Disincentivize Collusion in Verifiable Secret Sharing

Tiantian Gong[1](✉), Aniket Kate[2,3], Hemanta K. Maji[2], and Hai H. Nguyen[4]

[1] Yale University, New Haven, CT 06524, USA
tg@purdue.edu
[2] Purdue University, West Lafayette, IN 47906, USA
{aniket,hmaji}@purdue.edu
[3] Supra Research, Miami, USA
[4] ETH Zurich, Zürich, Switzerland
haihoang.nguyen@inf.ethz.ch

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 [2,38]. A user shares her transaction signing key among many servers using a secret sharing scheme,

T. Gong and Hai H. Nguyen—Completed most of this work at Purdue University.

S. Fehr and P.-A. Fouque (Eds.): EUROCRYPT 2025, LNCS 15605, pp. 34–64, 2025.
https://doi.org/10.1007/978-3-031-91092-0_2

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) [8,16,23,43,51], time-release cryptography [44],and distributed randomness beacons [1]: 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 [12,25] 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) [19] 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 (described in Sect. 4.1).

Gong et al. [24] recently studied collusion deterrence in multiserver private information retrieval with a singleton access structure.[1] 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.[2]

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

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

[2] In robust secret sharing [10,18], 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	$f^{(n,k)}(\omega)$

then define *trackable access structures* (TAS) and design a collusion deterrence mechanism $\mathbf{W}_1$ (described on page 7) 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 $\omega -$ *trackable* when any size-ω subset is contained in at most one minimal set. For example, an (n, k, ω)-design, as defined in Sect. 6.1, is ω-trackable; every size-ω subset appears in a unique minimal set in this design.

In Table 1, We compare the two mechanisms along the following metrics.

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 6. 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 [24]. 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.

Our Second Mechanism. For ω-trackable access structures ($1 \leqslant \omega < k$), we propose $(\mathbf{W}_1, \mathbf{P})$ on page 7. It is $(k-1)$-fair, $(k-1-\omega)$-effective and has optimal 0 fairness hazard. Its robustness is $f^{(n,k)}(\omega)$, an appropriate increasing function defined in Table 2, indicating a *trade-off between effectiveness and robustness* when setting ω. In Sect. 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.

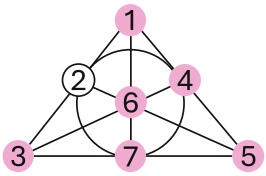
TAS and VSS. Given parameters n, k, ω, our objective is to construct efficient secret-sharing schemes on (n, k, ω)-TAS with high robustness. We systematically examine secret sharing on TAS, uncovering several connections with combinatorics: We establish links between TAS and well-known concepts in combinatorial 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 near-optimal size (Theorems 6 & 7).
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 [9,29] 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 $\Omega(n^{2-1/(k-1)}/k)$ for some



(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) 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. (Color figure online)

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. Figure 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.

2 Technical Overview

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 secret is 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 Sect. 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 parties 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 group 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 Sect. 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 [35] 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 [30] 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 Sect. 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.

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

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

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

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

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

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 ratio between the robustness and the number of parties n, called 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.

Remark 2. There are randomized constructions for near-optimal partial Steiner Systems and thus for TAS from combinatorial design [45]. 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.

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	?	?	$n - k^4 n^{1/2}$	$n - k^4 n^{\frac{2}{k}}$
$k = 4$	$n/4$	$n - n^{0.973}$	$n(1 - o(1))$		
$k = 3$	$n/3$	$n - \frac{n}{(\log n)^{1.01}}$			
$k = 2$	$n/2$				

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 [45], we show that there exists a randomized construction of (n, k, ω)-TAS such that the fractional robustness tends to 1 as n tends to infinity. Recently, in the context of randomness extractors, Chattopadhyay and Goodman [14] 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 [14], Chattopadhyay, Goodman, Goyal, and Li [15] 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 [9,29] 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 [39,40]. More generally, Martí-Farré and Padró [39] 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 [39] characterizes 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.[3] Here, an access structure is a *star* if a party is contained in every authorized set. Using this characterization [39] and decomposition techniques [9,50], we construct secret sharing on $(n, k, 2)$-TAS: (i) Decompose the (n, k, ω)-TAS into n stars, one for each party, and construct

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

an ideal secret sharing for each star. (ii) Applying the well-known decomposition techniques [9,50] 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 (zkSNARKs) in the full version.

