Cryptography from Planted Graphs: Security with Logarithmic-Size Messages

Damiano Abram
Aarhus University
damiano.abram@cs.au.dk

Amos Beimel
Ben-Gurion University
amos.beimel@gmail.com

Yuval Ishai
Technion
yuvali@cs.technion.ac.il

Eyal Kushilevitz
Technion
eyalk@cs.technion.ac.il

Varun Narayanan
UCLA
varunnkv@gmail.com

December 19, 2023

Abstract

We study the following broad question about cryptographic primitives: is it possible to achieve security against an arbitrary $\mathsf{poly}(n)$-time adversary with $O(\log n)$-size messages? It is common knowledge that the answer is "no" unless information-theoretic security is possible. In this work, we revisit this question by considering the setting of cryptography with *public information* and computational security.

We obtain the following results, assuming variants of well-studied intractability assumptions:

- A *private simultaneous messages* (PSM) protocol for every $f : [n] \times [n] \to \{0, 1\}$ requiring $(1+\varepsilon)\log n$-bit messages for most functions and $(2+\varepsilon)\log n$-bit messages for the remaining ones. We apply this towards *non-interactive* secure 3-party computation with similar message size in the preprocessing model, improving over previous 2-round protocols.
- A *secret-sharing scheme* for any "forbidden-graph" access structure on n nodes with $O(\log n)$ share size.
- On the negative side, we show that computational *threshold* secret-sharing schemes with public information require share size $\Omega(\log \log n)$. For arbitrary access structures, we show that computational security does not help with 1-bit shares.

The above positive results guarantee that any adversary of size $n^{o(\log n)}$ achieves an $n^{-\Omega(1)}$ distinguishing advantage. We show how to make the advantage negligible by slightly increasing the asymptotic message size, still improving over all known constructions.

The security of our constructions is based on the conjectured hardness of variants of the planted clique problem, which was extensively studied in the algorithms, statistical inference, and complexity theory communities. Our work provides the first applications of such assumptions improving the efficiency of mainstream cryptographic primitives, gives evidence for the *necessity* of such assumptions, and suggests new questions in this domain that may be of independent interest.

1 Introduction

We consider the following broad question about cryptographic primitives:

> Is it possible to achieve security against arbitrary $\mathsf{poly}(n)$-time adversaries with messages of size $O(\log n)$?

It is not hard to see that the answer is "no" unless information-theoretic security is possible. Indeed, a non-uniform adversary can simply apply a brute-force distinguisher implemented by a circuit of size $2^{O(\log n)} = \mathsf{poly}(n)$. A similar argument works for efficient uniform adversaries. In this work, we revisit this question by considering the setting of cryptography with *public information*. Public information may be viewed as a cheap resource: it can often be preprocessed (i.e., generated in an offline phase before the secret inputs are known), it does not require secure storage, and (under strong cryptographic assumptions) can even be generically compressed [HJK+16].

As a concrete example, consider the problem of 2-out-of-n secret sharing. It is known that in any such information-theoretic scheme, even when sharing a 1-bit secret, the bit-length of at least one share must be at least $\log n$ [KN90, CCX13].[1] We ask whether the share size can be reduced, in the computational setting, if the dealer is allowed to publish public information that is generated jointly with the shares. As argued above, this relaxation is *necessary* for breaking the $\log n$ lower bound even in the computational security setting. Moreover, by a simple conditioning argument, public information is not helpful at all in the information-theoretic setting.

We start with a seemingly unrelated observation that if such a 2-out-of-n scheme exists, then (a variant of) the planted clique problem is computationally hard. Specifically, for fixed public information I, we generate a polynomial-size, n-partite graph G whose nodes are all pairs (i, s_i), where $i \in [n]$ and s_i is a possible share for the i-th party (any string of the appropriate length). We put an edge in G between (i, s_i) and (j, s_j) (where $i \neq j$) if the parties i, j on shares s_i, s_j respectively, and with public information I, reconstruct the secret 1. Note that a legal sharing of the secret 1 forms a size-n clique in G between $(1, s_1), \ldots, (n, s_n)$. A legal sharing of the secret 0 forms instead a size-n independent set in G. The security of the secret-sharing scheme implies that if we pick a random secret b and apply the sharing algorithm to get $(I, s_1, \ldots, s_n)$ then, given a node (i, s_i) (corresponding to the view of the i-th party in the secret sharing), it is hard to decide whether it belongs to a size-n clique or a size-n independent set of G defined by the selected I.

The above observation suggests that the hardness of finding planted cliques is necessary for improved 2-out-of-n secret sharing in the public information setting. It is natural to ask whether it is also sufficient. Next, we outline an idea in this direction. This construction does not improve the share size, but it demonstrates in a simple way a high-level idea that we will apply to improve the efficiency of other primitives. First, sample an n-partite random graph, where each part is of size L, and each potential edge between two parts appears with probability 0.5. Then, if the secret is 1, plant in this graph a random n-partite clique (i.e., select one random node from each of the n parts and add to the graph all the edges between them, if they do not already exist); similarly, if the secret is 0, plant in the graph a random n-partite independent set. The resulting graph will be the public information. The share of party i will be the i-th node of the planted clique or independent set. The reconstruction is simple: given two shares (i, s_i) and (j, s_j) the share is determined by whether there is an edge between them in the public graph. For the security of the scheme, we assume that an adversary that sees the graph and gets the share of a party, i.e., a node in a clique or independent set, cannot distinguish between these two cases. Unfortunately, with the above simple planting procedure, the problem can be conjectured to be hard only if $L \geq n$ (see Section 2.1); hence the share size, which is $\log L$, is at least logarithmic.

[1] Here and elsewhere, $\log n$ stands for $\log_2 n$.

Generalizing the above example, in this work we systematically explore the possibility of obtaining computational security with logarithmic-size messages using public information. We show that plausible intractability conjectures about different variants of the planted clique problem, collectively referred to as "planted graph" problems, can be used to construct secret-sharing schemes and secure computation protocols that beat the best-known, and typically the best possible, information-theoretic protocols.

We apply our approach to several different problems. These include private simultaneous messages (PSM) protocols and secure 3-party computation, as well as secret sharing for "forbidden-graph" access structures. For all these primitives, we show how relaxing the standard model by allowing public information can improve over the communication complexity of the best-known solutions, assuming plausible hardness conjectures about planted graph problems. Similar results are not known under any other cryptographic assumptions, or even by using ideal forms of obfuscation. In fact, as in the above examples, assuming the hardness of natural computational problems involving planted graphs can be shown to be necessary. Finally, we also study the extent to which one can go below logarithmic-size messages. For the case of secret sharing, we obtain partial negative results about the access structures that can be realized using computational secret sharing schemes with public information and sub-logarithmic shares.

Different variants of the *planted clique* problem, introduced in [Jer92, Kuč95], were studied within the algorithms, statistical inference, and complexity-theory communities. While such problems have already found some cryptographic applications, these are either in the context of diversifying assumptions [JP00, ABW10, BKR23] or specialized tasks [GKVZ22]. Our work gives the first applications of planted graph problems to improving the efficiency of mainstream cryptographic tasks, and suggests new questions about such problems that may be of independent interest outside the cryptography community.

1.1 Our results

We now give a more detailed account of our results. For each result, we describe the task that we study, the previously known results, and our new results obtained by using hardness assumptions about planted graphs. For a more detailed and technical overview, see Section 2.

1.1.1 PSM protocols with public information

The private simultaneous messages (PSM) model, introduced in [FKN94], is a simple non-interactive model for secure computation in which Alice and Bob can evaluate a function $f(x, y)$ of their inputs by sending a single message to a referee Carol. More concretely, Alice is given a private input x and Bob a private input y. They are both given a common random string r, which is unknown to Carol. Alice and Bob simultaneously send messages to Carol, where each message only depends on the input of the sender and r. Carol should be able to compute $f(x, y)$ from the two messages she receives, but is required to learn no additional information about the inputs x, y. Most of the study of PSM protocols [FKN94, IK97, BIKK14, LVW17, AHMS18] focused on the information-theoretic setting, where the best-known protocols for arbitrary functions $f : [n] \times [n] \to \{0, 1\}$ has communication complexity $O(n^{0.5})$ [BIKK14] and the best known lower bound is $(1 + \Omega(1)) \log n$ [AHMS18].

We start by describing a simple and general method for using symmetric encryption to convert any PSM protocol Π into a computational PSM protocol Π' with public information and short messages. Sample shared randomness r for Alice and Bob in Π, and let the public information of Π' include, for each possible input $x \in [n]$ of Alice, the encryption of Alice's message on (x, r) under some secret key K_x, where the n encrypted messages are cyclically shifted by a random amount r_x. Similarly, the public information includes the n encryptions of the messages of Bob on inputs (y, r), shifted by a random r_y. Then, given the actual inputs x, y, Alice sends to Carol the key K_x and the location of the corresponding encryption (according to the secret shift r_x) and, similarly, Bob sends the key K_y and the location of the corresponding encryption. Carol then decrypts the two messages and computes the output, as in Π. The public information length of Π' is equal to $2n$ times the message length in Π (e.g., $O(n^{1.5})$ using [BIKK14]), and the message length of Π' is $O(\log n + \lambda)$, where λ denotes a security parameter for the underlying encryption scheme. However, to enable security against any $\mathsf{poly}(n)$-time Carol, λ must be super-logarithmic in n.

In this paper, we show how to use a planted graph assumption to construct a PSM protocol with messages of size $O(\log n)$. The PSM protocol proceeds in the following natural way. We first plant a graph obtained from the bipartite graph representing the function f, denoted by H, in a larger random N-node graph to obtain a graph G. The public information consists of G, and the shared randomness (only known by Alice and Bob) is the mapping of all nodes in H to the corresponding nodes in G. On input x, y, Alice and Bob send to Carol the corresponding nodes in G according to this mapping. Carol outputs 1 if and only if there is an edge between the two nodes in G she received.

The security of the protocol relies on the assumption that the planted graph H is hidden within the graph G. This assumption seems to be at least as plausible as the standard planted clique assumption that was studied extensively. More concretely, the flavor of the planted graph assumption that we use asserts that any efficient adversary, who receives a pair of nodes (x, y) in G, cannot distinguish between the case where (x, y) is an edge (resp., non-edge) of the public subgraph H planted in a random G, as above, and the case of a random graph G with a planted edge (resp., non-edge) (x, y) (see Section 2.1 for a detailed discussion of this planted graph assumption and its variants). In quantitative terms, we assume that any adversary of size $n^{o(\log n)}$ has an $n^{-\Omega(1)}$ distinguishing advantage. Under the above assumption, the PSM protocol has the same level of security. Note that we cannot hope for a negligible advantage of $n^{-\omega(1)}$, since a planted graph can be efficiently detected with an inverse-polynomial advantage.

We also present a variant of this construction that makes the adversary's advantage negligible. This comes at the cost of increasing the message length to be an arbitrary function in $\omega(\log n)$ and relying on a stronger assumption. We would like to stress that even when settling for $n^{-\Omega(1)}$ distinguishing advantage, PSM protocols with public information and $O(\log n)$ message size were not known based on any cryptographic assumption.

The above PSM protocols, like other primitives we construct from planted graph assumptions, only achieve conjectured security against (non-uniform) $n^{o(\log n)}$-time adversaries. This is weaker than the typical sub-exponential security achieved under standard cryptographic assumptions but stronger than fine-grained security [Mer78, BRSV18], where security holds against fixed poly-time adversaries. Note that a similar notion of security against quasi-polynomial time adversaries was also considered in other contexts (e.g., [ABW10, BLVW19, BKR23]).

On the plausibility of our assumptions. Our PSM protocols can offer different levels of efficiency depending on the strength of the underlying assumption. In the weaker version of the assumption, dubbed "weak planted subgraph with hints" (Weak-PSH), we assume that for every constant $\delta > 0$ and sequence of n-node graphs H_n, the distinguishing problem is hard when N – the number of nodes in G – is $n^{2+\delta}$; under this assumption the message size is $(2+\delta)\log n$. This assumption is a generalization of the planted clique assumption. On the one hand, we plant an arbitrary graph and not a clique; however, for the known attacks, the clique seems the easiest graph to detect. On the other hand, in our assumption, the adversary gets a "hint" consisting of two nodes from the planted graph H. We show that attacks by low-degree polynomials cannot break the assumption when $N = n^{2+\delta}$. In the domain of planted problems, low-degree polynomials are a powerful class of adversaries that capture known attacks on natural problems. It was even conjectured that, for a well-defined subclass of planted problems, security against low-degree polynomials implies security against general polynomial-time adversaries [Hop18, HW21].

In the stronger version of our assumption, dubbed "planted subgraph with hints" (PSH), we assume that for almost all graphs H the problem is hard when N – the number of nodes in G – is $n^{1+\delta}$ for any constant $\delta > 0$ (rather than $n^{2+\delta}$); under this assumption, the message size is $(1+\delta)\log n$. The evidence we have to support the PSH assumption when $N = o(n^2)$ is that the known attacks for the planted clique problem in this regime do not apply to most other graphs H. For example, consider two simple attacks based on the maximal degree or the total edge count. While planting an n-node clique H in a random N-node G increases the maximal degree and the total number of edges by $\omega(1)$ standard deviations, planting a "typical" H only changes these measures by $o(1)$ standard deviations. Our low-degree analysis provides further evidence: we show that, for any choice of n and N, low-degree attacks are most effective when the hidden graph H is a clique. We also obtain a formula characterising the effectiveness of low-degree attacks in detecting a given hidden graph H, but were not able to formally prove that the PSH assumption holds with respect to low-degree attacks.

Comparison with information-theoretic PSM. By [AHMS18], for almost all $f : [n] \times [n] \rightarrow \{0,1\}$, the message size in a PSM protocol with information-theoretic security is at least $(1+\Omega(1))\log n$ even when the error in the reconstruction and the indistinguishability are $n^{-\Omega(1)}$; for perfect PSM protocols the lower bound of [AHMS18] is $(1.5 - o(1))\log n$. The lower bound of $(1+\Omega(1))\log n$ holds also for computational PSM protocols without public information.[2] Our PSM protocol is much more efficient than the known information-theoretic PSM protocols and computational PSM protocols without public information, however, they do not beat the lower bound of [AHMS18].

1.1.2 Offline-online MPC

We apply the above PSM protocols to obtain offline-online protocols for secure multiparty computation (MPC) in which the online phase is non-interactive and has logarithmic communication. Concretely, consider MPC with 3 semi-honest parties, Alice and Bob who have inputs x and y, respectively, and Carol who has no input and should receive the output $f(x,y)$. We allow an offline stage (not depend on the inputs x, y) which generates correlated randomness to Alice and Bob and some public information. The goal is for the online stage to be non-interactive and highly efficient. That is, each of Alice and Bob sends a single short message to Carol. Based on these

[2] Any unbounded adversary against the PSM protocol can be simulated by a non-uniform polynomial-time adversary, where the adversary holds a polynomial-size table instructing the adversary for every pair of messages m_0, m_1 if to answer 0 (the messages were generated by the parties in the PSM protocol) or to answer 1 (the messages were generated by the simulator).

messages and the public information, Carol computes the output. As far as we know, the prior work that achieves the most efficient online phase in this setting is the one-time truth table protocol from [IKM+13]. This protocol, however, uses at least two rounds of online communication. Our new protocol, in contrast, uses only one online round and, similarly to [IKM+13], for every function $f : [n] \times [n] \to \{0, 1\}$ has message size of $O(\log n)$ bits. Being an application of the above PSM protocol, the MPC protocol relies on the same planted graph assumption.

1.1.3 Forbidden graph secret sharing

For a fixed graph Q with n nodes, a dealer is required to distribute a secret bit b to the n nodes (parties) so that any 2 nodes can reconstruct the secret if and only if they are connected by an edge (there is no additional requirements on sets of size different than 2). Forbidden Graph Secret-Sharing schemes (FGSS) were introduced in [SS97] and further studied in [BIKK14]. The best known information theoretic FGSS scheme has share size $2^{\tilde{O}(\sqrt{\log n})}$ [LVW17]. The best-known computational FGSS scheme (without public information) has share size $\mathsf{poly}(\log n)$, assuming the existence of one-way functions with sub-exponential security [ABI+23]. We show a computational FGSS scheme with public information and share size $O(\log n)$ based on the hardness of deciding whether a *random* graph H appears in a large random graph G with N nodes (both graphs are included in the public information of the FGSS scheme); again we assume that this hardness still holds when the adversary is given a "hint" of two nodes in H.

The security of our FGSS scheme can be based on the "weak planted subgraph with hints" (Weak-PSH) assumption discussed above; in this case $N = n^{2+\delta}$ and the share size is $(2 + \delta) \log n$. However, as we plant a random graph H that is independent of the graph Q representing the access structure, we can use a different assumption, dubbed "planted random subgraph with hints" (PRSH), where we assume that for every constant $\delta > 0$ the distinguishing problem is hard when H is a random n-node graph and N – the number of nodes in G – is $n^{1+\delta}$; under this assumption the share size for every graph Q is $(1 + \delta) \log n$. While the PRSH assumption is implied by the PSH assumption, the converse does not seem to hold. Indeed, in PSH the efficient (non-uniform) distinguisher can depend arbitrarily on the planted graph H, whereas in PRSH a random H is given as input to the distinguisher.

Finally, note that an FGSS scheme for bipartite graphs can be obtained from PSM protocols for general f via a transformation from [BIKK14]. However, a similar result for general graphs requires an extra $\log n$ multiplicative overhead. Our direct construction for FGSS avoids this overhead and moreover relies on a seemingly weaker assumption.

1.1.4 Negative results

We complement the above positive results by some negative results. In Section 8, we show that computational threshold secret-sharing schemes with public information require share size $\Omega(\log \log n)$. Closing the exponential gap between this lower bound and the $\log n$ upper bound given by Shamir's scheme is one of the most interesting questions left open by our work.

In Section 7, we establish a two-way connection between this question and a natural decision problem about planted graphs. Concretely, 2-out-of-n secret-sharing scheme with public information and share size $\delta \cdot \log n$ for $\delta < 1$ is equivalent to the existence of $\beta > 0.5$ and an efficiently samplable joint distribution (G, C, I), where G is an N-node graph, C is an N^{β}-sized clique in G and I is an N^{β}-sized independent set, such that it is hard to distinguish between (G, c) and

(G, i), where c is a random node in C and i is a random node in I. Note that Shamir's scheme implies such a distribution (G, C, I) with perfect indistinguishability when $\beta = 0.5$, and negative results for information-theoretic threshold secret sharing [KN90, CCX13] imply that even statistical indistinguishability is impossible for $\beta > 0.5$.

Finally, in Section 9, we show that, when considering secret-sharing schemes with *one-bit shares*, all access structures that can be realized with computational security with public information can also be realized information-theoretically.

1.1.5 Summary and open questions

A summary of our main positive and negative results is presented in Table 1 below.

	Information Theoretic		**Computational with Public Information**	
	Bound	*Ref.*	*Bound*	*Assumption & Ref.*
PSM	$\leq \sqrt{n}$ $\geq (1 + \Omega(1)) \cdot \log n^{\dagger}$	[BIKK14] [AHMS18]	$\leq 1.01 \cdot \log n^{\dagger}$ $\geq \log n$	PSH (Thm. 5.3)
Forbidden Graph Secret Sharing	$\leq 2^{\tilde{O}(\sqrt{\log n})}$ $\geq \log n$	[LVW17] [KN90, CCX13]	$\leq 1.01 \cdot \log n$ $\geq \frac{1}{5} \log \log n$	PRSH (Thm. 6.6) (Thm. 8.1)
2-out-of-n Secret Sharing	$\leq \log n$ $\geq \log n$	[Sha79] [KN90, CCX13]	$\leq \log n$ $\geq \frac{1}{5} \log \log n$	[Sha79] (Thm. 8.1)
Non-ideal Binary Secret Sharing	> 1	(by def.)	> 1	(Thm. 9.2)

[†] Both bounds hold for a $1 - o(1)$ fraction of the functions $f : [n] \times [n] \to \{0, 1\}$. In the case of *perfect* PSM protocols, the lower bound from [AHMS18] is $(1.5 - o(1)) \cdot \log n$.

Table 1: Bounds on the complexity of 2-party PSM protocols, n-party forbidden graph secret-sharing schemes, and 2-out-of-n secret-sharing schemes for the information-theoretic case and the computational case with public information. The values refer to constructions with perfect correctness and privacy error $\varepsilon = n^{-\Omega(1)}$ against non-uniform $n^{o(\log n)}$-time (resp., unbounded) adversaries in the computational (resp., information-theoretic) case. The complexity is defined as the maximum message size (resp., share size) for a single party. The PSH and PRSH assumptions are informally described in Section 2.1. The super-constant lower bounds ignore constant additive terms.

Our results leave several open questions about the succinctness of computationally secure PSM protocols and secret-sharing schemes with public information.

- **Stronger notions of indistinguishability.** Our positive results for PSM protocols and secret-sharing schemes with $O(\log n)$ message or share size are limited in two ways. First, security only holds against $n^{o(\log n)}$-time adversaries (instead of the typical subexponential $2^{n^{o(1)}}$ time). Second, the distinguishing advantage of such adversaries is only $n^{-\Omega(1)}$ (instead of the typical negligible $n^{-\omega(1)}$), where we can only make the advantage negligible at the price of a super-constant multiplicative overhead. The possibility of removing one of these limitations or both is left open.

