) All other groups have revealed their members. The parties in the remaining colluding group choose $\bar{\mathsf{R}}$ because R results in penalty λ_p after repeatedly applying the reasoning in case (b).
(d) One group has revealed its members. Any party in other colluding groups chooses $\bar{\mathsf{R}}$ because R results in penalty λ_p after repeatedly applying the reasoning in scenario (c).
(e) One group X has revealed $(k-1)$ of its members. Before any party i in other colluding groups report, the only remaining party $x \in X$ picks $\bar{\mathsf{R}}$ to avoid penalty by rule 1.B in $\mathbf{W}_1$. If any i takes action R, x would pick R. However, using backward induction, i picks $\bar{\mathsf{R}}$ to avoid the penalty by rule 1.A. As a result, x picks $\bar{\mathsf{R}}$.
(f) One group X has revealed ω of its members. Any party i not in X does not share a group with these ω parties by the definition of ω-trackability. The rest members in X and i choose $\bar{\mathsf{R}}$ by repeatedly applying the reasoning in scenario (e).
(g) One group X has revealed $u < \omega$ of its members. Any x in the same colluding group with the u reporters (not necessarily X) picks R because $\lambda_p > 0$. Consider any colluding party i that does not share a colluding group with the u reporters. i only picks R if along with $(\omega-1)$ parties in its collusion group, it can outrun the fastest remaining $(\omega-u)$ parties in any group containing the u existing reporters.

The implication of the case (g) is that in a group X with the fastest ω parties, these fast parties pick R in Stage 2. Other parties in other colluding groups choose action $\bar{\mathsf{R}}$. Then the slowest party in group X picks $\bar{\mathsf{C}}$ in Stage 1. Repeatedly applying backward induction, in each access group, the slowest parties are disincentivized to collude in Stage 1.

We next examine the fault tolerance of the mechanism for achieving fairness. Consider up to $(k-1)$ malicious parties. If they act rational when colluding with rational parties, then we achieve the non-collusion outcome and as a result, fairness. Otherwise, collusion takes place in access groups with at least $(k-\omega)$ malicious parties, and at most $(\omega-1)$ parties can co-exist in multiple groups. When $\omega = 1$, the malicious parties need to be in the same access group to cause successful collusion. The only remaining rational party picks R in Stage 2, which results in the malicious parties being marked as colluders. When $\omega = k-1$, the malicious parties can only possibly frame non-colluding parties by being in the same access group, leaving no additional party to facilitate collusion outside of this access group. In both cases, the fairness hazard is 0 regardless of the malicious parties' actions. When $\omega < \frac{k+2}{3}$ (i.e., $(k-1)-(k-\omega)+(\omega-1) < (k-\omega)$), there is at most one successful colluding group. When $\omega \geqslant \frac{k+2}{3}$, there are at least two successful colluding groups. In each of these groups, if there are less than ω rational parties, they do not report in Stage 2 as $\lambda_p > 0$, and $\mathbf{W}_1$ dismisses the reports if there are $< \omega$ total reports. For access groups with exactly ω rational parties, rational parties in the fastest such group pick R in Stage 2, become winners and receive λ_r. The reporters then implicate a single collusion group and only malicious parties are penalized. The mechanism still achieves fairness and has a fairness hazard of 0.

Optimally fair collusion deterrence in repeated VSS runs. Now we consider the d-secret reconstruction game in protocol $\Pi^{\mathcal{V},\mathbf{W}_1}_{\text{NI-VSS}}$ summarized in Fig. 7. We state the following result. The intuition is similar to the intuition behind Corollary 1.

Corollary 2. *Consider the d-secret reconstruction game $\mathcal{G}_d$ with a distributed host in protocol $\Pi^{\mathcal{V},\mathbf{W}_1}_{NI\text{-}VSS}$, where each secret is of worth at most V, and the most patient party has discount factor $\delta \in [0,1]$. If d is known and finite, Theorem 2 holds. Otherwise, given $(k-1-\omega)$ malicious parties, the SPE under mechanism $(\mathbf{W}_1, \mathbf{P})$ is each rational party playing $\bar{\mathsf{C}}$ in Stage 1 if $\lambda_p > 0$, $\frac{1}{\omega+1}(\omega\lambda_p + \omega\lambda_s - \lambda_r) > V$, and Eq. (5) holds. $(\mathbf{W}_1, \mathbf{P})$ is fair if there are up to $(k-1)$ malicious parties, yielding a fairness hazard of 0.*

$$\frac{\delta}{1-\delta}V < \frac{1}{\omega+1}(\lambda_r + \lambda_s - \omega\lambda_p) - \frac{\lambda_s}{1-\delta} \tag{5}$$

Proof (sketch). Similar to the proof of Corollary 1, we consider the alternative strategy of always colluding and never reporting. This strategy $(\mathsf{C}\ \bar{\mathsf{R}})$ gives returns

$$(V + \lambda_s)\sum_{i=0}^{\infty}\delta^i = \frac{V+\lambda_s}{1-\delta}$$

The alternative desired strategy of $(\mathsf{C}\ \mathsf{R})$ yields returns at least

$$V + p(\lambda_r + \lambda_s) - (1-p)\lambda_p$$

where p is the probability of a party becoming the winner. For the slowest rational party, $p \leqslant \frac{1}{\omega+1}$. Letting the second quantity be greater than the previous quantity gives us Eq. (5). This means that rational parties are discouraged from playing strategy $(\mathsf{C}\ \bar{\mathsf{R}})$. The rest follows from Theorem 2.