3.1 Secret Sharing Related 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.*

We later provide equivalent definitions in Sect. 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 [17,32]). *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.*

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

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 $\mathcal{F}_{\text{VSS}}^{t,\mathcal{A}}$. The simulator $\mathcal{S}$ controls the dealer in the same way if $\mathcal{S}$ corrupts it.

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

(i) Compute shares $\{s_i\}_{i\in[n]}$. Obtain the receipt of committing $s^\star$ and $\{s_i\}_{i\in[n]}$ from $\mathcal{F}_{\text{Commit}}$, denoted as cmt.

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

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

After step (i):

(ii) For $i \in [n]\backslash\mathcal{C}$, send s_i to i. If the dealer is not corrupted, also send $\{s_j\}_{j\in\mathcal{C}}$ to $\mathcal{S}$.

(iii) Store $\mathsf{tab}(i) = (s^\star, s_i)$ for all $i \in [n]$.

Verification. Upon receiving $(\mathsf{Verify}, i, \mathsf{cmt})$ from a party i, check if all shares are consistent via $\mathcal{F}_{\text{Commit}}$. If confirmed, send $(\mathsf{Verify}, i, \mathsf{valid})$ to all parties; otherwise, send $(\mathsf{Verify}, i, \mathsf{invalid})$ to all parties and abort.

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 access structure $\mathcal{A}$ 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 NI-VSS functionality $\mathcal{F}_{\text{VSS}}^{t,\mathcal{A}}$, parameterized by t and access structure $\mathcal{A}$. $\mathcal{F}_{\text{VSS}}^{t,\mathcal{A}}$ calls an ideal commitment functionality $\mathcal{F}_{\text{Commit}}$ such as [13].

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.

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 [47]), 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 [47]). *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 Sect. 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. 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 Sect. 4.2. We then extend the analysis to repeated runs of VSS in Sect. 4.3. We discuss the second mechanism providing the optimal fairness in Sect. 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: (1) The dealer shares the secret among the n parties with $\Pi_{\text{NI-VSS}}$ and sends the secret to the honest host H_h. (2) H_h then accepts reports of non-trivial knowledge about the secret through private authenticated channels. (3) 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 immediate issues about *framing* in the above simple protocol.

- First, 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 can collude with a share-holding party to frame others.

Dealing with False Alarms. To resolve the first issue, we penalize the sender of each wrong report the amount λ_p. Further, to discourage random guesses, we introduce the *non-triviality parameter* $\gamma \in [0, 1)$, which we provide details in the

full version. 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 label 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:

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

Here, $\frac{1}{q+1}$ is the probability of picking the share corresponding to user's actual secret (which the informed party has non-trivial prior knowledge of), and $\frac{q}{q+1}\gamma$ is the probability of picking one of the other shares and guessing some nontrivial information about a random secret correctly. In these cases, the informed party earns a reward λ_r. Otherwise, the party expects the penalty λ_p. Note that the random secrets are only to discourage false accusations from informed shareholding parties without participating in collusion. They are *not* used to hide the true secret when collusion actually takes place.

For the last problem, sampling random secrets and permuting shares are not sufficient anymore. 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 [25]. We provide details in the full version.

After adding the above components, we summarize the single-shot collusion deterrent NI-VSS protocol in Fig. 5. We next 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.

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.

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,

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

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 the full version) 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.

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 formal proof is deferred to the full version and 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.

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 [27] or Plonk [22] can be employed. Otherwise, collaborative zkSNARKs [42] 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 $(n-k)$.*

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 [28], 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 [42] only takes hundreds of microseconds per constraint. Overall, we can set Δ to be multiple weeks.

Online Dealer. If the dealer can be online periodically or at a known time (before a pre-determined proof collection window), 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,

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

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 detail in the full version), 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 $\mathbf{W}_1$.

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.

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 $\mathcal{V}$.

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 [24], there 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 $(n - k)$.

We present its proof in the full version. Intuitively, we 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.

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 depriving 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 [24].