- **Share size of threshold secret sharing.** What is the minimal share size of computationally secure 2-out-of-n secret sharing with public information? Is it possible to beat the information-theoretic $\log n$ bound, even by a constant factor? We were only able to prove an $\Omega(\log\log n)$ lower bound, and showed the equivalence of improving the upper bound to planting both a large clique and a large independent set in the same graph such that it is hard to distinguish a random node in the clique from a random node in the independent set.

- **Computational-statistical gaps for secret sharing with small shares.** We showed that for secret sharing with 1-bit secrets, 1-bit shares and public information, settling for computational security does help realize additional access structures beyond the ones realized by (ideal) information-theoretic schemes. Is this also the case for domain size 3? Note that with quasi-polynomial domain size, computational-statistical gaps were recently shown to exist based on the existence of a (subexponentially secure) one-way function [ABI+23].

2 Overview of Techniques

This paper studies the relation between cryptographic primitives, such as PSM protocols and secret-sharing schemes, and *planted subgraph problems*.

2.1 Planted subgraph assumptions

Suppose that G and H are graphs with N and n nodes respectively, where $N > n$. The operation of planting H into G consists in selecting a random subset S of n nodes in G and modifying the edges so that the subgraph induced by S is isomorphic to H. In other words, we are hiding a copy of H inside G. We are particularly interested in the case in which G is an Erdős-Rényi random graph, i.e. each edge is independently drawn with probability 1/2. We denote its distribution by $\mathcal{G}(N, 1/2)$.

We analyse three main subfamilies of assumptions: planted clique (PC), planted subgraph (PS), and planted subgraph with hints (PSH). The first one has a long history: it was introduced in the '90s [Jer92, Kuč95] and has been studied since then [AKS98, FK03, Ros08, Ros10, FGR+13, BHK+16, ABdR+18, MRS21]. The other two assumptions are introduced for the first time in this work. We describe them below.

The planted clique (PC) assumption. The PC assumption states that a random graph with a large planted clique looks random. Formally, for an appropriate choice of parameters N, T, and ε, it claims that, for every non-uniform $T(n)$-time adversary $\mathcal{A}$,

$$\left|\Pr\left[\mathcal{A}(G) = 1 \middle| G \xleftarrow{\$} \mathcal{G}(N, 1/2, n)\right] - \Pr\left[\mathcal{A}(G) = 1 \middle| G \xleftarrow{\$} \mathcal{G}(N, 1/2)\right]\right| \leq \varepsilon(n).$$

Above, $\mathcal{G}(N, 1/2, n)$ denotes the distribution that plants an n-node clique in a random N-node graph.

The assumption was independently introduced by Jerrum [Jer92] and Kučera [Kuč95] and has been studied since then. The hardness of the problem is supported by the NP-hardness of finding, or even approximating, the largest clique in a graph [Kar72, Hås96a].

Trivial attacks, such as counting the number of edges in G, break the assumption for any $\varepsilon = \mathsf{negl}(n)$. However, the assumption is believed to hold against non-uniform polynomial time adversaries when $\varepsilon = n^{-c}$ for a constant $c > 0$, and N is sufficiently large. Indeed, all the known attacks fail when $N = \omega(n^2)$ [Kuč95, AKS98, DM15a, CX16]. In this parameter setting, the assumption is also supported by many results proving hardness against particular classes of adversaries [FK03, Ros08, FGR+13, GS14, BHK+16, ABdR+18, FGN+20]. Finally, concerning the computational power of the attacker, it is known that $n^{O(\log n)}$-time algorithms can detect the planted clique with $\Theta(1)$ advantage [HK11]. This leads to the following conjecture.

Conjecture 2.1 (PC – Informal)**.** *For any constant $\delta > 0$, there exists a constant $0 < c < 1/2$ such that the PC assumption holds with $N = n^{2+\delta}$ and $\varepsilon = n^{-c}$ against all non-uniform $n^{o(\log n)}$-time adversaries.*

We refer to Section 4.1 for a more rigorous discussion about this assumption.

The planted subgraph (PS) assumption. The PS assumption generalizes what we described above: instead of hiding a clique in a random graph, we hide an n-node subgraph H coming from some distribution $\mathcal{D}$. The assumption asserts that the resulting graph looks random even when H is revealed. The concept is formalized similarly to the PC problem: for every non-uniform T-time adversary $\mathcal{A}$,

$$\left| \Pr\left[\mathcal{A}(G, H) = 1 \middle| \begin{matrix} H \xleftarrow{\$} \mathcal{D} \\ G \xleftarrow{\$} \mathcal{G}(N, 1/2, H) \end{matrix} \right] - \Pr\left[\mathcal{A}(G, H) = 1 \middle| \begin{matrix} H \xleftarrow{\$} \mathcal{D} \\ G \xleftarrow{\$} \mathcal{G}(N, 1/2) \end{matrix} \right] \right| \leq \varepsilon.$$

Above, $\mathcal{G}(N, 1/2, H)$ denotes the distribution that plants H in a random N-node graph.

We are particularly interested in two variants of the PS assumption: the case in which $\mathcal{D}$ is deterministic and the case in which the $\mathcal{D}$ outputs a random n-node graph. We refer to the latter as the *planted random subgraph* (PRS) assumption.

It is generally believed that breaking the PS assumption is easiest when $\mathcal{D}$ deterministically outputs an n-node clique. For instance, the successful attacks against the PC problem leverage the particular structure of cliques. If we plant a generic subgraph, these algorithms do not perform as well. It is therefore conjectured that, for an overwhelming fraction of subgraphs H, the PS assumption holds for $\mathcal{D} \equiv H$[3] with parameters $T = n^{o(\log n)}$ and $\varepsilon = n^{-c}$ even when $N = n^{1+\delta}$ (planted cliques needed $N = n^{2+\delta}$). This implies that the PRS assumption holds with similar parameters. We refer to Section 4.2 for more details.

The planted subgraph with hints (PSH) assumption. The PSH assumption is a variant of the PS assumption in which the adversary is provided with hints: we reveal where we hid a subset S of nodes of the planted subgraph. The size of S is bounded by a parameter t. Usually, t is small, e.g., $t = 2$. Clearly, after revealing the hints, the graph does not look random anymore: the adversary notices that G hides the subgraph induced by S. The PSH assumption claims, however, that the adversary cannot tell if G hides the whole graph H or just the subgraph induced by S.

[3] We use $\mathcal{D} \equiv H$ to denote the distribution that always outputs the subgraph H.

Formally, for any subset S with fewer than t nodes and every non-uniform T-time adversary $\mathcal{A}$,

$$\left|\Pr\left[\mathcal{A}\big(G_b, H, (u_i^b)_{i\in S}\big) = b \,\middle|\, \begin{array}{l} b \xleftarrow{\$} \{0,1\} \\ H \xleftarrow{\$} \mathcal{D},\ H' \leftarrow \mathsf{Subgraph}(H,S) \\ \big(G_1, (u_i^1)_{i\in S}\big) \xleftarrow{\$} \mathcal{G}(N, 1/2, H, S) \\ \big(G_0, (u_i^0)_{i\in S}\big) \xleftarrow{\$} \mathcal{G}(N, 1/2, H', S) \end{array}\right] - \frac{1}{2}\right| \le \varepsilon.$$

Above, $\mathcal{G}(N, 1/2, H, S)$ denotes the distribution that plants H in a random N-node graph and reveals where the nodes in S are hidden. The algorithm $\mathsf{Subgraph}(H, S)$ outputs instead the subgraph of H induced by S. When $\mathcal{D}$ outputs a random n-node graph, we refer to the assumption as *PRSH* (planted random subgraph with hints).

It is believed that revealing $t = O(1)$ nodes on the planted subgraph does not affect the security of the assumption. This leads to the following conjectures.

Conjecture 2.2 (Weak-PSH, PSH, PRSH – Informal)**.**

- ***(Weak-PSH).*** *Let $(H_n)_{n\in\mathbb{N}}$ be a family of n-node graphs. For any constants $\delta > 0$ and $t \in \mathbb{N}$, there exists a constant $0 < c < 1/2$ such that the PSH assumption holds for $\mathcal{D} \equiv H_n$ with $N = n^{2+\delta}$, t leaked nodes, and $\varepsilon = n^{-c}$ against all non-uniform $n^{o(\log n)}$-time adversaries.*
- ***(PSH).*** *For any constants $\delta > 0$ and $t \in \mathbb{N}$, there exists a constant $0 < c < 1/2$ such that the PSH assumption holds for most $\mathcal{D} \equiv H_n$ with $N = n^{1+\delta}$, t leaked nodes, and $\varepsilon = n^{-c}$ against all non-uniform $n^{o(\log n)}$-time adversaries.*
- ***(PRSH).*** *For any constants $\delta > 0$ and $t \in \mathbb{N}$, there exists a constant $0 < c < 1/2$ such that the PRSH assumption holds with $N = n^{1+\delta}$, t leaked nodes, and $\varepsilon = n^{-c}$ against all non-uniform $n^{o(\log n)}$-time adversaries.*

Note that the PRSH assumption is seemingly more conservative than the PRH assumption in that it requires the same efficient distinguisher to apply to almost every H_n, whereas in PRH the distinguisher can depend arbitrarily on H_n.

Also note that all the variations of planted problems we considered above are statistically hard only when $n = O(\log N)$ (e.g., the largest clique in a random N-node graph has $O(\log N)$ size [BE76]). In this parameter regime, our constructions would be outperformed by known information-theoretic upper bounds [BIKK14, LVW17].

We provide evidence supporting our conjectures: we show that the Weak-PSH assumption holds against any adversary that can be represented as a degree-D multivariate polynomial, where $D = (\log n)^{2-\varepsilon}$ and $\varepsilon > 0$ (a *low-degree polynomial*). We also show that, independently of N and the number of hints, cliques are the planted subgraph that are most easily detected by low-degree polynomials. In the domain of planted problems, low-degree polynomials are a powerful class of adversaries. For instance, all known attacks against the planted clique problem belong to this class. For this reason, it was even conjectured that, for planted problems, security against degree-D polynomials implies security against generic 2^D-time adversaries [Hop18, HW21].

In our low-degree analysis, we derive a formula describing the effectiveness of low-degree attacks in detecting a given planted subgraph H:

$$\sum_{0<w(\alpha)\le D} \binom{N - V(\alpha)}{n - V(\alpha)}^2 \cdot \sum_{\pi\in\mathsf{Sym}(n)} (-1)^{\langle \pi\circ H + H, \alpha\rangle}$$

The lower the above value, the harder detecting H becomes for degree-D polynomials. In the above formula, we represented the graphs H and $\pi \circ H$ as $\binom{n}{2}$-bit vectors encoding their adjacency matrix. We use $w(\alpha)$ to denote the Hamming weight of the $\binom{n}{2}$-bit vector α (which can be regarded also as a n-node graph). We use $\pi \circ H$ to denote the action of any permutation $\pi \in \mathsf{Sym}(n)$ on the nodes of H. Finally, $V(\alpha)$ denotes the number of non-isolated nodes in the graph encoded by α. We observe that our formula reaches the maximum when H is a clique: in that case, $\pi \circ H + H = 0$ for any $\pi \in \mathsf{Sym}(n)$. For generic, less symmetrical graphs H, we expect the inner product $\langle \pi \circ H + H, \alpha \rangle$ to assume the values 0 and 1 almost equally often over the choice of $\pi \in \mathsf{Sym}(n)$. That brings the sum $\sum_{\pi \in \mathsf{Sym}(n)} (-1)^{\langle \pi \circ H + H, \alpha \rangle}$ closer to 0. We refer to Section 4.3 for further details.

2.2 PSM protocols with logarithmic message size

We use the (Weak-)PSH conjecture to build a computational PSM protocol with $O(\log n)$ message size.

Private simultaneous messages protocols with public information. PSM protocols are a cryptographic primitive that specifies how two parties can simultaneously encode their inputs (each encoding only depends on the input of the party and a common random string) and non-interactively evaluate from the encodings a function f on the parties' inputs. An external observer that only sees the encoding of the inputs is guaranteed to learn no information beyond the output of the function.

We consider a computational version of the primitive in which a setup is used to generate common randomness for the parties (which is kept secret) along with some public information I. The latter is necessary for the reconstruction of the output, however, it does not help in learning additional information about the inputs.

We highlight that PSM protocol always needs an algorithm that sets up the randomness of the parties, no matter what[4]. The main novelty in this work is that we allow some information to be public. Since we are considering security in a computational setting, public information can help in decreasing the size of the message of the parties: I can hide all the information about the function f and its outputs. By revealing the encodings of their inputs x, y, the parties can make the extraction of $f(x, y)$ from I easy, while all other information remains secret. This is exactly the blueprint used by our constructions. In the paper, we focus our attention on functions of the form $f : [n] \times [n] \to \{0, 1\}$.

A trivial construction from OWFs. Before using techniques based on planted subgraphs, we linger for a moment on the notion of PSM protocols with public information and we check what can be achieved using more standard cryptographic primitives.

We can obtain a trivial construction using OWFs. Represent the function f as an $n \times n$ truth table T in which each row is associated with an input of the first party and each column is associated with an input of the second party. We permute all the rows and all the columns of T independently using permutations ϕ_0 and ϕ_1. Let T' be the result. For every i, we encrypt all the elements in the i-th row of T' using a key k_i^0. We then perform a similar operation on the already encrypted

[4] If the parties use independent randomness, an adversary can run a *residual function attack*. Check Section 5.1for more details.

matrix switching to columns: for every j, we encrypt all the elements in the j-th column using the key k_j^1. The public information I will consist of the resulting doubly-encrypted matrix.

In order for the parties to evaluate f on input x and y, they just need to send $(x', k_{x'}^0)$ and $(y', k_{y'}^1)$, where $x' = \phi_0(x)$ and $y' = \phi_1(y)$. The output is obtained by decrypting the element in position (x', y') in I using the keys sent by the parties. Observe that even if we assume exponentially secure OWFs and we opt for security against $n^{o(\log n)}$-time adversaries, this construction requires $\Omega(\log^2 n)$ message size.

PSM protocols with public information from PSH. Using planted subgraphs, we obtain a PSM protocol with public information where the message size is nearly optimal: under the PSH conjecture, for most functions $f : [n]\times[n] \to \{0,1\}$, the parties just need to communicate $(1+\delta)\cdot\log n$ bits where δ is an arbitrarily small positive constant. Under the Weak-PSH conjecture, we achieve instead $(2+\delta)\cdot\log n$ message size for all functions. Observe that there is an information-theoretic lower bound that requires at least $\log n$ bits of communication. Importantly, our construction achieves security against $n^{o(\log n)}$-time adversaries with *inverse-polynomial privacy error.*

The construction is rather simple: we represent the function f as a bipartite graph H with n nodes per part. Each node on the left will be associated with a different input for the first party. Similarly, each node on the right will be associated with a different input for the second party. We connect two nodes with an edge if the evaluation of f on the corresponding values gives 1. The public information will consist of a large random graph G in which we plant a copy of H. The setup will provide the parties with the position of the hidden subgraph. In order to evaluate the function, all the parties need to do is to reveal where the node associated with their input is hidden. The output of the function is 1 if and only if there is an edge connecting the broadcast nodes.

Under the PSH assumption with $t = 2$, the view of an external observer is as if it was given a random graph with a planted edge (if the output is 1) or a planted "non-edge" (if the output is 0). So, no information about the inputs is revealed beyond the result of the evaluation.

Theorem 2.3 (Informal). *Under the PSH conjecture for $t = 2$, for most functions $f : [n] \times [n] \to \{0,1\}$, the construction described above is a PSM protocol with public information that is secure against non-uniform $n^{o(\log n)}$-time adversaries with $\varepsilon = n^{-c}$ privacy error. The message size is $(1+\delta)\cdot\log n$ for a small positive constant δ.*

Under the Weak-PSH conjecture for $t = 2$, the construction is secure against the same class of non-uniform adversaries for every function $f : [n]\times[n] \to \{0,1\}$ and achieves $(2+\delta)\cdot\log n$ message size.

Privacy amplification. The disadvantage of the construction we just described is the inverse-polynomial privacy error ε. We therefore tried to amplify it to $\varepsilon = \mathsf{negl}(n)$. Unfortunately, techniques such as Yao's XOR lemma, do not seem to help. Another possible approach would have been the technique used in [BGIK22]. This solution, however, would have increased the message size to $\Omega(\log^2 n)$. We recall that the trivial solution from OWF achieves exactly this complexity.

In the end, we came up with a candidate construction that we believe to achieve negligible privacy error against non-uniform $n^{o(\log n)}$-time adversaries with $\omega(1)\cdot\log n$ message size. The idea is rather simple: we additively secret share the function f among $r = \omega(1)$ virtual parties. As we did for f in the previous paragraph, we can represent each share g_j as a $2n$-node graph H_j. The public information will consist of a vector $I = (G_1, \ldots, G_r)$ where G_j is a random N-node graph in which we planted H_j.

In order to evaluate the function, the parties encode their inputs as in the original construction with respect to each graph G_j. In particular, the parties reveal where the node associated with their input is hidden in G_j. For every $j \in [r]$, the parties obtain a different output bit z_j (z_j will be equal to 1, if the broadcast nodes in G_j are adjacent). By XORing all these values, they reconstruct the output of the evaluation.

To support our claim of security, observe that an adversary cannot learn where H_j is hidden by solely looking at G_j: it has to work on the joint distribution $(G_1, \dots, G_r)$. Indeed, each H_j is secret and uniformly distributed, so G_j is just a random graph. The natural attack would require the adversary to find a permutation of the graphs $G_1, \dots, G_r$, so that their "XOR" hides a copy of f[5]. However, only a negligible fraction of all permutations satisfies the desired property. In Section 5.3, we consider more sophisticated attacks.

Offline-online 2-input non-interactive 3-PC with logarithmic communication. Our PSM protocols give rise to very lightweight 2-input 3-party protocols with an offline phase. Our setting is the following: suppose that Alice and Bob have some input $x, y \in [n]$. After receiving some correlated randomness from a trusted dealer, in the so-called *offline phase*, they want to reveal the evaluation of a function $f : [n] \times [n] \to \{0, 1\}$ on their inputs to their friend, Carol. Carol should be the only one who learns such output. In our setting, Alice and Bob are, however, lazy: they want to send a single immediate message that is as short as possible.

PSM protocols with public information are the solution to this problem: the trusted dealer provides the common randomness to Alice and Bob and the public information to Carol. At that point, Alice and Bob independently encode their inputs using the PSM protocol and send their messages to Carol. The public information allows Carol to retrieve the output.

The construction withstands a semi-honest adversary that corrupts at most one party. Observe that the online phase requires a single round of interaction. Furthermore, our PSM protocols allow us to achieve $\omega(1) \cdot \log n$ communication. To our knowledge, the only solution that achieves lower communication complexity is the one-time truth table protocol of [IKM+13]. Such solution would, however, require more than one round of interaction.

Compressing the public information. In this work, we decreased the message size of PSM protocols by introducing public information. A natural question is how big the public information needs to be and whether this can be reused (e.g., the construction based on graphs cannot be used more than once).

A partial answer is given by *universal samplers* [HJK+16]. This primitive can be thought of as a small public obfuscated program that, on input the description of a distribution $\mathcal{D}$, outputs a sample from $\mathcal{D}$ without revealing any additional information about it. For instance, if $\mathcal{D}$ produces large random RSA moduli, nobody will learn the factorisation of the output of the universal sampler.

Now, suppose that a trusted dealer provides the parties of the PSM protocol with a key pair $(\mathsf{pk}, \mathsf{sk})$ and a universal sampler U. Everybody can evaluate U on input the distribution that generates the PSM public information I and encrypts the common randomness under pk. Everybody is able to retrieve I, but only the PSM participants can recover the common randomness using sk [ASY22].

The universal samplers presented in [HJK+16] set an upper-bound L on the size of the distributions that can be evaluated. In particular, the size of the sampler is $\mathsf{poly}(\lambda, L)$ where λ is a

[5] We "XOR" two graphs by XORing their adjacency matrices.

security parameter. In these constructions, the size of U would therefore be greater than the one of I. Using a sampler has nevertheless an advantage: if we rely on a programmable random oracle, U can be reused without limits. In other words, universal samplers allow us to compile a single-use PSM protocol into a reusable one.

The good news is that the issue with sizes can be fixed: Abram, Obremski and Scholl [AOS23] built an *unbounded universal sampler* (again, using a programmable random oracle). This is a universal sampler that sets no bound on the size of the distributions that can be given as input. The size of U is simply $\mathsf{poly}(\lambda)$. Notice that if we aim for security against $n^{o(\log n)}$-time adversaries, the size of the sampler is $\mathsf{polylog}\, n$.

We formalize our results about PSM protocols in Section 5.

2.3 Forbidden graph secret sharing with logarithmic share size

We use the PRSH assumption to build forbidden graph secret-sharing schemes with $O(\log n)$ share size.

Forbidden graph secret-sharing schemes. A secret-sharing scheme consists of a primitive that allows the sharing of a secret among n parties. In order to reconstruct the secret, the participants need to collaborate. Whether the reconstruction succeeds or not depends on the set of players that collaborate: some subsets are guaranteed to succeed, some of them are guaranteed to learn no information about the secret, some of them have no guarantee (they may get the whole secret, just some leakage or nothing at all). These reconstruction policies are described by the so-called *access structure*.

We are interested in a particular version of primitive called *forbidden graph secret-sharing schemes* [SS97]: the access structure is described by an n-node graph Q. Each party is associated with a different node. A pair of players is guaranteed to reconstruct the secret if and only if there is an edge connecting their nodes. If such edge does not exist, they learn no information about the secret. Finally, if a subset of more than 2 parties collaborates, the construction gives no guarantee on whether the secret can be recovered.