6 TAS and trackable secret sharing schemes

6.1 Section-specific preliminaries

An (n,k,ω)-*design* is an k-uniform hyper-graph with pairwise intersections of hyper-edges of size $< \omega$. The *independence number* α of a hypergraph is the maximum size of a set of vertices in the graph that contains no edges.
A *Steiner system* with parameters n, k, ω, denoted as $S(n,k,\omega)$, is an n-element set S together with a set of k-element subsets of S (called blocks) such that each ω-element subset of S is contained in exactly one block. The *partial Steiner system*, denoted $S_p(n,k,\omega)$ is obtained by relaxing the condition that each ω-subset is contained in a unique block to the condition that each ω-subset is contained in at most one block.

Example 1. The Fano plane (Fig. 1a) is a $S(7,3,2)$ Steiner system.

A secret sharing is *ideal* if the share size of every party equals to the secret size. An access structure is *ideal* if an ideal secret sharing realizes the access structure. An access structure is a *star* if a party is contained in every authorized set.

6.2 Bound on optimal TAS

This section presents an upper bound on the size of TAS. We begin with an equivalent formulation of TAS from a coding theory perspective. Recall that (n,k,ω)-TAS denotes a k-homogeneous ω-trackable access structure over $[n]$. Also recall that $\mathcal{A}^\star$ represents the minimal access structure of $\mathcal{A}$.

Lemma 1 (An equivalent definition of TAS). *An access structure $\mathcal{A}$ is (n,k,ω)-trackable if and only if $\mathcal{A}^\star$, represented as a subset of F_2^n, is a binary k-weight code with a Hamming distance at least $2k-2(\omega-1)$.*

Proof. Each element A in $\mathcal{A}$ is equivalently represented as an indicator vector of the set A - a codeword. For example, for $n = 4$, $A = \{1, 3\}$, and $B = \{1, 2, 4\}$, the codeword 1010 and 1101 are the equivalent representation of A and B, respectively. The lemma follows from the following simple observations. Every set of size k is represented as a code word in F_2^n with weight k. Furthermore, two sets of size k have an intersection at most $(\omega - 1)$ if and only if the two corresponding codewords in F_2^n have a distance at least $2k - 2(\omega - 1)$.

Theorem 3. *Let $\mathcal{A}$ be an (n, k, ω)-TAS. It holds that*

$$|\mathcal{A}^\star| \leqslant \binom{n}{\omega} \cdot \binom{k}{\omega}^{-1}$$

The equality happens if and only if a Steiner system $S(n, k, \omega)$ exists.

Proof. We shall employ the Johnson bound for binary code to derive the result. By Lemma 1, $\mathcal{A}^\star$ is a binary code with the following properties.

(1) Every codeword in $\mathcal{A}^\star$ has weight k.
(2) Distance between any two distinct codewords in $\mathcal{A}^\star$ is at least $2k - 2(\omega - 1)$.

Applying the well-known Johnson bound (Theorem 4 for the latter case) yields

$$\begin{aligned} |\mathcal{A}| \leqslant A(n, 2k - 2(\omega - 1), k) &\leqslant \left\lfloor \frac{n}{k} \left\lfloor \frac{(n-1)}{k-1} \cdots \left\lfloor \frac{(n - (\omega - 1))}{k - (\omega - 1)} \right\rfloor \cdots \right\rfloor \right\rfloor \\ &\leqslant \frac{n}{k} \cdot \frac{n-1}{k-1} \cdots \frac{n - (\omega - 1)}{k - (\omega - 1)} = \binom{n}{\omega} \cdot \binom{k}{\omega}^{-1}. \end{aligned}$$

The equality happens if and only if a Steiner $S(n, k, \omega)$ exists.

Johnson bound. Let $C(n, d, \omega)$ be the set of all binary codes with length n and minimum distance d. Let every codeword in $C(n, d, \omega)$ have weight ω. Let $A(n, d, \omega)$ be the largest size of a code in $C(n, d, \omega)$.

Theorem 4 (Johnson bound [31]). *Let $n, d, \omega \in \{1, 2, \dots, \}$ such that $d \leqslant n$ and $\omega \leqslant n$.*

1. *If $d > 2\omega$, then $A(n, d, \omega) = 1$.*
2. *Else ($d \leqslant 2\omega$), define*

$$a = \begin{cases} d/2, & \text{if } d \text{ is even} \\ (d+1)/2, & \text{otherwise.} \end{cases}$$

Then

$$A(n, d, \omega) \leqslant \left\lfloor \frac{n}{\omega} \left\lfloor \frac{(n-1)}{\omega - 1} \cdots \left\lfloor \frac{(n - \omega + a)}{a} \right\rfloor \cdots \right\rfloor \right\rfloor,$$

where $\lfloor \cdot \rfloor$ is the floor function. Furthermore, it holds that

$$A(n, 2\delta, \omega) \leqslant \frac{n}{k} \cdot \frac{n-1}{k-1} \cdots \frac{n - \omega + \delta}{\delta}$$

with equality if and only if a Steiner system $S(n, \omega, \omega - \delta + 1)$ exists.