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). *A k-homogeneous access structure $\mathcal{A}$ is untrackable if there exist a set X of $(k-1)$ parties such that any subset $X_s \subseteq X$ co-exist in at least two distinct minimal sets $A, B \in \mathcal{A}^\star$ where $A \cap B = X_s$.*

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.*

We present the proof by contradiction in the full version. Informally, let us assume such a mechanism exists. Denote the untrackable set of $(k-1)$ parties as S. By definition, they appear in two minimal sets. If the two non-overlapping parties in the two sets are slow, then colluding but not reporting gives them the highest utility. This is because the mechanism has 0 fairness hazard, and it cannot distinguish between exactly one of the two sets colluding or both sets colluding. Then for it to be effective, one party in S needs to suffer a utility loss, which results in this party being unwilling to report. We repeatedly apply this reasoning until arriving at the contradiction of the existence of such a 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 Sect. 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 $\Pi_{NI\text{-}VSS}^{\mathcal{V},\mathbf{W}_1}$ 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. More generally, when $1 < \omega < \frac{k+2}{3}$ (i.e., $(k-1)-(k-\omega)+(\omega-1) < (k-\omega)$), there is at most one successful colluding group. (This is because when a group has more than ω rational parties, they are disincentivized from collusion by the effectiveness of the mechanism.) Rational parties in this group are incentivized to collude if they are fast. After collusion, if there are exactly ω rational parties in this group, they 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. If this collusion group contains $< \omega$ rational parties, if only the rational parties report, $\mathbf{W}_1$ dismisses the reports. When the malicious parties also report, either this collusion group is located or a late (malicious) reporter (from another access group) is marked as a colluder.

When $\frac{k+2}{3} \leqslant \omega < k-1$, there can be at least two successful colluding groups. Same as before, rational parties in any of these groups are incentivized to collude if they are fast. If there are exactly ω rational parties in any of these groups, rational parties in the fastest such group pick R in Stage 2, become winners and receive λ_r. If all collusion groups contain less than ω rational parties, either this colluding group is located when sufficiently malicious parties in this group report or a late (malicious) reporter is marked as a colluder. Overall, the mechanism 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 and defer the proof to the full version. 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 $(n, k, \omega) - design$ is an k-uniform hypergraph with pairwise intersections of hyperedges 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-element subset is contained in a unique block to the condition that each ω-element subset is contained in at most one block.

Example 1. The Fano plane (Fig. 1a in Sect. 1) 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 and prove it in the full version. Recall that (n, k, ω)-TAS denotes a k-homogeneous ω-trackable access structure over $[n]$, and $\mathcal{A}^\star$ 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 $2k - 2(\omega - 1)$.*

Theorem 3. *Let $\mathcal{A}$ be an (n, k, ω)-TAS. It holds that $|\mathcal{A}^\star| \leqslant \binom{n}{\omega} \cdot \binom{k}{\omega}^{-1}$, where the equality holds 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

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

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 [30]). *Let $n, d, \omega \in \{1, 2, \ldots, \}$ such that $d \leqslant n$ and $\omega \leqslant n$. If $d > 2\omega$, then $A(n, d, \omega) = 1$. Otherwise ($d \leqslant 2\omega$), define $a = \begin{cases} d/2, & \text{if } d \text{ is even} \\ (d+1)/2, & \text{otherwise.} \end{cases}$. Then $A(n, d, \omega) \leqslant \lfloor \frac{n}{\omega} \lfloor \frac{(n-1)}{\omega-1} \cdots \lfloor \frac{(n-\omega+a)}{a} \rfloor \cdots \rfloor \rfloor$ where $\lfloor \cdot \rfloor$ is the floor function. Further, 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[4]. 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 for all n, and 3-trackable access structures with size-4 minimal sets for almost all values of n except for $n = 5 \mod 6$.

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

Theorem 5 ([31,37,49]). *The following statement holds.*

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

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

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 [33] 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 Fig. 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 [41]. 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 Fig. 3 based on algebraic geometry codes with a slight loss in other parameters. We emphasize that these constructions are efficient.

Theorem 6 (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 structure $\mathcal{A}$ satisfying*

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

Theorem 7 (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.

A Randomized Constructions from Combinatorial Design. Some variants of the Rodl nibble algorithm [45] are utilized to construct asymptotically optimal partial Steiner systems. Subsequent works [26,34] have improved the $o(1)$ term.