Secret-sharing schemes with public information. Similarly to what we did for PSM protocols, we consider security against computational adversaries and we augment the primitive with public information: in order to secret-share a value x, a player will broadcast large public information I along with small shares $s_1, \ldots, s_n$, one for each party. The public information will be necessary to reconstruct the secret, however, it will not help in learning x. Since we are in a computational setting, the public information can help in decreasing the size of the shares.

This version of the primitive is motivated by the fact that, in many settings, the cost of storing private information is higher than the one for public information. Moreover, in this kind of schemes, the reconstruction of the secret requires minimal communication. This is even more interesting whenever the public information is reusable.

A trivial construction from OWFs. Similarly to the case of PSM protocols, before presenting our solution based on graphs, we linger for a moment on the definition and we try to check what can be achieved using already known primitives. We can consider a forbidden graph secret-sharing scheme in which the share of each party P_i consists just of a λ-bit key k_i for a symmetric encryption scheme. The public information consists instead of a list of n ciphertexts, the i-th one of which is an encryption under k_i of the i-th share of an information-theoretic forbidden-graph secret-sharing (e.g. [BIKK14]) of our secret. It is trivial to see that this scheme is secure. If we opted for security against $n^{o(\log n)}$-time adversaries, the share size would be at least $\log^2 n$.

Forbidden graph secret-sharing schemes with public information from our PSM protocol. Beimel et al. [BIKK14] showed how to construct a forbidden graph secret-sharing scheme from a PSM protocol, increasing the share size by a factor of $O(\log n)$. Thus, our PSM protocol with public information implies, using the Weak-PSH assumption, a forbidden graph secret-sharing scheme with public information having $O(\log^2 n)$ share size and inverse-polynomial distinguishing advantage.

Forbidden graph secret-sharing schemes with public information from PRSH. Using planted subgraphs, we directly obtain a forbidden-graph secret-sharing scheme with public information in which the share size is $O(\log n)$. Under the PRSH conjecture, we obtain $(1 + \delta) \cdot \log n$ share size where δ is a small positive constant. Under the Weak-PSH conjecture, the complexity becomes instead $(2 + \delta) \cdot \log n$. Importantly, our construction achieves security against non-uniform $n^{o(\log n)}$-time adversaries with inverse-polynomial privacy error. Our direct construction reduces the share size and uses a weaker assumption compared to the construction using the PSM protocol.

The construction works as follows: we sample a random n-node graph H and we plant it in a larger random graph G. Each node in H is associated with a different party. Next, we modify H: we compare it to the graph access structure Q. For any edge that does not appear in Q, we remove the corresponding edge in H (if such edge exists). Let H' be the graph obtained in this way. In order to secret-share $b = 1$, we publish the pair (H', G) and we provide each party with the position of its node in G. In order to secret-share $b = 0$, we perform the same operations except that we publish $(H', \overline{G})$ where $\overline{G}$ is the complementary graph of G (i.e., $\overline{G}$ will have all the edges that do not appear in G).

If a pair of parties is allowed to reconstruct, they can recover b by just comparing the edge that connects their nodes in H' with the edge that connects their shares in G. If both edges exist or both do not, the secret is 1. Otherwise, it is 0.

Observe that under the PRSH assumption with $t = 2$, all the information the parties see in G is the edge (or non-edge) that connects their shares. All the rest looks random. If the pair is not allowed to reconstruct the secret, their edge in G will be independent of the graph H' (their edge was removed from H).

Theorem 2.4 (Informal)**.** *Under the PRSH conjecture for $t = 2$, the construction described above is a forbidden-graph secret-sharing scheme with public information that is secure against non-uniform $n^{o(\log n)}$-time adversaries with $\varepsilon = n^{-c}$ privacy error. The share size is $(1 + \delta) \cdot \log n$ for a small positive constant δ.*

In the context of secret sharing, amplifying privacy to a negligible error is easy. We just need to apply Yao's XOR lemma with $r = \omega(1)$ repetitions. The share size becomes therefore $\omega(1) \cdot \log n$ (we recall that the trivial VBB solution requires $\log^2 n$ share size).

Theorem 2.5 (Informal)**.** *Under the Weak-PSH conjecture, for every graph access structure, there exists a forbidden graph secret-sharing scheme with public information, a one-bit secret, and $\omega(1) \cdot$*

$\log n$ *share size. The scheme is secure against non-uniform* $n^{o(\log n)}$*-time adversaries with* $\varepsilon = \mathsf{negl}(n)$ *privacy error.*

Compressing the public information. Similarly to PSM protocols, we can use universal samplers to compress the public information and make it reusable. The technique requires the use of a programmable random oracle.

Suppose that a trusted setup provides the parties with an unbounded universal sampler U. Suppose also that each party P_i is associated with a key pair $(\mathsf{pk}_i, \mathsf{sk}_i)$. In order for P_1 to share a bit b, the players can run U on input the distribution that generates the secret-sharing of a random bit c and outputs the public information, the encryption of the share s_i under pk_i for every i and the encryption of c under pk_1. Each party can retrieve its share, P_1 also learns the random secret c. At that point, P_1 can simply broadcast $b \oplus c$. Observe that b can be recovered if and only the parties are able to reconstruct c.

This solution decreases the size of the public information and makes it reusable. A minor disadvantage is that the size of the private information stored by each party increases as the size of sk_i is at least λ bits where λ is a security parameter. The cost of such storage is however amortized over many executions of the secret-sharing scheme. Notice that the communication complexity of the reconstruction is as before: the parties just need to communicate $\omega(1) \cdot \log n$ bits.

We formalize our results about forbidden-graph secret-sharing schemes in Section 6.

2.4 On breaking the $\log n$ barrier for 2-out-of-n secret sharing

Unlike PSM protocols, in the context of secret-sharing schemes with public information, there is no obvious lower bound on the share size. In particular, we do not know whether there are schemes with $\delta \cdot \log n$ share size for any $\delta < 1$. We studied this question in the simplest setting: 2-out-n secret-sharing schemes. Unfortunately, we could not find an answer, however, we came up with necessary and sufficient conditions for this to happen.

From secret-sharing to graphs. We show that 2-out-of-n secret-sharing schemes with public information and share size ℓ are equivalent to a multipartite version of the planted clique problem: given the public information I, we can derive an n-partite graph with 2^ℓ nodes per part. Each of the nodes in the i-th partition corresponds to a different share for party P_i. We connect all the pairs of nodes that correspond to shares that reconstruct to 1.

By the correctness of the secret-sharing scheme, if the public information hides the secret $b = 1$, the graph we derived hides an n-node clique (the nodes containing the shares of the n parties with the secret 1 and a random string of the dealer generating the public information I). If instead the secret is $b = 0$, the graph hides an n-node independent set. Independently of the secret, each of the nodes in the hidden subgraph lies on a different part. The security of the 2-out-of-n secret-sharing scheme guarantees that the two distributions on graphs are indistinguishable even if we leak one of the nodes in the hidden subgraphs.

The above argument can be reversed to show that distributions over graphs with the described properties imply a 2-out-of-n secret-sharing scheme with public information. Finding them is however not simple when $\ell < \log n$. Indeed, we would need to hide an n-node clique in a graph that has less than n^2 nodes. In this parameter setting, the attacks of [Kuč95, AKS98] succeed in recovering the clique for all the graph distributions we tried. The multipartite nature of the graph seems to make the goal even harder.

A cleaner necessary and sufficient condition. By using a random partitioning argument, we can further simplify the above characterization and apply it to general, rather than multipartite, graphs. We look for a distribution $\mathcal{D}$ over N-node graphs G that contain both a (not necessarily unique) N^β-node clique and an N^β-node independent set, where $\beta < 1$ is a constant. We would like that, given G, it is infeasible to distinguish between a random node in the clique and a random node in the independent set. We prove that this problem is equivalent to 2-out-of-n secret sharing: the distribution $\mathcal{D}$ is possible for some $\beta > 1/2$ if and only if there is a 2-out-of-n secret-sharing schemes with $\delta \cdot \log n$ share size for some $\delta < 1$.

Theorem 2.6 (Planted subgraph formulation of 2-out-of-n secret sharing – Informal)**.** *The following are equivalent:*

- *There exists a constant $0 < \delta < 1$ for which there is a computational 2-out-of-n secret-sharing scheme with public information and $\delta \cdot \log n$ share size.*
- *There exists a a constant $1/2 < \beta < 1$ and a distribution $\mathcal{D}$ of triples (G, C, I), where G is an N-node graph, C is an N^β-node clique in G and I is an N^β-node independent set in G, such that it is hard to distinguish between (G, c) and (G, i), where c and i are random nodes in C and I respectively.*

We formalize this in Section 7.

Remark 2.7 (On search vs. decision). While the above condition has the flavour of a planted subgraph problem, it is different from traditional planted problems in the following way. In the traditional case, the planting is done in a way that guarantees (with high probability) that the planted object occurs only once. Thus, the natural search version of the problem is to find the single instance of this object. Here, the distribution $\mathcal{D}$ can be such that every node is involved in potentially many planted cliques and independent sets. In fact, for the multipartite version of the problem, there is an n-partite graph with only 4 nodes in each part such that every node is involved in both a clique and an independent set of size n [BF07]. Thus, it is not clear how to define a natural search problem. On the other hand, the information-theoretic impossibility of beating the $\log n$ share size [KN90, CCX13] implies that the above *decision* problem can be solved by a computationally unbounded distinguisher.

A lower bound on the share size. We prove a lower bound for 2-out-of-n secret-sharing schemes with public information: the share size needs to be at least $\frac{1}{5} \log \log n$.

The idea is rather simple: a 2-out-of-n secret-sharing scheme induces a 2-out-of-n' scheme for any $n' \leq n$. The security of the construction does not depend on n' but only on n. On the other hand, the size of the public information I decreases with n'. Indeed, as we discussed above, I can be represented as an n-partite graph. If we restrict the scheme to n' parties, we just need to consider n' of the parts.

Now, if the share size ℓ is smaller than $\frac{1}{5} \log \log n$, there exists an $n' > 2^\ell$ for which the size of the public information becomes $O(\log n)$. Such public information is too small to help against $\mathsf{poly}(n)$-time adversaries. Therefore, it must be that the induced scheme is statistically secure.

Lower bounds for the information-theoretic case require that $\ell \geq \log n'$. That contradicts our choice of n'.

We formalize the lower bound in Section 8.

On the relation between our primitives and planted subgraph problems. The discussion about breaking the $\log n$ barrier for 2-out-of-n secret-sharing schemes highlighted an important point: planted subgraph assumptions are not only sufficient to obtain PSM protocols and forbidden graph secret-sharing schemes with $O(\log n)$ share size, they are also necessary.

For instance, consider a function $f : [n] \times [n] \rightarrow \{0, 1\}$ and let H be the corresponding graph representation. We can reframe the security of any PSM protocol for f with $O(\log n)$ message size as a planted subgraph problem: we create a bipartite graph G as follows. Each node on the left side corresponds to a different message the first party can send. Similarly, each node on the right side corresponds to a different message for the second party. We connect any pair of nodes with an edge if the corresponding PSM messages give output 1. It is easy to see that the graph hides at least one copy of H. Breaking the security of the protocol roughly corresponds to recognising which nodes of H were broadcast by the parties. Here too, the fact that H is not necessarily unique requires formulating this using the joint distribution of G and H.

We can use an analogous argument to show that also forbidden graph secret-sharing schemes with $O(\log n)$ share size can be reframed as a planted subgraph problem.

Secret-sharing schemes with 1-bit shares. We study the following scenario: employing public information, when can we construct secret-sharing schemes with one-bit shares?

If an n-party gap access structure has at least $\omega(\log n)$-gap between the size of every qualified set and the size of every forbidden set, using virtual black box obfuscation (VBB), we can construct a secret-sharing scheme with one-bit shares as follows: The dealer with secret s distributes an independently and uniformly chosen bit r_i to each party P_i and publishes a VBB obfuscation of the function that, when queried with $Q, (r_i)_{i \in Q}$ for any qualified set Q, outputs s and outputs $\perp$ otherwise. Then, a computationally bounded adversary with shares $(r_i)_{i \in F}$ needs to correctly guess $\omega(\log n)$ random bits to recover the share, hence succeeds with negligible probability.

However, for a perfect access structure, where every set is either qualified or forbidden, we show that a secret-sharing scheme with one-bit shares even with public information achieves less than 1/6-indistinguishability advantage and perfect correctness only if it admits a perfectly secure secret-sharing scheme. In other words, access structures that are not binary ideal do not admit a secret-sharing scheme with one-bit shares even with public information.

To prove this impossibility, we develop an alternative characterization for binary ideal access structures: an access structure is binary ideal if and only if the set difference between any minimal qualified set and a maximal forbidden set is odd-sized. We then prove using a combinatorial argument that whenever this condition is not satisfied, there exists a minimal qualified set Q and a maximal forbidden set F such that $|Q \setminus F| = 2$. We show that an adversary who randomly corrupts one amongst the forbidden sets $F, Q \setminus \{i\}$ or $Q \setminus \{j\}$, where $(i, j) = Q \setminus F$ can recover the secret with 2/3 advantage. We formalize these results in Section 9.

3 Preliminaries

Notation. For any integer $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, \ldots, n\}$. For every $n, N \in \mathbb{N}$ such that $n \leq N$, we use $[n; N]$ to denote the set $\{n, n+1, \ldots, N\}$. Notice that $[0, 1]$ denotes the interval of real values $x \in \mathbb{R}$ such that $0 \leq x \leq 1$. For any $n, N \in \mathbb{N}$ such that $n \leq N$, $\mathsf{Inj}(n, N)$ represents the set of injective functions $[n] \to [N]$. We use $\mathsf{Sym}(n)$ to denote the set of permutations of $[n]$.

We use $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$ to denote a generic negligible function, i.e., $\mathsf{negl}(n) = o(n^{-c})$ for every constant $c \in \mathbb{N}$. We use $\mathsf{poly}(n)$ to denote a generic function that is $O(n^c)$ for some constant $c > 0$. Given a distribution μ over the space Ω and a function $f : \Omega \to \mathbb{R}$, we use $\mathbb{E}_\mu[f]$ to denote the expectation of $f(x)$ for $x \overset{\$}{\leftarrow} \mu$. In a similar way, we use $\mathsf{Var}_\mu[f]$ to denote the variance of $f(x)$ for $x \overset{\$}{\leftarrow} \mu$.

Give two vectors $x, y \in \{0, 1\}^n$, we use $\langle x, y \rangle$ to denote their inner-product. We use $w(x)$ to denote the Hamming weight of x, i.e., the number of non-zero entries of x.

Two ensembles of distributions $(\mathcal{D}_0(\mathbb{1}^n))_{n \in \mathbb{N}}$ and $(\mathcal{D}_1(\mathbb{1}^n))_{n \in \mathbb{N}}$ are said to be ε-indistinguishable, for $\varepsilon : \mathbb{N} \to [0, 1]$, by non-uniform $T(n)$-time adversaries if, for any non-uniform adversary $\mathcal{A}$ running in $T(n)$ time, for all sufficiently large n,

$$\left| \Pr\left[\mathcal{A}\big(\mathbb{1}^n, X\big) = b \,\middle|\, \begin{matrix} b \overset{\$}{\leftarrow} \{0,1\} \\ X \overset{\$}{\leftarrow} \mathcal{D}_b(\mathbb{1}^n) \end{matrix} \right] - \frac{1}{2} \right| \leq \varepsilon(n).$$

4 The Planted Subgraph Problem

In this section, we study the hardness of planted subgraph problems. Before presenting our assumptions, we introduce some notation. All the graphs in the paper are finite, undirected and simple. Furthermore, we assume that the set of nodes is $[n]$ for some $n \in \mathbb{N}$. Given a graph G, we denote its complementary by $\overline{G}$: this is the graph in which, for every $i \neq j$, the edge (i, j) appears if and only if (i, j) does not appear in G. We use $\mathcal{G}(n, 1/2)$ to denote a Erdős-Rényi random graph, i.e., the uniform distribution over n-node graphs. Observe that each edge appears independently of the others with probability $1/2$. We denote the clique with n-nodes by K_n. For any n-node graph H and $S \subseteq [n]$, $\mathsf{Subgraph}(H, S)$ denotes the subgraph of H induced by the nodes in S. Notice that this graph has only $|S|$ nodes and its edges are in one-to-one correspondence with the edges of H having both endpoints in S. We will make extensive use of the following *planting* experiment, where we sample a random graph R and then hide inside it a *public* random graph H.

Definition 4.1 (Planting). *Let $\mathcal{D}_R(\mathbb{1}^n)$ and $\mathcal{D}_H(\mathbb{1}^n)$ be distributions over graphs.*

We define the distribution $\mathcal{G}(\mathcal{D}_R, \mathcal{D}_H)$ as follows:

1. *$R \overset{\$}{\leftarrow} \mathcal{D}_R(\mathbb{1}^n)$.*
2. *$H \overset{\$}{\leftarrow} \mathcal{D}_H(\mathbb{1}^n)$.*
3. *Let N and ℓ be the number of nodes of R and H respectively.*
4. *If $\ell > N$, output $\perp$.*
5. *$\phi \overset{\$}{\leftarrow} \mathsf{Inj}(\ell, N)$.*

6. *$G \leftarrow R$.*

7. *For all $i, j \in [\ell]$, if (i, j) appears in H, add $(\phi(i), \phi(j))$ to G.*

8. *For all $i, j \in [\ell]$, if (i, j) does not appear in H, remove $(\phi(i), \phi(j))$ from G.*

9. *Output (G, H, ϕ).*

We often refer to the graph generated by $\mathcal{D}_R$ as the *ambient graph*. We call the output of $\mathcal{D}_H$ the *hidden graph*. Observe that $\mathcal{G}(\mathcal{D}_R, \mathcal{D}_H)$ hides a copy of H in the ambient graph. More specifically, the copy is the subgraph induced by $\phi\big([\ell]\big)$. In other words, the edge (i, j) will appear in H if and only if $(\phi(i), \phi(j))$ appears in G.

In the paper, we will rarely use the general notation $\mathcal{G}(\mathcal{D}_R, \mathcal{D}_H)$. Instead, we will typically refer to the following special cases:

- When $\mathcal{D}_R = \mathcal{G}(N, 1/2)$, we write $\mathcal{G}(N, 1/2, \mathcal{D}_H)$.
- When $\mathcal{D}_R = \mathcal{G}(N, 1/2)$ and $\mathcal{D}_H(\mathbb{1}^n) \equiv K_n$, we write $\mathcal{G}(N, 1/2, n)$.
- When $\mathcal{D}_R = \mathcal{G}(N, 1/2)$ and $\mathcal{D}_H(\mathbb{1}^n) \equiv H_n$ where H_n is a fixed graph, we write $\mathcal{G}(N, 1/2, H_n)$.
- When $\mathcal{D}_R = \mathcal{G}(N, 1/2)$ and $\mathcal{D}_H(\mathbb{1}^n) = \mathcal{G}(n, 1/2)$, we write $\mathcal{G}(N, 1/2, n, 1/2)$.

4.1 The planted clique assumption

We now present the assumptions we will use in this paper. We start by recalling the planted clique assumption, a problem that has been extensively studied by the computational complexity community over the last decades [Jer92, Kuč95, AKS98, FK03, BHK+16, MRS21]. The assumption states that it is hard to distinguish a random graph with a large planted clique from a random graph. The problem is related to the NP-hardness of finding or even approximating the largest clique contained in a graph [Kar72, ALM+92, AS92, BGLR93, BS94, BGS95, FGL+95, Hås96a, Hås96b].

Definition 4.2 (The planted clique assumption [Jer92, Kuč95])**.** *Let $N : \mathbb{N} \to \mathbb{N}$ be a function such that $N(n) \geq n$ for every $n \in \mathbb{N}$. Let $T : \mathbb{N} \to \mathbb{N}$ be a time bound and let $\varepsilon : \mathbb{N} \to [0, 1]$ be an indistinguishability bound. We say that the (N, T, ε)-planted clique (PC) assumption holds if the following distributions are $\varepsilon(n)$-computationally indistinguishable for any non-uniform $\big(T(n) \cdot \mathsf{poly}(n)\big)$-time adversary*

$$\left\{ G \middle| (G, R, \phi) \xleftarrow{\$} \mathcal{G}(N, 1/2, n) \right\} \quad \textit{and} \quad \left\{ G \middle| G \xleftarrow{\$} \mathcal{G}(N, 1/2) \right\}.$$

It is easy to see that the (N, T, ε)-PC assumption implies the (N', T', ε')-PC assumption for any functions $N' \geq N$, $T' \leq T$ and $\varepsilon' \geq \varepsilon$.

Attacks against the PC assumption. A result by Bollobás and Erdős [BE76] proves that the largest clique in an N-node random graph has almost always $\Theta(\log N)$ size. Therefore, the PC assumption cannot hold against computationally unbounded adversaries when $N = \mathsf{poly}(n)$.

The most natural attack against the PC assumption is *edge-counting*: if the graph G hides a clique, it will be denser on average. When $N = \mathsf{poly}(n)$, this leads to a polynomial-time attack with n^{-c} advantage (c is a positive constant).