6.3 Optimal TAS

This section presents results and constructions of optimal TAS. By Theorem 3, constructing such structure reduces to the constructions of binary constant-weight codes, particularly Steiner systems for some parameter regime. Constructing the maximum size of constant weight codes $A(n, d, \omega)$ and Steiner systems in general is a notoriously challenging problem[7]. The construction of optimal $A(n, d, \omega)$ for large values of n, d, ω is a long-standing open problem in coding theory. We first present some existing constructions on Steiner systems and binary constant weight codes. The following result characterizes the size of optimal 2-trackable access structures with minimal sets of size 3.

Theorem 5 ([40, 54]). *The following statement holds.*

$$|\mathcal{A}^\star(n,3,2)| = A(n,4,3) = \begin{cases} \lfloor \frac{n}{3} \lfloor \frac{n-1}{2} \rfloor \rfloor & \text{if } n \not\equiv 5 \mod 6 \\ \lfloor \frac{n}{3} \lfloor \frac{n-1}{2} \rfloor \rfloor - 1 & \text{if } n \equiv 5 \mod 6 \end{cases}$$

The next result characterizes the size of maximal 3-trackable access structures with size-4 minimal sets for almost all values of n except for $n = 5 \mod 6$. Determining $A(n, 4, 4)$ for $n = 5 \mod 6$ remains an interesting open problem in coding theory.

Theorem 6 ([32, 40]). *The following statement holds.*

$$|\mathcal{A}^*(n,4,3)| = A(n,4,4) = \begin{cases} \frac{n(n-1)(n-2)}{24} & \text{if } n \equiv 2 \text{ or } 4 \mod 6 \\ \frac{n(n-1)(n-3)}{24} & \text{if } n \equiv 1 \text{ or } 3 \mod 6 \\ \frac{n(n^2-3n-6)}{24} & \text{if } n \equiv 0 \mod 6 \end{cases}$$

More on Steiner systems. Steiner systems are not universally attainable for all parameters; their existence hinges on meeting specific natural divisibility criteria. For example, a $S(n, 3, 2)$ exists if and only if $n \equiv 1, 3 \mod 6$. Keevash [34] demonstrates that, for general $S(n, k, t)$ Steiner systems, these essential conditions also serve as sufficient conditions, granted that n is adequately large.

6.4 Near-optimal TAS

This section presents constructions of some near-optimal TAS.

Polynomial-based constructions. The first construction in Figure 2 is based on Reed-Solomon codes. This construction is common in combinatorial design (see Theorem 4.11 [3]). It is also used as a fundamental building block in Nisan-Wigderson pseudorandom generators [44]. The construction works when k is of order $\sqrt{n}$. We prove that the robustness of this access structure is n/k.To handle the case where k is larger, we naturally extend this construction to Figure 3 based on algebraic geometry codes with a slight loss in other parameters. We emphasize that these constructions are efficient.

Theorem 7 (Near-optimal TAS with Reed-Solomon codes [3]). *For every $k > \omega \geqslant 1$ and $n \geqslant 2k^2$, there is an efficient construction of (n, k, ω)-trackable access strucure $\mathcal{A}$ satisfying*

$$|\mathcal{A}^\star| \geqslant (n/2k)^\omega, \text{ and } r(\mathcal{A}) = n/k.$$

Theorem 8 (Near-optimal TAS with AG codes). *For every $k > \omega \geqslant 1$ and $k = \Theta(n)$, there is an efficient construction of (n, k, ω)-trackable access structure $\mathcal{A}$ satisfying*

$$|\mathcal{A}^\star| \geqslant p^{\omega - g}, \text{ and } r(\mathcal{A}) = n/k,$$

where g is the genus of the divisor used in the AG codes.

[7] For example, $A(111, 20, 11) \leqslant 111$, with equality if and only a projective plane of order 10 exists.

A randomized constructions from combinatorial design. Some variants of the Rodl nibble algorithm [48] are utilized to construct asymptotically optimal partial Steiner systems. Subsequent works [27,35] have improved the $o(1)$ term.

Theorem 9 ([27, 35, 48]). *For any fixed $k > t$, there is a partial Steiner system $S_p(n, k, t)$ of size at least*

$$(1 - o(1)) \cdot \binom{n}{t} \cdot \binom{k}{t}^{-1}.$$

6.5 TAS with optimal robustness

This section presents constructions of TAS with asymptotically optimal robustness based on partial Steiner systems with high independence numbers.

Equivalence between robustness and minimum vertex cover of hypergraphs. Recall that the robustness of an access structure is the minimum number of parties to corrupt to make reconstruction impossible. The robustness is equivalent to the minimum cover of the hypergraph representing the access structure – each party is a vertex, and each authorized set is a hyperedge. By the monotone property, the robustness equals the minimum cover of the hypergraph representing the minimal access structure.

It is well-known that the dual of the minimum cover problem is the maximum independence set problem. The sum of the minimum cover and the maximum independence set in a hypergraph equals the total number of vertices. Thus, a high robustness access structure is equivalent to a hypergraph with a small independence number – the size of the smallest independent set. We formalize the above observation as follows.

Proposition 3. *For any (monotone) access structure $\mathcal{A} \subseteq 2^{[n]}$,*

$$r(\mathcal{A}) = n - \alpha(\mathcal{A}^\star),$$

where $r(\mathcal{A})$ denotes the robustness of the structure $\mathcal{A}$, and $\alpha(\mathcal{A}^\star)$ denotes the independence number.

Our construction of high-robustness structures is then reduced to the construction of partial Steiner systems with small independence numbers, which are well-studied in combinatorial design literature.