Theorem 8 ([26,34,45]). *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 access structure $\mathcal{A} \subseteq 2^{[n]}$, $r(\mathcal{A}) = n - \alpha(\mathcal{A}^\star)$.*

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 9 ([46]). *For any $n > k > \omega \in \mathbb{N}$ with $\omega \geqslant 2$, there exists a 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 9 is tight up to the constant factor c. To prove it, Rödl and Sinajová utilize the Lovász Local Lemma to demonstrate that a randomly chosen k-uniform hypergraph qualifies as such a design. Thus, their finding establishes the existence of such designs but does not offer a direct construction method. In the context of randomness extractors, [14] presented a deterministic construction.

Theorem 10 ([14]). *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 $c_{k,\omega} = C \cdot k^4$ 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 [36] 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)-TAS $\mathcal{A}$ 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 Trackable Access Structures Through Generic Constructions. A straightforward method for constructing trackable secret sharing using generic approaches [9,29] 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 [39,40]. Martí-Farré and Padró [39] 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[5] 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 [9,50], we state the following result that improves the information ratio of generic constructions and present the proof in the full version.

Theorem 11. *There is an efficient construction of any 2-trackable secret sharing with information ratio $O(n)$, where n is the number of parties.*

7 Related Work

Traceable Secret Sharing. When given access to a private reconstruction program created by colluding parties, the traceable secret sharing (TSS) primitive [12,25] 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

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

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 [25] 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 [48] 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 [25]). Overall, the setting in TSS is more benign: computation circuits are known; one can query the pirate reconstruction program; and at least one share holder submits a share individually.

Utilize MPC Hardness. Assumeing that computing many hashes on a secret quickly is hard using MPC but feasible for an individual knowing the secret, Dziembowski et al. [21] design a secret sharing with snitching scheme under network synchrony. A snitching party and the dealer can prove to a judge that another party colluded by computing sufficiently many hashes in a short time. The scheme has an inefficient reconstruction algorithm that involves repeatedly computing many hashes. Besides, restricting the collusion method to MPC can be limiting.

Acknowledgment. We thank anonymous reviewers for their valuable feedback. This work was supported in part by the National Science Foundation (NSF) under grant CNS1846316. Hemanta K. Maji is partly supported by CNS–2055605 and CCF–2327981. Hai H. Nguyen is partly supported by the Zurich Information Security & Privacy Center (ZISC), ETH Zurich.

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, University of Chicago (1992)
4. Bai, G., Damgård, I., Orlandi, C., Xia, Yu.: Non-interactive verifiable secret sharing for monotone circuits. In: Pointcheval, D., Nitaj, A., Rachidi, T. (eds.) AFRICACRYPT 2016. LNCS, vol. 9646, pp. 225–244. Springer, Cham (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)
6. Beimel, A.: Lower bounds for secret-sharing schemes for k-hypergraphs. In: Chung, K. (ed.) 4th ITC Conference 2023, pp. 16:1–16:13 (2023)
7. Beimel, A., Farràs, O., Lasri, O.: Improved polynomial secret-sharing schemes. In: Rothblum, G.N., Wee, H. (eds.) TCC 2023. LNCS, vol. 14370, pp. 374–405. Springer, Cham (2023). 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

9. Benaloh, J., Leichter, J.: Generalized secret sharing and monotone functions. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 27–35. Springer, New York (1990). 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: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 58–86. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_3
11. Blakley, G.R.: Safeguarding cryptographic keys. In: International Workshop on Managing Requirements Knowledge, p. 313 (1979)
12. Boneh, D., Partap, A., Rotem, L.: Traceable secret sharing: strong security and efficient constructions. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024. LNCS, vol. 14924, pp. 221–256. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-68388-6_9
13. Canetti, R., Fischlin, M.: Universally composable commitments. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 19–40. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_2
14. Chattopadhyay, E., Goodman, J.: Improved extractors for small-space sources. In: 62nd FOCS, pp. 610–621. IEEE Computer Society Press, February 2022
15. Chattopadhyay, E., Goodman, J., Goyal, V., Li, X.: Extractors for adversarial sources via extremal hypergraphs. In: 52nd ACM STOC, pp. 1184–1197. ACM Press, June 2020
16. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (extended abstract). In: 20th ACM STOC, pp. 11–19. ACM Press, May 1988
17. Chen, Y.H., Lindell, Y.: Feldman's verifiable secret sharing for a dishonest majority. IACR Commun. Cryptol. **1**(1) (2024)
18. Cheraghchi, M.: Nearly optimal robust secret sharing. Des. Codes Crypt. **87**, 1777–1796 (2019)
19. 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)
20. Csirmaz, L.: The size of a share must be large. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 13–22. Springer, Heidelberg (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 CCS (2024)
22. Gabizon, A., Williamson, Z.J., Ciobotaru, O.: PLONK: permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive (2019)
23. 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
24. 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
25. Goyal, V., Song, Y., Srinivasan, A.: Traceable secret sharing and applications. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021. LNCS, vol. 12827, pp. 718–747. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84252-9_24
26. Grable, D.A.: More-than-nearly-perfect packings and partial designs. Combinatorica **19**(2), 221–239 (1999)