Another almost as straightforward attack is the *degree attack*: the planted nodes have on average higher degree. In a random N-node graph, the degree of the nodes is described by a binomial probability distribution with average $(N-1)/2$ and standard deviation $\Theta(\sqrt{N})$. After planting the clique, the distribution of the degree of the planted nodes is shifted by n. As noticed by Kučera in [Kuč95], this not only gives a probabilistic polynomial time attack with inverse-polynomial advantage: when $n = \Omega\big(\sqrt{N \cdot \log N}\big)$, it is possible to recover the planted clique with constant probability by simply picking the nodes with highest degree.

This approach can be generalized to a *common-neighbour attack*: for any constant $d > 0$, we consider all subsets of d pair-wise adjacent nodes and we count the number of common neighbours. In a random graph, the average number of common neighbours is $\Theta(N/2^d)$ and its standard deviation is still $\Theta(\sqrt{N})$. On the other hand, when the d nodes lie on the planted clique, the distribution of common neighbours is shifted by $n - d$.

In [HK11], it was also noticed that the PC assumption can be broken in time $n^{O(\log n)}$: the adversary can iterate through all subsets of $d = 3 \log n$ nodes. If the graph is random, with high probability, none of these subsets will form a clique.

The last common family of attacks relies on *spectral analysis*. For instance, in [AKS98], Alon *et al.* showed that the planted clique can be found with constant probability whenever $N < n^2/100$. Other attacks were studied in [FK00, McS01, FR10, AV11, DGGP14, DM15a, CX16]. To this day, none of the approaches discussed above succeeds in describing an $n^{o(\log n)}$-time attack with $o_n(1)$-advantage when $N = \omega(n^2)$.

Conjectured hardness. Motivated by the failed attacks described above, it is conjectured that, for $N = n^{2+\delta}$, the advantage of any $n^{o(\log n)}$ time adversary against the PC problem is dominated by n^{-c} for some constant $c > 0$ [MRS21]. As discussed in [BBB19], the assumption is also supported by its hardness against several classes of attacks: greedy algorithms [McD74, GM75, Kar76, Pit82, Jer92], local algorithms [GS14, COE15, RV17], query models [FGN+20], bounded-depth circuits [Ros08], monotone circuits [Ros10], statistical query algorithms [FGR+13] and resolution [ABdR+18]. Hardness was also proven in the Lovász-Shrijver [FK03] and Sum-of-Squares convex programming hierarchies [MPW15, BHK+16, DM15b, HKP+18].

Conjecture 4.3 (The PC assumption)**.** *For any constant $\delta > 0$, there exists a constant $c > 0$ such that the $(n^{2+\delta}, T, n^{-c})$-PC assumption holds for every $T = n^{o(\log n)}$.*

The PC assumption has been previously used in cryptography. Juels and Peinado [JP00] used a planted clique hardness assumption to build one-way functions, zero-knowledge proofs, and hierarchical key generation. More recently, in the context of machine learning, Goldwasser *et al.* [GKVZ22] used planted cliques to show how a malicious learner can hide a backdoor in a classifier. The assumption was also used to prove hardness of k-wise dependence testing [AAK+07], approximating Nash equilibria [HK11], sparse principal component detection [BR13, BBH18, BB19], restricted isometry sensing [KZ14, WBP16], community detection [HWX15], adaptive estimators [SBW19], matrix completion [Che15], and submatrix detection [MW15, CLR17, BBH19, MRS21]. In [ERSY22], the PC assumption was used to prove that the NP-Complete Clique problem admits a non-adaptive pseudorandom self-reduction.

4.2 The planted subgraph assumption

We now generalize the PC assumption: instead of planting an n-sized clique in a random graph, we plant a generic n-node graph coming from a distribution $\mathcal{D}(\mathbb{1}^n)$. We say that the planted subgraph assumption holds for $\mathcal{D}$ if the resulting graph looks random even if we reveal the output of $\mathcal{D}$.

The idea of generalizing the PC problem to a different distribution of hidden subgraphs is not new. For instance, the planted dense subgraph assumption, which hides a dense subgraph in a large and sparser ambient graph, has been used in learning theory [HWX15, BBH19]. The DUE assumption, introduced by Applebaum *et al.* [ABW10] to build PKE, is also somewhat related: it conjectures the hardness of detecting a subset of nodes with a small number of neighbours hidden in a random regular bipartite graph.

Definition 4.4 (The planted subgraph assumption)**.** *Let $\mathcal{D}(\mathbb{1}^n)$ be an efficient distribution outputting an n-node graph. Let $N : \mathbb{N} \to \mathbb{N}$ be a function such that $N(n) \geq n$ for every $n \in \mathbb{N}$. Let $T : \mathbb{N} \to \mathbb{N}$ be a time bound and let $\varepsilon : \mathbb{N} \to [0, 1]$ be an indistinguishability bound. We say that the $(\mathcal{D}, N, T, \varepsilon)$-planted subgraph (PS) assumption holds if the distributions*

$$\left\{(G, H)\middle|(G, H, \phi) \xleftarrow{\$} \mathcal{G}(N, 1/2, \mathcal{D})\right\} \text{ and } \left\{(G, H)\middle|G \xleftarrow{\$} \mathcal{G}(N, 1/2), H \xleftarrow{\$} \mathcal{D}(\mathbb{1}^n)\right\}$$

are $\varepsilon(n)$-computationally indistinguishable for any non-uniform $\big(T(n) \cdot \mathsf{poly}(n)\big)$-time adversary We say that the (N, T, ε)-planted random subgraph (PRS) assumption holds if the $(\mathcal{D}, N, T, \varepsilon)$-PS assumption holds for $\mathcal{D} = \mathcal{G}(n, 1/2)$.

Observe that if $\mathcal{D}(\mathbb{1}^n) \equiv K_n$, we obtain exactly the PC assumption. Once again, it is easy to see that, for any distribution $\mathcal{D}$, the $(\mathcal{D}, N, T, \varepsilon)$-PS assumption implies the $(\mathcal{D}, N', T', \varepsilon')$-PS whenever $N' \geq N$, $T' \leq T$ and $\varepsilon' \geq \varepsilon$.

Planting fixed families of graphs. We are particularly interested in a variation of the PS assumption: the case in which $\mathcal{D}(\mathbb{1}^n) \equiv H_n$ where $(H_n)_{n\in\mathbb{N}}$ is a fixed family of n-node graphs. It is believed that hiding any subgraph H_n is at least as easy as hiding a clique, i.e., detecting H_n is harder. Indeed, cliques have easily recognisable characteristics that do not occur on most graphs: they are extremely dense, their nodes have large degree, and any subset of their nodes has a lot of common neighbours. The common approaches to solve the PC problem try to leverage these traits.

On the other hand, the vast majority of n-node graphs do not satisfy any of these properties. It is conjectured that, for all $\delta > 0$, the PS assumption holds for every family $(H_n)_{n\in\mathbb{N}}$ with parameters $N = n^{2+\delta}$, $\varepsilon = n^{-c}$ and $T = n^{o(\log n)}$. In Section 4.3, we provide some evidence to support this claim: we show that the assumption holds against any adversary that can be represented as a degree-$(\log n)^{2-\varepsilon}$ polynomial where $\varepsilon > 0$. In the domain of planted problems, this kind of adversaries have always turned out to lead to the best known attacks. For this reason, it was even conjectured that if no degree-D polynomial can distinguish, then the planted assumption holds against any adversary running in time $2^{O(D)}$ [Hop18, Conjecture 2.2.4].

We additionally conjecture that, except for an inverse-polynomial fraction of n-node graphs H_n, the PS assumption holds, for every $\delta > 0$, with parameters $N = n^{1+\delta}$, $\varepsilon = n^{-c}$ and $T = n^{o(\log n)}$. In other words, the size of the graph is $n^{1+\delta}$ (compared to $n^{2+\delta}$ for hiding cliques). In some sense, these conjectures give information about the worst-case hardness and the average-case hardness of detecting a subgraph planted in a random graph.

Planting random graphs. Another interesting version of the PS problem is the case in which $\mathcal{D}(\mathbb{1}^n) = \mathcal{G}(n, 1/2)$. We refer to this variation as the *PRS assumption*. Even in this case, the assumption is believed to hold with parameters $N = n^{1+\delta}$, $\varepsilon = n^{-c}$ and $T = n^{o(\log n)}$. This fact is actually implied by the conjectured average-case hardness of detecting planted subgraphs. The PRS assumption is however strictly weaker: if we plant a a fixed graph H_n, the adversary receives an H_n-dependent non-uniform advice. The same would not happen when H_n is sampled at random.

4.3 The planted subgraph assumption with hints

We finally present a variation of the PS assumption in which we provide the adversary with hints: we leak the position of t nodes in the hidden subgraph. Formally, the assumption states that even if we reveal where a subset S of t nodes is hidden, then we cannot distinguish between a graph in which we plant $H \xleftarrow{\$} \mathcal{D}(\mathbb{1}^n)$ and a random graph in which we hide the subgraph of H induced by S. We will consider small t, e.g., $t = 2$.

The PC assumption is considered robust against leakage. For instance, Brennan and Bresler [BB20] studied several variations of the PC problem in which the adversary is provided with leakage about the position of the planted clique. The authors consider e.g. the case in which the clique is planted in a multipartite graph (the clique will have a single node in each part of the graph).

Definition 4.5 (The planted subgraph assumption with hints)**.** *Let $\mathcal{D}(\mathbb{1}^n)$ be an efficient distribution outputting an n-node graph. Let $N, t : \mathbb{N} \to \mathbb{N}$ be functions such that $N(n) \geq n \geq t(n)$ for every $n \in \mathbb{N}$. Let $T : \mathbb{N} \to \mathbb{N}$ be a time bound and let $\varepsilon : \mathbb{N} \to [0, 1]$ be an indistinguishability bound. We say that the $(\mathcal{D}, N, t, T, \varepsilon)$-planted subgraph with hints (PSH) assumption holds if, for every sequence of subsets $(S_n)_{n\in\mathbb{N}}$ such that $S_n \subseteq [n]$, $|S_n| \leq t(n)$ for every $n \in \mathbb{N}$, the following ensembles of distributions are $\varepsilon(n)$-computationally indistinguishable for any non-uniform $\big(T(n) \cdot \mathsf{poly}(n)\big)$-time adversary:*

$$\left(\left\{(G, H, (u_i)_{i\in S_n}) \middle| \begin{array}{l} (G, H, \phi) \xleftarrow{\$} \mathcal{G}(N, 1/2, \mathcal{D}) \\ \forall i \in S_n : \quad u_i \leftarrow \phi(i) \end{array}\right\}\right)_{n\in\mathbb{N}}$$

and

$$\left(\left\{(G, H, (u_i)_{i\in S_n}) \middle| \begin{array}{l} H \xleftarrow{\$} \mathcal{D}(\mathbb{1}^n) \\ H' \leftarrow \mathsf{Subgraph}(H, S_n) \\ (G, H', \phi) \xleftarrow{\$} \mathcal{G}\big(N, 1/2, H'\big) \\ \forall i \in S_n : \quad u_i \leftarrow \phi(i) \end{array}\right\}\right)_{n\in\mathbb{N}} .$$

We say that the (N, t, T, ε)-planted random subgraph with hints (PRSH) assumption holds if the $(\mathcal{D}, N, t, T, \varepsilon)$-PSH assumption holds for $\mathcal{D} = \mathcal{G}(n, 1/2)$.

Observe that if $t = 0$, we obtain exactly the PS assumption. Once again, it is easy to see that, for any distribution $\mathcal{D}$, the $(\mathcal{D}, N, t, T, \varepsilon)$-PSH assumption implies the $(\mathcal{D}, N', t', T', \varepsilon')$-PSH whenever $N' \geq N$, $t' \leq t$, $T' \leq T$ and $\varepsilon' \geq \varepsilon$.

Conjectured hardness. Revealing $t(n) = O(1)$ nodes on the planted graph is believed to not affect the overall security of the planted subgraph assumptions.

When t is super-constant, the hardness of the problem becomes, however, less clear. For instance, revealing $t = \log n$ nodes on a planted clique would allow distinguishing the graph from a random one with constant advantage by simply counting the number of common neighbours of the t nodes. This leads to the following conjectures.

Conjecture 4.6 (wPSH, PSH, and PRSH)**.**

Weak-PSH Conjecture (Weak Planted Subgraph with Hints). *For every constants $\delta > 0$ and $t \in \mathbb{N}$ there exists a constant $c > 0$, such that for every sequence of graphs $(H_n)_{n\in\mathbb{N}}$ (where H_n is a n-node graph), the $(\mathcal{D}_H, n^{2+\delta}, t, T, n^{-c})$-PSH assumption holds for $\mathcal{D}_H(\mathbb{1}^n) \equiv H_n$, for all $T = n^{o(\log n)}$.*

PSH Conjecture (Planted Subgraph with Hints). *There exists a sequence $(\mathcal{R}_n)_{n\in\mathbb{N}}$, where $\mathcal{R}_n$ is a set of n-node graphs, with the following properties:*

1. *With overwhelming probability a graph is in $\mathcal{R}_n$, that is, there exists a negligible function $\mathsf{negl}(n)$ such that $|\mathcal{R}_n| \geq (1 - \mathsf{negl}(n)) \cdot 2^{\binom{n}{2}}$ for every $n \in \mathbb{N}$.*
2. *For every constants $\delta > 0$ and $t \in \mathbb{N}$, there exists a constant $c > 0$ such that, for all $T = n^{o(\log n)}$ and all sequences $(H_n)_{n\in\mathbb{N}}$ such that $H_n \in \mathcal{R}_n$ for every $n \in \mathbb{N}$, the $(\mathcal{D}_H, n^{1+\delta}, t, T, n^{-c})$-PSH assumption holds for $\mathcal{D}_H(\mathbb{1}^n) \equiv H_n$.*

PRSH Conjecture (Planted Random Subgraph with Hints). *For every constants $\delta > 0$ and $t \in \mathbb{N}$, there exists a constant $c > 0$ such that the $(n^{1+\delta}, t, T, n^{-c})$-PRSH assumption holds for all $T = n^{o(\log n)}$.*

4.4 Security against low-degree polynomials

We now provide some evidence to support our conjectures: we show that the weak PSH assumption holds for any graph H_n against all adversaries that can be represented as degree-$(\log n)^{1+\varepsilon}$ polynomials where $\varepsilon < 1$. In the domain of planted problems, interestingly, all known successful attacks belong to this class [Hop18]. Low-degree polynomials can be incredibly useful in detecting structures planted in large objects. To give an example, given a graph H with D edges, we can use a degree-D polynomial p to tell how many copies of H are hidden in another larger graph. Moreover, if the polynomial p is M-variate, we can always evaluate it in $D \cdot \binom{M}{D}$ arithmetic operations. For these reasons, Hopkins conjectured that if there exists no degree-D distinguisher, then the planted assumption holds against generic $2^{O(D)}$-time adversaries [Hop18, Conjecture 2.2.4]. In [HW21], Holmgren and Wein presented counterexamples to the conjecture of Hopkins. Their techniques, however, crucially rely on the size of the alphabet used to encode the problem instance to be large. For this reason, they suggest that the conjecture is still likely to hold when the alphabet size is constant (this is always the case in planted graph problems) [HW21, Remark 3.3].

We highlight that the conjecture of Hopkins was introduced in the context of the study of algorithms, where a different notion of indistinguishability is in use: two distributions are "algorithmically indistinguishable" if the advantage of any efficient adversary is $1 - \Omega(1)$ (i.e., it is impossible to efficiently distinguish with vanishing error probability, or equivalently with advantage that approaches 1 at the limit). It is, however, reasonable to assume that the conjecture scales to other notions of indistinguishability, e.g., the main notion we adopt here, where the advantage is required to be $n^{-\Omega(1)}$.

Theorem 4.7. *Let $(H_n)_{n\in\mathbb{N}}$ be a sequence of graphs where H_n has n nodes and let $t \in \mathbb{N}$ be a constant. Let $(S_n)_{n\in\mathbb{N}}$ be a sequence of sets where $S_n = \{u_{n,1}, \dots, u_{n,\ell_n}\} \subseteq [n]$ and $\ell_n \leq t$. Let $N(n) := n^{2+\delta}$ where $\delta > 0$ is a constant. Let μ_n be the distribution that samples $(G, H'_n, \phi) \overset{\$}{\leftarrow} \mathcal{G}(N(n), 1/2, H'_n)$, where $H'_n \leftarrow \mathsf{Subgraph}(H_n, S_n)$, then reorders the nodes in G so that $\phi(u_i)$ ends up in the i-th position and, finally, outputs a bit string encoding the edges of the graph except those that have both endpoints in $\phi(S_n)$. Let ν_n be the analogous distribution where, instead, we sample G from $\mathcal{G}(N(n), 1/2, H_n)$. Let $M(n)$ be the length of the strings generated by μ_n and ν_n.*

For any constant $0 < \varepsilon \leq 2$ and sequence of polynomials $(p_n)_{n\in\mathbb{N}}$, where $p_n \in \mathbb{R}[X_1, \dots, X_M]$ has degree at most $D(n) := (\log n)^{2-\varepsilon}$, we have

$$\mathsf{Adv}(p_n) := \frac{|\mathbb{E}_{\nu_n}[p_n] - \mathbb{E}_{\mu_n}[p_n]|}{\sqrt{\mathsf{Var}_{\mu_n}[p_n]}} \leq n^{-\Omega(1)}. \tag{1}$$

The notion of advantage used for low-degree polynomials may first look a bit odd. We explain why it is a meaningful definition. Suppose that $\mathsf{Var}_{\mu_n}[p_n] \sim \mathsf{Var}_{\nu_n}[p_n]$. When (1) does not hold, then, it is usually easy to distinguish between μ_n and ν_n just based on the result of the evaluation of p_n: since the distributions $p_n(\mu_n)$ and $p_n(\nu_n)$ are concentrated around their mean, one can effectively distinguish between the distributions by determining whether the sample is "large" or "small." Conversely, if (1) holds, an attacker has a hard time distinguishing between μ_n and ν_n just based on the evaluation of p_n: if the result is close to $\mathbb{E}_{\nu_n}[p_n]$, it could be that we actually received a sample from ν_n, or it could be that, due to its variance, $p_n(\mu_n)$ produced a sample that is relatively far away from its expectation. Since $\mathsf{Var}_{\mu_n}[p_n] \sim \mathsf{Var}_{\nu_n}[p_n]$, the adversary faces a similar dilemma even if we obtain a value that is close to $\mathbb{E}_{\mu_n}[p_n]$ or far from both $\mathbb{E}_{\mu_n}[p_n]$ and $\mathbb{E}_{\nu_n}[p_n]$. If instead $\mathsf{Var}_{\mu_n}[p_n]$ and $\mathsf{Var}_{\nu_n}[p_n]$ are far apart, the polynomials $q_n := \big(p_n(X) - \mathbb{E}_{\mu_n}[p_n]\big)^2$ and $q'_n := \big(p_n(X) - \mathbb{E}_{\nu_n}[p_n]\big)^2$ most likely do not satisfy (1).

Example 4.8. To demonstrate the notion of an advantage, we next give an example of a degree-1 polynomial that distinguishes with constant advantage between a random graph with n^2 nodes and a random graph with a planted clique (without any hints). Let G be a graph with n^2 nodes. For each possible edge between the i-th node and the j-th node we have a variable $x_{i,j}$ and consider the polynomial $p((x_{i,j})_{1\leq i<j\leq n^2}) = \sum_{1\leq i<j\leq n^2} x_{i,j}$. If G is a random n^2-node graph, then the expected value of the polynomial p is $0.5\binom{n^2}{2}$. On the other hand, if G is a random graph with a planted clique of size n, then the expected value of p is $0.5(\binom{n^2}{2} + \binom{n}{2})$. Recall that the variance of the number of edges in a n^2-node random graph is $0.25\binom{n^2}{2}$. Thus, the advantage is constant. On the other hand, if we plant an n-node clique in a random $n^{2+\delta}$-node graph, then the advantage is

$$O\left(\frac{0.5(\binom{n^{2+\delta}}{2} + \binom{n}{2}) - 0.5\binom{n^{2+\delta}}{2}}{\sqrt{0.25\binom{n^{2+\delta}}{2}}}\right) = O\left(\frac{1}{n^{\delta}}\right).$$

Below, we prove Theorem 4.7

Proof. We follow the blueprint of [Hop18, Chapter 2.4]. Without loss of generality, we can assume that $\mathbb{E}_{\nu_n}[p_n] \geq 0$, $\mathbb{E}_{\mu_n}[p_n] = 0$ and $\mathsf{Var}_{\mu_n}[p_n] = 1$. Indeed, observe that

$$\frac{|\mathbb{E}_{\nu_n}[p_n] - \mathbb{E}_{\mu_n}[p_n]|}{\sqrt{\mathsf{Var}_{\mu_n}[p_n]}} = \mathbb{E}_{\nu_n}[p'_n]$$

where $p_n'(X) := \sigma \cdot (p_n(X) - \mathbb{E}_{\mu_n}[p_n]) / \sqrt{\mathsf{Var}_{\mu_n}[p_n]}$ and $\sigma \in \{-1, 1\}$ is positive if and only if $\mathbb{E}_{\nu_n}[p_n] \geq \mathbb{E}_{\mu_n}[p_n]$. Let Ξ_n be the set of all polynomials $p_n \in \mathbb{R}[X_1, \dots, X_M]$ of degree at most $D(n)$ such that $\mathbb{E}_{\mu_n}[p_n] = 0$ and $\mathsf{Var}_{\mu_n}[p_n] = 1$.