Theorem 10 ([49]). *For any $n > k > \omega \in \mathbb{N}$ with $\omega \geqslant 2$, there exists an partial Steiner (n, k, ω)-system with independence number*

$$\alpha(G) \leqslant c \cdot n^{\frac{k-\omega}{k-1}} ((\log n)^{\frac{1}{k-1}}),$$

where $c = c(k, \omega)$ is a constant depend only on k, ω.

Theorem 10 is tight up to the constant factor c. To prove Theorem 10, Rödl and Sinajová utilize the Lovász Local Lemma to demonstrate that a randomly chosen k-uniform hypergraph qualifies as such a design. Thus, although their finding establishes the existence of such designs, it does not offer a direct method for their construction. Recently, in the context of randomness extractors, [13] presented a deterministic construction.

Theorem 11 ([13]). *For any constants $k > \omega \in \mathbb{N}$, there are explicit partial Steiner (n, k, ω)-systems $(G_n)_{n\in\mathbb{N}}$ with independence number*

$$\alpha(G_n) = \begin{cases} c_{k,\omega} \cdot n^{\frac{2(k-\omega)}{k}} & \text{if } k \text{ is even,} \\ c_{k+1,\omega} \cdot n^{\frac{2(k+1-\omega)}{k+1}}) & \text{if } k \text{ is odd,} \end{cases}$$

where $c_{k,\omega}$ is a constant depend only on k, ω.

The constant $c_{k,\omega} = C \cdot k^4$ for some global constant C. The construction for the odd case is based on the construction of the even one. There is a slight loss in the parameters. Observe that the independence number is sublinear in n if $k \leqslant 2\omega$. Liu and Mubayi [38] provide constructions when $k \geqslant 2\omega$ for certain pairs of (k, ω). Their proof is based on a recent result about the maximum size of a set in $\mathbb{Z}_6^n$, avoiding 6-term arithmetic progression.

Consequently, TAS with asymptotically optimal robustness can be obtained from the above constructions.

Corollary 3 (Asymptotically optimal robustness). *For any $n > k > \omega \in \mathbb{N}$, there is a (n, k, ω)-trackable access structure $\mathcal{A}$ with robustness*

$$r(\mathcal{A}) \geqslant n(1 - c \cdot n^{-\varepsilon}),$$

where $c = c(k, \omega)$ and $\varepsilon = \varepsilon(k, \omega)$ are constants depend only on k, ω.

6.6 Secret sharing and VSS schemes

This section presents our constructions of secret sharing schemes.

Realizing secret sharing on TAS through generic constructions. A straightforward method for constructing trackable secret sharing using generic approaches [9, 30] would entail each party's share size being proportionate to the minimal access structure's size. For instance, Benaloh and Leichter's method for any access structure relies on monotone formulae [9]. It's evident that a formula with a size equal to the minimal access structure can represent any access structure. Employing the Benaloh-Leichter construction, the secret share's size is proportional to the minimal access structure's size, which can be approximately $\binom{n}{\omega}$ for (n, k, ω)-trackable access structures with the largest size.

Some ideal secret sharing constructions. The above construction works for any access structure, but the information ratio can be huge for access structures of large size. Our objective is to construct secret-sharing schemes more efficiently.

The Fano plane access structure (the $S(n, 3, 2)$ Steiner triple system) admits an ideal secret-sharing scheme [42, 43]. Martí-Farré and Padró [42] provide a complete characterization of the ideal access structures with an intersection number equal to one—structures in which at most one participant is in the intersection of any two distinct minimal qualified subsets. Note that this is equivalent to characterizing which $(n, *, \omega = 2)$-TAS[8] admit an ideal secret sharing. An $(n, *, 2)$-TAS $\mathcal{A}$ is ideal if and only if every connected component of $\mathcal{A}$ is a complete bipartite graph, a star, the Fano plane access structure, or some specific small graphs. This implies that most $(n, k, 2)$-TAS are not ideal. However, using this characterization and decomposition techniques [9, 55], we prove the following result, improving the information rate compared to the generic constructions.

Theorem 12. *There is an efficient construction of any 2-trackable secret sharing with information ratio $O(n)$, where n is the number of parties.*

Proof. (1) It follows from [42] that a star access structure with an intersection number equal to one admits a vector space construction. This implies a star $(n, k, 2)$-TAS admits an ideal secret sharing.

(2) Decompose the access structure into stars, a star for each party. There will be n stars.

(3) Applying the decomposition technique [9, 55] yields a linear secret sharing schemes with information ratio is n, which is $O(n^2)$ if using the generic constructions.

[8] The size of minimal sets is not necessarily the same.

7 Measure non-trivial information gain

Let X be a discrete random variable (for a secret x) on a finite alphabet $\mathcal{X} = \{x_1, \ldots, x_n\}$ according to a probability distribution $\mathbf{p} = (p_1, \ldots, p_n)$ where $p_i = \mathbb{P}[X = x_i]$ for $i \in [n]$. Let $f : \mathcal{X} \mapsto \mathcal{Y}$ be any function, e.g., $f(x) = x$ or $f(x)$ outputs the most significant bit or Hamming weight of x. We are interested in the following question:

$$\text{Given a constant } \gamma \in (0,1), \text{ is } \mathbb{P}[\text{guessing } f(x) \text{ correctly in one shot}] < \gamma?$$

If f is a bijective function, then the best guess one can come up with is when f is evaluated at the most likely value for x. Otherwise, one naive approach is to evaluate f on all possible inputs, or at least the significant ones. Another approach is to connect the guessability notion with existing entropy measures.