27. Groth, J.: On the size of pairing-based non-interactive arguments. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 305–326. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_11
28. Cloudflare Inc.: DDoS threat report for 2024 Q2 by Cloudflare (2024). https://blog.cloudflare.com/ddos-threat-report-for-2024-q2/
29. Ito, M., Saito, A., Nishizeki, T.: Secret sharing scheme realizing general access structure. Electroni. Commun. Japan (Part III: Fundam. Electron. Sci.) **72**(9), 56–64 (1989)
30. Johnson, S.: A new upper bound for error-correcting codes. IRE Trans. Inf. Theor. **8**(3), 203–207 (1962)
31. Kalbfleisch, J., Stanton, R.: Maximal and minimal coverings of (k- 1)-tuples by k-tuples. Pac. J. Math. **26**(1), 131–140 (1968)
32. Kate, A., Mangipudi, E.V., Mukherjee, P., Saleem, H., Thyagarajan, S.A.K.: Non-interactive VSS using class groups and application to DKG. In: ACM CCS (2024)
33. Keevash, P.: The existence of designs. arXiv preprint arXiv:1401.3665 (2014)
34. Kim, J.H.: Nearly optimal partial Steiner systems. Electron. Notes Discret. Math. **7**, 74–77 (2001)
35. 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, June 2018
36. Liu, X., Mubayi, D.: On explicit constructions of designs. Electron. J. Comb. **29**(1) (2022)
37. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes, vol. 16. Elsevier (1977)
38. 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)
39. Martí-Farré, J., Padró, C.: Secret sharing schemes on access structures with intersection number equal to one. In: Cimato, S., Persiano, G., Galdi, C. (eds.) SCN 2002. LNCS, vol. 2576, pp. 354–363. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36413-7_26
40. 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)
41. Nisan, N., Wigderson, A.: Hardness vs randomness. J. Comput. Syst. Sci. **49**(2), 149–167 (1994)
42. Ozdemir, A., Boneh, D.: Experimenting with collaborative zk-SNARKs: zero-knowledge proofs for distributed secrets. In: USENIX Security 2022, pp. 4291–4308 (2022)
43. 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
44. Rivest, R.L., Shamir, A., Wagner, D.A.: Time-lock puzzles and timed-release crypto. Technical report, USA (1996)
45. Rödl, V.: On a packing and covering problem. Eur. J. Comb. **6**(1), 69–78 (1985)
46. Rödl, V., Šinajová, E.: Note on independent sets in Steiner systems. Random Struct. Algorithms **5**(1), 183–190 (1994)
47. Selten, R.: Spieltheoretische behandlung eines oligopolmodells mit nachfrageträgheit: Teil i: Bestimmung des dynamischen preisgleichgewichts. Zeitschrift für die gesamte Staatswissenschaft/J. Instit. Theor. Econ. (H. 2), 301–324 (1965)
48. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)

49. Spencer, J.: Maximal consistent families of triples. J. Comb. Theory **5**(1), 1–8 (1968)
50. Blundo, C., De Santis, A., Gargano, L., Vaccaro, U.: On the information rate of secret sharing schemes. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 148–167. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-48071-4_11
51. Yao, A.C.C.: Theory and applications of trapdoor functions (extended abstract). In: 23rd FOCS, pp. 80–91. IEEE Computer Society Press, November 1982