Observe that μ_n outputs a uniformly random string. Therefore, as proven in [Hop18, Chapter 2.3], we know that

$$\max_{p_n \in \Xi_n} \mathbb{E}_{\nu_n}[p_n] = \sqrt{\sum_{\substack{w(\alpha) \leq D(n) \\ \alpha \neq 0}} (\mathbb{E}_{\nu_n}[\chi_\alpha])^2}$$

where $\chi_\alpha(x) = (-1)^{\langle \alpha, x \rangle}$. Now, let U denote the ordered list containing $\phi(i)$ for every $i \notin S_n$ and let Ω denote all possible ordered subsets of $n - \ell_n$ distinct elements in $[N] \setminus \phi(S_n)$. Let x_i be the i-th bit in the vector output by ν_n (we recall that such bit describes whether a certain edge appears in G or not). We observe that, conditioned on U, all x_is are independently distributed. Indeed, if both $u, v \in U \cup \phi(S_n)$, the edge (u, v) will appear in G either with probability 0 or 1 (depending on the graph H_n). If, instead, either u or v (or both) do not belong to $U \cup \phi(S_n)$, the edge (u, v) will appear in G with probability $1/2$ independently of everything else. Therefore, for any $\alpha \in \{0, 1\}^n$, we have

$$\begin{aligned}\mathbb{E}_{\nu_n}[\chi_\alpha] &= \sum_{\omega \in \Omega} \Pr[U = \omega] \cdot \mathbb{E}_{\nu_n}[\chi_\alpha | U = \omega] = \\ &= \sum_{\omega \in \Omega} \Pr[U = \omega] \cdot \mathbb{E}_{\nu_n}\left[\prod_{i \in [M]} (-1)^{x_i \cdot \alpha_i} \middle| U = \omega\right] = \\ &= \sum_{\omega \in \Omega} \Pr[U = \omega] \cdot \prod_{i \in [M]} \mathbb{E}_{\nu_n}\left[(-1)^{x_i \cdot \alpha_i} | U = \omega\right] = \\ &= \frac{(N-n)!}{(N-\ell_n)!} \cdot \sum_{\omega \in \Omega} \prod_{i \in [M]} \mathbb{E}_{\nu_n}\left[(-1)^{x_i \cdot \alpha_i} | U = \omega\right].\end{aligned}$$

Now, suppose that there exists an i such that $\alpha_i = 1$ and at least one of the endpoints of the i-th edge do not belong to $\omega \cup \phi(S_n)$. Then, $\mathbb{E}_{\nu_n}\left[(-1)^{x_i \cdot \alpha_i} | U = \omega\right] = 0$ as x_i is uniformly distributed. Furthermore, if both endpoints of the edge are in $\omega \cup \phi(S_n)$, either $\mathbb{E}_{\nu_n}\left[(-1)^{x_i \cdot \alpha_i} | U = \omega\right] = 1$ or $\mathbb{E}_{\nu_n}\left[(-1)^{x_i \cdot \alpha_i} | U = \omega\right] = -1$ depending on the edge of H_n that we planted there. By using $V(\alpha)$ to denote the number of endpoints of the edges described by α (ignoring those in $\phi(S_n)$), we obtain that

$$\mathbb{E}_{\nu_n}[\chi_\alpha] = \frac{(N-n)!}{(N-\ell_n)!} \cdot \binom{N - V(\alpha) - \ell_n}{n - V(\alpha) - \ell_n} \cdot \sum_{\pi \in \mathsf{Sym}(n - \ell_n)} (-1)^{\langle \pi \circ h, \alpha' \rangle}.$$

Above, h denotes the bit vector encoding of the edges of H_n except those that have both endpoints in S_n. We use instead $\pi \circ h$ to denote the vector encoding the graph obtained by permuting the nodes of $H_n \setminus S_n$ according to π (again, we ignore the edges that have both endpoints in S_n). We highlight that π permutes the nodes of H_n, not the edges. As a consequence, $\pi \circ h$ is a permutation of h, but not all permutations of h correspond to a $\pi \in \mathsf{Sym}(n - \ell_n)$. Finally, α' is a vector obtained by taking the graph representation of α (we recall that α represents a set of at most D edges in an N-node graph), we remove $N - n$ isolated nodes from this graph, but not those in S_n (here are enough isolated nodes as $V(\alpha) + t \leq 2D + t < n$.) and we encode the result as a vector as we did for H_n. The above formula is obtained by counting the number of (unordered) subsets of nodes in $G \setminus \phi(S_n)$ that contain all the endpoints of the edges in α, ignoring those in $\phi(S_n)$. This number is $\binom{N-V(\alpha)-\ell_n}{n-V(\alpha)-\ell_n}$. We recall that

$$\prod_{i\in[M]} \mathbb{E}_{\nu_n}\left[(-1)^{x_i\cdot\alpha_i}|U=\omega\right] \neq 0 \tag{2}$$

only if ω is obtained by reordering any of these subsets. Moreover, the exact value of (2) depends only on such reordering (as the reordering univocally determines how H_n is planted).

To conclude, we have that, for any $p_n \in \Xi_n$, $\mathbb{E}_{\nu_n}[p_n]$ is at most

$$\frac{(N-n)!}{(N-\ell_n)!}\sqrt{\sum_{\substack{w(\alpha)\leq D(n)\\ \alpha\neq 0}} \binom{N-V(\alpha)-\ell_n}{n-V(\alpha)-\ell_n}^2 \cdot \sum_{\pi_1,\pi_2\in\mathsf{Sym}(n-\ell_n)} (-1)^{\langle \pi_1\circ h+\pi_2\circ h,\alpha'\rangle}} =$$

$$=\frac{(N-n)!\cdot\sqrt{(n-\ell_n)!}}{(N-\ell_n)!}\sqrt{\sum_{\substack{w(\alpha)\leq D(n)\\ \alpha\neq 0}} \binom{N-V(\alpha)-\ell_n}{n-V(\alpha)-\ell_n}^2 \cdot \sum_{\pi\in\mathsf{Sym}(n-\ell_n)} (-1)^{\langle \pi\circ h+h,\alpha'\rangle}}$$

It is easy to see that when H_n is a clique, the above formula reaches its maximum as $\pi\circ h = h$ for any π. The value of such maximum is

$$\binom{N-\ell_n}{n-\ell_n}^{-1}\cdot\sqrt{\sum_{\substack{w(\alpha)\leq D(n)\\ \alpha\neq 0}} \binom{N-V(\alpha)-\ell_n}{n-V(\alpha)-\ell_n}^2}$$

Observe that

$$\frac{\binom{N-V(\alpha)-\ell_n}{n-V(\alpha)-\ell_n}}{\binom{N-\ell_n}{n-\ell_n}} \leq \left(\frac{n-\ell_n}{N-\ell_n}\right)^{V(\alpha)} \leq \left(\frac{1}{n^{1+\delta}}\right)^{V(\alpha)}$$

Furthermore, for every α such that $0 < w(\alpha) \leq D(n)$, we have $0 < V(\alpha) \leq 2D(n)$. Now, if we take any $m \leq 2D(n)$, $m > 0$, there are most

$$(N-\ell_n)^m\cdot(m+\ell_n)^{\min\{2D,2(m+\ell_n)^2\}} \leq N^m\cdot(m+t)^{\min\{2D,2(m+t)^2\}}$$

different α such that $V(\alpha) = m$ and $w(\alpha) \leq D(n)$. Therefore, $\max_{p_n\in\Xi_n}\mathbb{E}_{\nu_n}[p_n]$ is upper-bounded by

$$\sqrt{\sum_{0<m\leq\sqrt{D}-t}\frac{N^m\cdot(m+t)^{2(m+t)^2}}{n^{2m(1+\delta)}} + \sum_{\sqrt{D}-t<m\leq 2D}\frac{N^m\cdot(m+t)^{2D}}{n^{2m(1+\delta)}}} \leq$$

$$\leq \sqrt{\sum_{0<m\leq\sqrt{D}-t}\frac{(m+t)^{2(m+t)^2}}{n^{m\cdot\delta}} + \sum_{\sqrt{D}-t<m\leq 2D}\frac{(m+t)^{2D}}{n^{m\cdot\delta}}}$$

Observe that

$$\sum_{\sqrt{D}-t<m\leq 2D} \frac{(m+t)^{2D}}{n^{m\cdot\delta}} \leq 2D \cdot \frac{(4D)^{2D}}{n^{\delta\cdot(\sqrt{D}-t)}} = 2^{O\left((\log n)^{2-\varepsilon}\cdot \log\log n\right)-\Omega\left((\log n)^{2-\frac{\varepsilon}{2}}\right)}$$

Since $\varepsilon > 0$, the above is a negligible function in n. As for the first sum, we have

$$\begin{aligned}\sum_{0<m\leq\sqrt{D}-t} \frac{(m+t)^{2(m+t)^2}}{n^{m\cdot\delta}} \leq \sum_{0<m\leq\sqrt{D}-t} \frac{\left(\sqrt{D}^{2\sqrt{D}}\right)^{m+t}}{n^{m\cdot\delta}} =& \\ =&D^{t\cdot\sqrt{D}} \cdot \sum_{0<m\leq\sqrt{D}-t} \left(\frac{D^{\sqrt{D}}}{n^{\delta}}\right)^{m}\end{aligned}$$

Observe that, since $\varepsilon > 0$,

$$D^{\sqrt{D}} = 2^{O\left(\log\log n\cdot(\log n)^{1-\frac{\varepsilon}{2}}\right)} = n^{o(1)}$$

Therefore, $\frac{D^{\sqrt{D}}}{n^{\delta}}$ is asymptotically smaller than $1/2$. We can therefore conclude that

$$\begin{aligned}\sum_{0<m\leq\sqrt{D}-t} \frac{(m+t)^{2(m+t)^2}}{n^{m\cdot\delta}} \leq& D^{t\cdot\sqrt{D}} \cdot \sum_{0<m\leq\sqrt{D}-t} \left(\frac{D^{\sqrt{D}}}{n^{\delta}}\right)^{m} \leq \\ \leq& D^{t\cdot\sqrt{D}} \cdot 2 \cdot \frac{D^{\sqrt{D}}}{n^{\delta}} = \frac{n^{o(1)}}{n^{\delta}} = n^{-\Omega(1)}\end{aligned}$$

We have just proven that $\max_{p_n\in\Xi_n} \mathbb{E}_{\nu_n}[p_n] = n^{-\Omega(1)}$. □

The proof of Theorem 4.7 highlights some important facts. First of all, it confirms the intuition that cliques are the easiest subgraph we can detect (independently of N, t and D). Moreover, it provides a formula that describes how easy it is for a degree-D polynomial to detect a planted n-node subgraph H in a random N-node graph: if we provide no hints, the formula is the following:

$$\sum_{0<w(\alpha)\leq D} \binom{N-V(\alpha)}{n-V(\alpha)}^2 \cdot \sum_{\pi\in\mathsf{Sym}(n)} (-1)^{\langle \pi\circ H+H,\alpha\rangle} \tag{3}$$

The lower the value, the harder detecting H becomes.

We recall that, above, α denotes a subset of at most D edges in an n-node graph. We encode α as a vector of bits (the i-th bit indicates whether the i-th edge is in the subset or not). We use a similar representation for H (therefore, the inner-product is well-defined). We use $\pi \circ H$ to denote the graph obtained by permuting the nodes (n.b. not the edges) of H according to π (once again we represent this graph as a vector). Finally, $V(\alpha)$ denotes the total number of nodes touched by the edges in α.We observe that the vectors $(\pi\circ H + H)_{\pi\in\mathsf{Sym}(n)}$ give a good description of how "structured" H is (e.g., if H is a clique or an independent set, all these vectors are 0). We conjecture that, for most graphs H, the sum $\sum_{\pi\in\mathsf{Sym}(n)}(-1)^{\langle\pi\circ H+H,\alpha\rangle}$ should be small for all choices of α, as $\langle\pi\circ H + H,\alpha\rangle$ should assume the values 0 and 1 almost equally often. We leave the rigorous study of this problem to future work.

The last important observation is that the proof stops working as soon as D reaches $\log^2 n$. This is no coincidence: the largest clique in a random N-node graph has at most $2\log n$ nodes with high probability [BE76]. Therefore, we can distinguish between $\mathcal{G}(N, 1/2)$ and $\mathcal{G}(N, 1/2, n)$ using a degree-D' polynomial where $D' := (3\log n)^2$. Such polynomial will simply count the number of cliques of size $3\log n$ hidden in the graph.

The PRSH conjecture. The PSH conjecture tells us information about the average-case hardness of detecting planted subgraphs: if $N = n^{1+\delta}$, the assumption holds for an overwhelming fraction of n-node graphs. The conjecture is therefore related to the hardness of the PRSH problem: in the following theorem, we prove that the former implies the latter.

Theorem 4.9. *The PSH conjecture implies the PRSH conjecture.*

Proof. Suppose this is not the case: there exists a $\overline{\delta} > 0$ and $\overline{t} \in \mathbb{N}$ such that, for every $c > 0$, there exists a non-uniform adversary $\mathcal{A}$ that breaks the PRSH assumption for $N = n^{1+\overline{\delta}}$ with advantage asymptotically greater than n^{-c}. In the context of this proof, we say that an n-node graph H_n is good if $H_n \in \mathcal{R}_n$. Now, consider the PSH conjecture and let $\overline{c} > 0$ be the constant associated with $\overline{\delta}$ and $\overline{t}$. Consider the non-uniform adversary $\mathcal{A}$ that breaks the PRSH assumption for parameters $\overline{\delta}$, $\overline{t}$ and $\overline{c}/2$. For any $n \in \mathbb{N}$, we consider the good graph H_n for which the advantage of the adversary $\mathcal{A}$ in the PRSH game conditioned on the hidden subgraph being H_n is greatest (the maximum exists as there are only a finite number of n-node graphs for a fixed n). Since a random graph is good with probability $1 - \mathsf{negl}(n)$ and since $\mathcal{A}$ has advantage asymptotically greater than $n^{-\overline{c}/2}$ against the PRSH game, the advantage of the adversary $\mathcal{A}$ in the PRSH game conditioned on the hidden subgraph being H_n must be asymptotically greater than $n^{-\overline{c}/2} - \mathsf{negl}(n) > n^{-\overline{c}}$. Such adversary would therefore contradict the PSH conjecture for the graph family $(H_n)_{n\in\mathbb{N}}$. □

5 Private Simultaneous Messages with Logarithmic Message Size

In this section, we present a computational 2-party private simultaneous message (PSM) protocol with public information achieving logarithmic message size (with inverse-polynomial security). The construction is based on the PSH assumption.

5.1 PSM protocol with public information

We start by formalizing the definition of private simultaneous message protocol with public information. As in standard PSM protocols, the primitive allows a pair of parties to encode their inputs x and y and non-interactively evaluate a function f of the inputs from the encodings. Any external observer that intercepts the exchanged messages is guaranteed to learn no information beyond the output of the function.[6] A PSM protocol with public information differs in that the setup outputs also public information that is needed to reconstruct the output.

Definition 5.1 (PSM protocols with public information)**.** *Let $\mathcal{F} := \{f_n\}_{n\in\mathbb{N}}$ be a family of functions such that $f_n : [n] \times [n] \to \{0, 1\}$. A (T, ε)-secure private simultaneous messages (PSM)*

[6] Observe that the randomness used by the parties to encode their inputs cannot be independent. If that was not the case, an external attacker would be able to mount a *residual function attack*: given the encoding of the input x of one of the parties, it can generate the encoding of any value y' for the other party and learn $f(x, y')$. This guarantees that in every PSM protocol, the parties need to share some common secret provided to them by a setup procedure.

protocol with public information for $\mathcal{F}$ is a triple of PPT algorithms ($\mathsf{Setup}, \mathsf{Encode}, \mathsf{Output}$) *with the following syntax:*

- Setup *is randomized and takes as input $\mathbb{1}^n$ and a description of f_n. The output is a triple (I, s_0, s_1) where I is public information, and s_0, s_1 are private values.*
- Encode *is randomized and takes as input $\mathbb{1}^n$, an index $i \in \{0,1\}$, a public information I, a private value s_i, and an input $x_i \in [n]$. The output is a message m_i.*
- Output *is deterministic and takes as input two messages m_0, m_1 and public information I. The output is a bit $z \in \{0,1\}$.*

We require the following properties:

Perfect correctness. *For every inputs $x, y \in [n]$ and random strings r, r_0, r_1, if $(I, s_0, s_1) \xleftarrow{\$} \mathsf{Setup}(\mathbb{1}^n, f_n; r)$, $m_0 \xleftarrow{\$} \mathsf{Encode}(\mathbb{1}^n, 0, s_0, x; r_0)$, and $m_1 \xleftarrow{\$} \mathsf{Encode}(\mathbb{1}^n, 1, s_1, y; r_1)$, then the output is correct, i.e.,* $\mathsf{Output}(m_0, m_1, I) = f_n(x,y)$.

Security. *There exists a non-uniform polynomial-time simulator* PSMSim *such that, for every non-uniform $\big(T(n) \cdot \mathsf{poly}(n)\big)$-time adversary $\mathcal{A}$, sufficiently large n, and $x, y \in [n]$, we have*

$$\left| \Pr\left[\mathcal{A}(\mathbb{1}^n, m_0^b, m_1^b, I_b) = b \middle| \begin{array}{l} b \xleftarrow{\$} \{0,1\} \\ (I_0, s_0, s_1) \xleftarrow{\$} \mathsf{Setup}(\mathbb{1}^n, f_n) \\ m_0^0 \xleftarrow{\$} \mathsf{Encode}(\mathbb{1}^n, 0, s_0, x) \\ m_1^0 \xleftarrow{\$} \mathsf{Encode}(\mathbb{1}^n, 1, s_1, y) \\ (m_0^1, m_1^1, I_1) \xleftarrow{\$} \mathsf{PSMSim}\big(\mathbb{1}^n, f_n(x,y)\big) \end{array} \right] - \frac{1}{2} \right| \leq \varepsilon(n),$$

that is, the adversary cannot distinguish with advantage greater than $\varepsilon(n)$ if the messages and public information were generated as in the PSM protocol with inputs x, y or by the simulator only holding the output $f(x,y)$.

The message size of a PSM protocol with public information is

$$\ell(n) := \max_{i,r,r',x \in [n]} \left\{ |m_i| \middle| \begin{array}{l} (I, s_0, s_1) \leftarrow \mathsf{Setup}(\mathbb{1}^n;\, r) \\ m_i \leftarrow \mathsf{Encode}(\mathbb{1}^n, i, s_i, x;\, r') \end{array} \right\}.$$

5.2 PSM protocols with public information from planted subgraphs

In this section, we present a PSM protocol with public information achieving $(n^{o(\log n)}, n^{-c})$-security for a constant $c > 0$ with $O(\log n)$ message size. The scheme is based on the PSH assumption. The construction is formalized in Figure 1. Before explaining the idea at the base of the construction, we introduce the following definition.

Definition 5.2. *Let $\mathcal{F} = (f_n)_{n \in \mathbb{N}}$ be a family of functions such that $f_n : [n] \times [n] \to \{0,1\}$. We define the graph H_n as follows:*

- *The graph is bipartite and has n nodes per part. We index the nodes with the elements in $[2n]$. The first partition will consist of the nodes in $[n]$. The node $1 \leq j \leq n$ represents the input j for the first party. The node $n+1 \leq j \leq 2n$ represents the input j for the second party.*

- *For every $i, j \in [n]$, we draw the edge $(i, n+j)$ if and only if $f_n(i,j) = 1$.*

In the PSM protocol, the public information consists of a large random graph in which we plant a copy of H_n. Observe that the latter is a smaller bipartite graph that describes the behaviour of f_n: two nodes are connected if and only if the evaluation of f_n on the associated values gives 1. The private information received by the parties consists of the injective map that indicates where H_n was hidden.

In order to encode its input, a party just needs to reveal where the corresponding node was hidden in the big graph. The output of the evaluation can be recovered by simply checking whether the nodes revealed by the parties are adjacent in the public graph. Under the PSH assumption, we can argue that an external adversary has no clue about where the other $2n-2$ nodes of H_n are hidden. That ensures that an attacker cannot learn anything about the inputs beyond the output of the evaluation.

A PSM PROTOCOL WITH PUBLIC INFORMATION BASED ON PLANTED SUBGRAPHS

Let $\mathcal{F} = (f_n)_{n \in \mathbb{N}}$ be a family of functions such that $f_n : [n] \times [n] \to \{0,1\}$. Let H_n be the graph representation of $\mathcal{F}$. Let $N : \mathbb{N} \to \mathbb{N}$ be a function such that $N(n) \geq 2n$ for every $n \in \mathbb{N}$.

$\mathsf{Setup}(\mathbb{1}^n, f_n)$

1. $(G, H, \phi) \overset{\$}{\leftarrow} \mathcal{G}(N, 1/2, H_n)$.
2. Output $I = G$, and $s_0 = s_1 = \phi$.

$\mathsf{Encode}(\mathbb{1}^n, i, s_i = \phi, x)$

1. Output $m_i = \phi(x + n \cdot i)$.

$\mathsf{Output}(m_0, m_1, I = G)$

1. Output 1 if m_0 and m_1 are adjacent in G. Otherwise, output 0.

Figure 1: A PSM protocol with public information based on planted subgraphs.