Entropy of a function of a random variable. First, consider Shannon entropy (Definition 9). If f is a bijection, then

$$H(f(X)) = H(X)$$

where $H(\cdot)$ computes the Shannon entropy.

If f is a surjection, $H(f(X)) < H(X)$. Cicalese et al. [19] provides tighter upper and lower bounds for the entropy of these functions as follows. Recall the probability distribution $\mathbf{p} = (p_1, \ldots, p_n)$. Without loss of generality, we let $p_1 \geqslant p_2 \ldots \geqslant p_n \geqslant 0$. Let m be an integer and $2 \leqslant m < n$. Let set $\mathcal{Y}_m = \{y_1, \ldots, y_m\}$. Denote the family of surjective functions with m possible outputs as $\mathcal{F}_m = \{f | f : \mathcal{X} \mapsto \mathcal{Y}_m, |f(\mathcal{X})| = m\}$. Define $R_m(\mathbf{p}) = (r_1, \ldots, r_m)$ where $\forall i \in [m], r_i = \frac{1}{m}$ if the maximum probability $p_1 < \frac{1}{m}$, and

$$r_i = \begin{cases} p_i & i = 1, \ldots, i^* \\ \frac{\sum_{j=i^*+1}^{n} p_j}{m - i^*} & i = i^* + 1, \ldots, m \end{cases}$$

if $p_1 \geqslant \frac{1}{m}$. Here $i^* = \max\{i \in [m-1] : p_i \geqslant \frac{\sum_{j=i+1}^{n} p_j}{m-i}\}$.

Define $Q_m(\mathbf{p}) = (q_1, \ldots, q_m)$ where

$$q_i = \begin{cases} \sum_{j=1}^{n-m+1} p_j & i = 1 \\ p_{n-m+i} & i = 2, \ldots, m \end{cases}$$

Then

$$\max_{f \in \mathcal{F}_m} H(f(X)) \in [H(R_m(\mathbf{p})) - \mu, H(R_m(\mathbf{p}))]$$

where $\mu = 1 - \frac{1+\ln(\ln 2)}{\ln 2} < 0.09$ and

$$\min_{f \in \mathcal{F}_m} H(f(X)) = H(Q_m(\mathbf{p}))$$

Remark 3. Computing the exact lower bound is easy. Computing the exact upper bound of the Shannon entropy of any surjective function on a random variable is NP-hard [19], which can be shown with a reduction from partition problem. But a polynomial time approximation algorithm with additive approximation factor μ exists.

Remark 4. Computing the Shannon entropy of a specific function on a random variable can be done explicitly on all inputs or sampled inputs (Definition 2 and Theorem 1 in Lorentz [39]).

Sason [50] considers Rényi entropy (Definition 10) and proves the following upper and lower bounds

$$\max_{f \in \mathcal{F}_m} H(f(X)) \in [H(R_m(\mathbf{p})) - e(d), H(R_m(\mathbf{p}))]$$

$$\min_{f \in \mathcal{F}_m} H_d(f(X)) = H_d(Q_m(\mathbf{p}))$$

where $e(d) = 0.09$ for $d = 1$ (Shannon entropy case) and $e(d) = \log \frac{d-1}{2^d - 2} - \frac{d-1}{d} \log \frac{d}{2^d - 1}$ otherwise. As d increases, the error $e(d)$ increases.

Relationship between entropy and guessability. Entropy in general captures the degree of uncertainty.

Definition 9 (Shannon entropy). *Let X be a discrete random variable which takes values in a finite alphabet $\mathcal{X}$. For $x \in \mathcal{X}$, let $p(x) = \mathbb{P}[X = x]$, and $p(x) > 0$. The Shannon entropy of X is*

$$H(X) = -\sum_{x \in \mathcal{X}} (p(x) \log_2(p(x)))$$

Definition 10 (Rényi entropy). *Let X be a discrete random variable which takes values in a finite alphabet $\mathcal{X}$. For $x \in \mathcal{X}$, let $p(x) = \mathbb{P}[X = x]$, and $p(x) > 0$. The Rényi entropy of order $d \in (0, 1) \cup (1, \infty)$ is*

$$H_d(X) = \frac{1}{1-d} \log(\sum_{x \in \mathcal{X}} p(x)^d)$$

When guessing the output, the optimal strategy that minimizes the average number of guesses is to guess the values in order of decreasing probabilities. When parties are restricted to *one guess*, we desire the probability of guessing a non-trivial function output correctly to be small. In other words, to show a function f is non-trivial, it is sufficient to show that its min-entropy ($d = \infty$ in Rényi entropy) is above a known constant threshold (set by a protocol designer). More specifically, consider a threshold γ (preferably closer to 1).

(1) When proving the *non-triviality of a function*, one utilizes the lower bound for the min-entropy and shows that it is $> \log \frac{1}{\gamma}$.
(2) When proving the triviality of a function, one utilizes the upper bound and shows that it is $\leqslant \log \frac{1}{\gamma}$, though it might be harder to prove due to the higher approximation error as d approaches ∞. In this case, one can resort to prove by evaluating the entropy of the function explicitly.
(3) When the min-entropy lower bound is lower than the threshold and its upper bound is higher, one can resort to evaluating the entropy directly.