Theorem 5.3. *Let $\mathcal{F} = (f_n)_{n \in \mathbb{N}}$ be a family of functions such that $f_n : [n] \times [n] \to \{0,1\}$. If the $(N, 2, T, \varepsilon)$-PSH assumption holds for $\mathcal{D}(\mathbb{1}^n) \equiv H_n$, then the construction in Figure 1 is a (T, ε)-secure PSM protocol with public information for $\mathcal{F}$. The message size of each party is $\log N$, the public information size is $N(N-1)/2$.*

Moreover, assume that the Weak-PSH conjecture holds (see Conjecture 4.6). Then, there exists a constant $c > 0$, such that for every constant $\delta > 0$ the construction with $M(n) = n^{2+\delta}$ achieves (T, n^{-c})-security with $(2+\delta) \cdot \log n$ message size and $O(n^{4+2\delta})$ public information size for any $T = n^{o(\log n)}$. Finally, if the PSH conjecture holds and $H_n \in \mathcal{R}_n$ for all sufficiently large n, then, *for every constant $\delta > 0$, there exists a constant $c > 0$ such that the construction with $M(n) = n^{1+\delta}$ achieves (T, n^{-c})-security with $(1+\delta) \cdot \log n$ message size and $O(n^{2+2\delta})$ public information size for any $T = n^{o(\log n)}$.*

Proof. Proving correctness is straightforward. We therefore focus on security. We consider the simulator PSMSim that, on input $z \in \{0, 1\}$, performs the following operations:

1. Let H' be the 2-node graph that is connected if and only if $z = 1$.
2. $(G, H', \phi) \xleftarrow{\$} \mathcal{G}(N, 1/2, H')$
3. $m_0 \leftarrow \phi(1)$
4. $m_1 \leftarrow \phi(2)$
5. Output $(m_0, m_1, I = G)$

Let $x, y \in [n]$. We prove that no non-uniform $\big(T(n) \cdot \mathsf{poly}(n)\big)$-time adversary can distinguish between the encoding of (x, y) and the output of the simulator with advantage greater than $\varepsilon(n)$. We use a hybrid argument.

Hybrid 0. In this hybrid, we provide the adversary with an encoding of (x, y), i.e., with the triple (m_0, m_1, G) where

1. $(G, H, \phi) \xleftarrow{\$} \mathcal{G}(N, 1/2, H_n)$
2. $m_0 \leftarrow \phi(x)$
3. $m_1 \leftarrow \phi(y + n)$

Hybrid 1. In this hybrid, instead of planting H_n in G, we just plant the subgraph induced by the nodes x and $n + y$. In other words, we plant either the two node graph with just an edge (if $f_n(x, y) = 1$) or the two node graph with no edge. Formally, we provide the adversary with the triple (m_0, m_1, I) generated as follows

1. $H' \leftarrow \mathsf{Subgraph}\big(H_n, (x, n + y)\big)$
2. $(G, H, \phi) \xleftarrow{\$} \mathcal{G}(N, 1/2, H')$
3. $m_0 \leftarrow \phi(x)$
4. $m_1 \leftarrow \phi(y + n)$

This hybrid is (T, ε)-indistinguishable from the previous one due to the $(N, 2, T, \varepsilon)$-PSH assumption. Observe that, in Hybrid 1, the triple (m_0, m_1, I) is distributed as the output of $\mathsf{PSMSim}\big(\mathbb{1}^n, f(x, y)\big)$. This concludes the proof. $\square$

5.3 Privacy amplification of the PSM construction

We now try to amplify the privacy of the our PSM protocol from $\varepsilon = n^{-c}$ for a constant $c > 0$ to $\varepsilon = \mathsf{negl}(n)$. Unfortunately, techniques such as Yao's XOR lemma [Yao82, GNW11] do not seem to help us if we want to rely on the planted graph assumptions of Conjecture 4.6. A possible solution would be the approach used by Boyle *et al.* to amplify the privacy of programmable distributed point functions [BGIK22]. That would however lead to a PSM protocol with $\Omega(\log^2 n)$ message size. The same complexity could be achieved using one-way functions. Our construction expresses the graph H_n of the function as an exclusive-or of random graphs $R_1, \ldots, R_r$ and applies the PSM protocol to each R_j; however, we need to rely on non-standard assumptions for the security.

A PSM protocol with negligible privacy error and slightly superlogarithmic messages

Let $\mathcal{F} = (f_n)_{n\in\mathbb{N}}$ be a family of functions such that $f_n : [n] \times [n] \to \{0,1\}$. Let H_n be the graph representation of $\mathcal{F}$. Let $N : \mathbb{N} \to \mathbb{N}$ be a function such that $N(n) \geq 2n$ for every $n \in \mathbb{N}$. Let $r : \mathbb{N} \to \mathbb{N}$ be a function such that $r = \omega_n(1)$.

```
$\mathsf{Setup}(\mathbb{1}^n, f_n)$
    1. $\forall j \in [r-1]: \ R_j \xleftarrow{\$} \mathcal{G}(2n, 1/2)$.
    2. $R_r \leftarrow H_n \oplus R_1 \oplus \cdots \oplus R_{r-1}$.
    3. $\forall j \in [r]: \ (G_j, R_j, \phi_j) \xleftarrow{\$} \mathcal{G}(N, 1/2, R_j)$.
    4. Output $I := (G_1, \ldots, G_r)$ and $s_0 := s_1 := (\phi_1, \ldots, \phi_r)$.
$\mathsf{Encode}(\mathbb{1}^n, i, s_i = (\phi_1, \ldots, \phi_r), x)$
    1. $\forall j \in [r]: \ m_i^j = \phi_j(x + n \cdot i)$.
    2. Output $m_i := (m_i^1, \ldots, m_i^r)$.
$\mathsf{Output}\big(m_0 = (m_0^j)_{j\in[r]}, m_1 = (m_1^j)_{j\in[r]}, I = (G_j)_{j\in[r]}\big)$
    1. For every $j \in [r]$, let $z_j$ be 1 if $m_0^j$ and $m_1^j$ are adjacent in $G_j$. Otherwise, let $z_j$ be 0.
    2. Output $z \leftarrow z_1 \oplus \cdots \oplus z_r$.
```

Figure 2: A PSM protocol with negligible privacy error and slightly superlogarithmic messages.

We therefore present a candidate construction that we conjecture to achieve $\varepsilon = \mathsf{negl}(n)$ privacy error with $\omega_n(1) \cdot \log n$ message size. The scheme is formally described in Figure 2. Before explaining the idea, we need to introduce the following notation.

Definition 5.4 (Graph XORing)**.** *Let G_0 and G_1 be n-node graphs. We define $G_0 \oplus G_1$ as the n-node graph in which, for every $i, j \in [n]$, we draw the edge (i, j) if and only if such edge appears in only one among G_0 and G_1. Observe that the adjacency matrix of $G_0 \oplus G_1$ corresponds to the XOR of the adjacency matrices of G_0 and G_1.*

The idea at the base of our construction is rather simple: we generate a random additive secret-sharing scheme of the function f_n using $r = \omega_n(1)$ shares. The function is secret shared at graph level: we sample r random $2n$-node graphs whose XOR is H_n, the graph describing f_n. Then, we essentially apply the n^{-c}-secure PSM protocol described in Section 5.2 on each of the shares. After encoding the inputs x and y of the parties, it is possible to recover the evaluation of the shares on such values. By XORing the results of the evaluation, we obtain $f_n(x, y)$.

Theorem 5.5. *The construction in Figure 2 satisfies the PSM correctness. Moreover, the message size is $r \cdot \log N$ and the public information size is $r \cdot N(N-1)/2$.*

We conjecture that the construction in Figure 2 is $(\mathsf{poly}(n), \mathsf{negl}(n))$-secure according to Definition 5.1 when $N(n) \geq n^{2+\delta}$ for some constant δ; we do not know how to prove this conjecture using a more standard planting assumption.

Failed attacks. Observe that an adversary cannot hope to recover where R_j is hidden by simply looking at G_j, m_0^j and m_1^j. Indeed, R_j is never revealed: m_0^j and m_1^j are simply random nodes in a random graph G_j. Even more generally, for any proper subset $S \subsetneq [r]$, the tuple $(G_j, m_0^j, m_1^j)_{j \in S}$ is independent of the inputs x and y.

The most natural attack to the construction would therefore be to find permutations $\psi_1, \ldots, \psi_r \in \mathsf{Sym}(2n)$ such that $\psi_1(G_1) \oplus \cdots \oplus \psi_r(G_r)$ hides a copy of H_n. Such approach fails as only a negligible fraction of all $(\psi_1, \ldots, \psi_r)$ satisfies this property.

A more sophisticated attack is the *graph extension attack*: we build a large graph G whose nodes are in one-to-one correspondence with the tuples $(u_1, \ldots, u_r)$ where u_j is a node of G_j for every $j \in [r]$. We connect the nodes $(u_1, \ldots, u_r)$ and $(v_1, \ldots, v_r)$ in G if and only if $b_1 \oplus \cdots \oplus b_r = 1$, where b_j indicates where the edge (u_j, v_j) appears in G_j. Observe that G hides a copy of H_n. Furthermore, the nodes associated with the inputs x and y are exactly $(m_0^1, \ldots, m_0^r)$ and $(m_1^1, \ldots, m_1^r)$. An adversary could try to leverage the graph G to determine x and y. Luckily, for $r = \omega_n(1)$, the graph G has superpolymomial size: not only does this prevent a non-uniform polynomial-time adversary from reconstructing G, but it also makes H_n harder to recover for superpolynomial-time adversaries.

5.4 Offline-online non-interactive 2-input 3-pc with logarithmic communication

Using the PSM protocol with public information described in the previous subsection, we build an offline-online 2-input 3-party protocol, where the online phase is non-interactive and requires logarithmic communication. More formally, our protocol implements the functionality $\mathcal{F}_f$ described in Figure 3: the first two parties, Alice and Bob, provide inputs x and y. The last party, Carol, receives the output $f_n(x, y)$, where $f_n : [n] \times [n] \to \{0, 1\}$ is the function we want to compute. The construction withstands a semi-honest adversary corrupting a single player. Assuming the PSH conjecture (see Conjecture 4.6), for most functions $(f_n)_{n \in \mathbb{N}}$, the total communication in the online phase becomes $2(1 + \delta) \cdot \log n$ for an arbitrary small constant $\delta > 0$ (the privacy error is inverse-polynomial in n). If we want to achieve $\mathsf{negl}(n)$ privacy error, the communication complexity of the online phase becomes $\omega_n(1) \cdot \log(n)$. As far as we know, the only construction in the literature that achieves a similar lightweight online phase is the one-time truth table protocol of Ishai *et al.* [IKM+13]. Such solution would, however, require at least two rounds of communication. Our construction requires instead one.

THE 3-PC FUNCTIONALITY $\mathcal{F}_f$

Let $(f_n)_{n\in\mathbb{N}}$ be a family of functions such that $f_n : [n] \times [n] \to \{0,1\}$.
Evaluation: On input $x \in [n]$ from Alice and $y \in [n]$ from Bob, output $f_n(x,y)$ to Carol.

Figure 3: The 2-input 3-PC functionality $\mathcal{F}_f$.

THE 3-PC PROTOCOL Π_f

Let $(f_n)_{n\in\mathbb{N}}$ be a family of functions such that $f_n : [n] \times [n] \to \{0,1\}$. Let $(\mathsf{Setup}, \mathsf{Encode}, \mathsf{Output})$ be a PSM protocol with public information for $(f_n)_{n\in\mathbb{N}}$.

Offline phase:

1. $(I, s_0, s_1) \overset{\$}{\leftarrow} \mathsf{Setup}(\mathbb{1}^n)$.
2. Provide s_0 to Alice, s_1 to Bob, and I to Alice, Bob, and Carol.

Online phase:
Let $x \in [n]$ be Alice's input and $y \in [n]$ be Bob's input.

1. Alice sends $m_0 \overset{\$}{\leftarrow} \mathsf{Encode}(\mathbb{1}^n, 0, s_0, x)$ to Carol.
2. Bob sends $m_1 \overset{\$}{\leftarrow} \mathsf{Encode}(\mathbb{1}^n, 1, s_1, y)$ to Carol.
3. Carol outputs $z \leftarrow \mathsf{Output}(m_0, m_1, I)$.

Figure 4: The 3-PC protocol Π_f.

The protocol is formally described in Figure 4. The idea is very simple: in the offline phase, a trusted dealer runs the setup for a PSM protocol with public information. It distributes the secret information s_0 and s_1 to Alice and Bob. Carol instead receives the public information I. In the online phase, Alice and Bob just need to encode their inputs using the PSM protocol and send the resulting messages to Carol. Observe that, by the security of the PSM protocol, Carol can reconstruct the output without learning any other information. Alice and Bob instead learn no information about the other party's input as they receive no communication.

Theorem 5.6. *Assume the existence of authenticated and private point-to-point channels among the parties. Let $T : \mathbb{N} \to \mathbb{N}$ be a time bound and let $\varepsilon : \mathbb{N} \to [0,1]$ be an indistinguishability bound. Let $(f_n)_{n\in\mathbb{N}}$ be a family of functions such that $f_n : [n] \times [n] \to \{0,1\}$. Let $(\mathsf{Setup}, \mathsf{Encode}, \mathsf{Output})$ be a (T, ε)-secure PSM protocol with public information for $(f_n)_{n\in\mathbb{N}}$. Then, the protocol Π_f (see Figure 4) is an offline-online non-interactive protocol that (T, ε)-implements[7] the functionality $\mathcal{F}_f$ (see Figure 3) against a semi-honest adversary corrupting at most one party. Moreover, the total communication complexity in the online phase is at most twice the message size of the PSM protocol.*

Proof. The protocol is trivially secure against a corrupted semi-honest Alice or Bob (indeed, Alice

[7] I.e. for sufficiently large n, no non-uniform $(T(n) \cdot \mathsf{poly}(n))$-time adversary can distinguish the protocol from the simulation with advantage greater than $\varepsilon(n)$.

and Bob receive no communication). We can therefore focus on the case in which Carol is corrupt. In such case, the simulator, after receiving the output z, simply runs the PSM simulator $(m_0, m_1, I) \overset{\$}{\leftarrow} \mathsf{PSMSim}(\mathbb{1}^n, z)$. It then provides the result to the adversary. Observe that if any $\big(T(n) \cdot \mathsf{poly}(n)\big)$-time adversary distinguishes between the protocol and the simulation with advantage greater than $\varepsilon(n)$, then it breaks the security of the PSM protocol. □

6 Secret Sharing with Logarithmic Share Size

In a secret sharing for a forbidden graph G the parties are the vertices of the graph, every pair of parties connected by an edge should learn the secret, and every pair of parties not connected by an edge should get no information on the secret. Using the PRSH assumption, we show how to obtain $O(\log n)$ share size (with inverse-polynomial distinguishing advantage). We start by formalizing the definition of secret-sharing scheme with public information. We then present our constructions and prove their security under the assumptions of Section 4.

6.1 Secret-sharing schemes with public information

As already mentioned in the introduction, the goal of this work is to improve the complexity of secret-sharing schemes. We do this by differentiating between secret information and public information: in many settings, the cost of storing private information is indeed higher than the one of public information. So, is it possible to design computational secret-sharing schemes in which the information given to each party consists of a large public part and a private part smaller than the share of any known construction?

In this subsection, we formalize the notion of secret-sharing scheme with public information. As standard secret-sharing schemes, the primitive allows sharing a secret x between n parties. In order to reconstruct the secret, the parties need to collaborate by pooling their shares together. Whether the reconstruction succeeds or fails depends on the subset of parties that collaborate: the scheme is associated with an access structure. The latter describes which subsets succeeds in the reconstruction and which are guaranteed to learn no information about the secret.

Definition 6.1 (Access structures and promise access structures)**.** *An n-party* access structure *is a family of sets $(\mathcal{Q}_n)_{n\in\mathbb{N}}$ where, for every $n \in \mathbb{N}$,*

- $S \subseteq [n]$ *for every* $S \in \mathcal{Q}_n$ *and*
- *if* $S_1 \subseteq S_2$ *and* $S_1 \in \mathcal{Q}_n$*, then* $S_2 \in \mathcal{Q}_n$.

An n-party promise access structure *is a family of sets $(\mathcal{Q}_n, \mathcal{F}_n)_{n\in\mathbb{N}}$ where, for every $n \in \mathbb{N}$,*

- $\mathcal{Q}_n \cap \mathcal{F}_n = \emptyset$,
- $S \subseteq [n]$ *for every* $S \in \mathcal{Q}_n \cup \mathcal{F}_n$,
- *if* $S_1 \in \mathcal{Q}_n$ *and* $S_2 \in \mathcal{F}_n$*, then* $S_1 \not\subseteq S_2$.

The set $\mathcal{Q}_n$ represents the subset of parties that succeed in recovering the secret (i.e., the subsets of qualified parties). On the other hand, the set $\mathcal{F}_n$ represents the subsets of parties that should get no information about the secret (i.e., the forbidden subsets of parties). Observe that there may be subsets of $[n]$ that are neither in $\mathcal{Q}_n$ nor $\mathcal{F}_n$. In these cases, the secret-sharing scheme does not give any guarantee: the parties might be able to recover the full secret, they might get only some leakage or no information at all.

In this work, we consider a particular type of access structure called *forbidden graph* access structured.

Definition 6.2 (Forbidden graph access structures [SS97])**.** *An n-party forbidden graph access structure is a family of sets $(\mathcal{Q}_n, \mathcal{F}_n)_{n\in\mathbb{N}}$ such that*

- $\mathcal{Q}_n \cup \mathcal{F}_n = \left\{S \subseteq [n] \middle| |S| \leq 2\right\}$.
- $|S| = 2$ *for every* $S \in \mathcal{Q}_n$.

As the name suggests, a forbidden graph access structure can be represented as an n-node graph, where we draw an edge between nodes i and j if and only if $\{i, j\} \in \mathcal{Q}_n$. In other words, each party is associated with a node. A pair of parties can reconstruct the secret only when their nodes are connected by an edge and each party is forbidden.

We define secret-sharing schemes with public information. The definition is analogous to that of any secret-sharing scheme with the exception that the sharing algorithm outputs public information I along with the shares. The public information is disclosed to all the parties and is it used for the reconstruction.

Definition 6.3 (Secret-sharing schemes with public information)**.** *Let $T : \mathbb{N} \to \mathbb{N}$ be a time bound and let $\varepsilon : \mathbb{N} \to [0, 1]$ be an indistinguishability bound. A (T, ε)-secure secret-sharing scheme with public information for the promise access structure $(\mathcal{Q}_n, \mathcal{F}_n)_{n\in\mathbb{N}}$ is a pair of uniform PPT algorithms* $(\mathsf{Share}, \mathsf{Recover})$ *with the following syntax:*

- Share *is a randomized algorithm that takes as input $\mathbb{1}^n$ and a secret $x \in \{0, 1\}$. Its output is public information I and n strings, called shares, $s_1, \ldots, s_n$, one for each party.*
- $\mathsf{Recover}$ *is a deterministic algorithm that takes as input $\mathbb{1}^n$, a set $S \subseteq [n]$, public information I, and shares $(s_i)_{i\in S}$. Its output is a value $x' \in \{0, 1\} \cup \{\bot\}$.*

We require the following properties:

Perfect correctness. *For every $n \in \mathbb{N}$, $S \in \mathcal{Q}_n$, $b \in \{0, 1\}$, and random string r,*

$$\text{If } (I, s_1, \ldots, s_n) \leftarrow \mathsf{Share}(\mathbb{1}^n, b; r), \text{ then } \mathsf{Recover}\big(\mathbb{1}^n, S, I, (s_i)_{i\in S}\big) = b.$$

Security. *For every non-uniform $\big(T(n) \cdot \mathsf{poly}(n)\big)$-time adversary $\mathcal{A}$, sufficiently large n, and $S \in \mathcal{F}_n$,*

$$\left|\Pr\left[\mathcal{A}\big(\mathbb{1}^n, I, (s_i)_{i\in S}\big) = b \middle| \begin{array}{l} b \xleftarrow{\$} \{0,1\} \\ (I, s_1, \ldots, s_n) \xleftarrow{\$} \mathsf{Share}(\mathbb{1}^n, b) \end{array}\right] - \frac{1}{2}\right| \leq \varepsilon(n).$$

We define the share of the scheme as

$$\ell(n) := \max_{i\in[n], x\in\{0,1\}, r} \left\{|s_i| \middle| (I, s_1, \ldots, s_n) \leftarrow \mathsf{Share}(\mathbb{1}^n, x;\, r)\right\}.$$

Remark 6.4. We next discuss some variants of Definition 6.3.

- In Definition 6.3, we define the domain of secrets as $\{0,1\}$. To share a secret from a larger domain, one can share each bit of the secret independently. Furthermore, one can define and directly construct secret-sharing schemes with domains of secrets $(D_n)_{n\in\mathbb{N}}$, where the domain of the secrets with parameter $\mathbb{1}^n$ is D_n.

- In Definition 6.3, we define secret-sharing schemes with perfect correctness, as all the schemes we construct have this property. For constructions, one can also consider schemes which have a negligible error. In our lower bounds, we rule out weaker secret-sharing schemes that are allowed a non-negligible error probability. This only makes our lower bounds stronger.

- One can define secret-sharing schemes that are only secure against uniform adversaries. Our constructions are secure against uniform adversaries under the weaker assumptions that the planted subgraph with hints assumptions are only secure against uniform adversaries.