8 Related work

Traceable secret sharing. When given access to a private reconstruction program created by colluding parties, the traceable secret sharing (TSS) primitive [12, 26] allows tracing at least 1 colluding party with non-negligible probability (defined as *traceability*) and never implicates non-colluding parties (defined as *non-imputability*). Aside from a sharing and a reconstruction algorithm, TSS specifies a tracing algorithm for generating proofs of guilt from a reconstruction program and a judging algorithm for verifying the proof. To ensure non-imputability, the sharing phase in [26] utilizes a secure 2PC such that the dealer does not learn the complete shares of a party. The size of shares in the constructed TSS is quadratic in the size of the secret. [12] improves the quadratic overhead to linear for the widely used Shamir [53] and Blakley [11] secret sharing schemes but only if the reconstruction box outputs the entire secret (instead of anything non-trivial about the secret as in [26]). Overall, the setting in TSS is more benign, e.g., the computation circuits are known, one can query the pirate reconstruction program, and at least one share holder submits a share individually.

Utilize MPC hardness. [21] assumes that computing many hashes on a secret value quickly is hard using MPC but feasible for an individual who knows the secret. They propose a secret sharing with snitching (SSS) scheme under network synchrony where a snitching party and the dealer can prove to a judge that another party colluded by computing sufficiently many hashes in a short time frame. The SSS scheme has an inefficient reconstruction algorithm that involves repeatedly computing a large number of hashes. Besides, restricting the collusion method to MPC can be limiting.

References

1. Drand: Distributed randomness beacon, https://drand.love/
2. Zengo threshold wallet, https://github.com/ZenGo-X/gotham-city/blob/master/white-paper/white-paper.pdf
3. Babai, L., Frankl, P.: Linear algebra methods in combinatorics: with applications to geometry and computer science. Department of Computer Science, univ. of Chicag (1992)
4. Bai, G., Damgård, I., Orlandi, C., Xia, Y.: Non-interactive verifiable secret sharing for monotone circuits. In: Pointcheval, D., Nitaj, A., Rachidi, T. (eds.) AFRICACRYPT 16. LNCS, vol. 9646, pp. 225–244. Springer, Heidelberg (Apr 2016). https://doi.org/10.1007/978-3-319-31517-1_12
5. Bai, G., Damgård, I., Orlandi, C., Xia, Y.: Non-interactive verifiable secret sharing for monotone circuits. Cryptology ePrint Archive, Paper 2016/078 (2016), https://eprint.iacr.org/2016/078
6. Beimel, A.: Lower bounds for secret-sharing schemes for k-hypergraphs. In: Chung, K. (ed.) 4th Conference on Information-Theoretic Cryptography, ITC 2023, June 6-8, 2023, Aarhus University, Aarhus, Denmark. LIPIcs, vol. 267, pp. 16:1–16:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2023). https://doi.org/10.4230/LIPICS.ITC.2023.16, https://doi.org/10.4230/LIPIcs.ITC.2023.16
7. Beimel, A., Farràs, O., Lasri, O.: Improved polynomial secret-sharing schemes. In: Rothblum, G.N., Wee, H. (eds.) Theory of Cryptography - 21st International Conference, TCC 2023, Taipei, Taiwan, November 29 - December 2, 2023, Proceedings, Part II. Lecture Notes in Computer Science, vol. 14370, pp. 374–405. Springer (2023). https://doi.org/10.1007/978-3-031-48618-0_13, https://doi.org/10.1007/978-3-031-48618-0_13
8. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: 20th ACM STOC. pp. 1–10. ACM Press (May 1988). https://doi.org/10.1145/62212.62213
9. Benaloh, J.C., Leichter, J.: Generalized secret sharing and monotone functions. In: Goldwasser, S. (ed.) Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings. Lecture Notes in Computer Science, vol. 403, pp. 27–35. Springer (1988). https://doi.org/10.1007/0-387-34799-2_3, https://doi.org/10.1007/0-387-34799-2_3
10. Bishop, A., Pastro, V., Rajaraman, R., Wichs, D.: Essentially optimal robust secret sharing with maximal corruptions. In: Advances in Cryptology–EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I 35. pp. 58–86. Springer (2016), https://doi.org/10.1007/978-3-662-49890-3_3
11. Blakley, G.R.: Safeguarding cryptographic keys. In: Managing Requirements Knowledge, International Workshop on. pp. 313–313. IEEE Computer Society (1979)
12. Boneh, D., Partap, A., Rotem, L.: Traceable secret sharing: Strong security and efficient constructions. In: Annual International Cryptology Conference. pp. 221–256. Springer (2024), https://doi.org/10.1007/978-3-031-68388-6_9
13. Chattopadhyay, E., Goodman, J.: Improved extractors for small-space sources. In: 62nd FOCS. pp. 610–621. IEEE Computer Society Press (Feb 2022). https://doi.org/10.1109/FOCS52979.2021.00066
14. Chattopadhyay, E., Goodman, J., Goyal, V., Li, X.: Extractors for adversarial sources via extremal hypergraphs. In: Makarychev, K., Makarychev, Y., Tulsiani, M., Kamath, G., Chuzhoy, J. (eds.) 52nd ACM STOC. pp. 1184–1197. ACM Press (Jun 2020). https://doi.org/10.1145/3357713.3384339
15. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (extended abstract). In: 20th ACM STOC. pp. 11–19. ACM Press (May 1988). https://doi.org/10.1145/62212.62214
16. Chen, Y.H., Lindell, Y.: Feldman's verifiable secret sharing for a dishonest majority. IACR Communications in Cryptology **1**(1) (2024), https://eprint.iacr.org/2024/031
17. Cheraghchi, M.: Nearly optimal robust secret sharing. Designs, Codes and Cryptography **87**, 1777–1796 (2019), https://doi.org/10.1007/s10623-018-0578-y
18. Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In: FOCS. pp. 383–395 (1985)
19. Cicalese, F., Gargano, L., Vaccaro, U.: Bounds on the entropy of a function of a random variable and their applications. IEEE Transactions on Information Theory **64**(4), 2220–2230 (2017), https://doi.org/10.1109/TIT.2017.2787181
20. Csirmaz, L.: The size of a share must be large. In: Santis, A.D. (ed.) EUROCRYPT'94. LNCS, vol. 950, pp. 13–22. Springer, Heidelberg (May 1995). https://doi.org/10.1007/BFb0053420
21. Dziembowski, S., Faust, S., Lizurej, T., Mielniczuk, M.: Secret sharing with snitching. In: Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (2024)