6.2 Forbidden graph secret sharing from planted random subgraphs

We now present a forbidden graph secret-sharing scheme based on the PRSH problem. We first informally describe a simpler version of the secret-sharing scheme, which is similar to the PSM protocol described in Section 5.2. Given a graph $Q_n = \big([n], E_Q\big)$ representing a forbidden graph access structure, we construct a graph $H = ([n], E)$, where if $(i,j) \in E_Q$, then $(i,j) \in E$ and if $(i,j) \notin E_Q$, then $(i,j) \in E$ with probability $1/2$. When the secret is 1, we plant H in a random graph with $N \gg n$ nodes and when the secret is 0 we plant $\overline{H}$ in the random graph. Let $G_1 = ([N], E_1)$ be the N-node graph with a copy of H or $\overline{H}$. The share of party i is $\phi(i)$ and the public information is G_1. The reconstruction of the secret by a set $\{i,j\} \in Q_n$ (i.e., $(i,j) \in E_Q$ and (i,j) is an edge in H) is simple: if $(\phi(i),\phi(j)) \in E_1$, output 1 and otherwise output 0. On the other hand, if $\{i,j\} \notin Q_n$, then (i,j) is an edge in H with probability $1/2$ hence an edge in $\overline{H}$ with probability $1/2$; it can be shown that by the Weak-PSH assumption, the parties cannot distinguish if H or $\overline{H}$ is planted in G_1 (although they have a hint of two nodes). Our construction is more complicated as we want to rely on the PRSH assumption. The construction is formally described in Figure 5. The scheme uses the procedure defined below.

Definition 6.5. *Let $H = \big([n], E_0\big)$ and $Q_n = \big([n], E_1\big)$ be graphs with n-nodes. We define the distribution* $\mathsf{Cut}(H, Q)$ *as follows:*

1. *$G \leftarrow H$.*
2. *For every $i,j \in [n]$ such that $i \neq j$, if (i,j) does not appears in Q_n, remove (i,j) from G (if the edge exists).*
3. *Output G.*

Let Q_n denote the access structure graph. In the scheme, the public information consists of two graphs G_0 and G_1. The first graph has n nodes, the second one is instead larger: it has $N \gg n$ nodes. We start from an N-node graph G that hides a random graph H. The smaller graph G_0 is obtained by comparing H to Q_n: we draw the edge (i,j) in G_0 if and only if (i,j) appears in both Q_n and H. When the secret is 1, G_1 is equal to G, otherwise, G is equal to the complement $\overline{G}$. In other words, if $b = 1$, G_1 hides a subgraph isomorphic to G_0. When the secret is 0, G_0 is instead hidden in the complementary graph. The private share of each party consists of the corresponding node in the hidden subgraph. Observe that the share size is $\log N = O(\log n)$.

```
FORBIDDEN GRAPH SECRET-SHARING SCHEMES BASED ON PLANTED RANDOM SUBGRAPHS
Let $Q_n$= $([n], E)$ be the graph representing the access structure.
$\mathsf{Share}(\mathbb{1}^n, x)$:
    1. $H \xleftarrow{\$} \mathcal{G}(n, 1/2)$.
    2. $G_0 \xleftarrow{\$} \mathsf{Cut}(H, Q_n)$.
    3. $(G_1, H, \phi) \xleftarrow{\$} \mathcal{G}(N, 1/2, H)$.
    4. If $x = 1$, output $I := (G_0, G_1)$ and, for every $i \in [n]$, $s_i := \phi(i)$.
    5. If $x = 0$, output $I := (G_0, \overline{G}_1)$ and, for every $i \in [n]$, $s_i := \phi(i)$.
$\mathsf{Recover}\big(\mathbb{1}^n, S = (i, j), I, (s_k)_{k \in S}\big)$
    1. If $|S| \neq 2$ or $(i, j) \notin E$, then output $\perp$.
    2. Rewrite $I$ as $(G_0, G_1)$ where $G_0$ and $G_1$ are graphs with $n$ and $N$ nodes respectively.
    3. Output 1 if $(s_i, s_j)$ is an edge in $G_1$ and $(i, j)$ is an edge in $G_0$. Output the same value
       if $(s_i, s_j)$ is not an edge in $G_1$ and $(i, j)$ is not an edge in $G_0$. Otherwise, output 0.
```

Figure 5: A forbidden graph secret-sharing scheme based on planted random subgraphs.

Theorem 6.6. *If the $(N, 2, T, \varepsilon)$-PRSH assumption holds, the construction in Figure 5 is a $(T, 2\varepsilon)$-secure forbidden graph secret-sharing scheme with $\log N$ share size and $N(N-1)/2 + n(n-1)/2$ public information size. Moreover, assume that PRSH conjecture holds (see Conjecture 4.6), then, for every constant $\delta > 0$, there exits a constant $c > 0$ such that the construction achieves (T, n^{-c})-security with $(1+\delta) \cdot \log n$ share size and $O(n^{2+2\delta})$ public information size for every $T = n^{o(\log n)}$.*

Remark 6.7. Under the more conservative Weak-PSH conjecture (see Conjecture 4.6), for every $\delta > 0$, the construction achieves (T, n^{-c})-security with $(2+\delta) \cdot \log n$ share size and $O(n^{4+2\delta})$ public information size for any $T = n^{o(\log n)}$.

Proof. First, suppose that i and j are allowed to reconstruct. If the secret is 1, the edges (i, j) and (s_i, s_j) will appear in both G_0 and G_1 or in none of them. If instead the secret is 0, only one of the edges (i, j) and (s_i, s_j) will appear. Thus, the scheme is perfectly correct.

We next focus on security. Suppose there exists a non-uniform $\big(T(n) \cdot \mathsf{poly}(n)\big)$-time adversary $\mathcal{A}$ that breaks the security of the secret-sharing scheme. Then, there exist a subsequence $(n_k)_{k \in \mathbb{N}}$ and sets $(S_n)_{n \in \mathbb{N}}$ such that $S_n \in \mathcal{F}_n$, $|S_n| \leq 2$ and

$$\Pr\left[\mathcal{A}\big(\mathbb{1}^{n_k}, I, (s_i)_{i \in S_{n_k}}\big) = b \,\middle|\, \begin{array}{l} b \xleftarrow{\$} \{0,1\} \\ (I, s_1, \ldots, s_{n_k}) \xleftarrow{\$} \mathsf{Share}(\mathbb{1}^{n_k}, b) \end{array}\right] > \frac{1}{2} + 2\varepsilon(n_k)$$

for every $k \in \mathbb{N}$.

We can use $\mathcal{A}$ to attack the PRSH assumption. We consider indeed the adversary $\mathcal{B}$ that, given graphs H and G with n and N nodes respectively and nodes $(u_i)_{i \in S_n}$, performs the following operations:

1. If $n \neq n_k$ for every $k \in \mathbb{N}$, it outputs 0.
2. It samples $b \overset{\$}{\leftarrow} \{0,1\}$.
3. $G_0 \overset{\$}{\leftarrow} \mathsf{Cut}(H, Q_n)$.
4. If $b = 1$, it runs $\mathcal{A}$ on input $I := (G_0, G_1 = G)$ and $(u_i)_{i \in S_n}$.
5. If $b = 0$, it runs $\mathcal{A}$ on input $I := (G_0, G_1 = \overline{G})$ and $(u_i)_{i \in S_n}$.
6. If $\mathcal{A}$ guesses b, $\mathcal{B}$ outputs 1, otherwise, it outputs 0.

Observe that if H is not planted in G, the view of $\mathcal{A}$ is independent of b. So, $\mathcal{A}$ guesses its value exactly with probability $1/2$. If instead G hid H, the view of $\mathcal{A}$ is as in the secret-sharing security game. So, $\mathcal{A}$ guesses b with probability greater than $1/2 + 2\varepsilon(n_k)$ for every $n = n_k$. The advantage of $\mathcal{B}$ against the PRSH assumption would therefore be greater $\varepsilon(n_k)$ for every n_k. Since the adversary $\mathcal{B}$ runs in time $T(n) \cdot \mathsf{poly}(n)$, the existence of $\mathcal{A}$ contradicts the $(N, 2, T, \varepsilon)$-PRSH assumption. This ends the proof. □

Corollary 6.8. *Let $T : \mathbb{N} \to \mathbb{N}$ be a time bound. Let $\varepsilon(n) = n^{-c}$ for a constant $c > 0$ and $N = \mathsf{poly}(n)$. If the $(N, 2, T, \varepsilon)$-PRSH assumption holds, then there exists a (T, ε')-secure forbidden graph secret-sharing scheme with public information where $\varepsilon' = \mathsf{negl}(n)$, the share size is $\omega_n(1) \cdot \log(n)$ and the public information size is $\omega_n(1) \cdot O(N^2)$.*

Proof. If the $(N, 2, T, \varepsilon)$-PRSH assumption holds, by Theorem 6.6, there exists a forbidden graph secret-sharing scheme with public information with (T, n^{-c})-security and $O(\log n)$ share size. Using Yao's XOR lemma with $r = \omega_n(1)$ repetitions, we can obtain a secret-sharing scheme with $\omega_n(1) \cdot \log n$ share size and $(T, \mathsf{negl}(n))$-security. The scheme consists in generating a random additive secret sharing of the secret $x = x_1 \oplus \cdots \oplus x_r$ and then secret-share each bit x_i among the n parties using the (T, n^{-c})-secure scheme with public information. □

Remark 6.9. By generalizing the hidden-subgraph assumption to multigraphs it is possible to design secret-sharing schemes with public information for more general slice access structures. For example we can consider a promise 3-slice access structure, where $\mathcal{Q}_n \cup \mathcal{F}_n = \{S \subset [n] | |S| = 3\}$, represent it by a 3-hypergraph Q_n, and plant an n-node graph obtained from Q_n in a bigger random 3 hypergraph.

Why did we not build PSM under the PRSH assumption? The PSM construction we described in Section 5.2 relies on the PSH assumption (see Conjecture 4.6). As a consequence, we achieved $(1 + \delta) \cdot \log n$ message size only for most but not all functions $f_n : [n] \times [n] \to \{0,1\}$. A natural question is whether we could have used the techniques in this section to obtain $(1+\delta) \cdot \log n$ message size for all functions under the PRSH assumption. For instance, instead of hiding a copy of H_n in the public graph G, we could have hidden a random graph R and published $R \oplus H_n$ along with G. It would be possible to prove the security of such construction under the PRSH assumption. However, this solution would have achieved $(2 + \varepsilon) \cdot \log n$ message size: in order to encode its input, a party would need to communicate two nodes, one in the big graph G and one in $R \oplus H_n$. In the context of secret sharing, one of the two nodes is instead public, so we can achieve lower complexity.

7 On Breaking the $\log n$ Barrier for 2-out-of-n Secret Sharing

In Section 6, we observed that, for forbidden graph access structures, secret-sharing schemes with public information can achieve better complexity than any previously known construction. One could ask whether, with a similar approach, it is possible to obtain schemes with share size strictly smaller than $\log n$. We study this question for a 2-out-of-n secret-sharing scheme with public information, i.e., a secret-sharing scheme for the access structure $(\mathcal{Q}_n)_{n\in\mathbb{N}}$, where $\mathcal{Q}_n = \{S \subseteq [n] || S| \geq 2\}$ for every $n \in \mathbb{N}$ (that is, the minimal authorized sets are all sets of size 2). Unfortunately, we could not answer this question even under the same type of conjectures we made in previous sections. However, in this section, we study the problem and come up with sufficient and necessary conditions for a positive answer.

7.1 An equivalent formulation via planting in multipartite graphs

We start by showing the equivalence between 2-out-of-n secret-sharing schemes with $O(\log n)$ share size and a multipartite version of the planted clique problem.

Lemma 7.1. *There exists a 2-out-of-n secret-sharing scheme with $\ell = O(\log n)$ share size, one-bit secret, and (T, ε)-security if and only if there exists a pair of distributions $\mathcal{D}_0$, $\mathcal{D}_1$ satisfying the following properties:*

- *For $b \in \{0, 1\}$, $\mathcal{D}_b(\mathbb{1}^n)$ outputs an n-partite graph G with 2^ℓ nodes per part. In addition, $\mathcal{D}_b(\mathbb{1}^n)$ outputs a node u_i in each part U_i of the graph.*
- *The nodes $u_1, \ldots, u_n$ output by $\mathcal{D}_0$ are an independent set of G with probability 1.*
- *The nodes $u_1, \ldots, u_n$ output by $\mathcal{D}_1$ are a clique of G with probability 1.*
- *For every non-uniform $\big(T(n) \cdot \mathsf{poly}(n)\big)$-time adversary $\mathcal{A}$, sufficiently large n, and $i \in [n]$,*

$$\left| \Pr\left[\mathcal{A}(\mathbb{1}^n, G, u_i) = b \,\middle|\, \begin{matrix} b \xleftarrow{\$} \{0,1\} \\ (G, u_1, \ldots, u_n) \xleftarrow{\$} \mathcal{D}_b(\mathbb{1}^n) \end{matrix} \right] - \frac{1}{2} \right| \leq \varepsilon(n).$$

Proof. Suppose that there exist distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ as the one described in the theorem. We build a 2-out-of-n secret-sharing scheme with public information as follows.

- $\mathsf{Share}(\mathbb{1}^n, b)$: compute $(G, u_1, \ldots, u_n) \xleftarrow{\$} \mathcal{D}_b(\mathbb{1}^n)$. Output $I := G$ and $s_i := u_i$ for every $i \in [n]$.
- $\mathsf{Recover}\big(\mathbb{1}^n, S, G, (s_i)_{i\in S}\big)$: if $|S| \leq 1$, output $\perp$. Otherwise, output 1 if and only if there exist $i, j \in S$ such that s_i and s_j are adjacent in G.

It is trivial to see that the construction is correct and (T, ε)-secure. Furthermore, since u_i is hidden among $|U_i| = 2^\ell$ nodes, it can be encoded using ℓ bits.

We now prove the opposite implication. Consider a (T, ε)-secure secret-sharing scheme with shares of size ℓ. For every $b \in \{0, 1\}$, we define the distribution $\mathcal{D}_b(\mathbb{1}^n)$ as follows:

1. $(I, s_1, \ldots, s_n) \overset{\$}{\leftarrow} \mathsf{Share}(\mathbb{1}^n, b)$.
2. Draw an n-partite graph $G = (U_1, \ldots, U_n, E)$ with 2^ℓ nodes per part, where the i-th part is $U_i = \{\, (i, s) | s \in \{0,1\}^\ell \,\}$.
3. For every $i \neq j$ and $s, s' \in \{0,1\}^\ell$, add $((i, s), (j, s')) \in E$ iff $\mathsf{Recover}(\mathbb{1}^n, \{i, j\}, I, \{s, s'\}) = 1$.
4. Output $(G, (1, s_1), \ldots, (n, s_n))$.

By the perfect correctness of the secret-sharing scheme, $(1, s_1), \ldots, (n, s_n)$ form an independent set with probability 1 when $b = 0$. If instead $b = 1$, $(1, s_1), \ldots, (n, s_n)$ form a clique. It is also immediate to see that if a non-uniform $\big(T(n) \cdot \mathsf{poly}(n)\big)$-time adversary can guess b with probability greater than $1/2 + \varepsilon(n)$ given only G and (i, s_i) for any $i \in [n]$, then we would be able to break the (T, ε)-security of the secret-sharing scheme. □

Planting in random graphs does not work. Unfortunately, if $\mathcal{D}_1$ is the distribution that plants a clique in a random multipartite graph and $\mathcal{D}_0$ is the distribution that outputs the complementary graph, we cannot hope to obtain a $(T, o_n(1))$-secure scheme with share size smaller than $\log n$. Indeed, we would be planting an n-node clique in a random graph with $N < n^2$ nodes. As mentioned in Section 4, in this setting, it is possible to recover the clique with $\Omega_n(1)$ probability by just picking the nodes with the highest degree. The task becomes even easier when the graph is multipartite and we leak one of the nodes.

7.2 An equivalent formulation via planting in general graphs

We next provide a variant of Lemma 7.1 that applies to general rather than multipartite graphs: We show that there exists a 2-out-of-n secret-sharing scheme with public information and share size $\delta \cdot \log n$ for some constant δ if and only if for every $N \in \mathbb{N}$ there is a distribution on N-node graphs that hide (not necessarily unique) N^β-sized clique and N^β-sized independent set for some $\beta > 0.5$ and it is hard to distinguish a random node in the clique from a random node in the independent set.

Theorem 7.2 (A necessary and sufficient condition)**.** *Let $T : \mathbb{N} \to\to \mathbb{N}$ be time bound and $\varepsilon : \mathbb{N} \to [0, 1]$ be an indistinguishability bound. There exists, for some constant $0 < \delta < 1$ and some negligible function $\mathsf{negl}_1(n)$, a 2-out-of-n secret-sharing scheme with $\delta \cdot \log n$ share size and $(T(n), \varepsilon(n) + \mathsf{negl}_1(n))$-security if and only if there exists a constant $1/2 < \beta < 1$ and an ensemble of distributions $(\mathcal{D}(\mathbb{1}^n))_{n \in \mathbb{N}}$ outputting a tuple $(G, x_1, \ldots, x_n, y_1, \ldots, y_n)$ with the following properties:*

- *G is an N-node graph, where $n \geq N^\beta$.*
- *$x_1, \ldots, x_n$ form an n-sized clique of G.*
- *$y_1, \ldots, y_n$ form an n-sized independent set of G.*
- *For some negligible function $\mathsf{negl}_2(n)$ for every non-uniform $\big(T(n) \cdot \mathsf{poly}(n)\big)$-time polynomial-*

time adversary $\mathcal{A}$ and sufficiently large n,

$$\left| \Pr\left[\mathcal{A}(\mathbb{1}^n, G, z) = b \middle| \begin{array}{l} b \xleftarrow{\$} \{0,1\} \\ i \xleftarrow{\$} [n] \\ \big(G, (x_j, y_j)_{j\in[n]}\big) \xleftarrow{\$} \mathcal{D}(\mathbb{1}^n) \\ \textit{If } b = 1 : z \leftarrow x_i \\ \textit{If } b = 0 : z \leftarrow y_i \end{array} \right] - \frac{1}{2} \right| \leq \varepsilon(n) + \mathsf{negl}_2(n).$$

The proof of Theorem 7.2 is implied by Lemmas 7.3 and 7.4.

Lemma 7.3 (Necessary condition)**.** *Suppose there exists a 2-out-of-n secret-sharing scheme with $\delta \cdot \log n$ share size and (T, ε)-security. Then, there exists a distribution $\mathcal{D}(\mathbb{1}^n)$ outputting a tuple $(G, x_1, \ldots, x_n, y_1, \ldots, y_n)$ with the following properties:*

- *G is a $2n^{1+\delta}$-node graph.*
- *$x_1, \ldots, x_n$ form an n-sized clique of G.*
- *$y_1, \ldots, y_n$ form an n-sized independent set of G.*
- *For every non-uniform $\big(T(n) \cdot \mathsf{poly}(n)\big)$-time polynomial-time adversary $\mathcal{A}$ and sufficiently large n,*

$$\left| \Pr\left[\mathcal{A}(\mathbb{1}^n, G, z) = b \middle| \begin{array}{l} b \xleftarrow{\$} \{0,1\} \\ i \xleftarrow{\$} [n] \\ \big(G, (x_j, y_j)_{j\in[n]}\big) \xleftarrow{\$} \mathcal{D}(\mathbb{1}^n) \\ \textit{If } b = 1 : z \leftarrow x_i \\ \textit{If } b = 0 : z \leftarrow y_i \end{array} \right] - \frac{1}{2} \right| \leq \varepsilon(n).$$

Notice that, if we denote the number of nodes in G by N, then it is necessary that the size of the clique and independent set is N^β for any constant $\beta > 1$.

Proof. If there exists a 2-out-of-n secret-sharing scheme with $\delta \cdot \log n$ share size and (T, ε)-security, then, there exist two distribution $\mathcal{D}_0$ and $\mathcal{D}_1$ satisfying the properties described in Lemma 7.1. Both of them output graphs with $n^{1+\delta}$ nodes. In the first case, the graph hides an n-sized independent set. In the other case, it hides an n-sized clique. Moreover, the graphs are (T, ε)-indistinguishable even if we reveal one of the nodes in the clique or the independent set respectively.

We build our distribution $\mathcal{D}$ by "glueing" the graphs G_0 and G_1 output by $\mathcal{D}_0$ and $\mathcal{D}_1$ respectively. After that, we permute the nodes. Along with the graph, the distribution outputs the nodes on the clique of G_1 and the nodes on the independent set of G_0. It is easy to see that $\mathcal{D}$ satisfies the right property. □

Lemma 7.4 (Sufficient condition)**.** *Suppose that there exists a distribution $\mathcal{D}(\mathbb{1}^n)$, functions $T : \mathbb{N} \to \mathbb{N}$ and $\varepsilon : \mathbb{N} \to [0,1]$, and constants $0 < \gamma < \delta < 1$ with the following properties:*

- *$\mathcal{D}(\mathbb{1}^n)$ outputs an $n^{1+\delta}$-node graph G along with $2n^{1+\gamma}$ nodes $x_1, \ldots, x_{n^{1+\gamma}}$ and $y_1, \ldots, y_{n^{1+\gamma}}$.*

- *$x_1, \dots, x_{n^{1+\gamma}}$ form an $n^{1+\gamma}$-sized clique in G.*
- *$y_1, \dots, y_{n^{1+\gamma}}$ form an $n^{1+\gamma}$-sized independent set in G.*
- *For every non-uniform $\big(T(n) \cdot \mathsf{poly}(n)\big)$-time adversary $\mathcal{A}$ and sufficiently large n,*

$$\left| \Pr\left[\mathcal{A}(\mathbb{1}^n, G, z) = b \;\middle|\; \begin{array}{l} b \xleftarrow{\$} \{0,1\} \\ m \xleftarrow{\$} [n^{1+\gamma}] \\ \big(G, (x_j, y_j)_{j \in [n^{1+\gamma}]}\big) \xleftarrow{\$} \mathcal{D}(\mathbb{1}^n) \\ \textit{If } b = 1 : z \leftarrow x_m \\ \textit{If } b = 0 : z \leftarrow y_m \end{array} \right] - \frac{1}{2} \right| \leq \varepsilon(n).$$

Then, there exists an (T, ε')-secure 2-out-of-n secret-sharing scheme with public information having $\delta \cdot \log n$ share size where $\varepsilon' = \varepsilon(n) + \mathsf{negl}(n)$.