22. Gabizon, A., Williamson, Z.J., Ciobotaru, O.: Plonk: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive (2019), https://eprint.iacr.org/2019/953
23. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Secure distributed key generation for discrete-log based cryptosystems. Journal of Cryptology **20**, 51–83 (2007)
24. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or A completeness theorem for protocols with honest majority. In: Aho, A. (ed.) 19th ACM STOC. pp. 218–229. ACM Press (May 1987). https://doi.org/10.1145/28395.28420
25. Gong, T., Henry, R., Psomas, A., Kate, A.: More is merrier: Relax the non-collusion assumption in multi-server pir. In: 2024 IEEE Symposium on Security and Privacy (SP). pp. 95–95. IEEE Computer Society, Los Alamitos, CA, USA (may 2024). https://doi.org/10.1109/SP54263.2024.00095, https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00095
26. Goyal, V., Song, Y., Srinivasan, A.: Traceable secret sharing and applications. In: Annual International Cryptology Conference. pp. 718–747. Springer (2021), https://doi.org/10.1007/978-3-030-84252-9_24
27. Grable, D.A.: More-than-nearly-perfect packings and partial designs. Comb. **19**(2), 221–239 (1999). https://doi.org/10.1007/S004930050053, https://doi.org/10.1007/s004930050053
28. Groth, J.: On the size of pairing-based non-interactive arguments. In: Annual international conference on the theory and applications of cryptographic techniques. pp. 305–326. Springer (2016), https://doi.org/10.1007/978-3-662-49896-5_11
29. Inc, C.: Ddos threat report for 2024 q2 by cloudflare (2024), https://blog.cloudflare.com/ddos-threat-report-for-2024-q2/
30. Ito, M., Saito, A., Nishizeki, T.: Secret sharing scheme realizing general access structure. Electronics and Communications in Japan (Part III: Fundamental Electronic Science) **72**(9), 56–64 (1989), https://doi.org/10.1002/ecjc.4430720906
31. Johnson, S.: A new upper bound for error-correcting codes. IRE Transactions on Information Theory **8**(3), 203–207 (1962), https://doi.org/10.1109/TIT.1962.1057714
32. Kalbfleisch, J., Stanton, R.: Maximal and minimal coverings of (k- 1)-tuples by k-tuples. Pacific journal of mathematics **26**(1), 131–140 (1968), http://dx.doi.org/10.2140/pjm.1968.26.131
33. Kate, A., Mangipudi, E.V., Mukherjee, P., Saleem, H., Thyagarajan, S.A.K.: Non-interactive vss using class groups and application to dkg. In: Proceedings of the 2024 ACM conference on Computer and communications security (2024), https://eprint.iacr.org/2023/451
34. Keevash, P.: The existence of designs. arXiv preprint arXiv:1401.3665 (2014), https://doi.org/10.48550/arXiv.1401.3665
35. Kim, J.H.: Nearly optimal partial steiner systems. Electron. Notes Discret. Math. **7**, 74–77 (2001). https://doi.org/10.1016/S1571-0653(04)00228-8, https://doi.org/10.1016/S1571-0653(04)00228-8
36. Komlo, C., Goldberg, I., Stebila, D.: A formal treatment of distributed key generation, and new constructions. Cryptology ePrint Archive (2023)
37. Liu, T., Vaikuntanathan, V.: Breaking the circuit-size barrier in secret sharing. In: Diakonikolas, I., Kempe, D., Henzinger, M. (eds.) 50th ACM STOC. pp. 699–708. ACM Press (Jun 2018). https://doi.org/10.1145/3188745.3188936
38. Liu, X., Mubayi, D.: On explicit constructions of designs. Electron. J. Comb. **29**(1) (2022). https://doi.org/10.37236/10513, https://doi.org/10.37236/10513
39. Lorentz, R.A.: On the entropy of a function. Journal of Approximation Theory **158**(2), 145–150 (2009), https://doi.org/10.1016/j.jat.2008.07.004
40. MacWilliams, F.J., Sloane, N.J.A.: The theory of error-correcting codes, vol. 16. Elsevier (1977)
41. Mangipudi, E.V., Desai, U., Minaei, M., Mondal, M., Kate, A.: Uncovering impact of mental models towards adoption of multi-device crypto-wallets. In: ACM CCS. pp. 3153–3167. ACM (2023)
42. Martí-Farré, J., Padró, C.: Secret sharing schemes on access structures with intersection number equal to one. In: International Conference on Security in Communication Networks. pp. 354–363. Springer (2002), https://doi.org/10.1007/3-540-36413-7_26
43. Martí-Farré, J., Padró, C.: Secret sharing schemes on access structures with intersection number equal to one. Discret. Appl. Math. **154**(3), 552–563 (2006). https://doi.org/10.1016/J.DAM.2005.09.003, https://doi.org/10.1016/j.dam.2005.09.003
44. Nisan, N., Wigderson, A.: Hardness vs randomness. J. Comput. Syst. Sci. **49**(2), 149–167 (1994). https://doi.org/10.1016/S0022-0000(05)80043-1, https://doi.org/10.1016/S0022-0000(05)80043-1
45. Ozdemir, A., Boneh, D.: Experimenting with collaborative {zk-SNARKs}:{Zero-Knowledge} proofs for distributed secrets. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 4291–4308 (2022), https://eprint.iacr.org/2021/1530

46. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In: 21st ACM STOC. pp. 73–85. ACM Press (May 1989). https://doi.org/10.1145/73007.73014
47. Rivest, R.L., Shamir, A., Wagner, D.A.: Time-lock puzzles and timed-release crypto. Tech. rep., USA (1996)
48. Rödl, V.: On a packing and covering problem. European Journal of Combinatorics **6**(1), 69–78 (1985), https://doi.org/10.1016/S0195-6698(85)80023-8
49. Rödl, V., Šinajová, E.: Note on independent sets in steiner systems. Random Structures & Algorithms **5**(1), 183–190 (1994), https://doi.org/10.1002/rsa.3240050117
50. Sason, I.: Tight bounds on the rényi entropy via majorization with applications to guessing and compression. Entropy **20**(12), 896 (2018), https://doi.org/10.3390/e20120896
51. Schoenmakers, B.: A simple publicly verifiable secret sharing scheme and its application to electronic voting. In: Annual International Cryptology Conference. pp. 148–164. Springer (1999)
52. Selten, R.: Spieltheoretische behandlung eines oligopolmodells mit nachfrageträgheit: Teil i: Bestimmung des dynamischen preisgleichgewichts. Zeitschrift für die gesamte Staatswissenschaft/Journal of Institutional and Theoretical Economics (H. 2), 301–324 (1965), https://www.jstor.org/stable/40748884
53. Shamir, A.: How to share a secret. Communications of the ACM **22**(11), 612–613 (1979)
54. Spencer, J.: Maximal consistent families of triples. Journal of Combinatorial Theory **5**(1), 1–8 (1968), https://doi.org/10.1016/S0021-9800(68)80023-7
55. Stinson, D.R.: New general lower bounds on the information rate of secret sharing schemes. In: Brickell, E.F. (ed.) CRYPTO'92. LNCS, vol. 740, pp. 168–182. Springer, Heidelberg (Aug 1993). https://doi.org/10.1007/3-540-48071-4_12
56. Yao, A.C.C.: Theory and applications of trapdoor functions (extended abstract). In: 23rd FOCS. pp. 80–91. IEEE Computer Society Press (Nov 1982). https://doi.org/10.1109/SFCS.1982.45

A Additional preliminaries

Commitment scheme. A commitment scheme is a tuple of two algorithms $(\mathsf{Comm}(\cdot), \mathsf{Reveal}(\cdot))$:

- $c \leftarrow \mathsf{Comm}(v)$: Given input value v, output a commitment c on v.
- $v' \leftarrow \mathsf{Reveal}(c)$: Given a commitment c, reveal the committed value c'.

A secure commitment scheme achieves hiding and binding properties. Hiding requires that by observing the commitment c, one does not learn more about the committed value v. Binding requires that the committer cannot open the commitment to another value $v' \neq v$. Both guarantees can be either information-theoretic or computational but cannot be information-theoretic simultaneously.

zkSNARKs. Informally, a *proof* for a relation $\mathcal{R}$ is a protocol that allows a prover $\mathcal{P}$ with an instance x to prove to an efficient verifier that there exists a witness w such that $\mathcal{R}(x, w) = 1$. If the proof avoids sending the entire witness w to $\mathcal{V}$, it is called *succint*. If the proof contains a single message from $\mathcal{P}$ to $\mathcal{V}$, it is called *non-interactive*, which can be described with a tuple of three algorithms:

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{R})$: Given the security parameter and the relation, output public parameters.
- $\pi/\perp \leftarrow \mathsf{Prove}(\mathsf{pp}, x, w)$: Given the public parameters, instance x and witness w, if $\mathcal{R}(x, w) = 1$, output a proof π. Otherwise, output $\perp$.
- $\{1, 0\} \leftarrow \mathsf{Verify}(\mathsf{pp}, x, \pi)$: Verify proof π of instance x.

Informally, a zkSNARK is a proof (or more suitably, an argument) that satisfies the following properties [45]:

- (Completeness) If $\mathcal{R}(x, w) = 1$, then an honest $\mathcal{P}$ convinces $\mathcal{V}$ except with negligible probability. If $\mathcal{R}(x, w) = 0$, Prove outputs $\perp$.
- (Computational knowledge soundness) For every prover $\mathcal{P}$ that convinces $\mathcal{V}$ to output 1, there exists a polynomial time extractor that can use $\mathcal{P}$ to output w such that $\mathcal{R}(x, w) = 1$.
- (Zero-knowledge) The triple (pp, x, π) reveals nothing about witness w.
- (Sunccintness) Proof size and verification time are $o(|\mathcal{R}|)$.