Proof. Let $\mathsf{Permute}(G)$ be the algorithm that, on input an N-node graph, outputs a pair (G', ϕ) where $\phi \xleftarrow{\$} \mathsf{Sym}(N)$ and G' is obtained by permuting the nodes of G according to ϕ. In particular, the edge (i,j) appears in G if and only if $(\phi(i), \phi(j))$ appears in G'.

We construct the algorithm $\mathsf{Share}(\mathbb{1}^n, b)$ as follows:

1. $\big(G, (x_j, y_j)_{j \in [n^{1+\gamma}]}\big) \xleftarrow{\$} \mathcal{D}(\mathbb{1}^n)$.
2. $(G', \phi) \xleftarrow{\$} \mathsf{Permute}(G)$.
3. For every $i \in [n]$, define the bucket $B_i := \{n^\delta \cdot (i-1) + 1, \dots, i \cdot n^\delta\}$.
4. If there is some $i \in [n]$ such that $B_i \cap \big\{\phi(x_j) \big| j \in [n^{1+\gamma}]\big\} = \emptyset$ or $B_i \cap \big\{\phi(y_j) \big| j \in [n^{1+\gamma}]\big\} = \emptyset$, then let $s_1 = \dots = s_n = b$, output $I =$ "FAIL", $s_1, \dots, s_n$, and halt.
5. For every $i \in [n]$, if $b = 1$, set $s_i \xleftarrow{\$} B_i \cap \big\{\phi(x_j) \big| j \in [n^{1+\gamma}]\big\}$.
6. For every $i \in [n]$, if $b = 0$, set $s_i \xleftarrow{\$} B_i \cap \big\{\phi(y_j) \big| j \in [n^{1+\gamma}]\big\}$.
7. Output $I := G'$ and $s_1, \dots, s_n$.

In order to reconstruct the secret (when $I \neq$ "FAIL"), a pair of parties just needs to check their shares s_i and s_j and check if there exists an edge connecting them in G'. That occurs if and only if the secret is 1. Clearly, the scheme is perfectly correct.

It is easy to observe that the share size is $\delta \cdot \log n$. Indeed, for every $i \in [n]$, s_i is hidden in B_i which has n^δ elements.

We next analyze the security of the scheme. Let $\mathcal{E}$ be the event that there is some $i \in [n]$ such that $B_i \cap \big\{\phi(x_j) \big| j \in [n^{1+\gamma}]\big\} = \emptyset$ or $B_i \cap \big\{\phi(y_j) \big| j \in [n^{1+\gamma}]\big\} = \emptyset$ and let $\varepsilon''(n)$ be the probability of $\mathcal{E}$. We next provide a simple upper bound on $\varepsilon''(n)$. Fix $i \in [n]$ and consider the event that $B_i \cap \big\{\phi(x_j) \big| j \in [n^{1+\gamma}]\big\} = \emptyset$. We give an upper bound on the probability of this event using a "birthday paradox" analysis. The elements $\phi(x_1),\ldots,\phi(x_{n^{1+\gamma}})$ are chosen at random without replacements from $[n^{1+\delta}]$, thus,

$$\begin{aligned}\Pr[B_i \cap \{\phi(x_j) \big| j \in [n^{1+\gamma}]\} = \emptyset] &= \prod_{j=1}^{n^{1+\gamma}} \Pr[\phi(x_j) \notin B_i | \phi(x_1),\ldots,\phi(x_{j-1}) \notin B_i] \\ &= \prod_{j=1}^{n^{1+\gamma}} \left(1 - \frac{n^\delta}{n^{1+\delta}-j+1}\right) \\ &< \left(1-\frac{n^\delta}{n^{1+\delta}}\right)^{n^{1+\gamma}} \leq e^{-n^\gamma}.\end{aligned}$$

Thus, by a union bound, $\varepsilon''(n)$ – the probability of $\mathcal{E}$ – is negligible.

Suppose that there exists a non-uniform $\big(T(n)\cdot \mathsf{poly}(n)\big)$-time adversary $\mathcal{A}_{\mathrm{ss}}$ that breaks the security of the secret-sharing scheme with advantage greater than $\varepsilon'(n) := \varepsilon(n) + \varepsilon''(n)$. Let i be the index of the party corrupted by $\mathcal{A}_{\mathrm{ss}}$. We build an adversary $\mathcal{A}$ that can easily tell whether a point belongs to the clique or the independent set of the graph output by $\mathcal{D}$. The reduction is simple; given G and z, the adversary does the following:

- Permutes the nodes of G so that $\mathcal{E}$ does not occur and z ends in a random position of the bucket B_i.
- Provides $\mathcal{A}_{\mathrm{ss}}$ with the permuted graph and permuted z.
- Outputs the output of $\mathcal{A}_{\mathrm{ss}}$.

The view of $\mathcal{A}_{\mathrm{ss}}$ is the same as in the secret-sharing security game condition on $\mathcal{E}$ not occurring. The advantage of $\mathcal{A}_{\mathrm{ss}}$ in guessing b is strictly greater than $\varepsilon'(n) - \varepsilon''(n) \geq \varepsilon(n)$.

□

8 A Lower Bound for Threshold Secret Sharing

In this section, we prove a lower bound for computational 2-out-of-n secret-sharing with public information: the share size has to be at least $\frac{1}{5}\log\log n$.

The idea is rather simple: a 2-out-of-n secret-sharing scheme induces a 2-out-of-n' secret-sharing scheme for any $n' < n$. Since we can represent the public information as an n-partite graph (see Lemma 7.1), the size of the public information decreases as n' becomes smaller. We show that if the share size ℓ is smaller than $\frac{1}{5}\log\log n$, there exists an $n' > 2^{3/2\ell}$ for which the graph has size $O(\log n)$. At that point, the public information would be so small that the scheme is necessarily statistically secure. By Theorem A.2 (proved in Appendix A), that would imply that $\ell \geq \log n'$, reaching a contradiction.

Theorem 8.1. *Let $T : \mathbb{N} \to \mathbb{N}$ be an increasing time bound and let $\varepsilon : \mathbb{N} \to [0,1]$ be an indistinguishability bound that such that $\varepsilon(n) \leq 1/12$ for every $n \in \mathbb{N}$. For sufficiently large n, in every (T,ε)-secure 2-out-of-n secret-sharing scheme with public information and reconstruction error at most $1/(4\log^{3/5} T(n))$ the share size is at least $\frac{1}{5}\log\log T(n)$. In particular, if the scheme is secure against non-uniform polynomial-time adversaries, the share size is at least $\frac{1}{5}\log\log n$.*

Proof. Suppose that our claim is false for infinitely many values of n: let $(\mathsf{Share}, \mathsf{Recover})$ be a scheme that contradicts it. Let $\ell = \frac{1}{5}\log\log T(n)$ be an upper bound on its share size. We construct a 2-out-of-n' secret-sharing scheme where $n' = 2^{3\ell/2}$. In order to share $b \in \{0,1\}$, we perform the following operations.

- $(I, s_1, \ldots, s_n) \xleftarrow{\$} \mathsf{Share}(\mathbb{1}^n, b)$.
- Build an n'-partite graph $G = (U_1, \ldots, U_{n'}, E)$ with 2^ℓ nodes in each partition (similar to the construction in Lemma 7.1):
  - $U_i = \{(i,s) | s \in \{0,1\}^\ell\}$ for $1 \le i \le n'$.
  - For every $i \neq j$ and $s, s' \in \{0,1\}^\ell$, draw the edge $((i,s),(j,s'))$ if and only if
  $$\mathsf{Recover}(\mathbb{1}^n, \{i,j\}, I, \{s, s'\}) = 1.$$
- Output $(G, s_1, \ldots, s_{n'})$.

In order to reconstruct the secret, two parties i, j holding shares s_i, s_j respectively just need to check if $((i, s_i),(j, s_j))$ is in E.

The new scheme is a 2-out-of-n' secret-sharing scheme with the following properties:

- The reconstruction error is at most $1/(4\log^{3/5} T(n)) = 1/(4(n')^2)$,
- It is $\varepsilon(n)$-secure against non-uniform adversaries running in $T(n)$ time (as the public information G can be efficiently computed from the original public information I),
- The share size is $\ell = \frac{2}{3}\log n' < \log n' - 6$ for sufficiently large n (which depends on the function $T(\cdot)$),
- The public information has size $(n')^2 \cdot 2^{2\ell} = 2^{5\ell} = \log T$.

Since the share size in the scheme is smaller than $\log n' - 6$, by Theorem A.2 the scheme is not statistically secure, that is there exists a party i such that the statistical distance between the shares of the i-th party for the secret 0 and its shares for for the secret 0 is at least $1/12$. Consider a non-uniform adversary that given the public information I and the share s_i returns 1 if the probability of G, s_i given the secret 1 is at least the probability of I, s_i given the secret 0 and 0 otherwise; this adversary guesses the correct secret with probability at least $7/12$.

We show that a non-uniform $T(n) \cdot \mathsf{poly}(n)$-time adversary can break the $1/12$-security of the scheme of the original scheme (for infinitely many values of n). For every n for which the share size is at most $\frac{1}{5}\log\log T(n)$, the non-uniform has a party i as above and a table such that for every G, s_i states if the G, s_i are more likely given 1 or given 0. The adversary is given I, s_i, where I is the public information for the original scheme does the following:

- Builds the graph G for the n'-party secret-sharing scheme,
- If the probability of G, s_i given the secret 1 is at least the probability of G, s_i given the secret 0, then it returns 1; otherwise it returns 0.

Since the statistical distance between G, s_i for the secret 1 and the secret 0 is at least $1/12$, the adversary guesses the correct secret with probability at least $7/12$. Furthermore, the size of the table is at most $2^{\log T(n)} \cdot 2^\ell = T(n) \cdot \log^{1/5} T(n)$. This contradicts the $(T(n), 1/12)$-security of the scheme. □

Any $t(n)$-out-of-n secret-sharing scheme with public information implies a 2-out-of-$(n-t(n)+2)$ secret-sharing scheme with public information and with the same share size; this is done publishing shares of $t(n)-2$ parties as part of the public information. Thus, Theorem 8.1 implies a lower bound for $t(n)$-out-of-n secret-sharing schemes.

Corollary 8.2. *Let $t : \mathbb{N} \to \mathbb{N}$ be a function such that $2 \leq t(n) \leq n-1$. In every $t(n)$-out-of-n secret-sharing scheme with public information that is secure against non-uniform polynomial-time adversaries, the share size is at least $\frac{1}{5}\log\log(n-t(n)+2)$.*

9 No Computational-Statistical Gap with 1-Bit Shares

In this section, we address the most optimistic scenario – employing public information and achieving secure sharing by distributing single bit shares. We prove that this scenario is indeed too good to be true, that is, we prove that any n-party access structure $\mathcal{Q}$ can be realized by a secret-sharing schemes with public information and 1-bit shares if and only if the access structure can be realized by a perfect secret-sharing scheme with one-bit shares and without public information. Thus, there is no gap between computational and statistical secret-sharing scheme with 1-bit shares. Furthermore, public information does not help in this scenario.

In our result, we will consider an n-party access structure $\mathcal{Q}_n$, which we will denote by $\mathcal{Q}$, for some $n \in \mathbb{N}$, rather than considering a sequence $(\mathcal{Q}_n)_{n\in\mathbb{N}}$. The running time of the adversaries we construct for $\mathcal{Q}$ is polynomial in n; thus, our result will translate to a non-uniform polynomial-time against a sequence of access structure. Recall that a perfect (information-theoretic) secret-sharing scheme is a secret-sharing scheme as defined in Definition 6.3 with the following differences: (1) $(\mathsf{Share}, \mathsf{Recover})$ are not required to be efficient algorithms, (2) correctness should hold with probability 1, and (3) security should hold for unbounded adversaries and $\varepsilon(n) = 0$ (alternatively, the shares are equally distributed when the secret is 0 and when the secret is 1). We will use the notions of minimal qualified sets and maximal forbidden sets of an access structure, which we define below:

Definition 9.1. *The minimal qualified sets of an n-party access structure $\mathcal{Q}$, denoted by* $\min(\mathcal{Q})$*, is defined as*

$$Q \in \min(\mathcal{Q}) \text{ if and only if } Q \in \mathcal{Q} \text{ and } \nexists Q^* \in \mathcal{Q} \text{ s.t. } Q^* \subsetneq Q.$$

The maximal forbidden/unqualified sets of $\mathcal{Q}$, denoted is $\max\left(\overline{\mathcal{Q}}\right)$ *(where $\overline{\mathcal{Q}} = \left(2^{[n]} \setminus \mathcal{Q}\right)$), is defined as*

$$F \in \max\left(\overline{\mathcal{Q}}\right) \text{ if and only if } F \notin \mathcal{Q} \text{ and } \nexists F^* \notin \mathcal{Q} \text{ s.t. } F \subsetneq F^*.$$

The main result of this section is stated as follows:

Theorem 9.2. *An n-party access structure can be realized by a secret-sharing scheme with public information, 1-bit shares, $\varepsilon(n) = 0.03$-indistinguishability, and reconstruction error at most 0.01 if and only if it can be realized by a perfect secret-sharing scheme with 1-bit shares and without public information.*

Access structures that admit a perfect secret-sharing scheme with 1-bit shares are called binary ideal access structures. In [BC93, Mat95, Gol98], it was shown that an access structure $\mathcal{Q}$ admits a perfect secret-sharing scheme with one-bit shares and without public information – i.e., $\mathcal{Q}$ is binary ideal – if and only if $\mathcal{Q}$ is a port of a binary matroid. We devise an alternative characterization of binary ideal access structures to prove our result. Theorem 9.2 is directly implied by Lemma 9.3 and Lemma 9.7.

Lemma 9.3. *Le $\mathcal{Q}$ be an n-party access structure. If $|Q \setminus F|$ is odd for every minimal qualified $Q \in \min(\mathcal{Q})$ and maximal forbidden $F \in \max\left(\overline{\mathcal{Q}}\right)$, then $\mathcal{Q}$ can be realized by a linear perfectly secure secret-sharing scheme with 1-bit shares and without public information.*

Proof. Suppose the access structure satisfies the condition in the lemma. We construct a perfect secret-sharing scheme with public information and 1-bit shares and argue that the scheme is perfectly secure. By conditioning on any fixed public information, we obtain a perfect secret-sharing scheme with one bit shares and without public information.[8]

A secret-sharing scheme with public information and 1-bit shares is constructed as follows:

- For $b \in \{0,1\}$, let $\mathsf{Share}(b) = (I, r_1, \ldots, r_n)$, where $r_1, \ldots, r_n$ are uniform and independent bits, and I contains a bit I_Q for each $Q \in \min(\mathcal{Q})$, where $I_Q = b \oplus \bigoplus_{i \in Q} r_i$.

Perfect correctness is immediate; for any minimal qualified set $Q \in \min(\mathcal{Q})$,

$$\mathsf{Recover}(Q, I, (r_i)_{i \in Q}) = I_Q \oplus \bigoplus_{i \in Q} r_i.$$

To show that the scheme is perfectly secure, it suffices to show that the scheme is secure against an adversary corrupting any maximally unqualified set. Fix $F \in \max\left(\overline{\mathcal{Q}}\right)$. For all $Q \in \min(\mathcal{Q})$, $|Q \setminus F|$ is odd. Let $I, r_1, \ldots, r_n$ be shares and public information generated for a secret $b \in \{0,1\}$. Let $I' = I$ and $r'_1, \ldots, r'_n$ such that $r'_i = r_i$ if $i \in F$ and $r'_i = 1 \oplus r_i$ otherwise. We claim that $I' = I$ and $r'_1, \ldots, r'_n$ are sharing of $\bar{b}$: For every $Q \in \min(\mathcal{Q})$, since $|Q \setminus F|$ is odd,

$$\begin{aligned} I'_Q = I_Q &= b \oplus \bigoplus_{i \in Q} r_i \\ &= b \oplus \left(\bigoplus_{i \in Q \cap F} r'_i \right) \oplus \left(\bigoplus_{i \in Q \setminus F} (1 \oplus r'_i) \right) \\ &= b \oplus \left(\bigoplus_{i \in Q} r'_i \right) \oplus \left(\bigoplus_{i \in Q \setminus F} 1 \right) \\ &= b \oplus \bigoplus_{i \in Q} r'_i \oplus 1 = \bar{b} \oplus \bigoplus_{i \in Q} r'_i . \end{aligned}$$

Since the mapping from $I, r_1, \ldots, r_n$ to $I', r'_1, \ldots, r'_n$ is invertible, and in both cases the adversary sees the same shares and public information (i.e., $I, (r_i)_{i \in F} = I', (r'_i)_{i \in F}$), the perfect security follows. □

We next prove in Lemma 9.7 that if for some access structure there is a minimal qualified set Q and a maximal forbidden set F such that $|Q \cap F|$ is even, then there is no secret-sharing scheme with public information and 1-bit shares that realizes the access structure. As a warm-up, we prove

[8] Since we obtain a linear secret-sharing scheme with one bit shares, the resulting scheme is necessarily efficient (i.e., the sharing and reconstruction can be computed in polynomial time using the monotone span program of the scheme).

in Lemma 9.5, that under the assumptions of Lemma 9.7, there is no perfect secret-sharing scheme with 1-bit shares that realizes the access structure. To prove Lemma 9.7 (and Lemma 9.5), we need the following lemma. The proof of this lemma is quite involved and is deferred to Section 9.1.

Lemma 9.4. *If there exists $Q \in \min(\mathcal{Q})$ and $F \in \max\left(\overline{\mathcal{Q}}\right)$, such that $|Q \setminus F|$ is even, then there exists $Q' \in \min(\mathcal{Q})$ and $F' \in \max\left(\overline{\mathcal{Q}}\right)$, such that $|Q' \setminus F'| = 2$.*

Lemma 9.5. *Suppose $\mathcal{Q}$ is an n-party access structure such that $|Q \setminus F|$ is even for some minimal qualified set $Q \in \min(\mathcal{Q})$ and maximal forbidden set $F \in \max\left(\overline{\mathcal{Q}}\right)$. Then, there is no perfect secret-sharing scheme with 1-bit shares realizing $\mathcal{Q}$.*

Proof. By Lemma 9.4, we can assume that $Q \setminus F = \{i, j\}$ for some parties $i \neq j$. Suppose that there is a perfect secret-sharing scheme realizing $\mathcal{Q}$. For a secret $b \in \{0,1\}$, let $(I, s_1, \ldots, s_n) \overset{\$}{\leftarrow} \mathsf{Share}(b)$ be possible public information and shares. Since F is a maximal forbidden set, $F \cup \{i\}, F \cup \{j\} \in \mathcal{Q}$. If $\mathsf{Recover}\left(F \cup \{i\}, I, (s_i = 0, (s_k)_{k \in F})\right) = \mathsf{Recover}\left(F \cup \{i\}, I, (s_i = 1, (s_k)_{k \in F})\right)$, then an adversary controlling F and knowing $(s_k)_{k \in F}$ can learn that the secret is b. Thus, by possibly permuting the secrets,

$$\mathsf{Recover}\left(F \cup \{i\}, I, (s_i = 0, (s_k)_{k \in F})\right) = 0 \text{ and } \mathsf{Recover}\left(F \cup \{i\}, I, (s_i = 1, (s_k)_{k \in F})\right) = 1. \quad (4)$$

Similarly, by possibly permuting the share of party j,

$$\mathsf{Recover}\left(F \cup \{j\}, I, (s_j = 0, (s_k)_{k \in F})\right) = 0 \text{ and } \mathsf{Recover}\left(F \cup \{j\}, I, (s_j = 1, (s_k)_{k \in F})\right) = 1. \quad (5)$$

Thus, by the perfect correctness, given the shares $(s_k)_{k \in F \setminus \{i,j\}}$, the shares of i, j and the secret are, respectively, either $0, 0$, and 0, or $1, 1$, and 1. This, along with the perfect correctness and the fact that $Q \subseteq F \cup \{i, j\}$ imply that

$$\mathsf{Recover}\left(Q, I, (s_i = 0, s_j = 0, (s_k)_{k \in Q \setminus \{i,j\}})\right) = 0$$

and

$$\mathsf{Recover}\left(Q, I, (s_i = 1, s_j = 1, (s_k)_{k \in Q \setminus \{i,j\}})\right) = 1.$$

There are two cases for $\mathsf{Recover}\left(Q, I, (s_i = 0, s_j = 1, (s_k)_{k \in Q \setminus \{i,j\}})\right)$.

- If $\mathsf{Recover}\left(Q, I, (s_i = 0, s_j = 1, (s_k)_{k \in Q \setminus \{i,j\}})\right) = 0$, then,

  $$\mathsf{Recover}\left(Q, I, (s_i = 0, s_j = 0, (s_k)_{k \in Q \setminus \{i,j\}})\right) = \mathsf{Recover}\left(Q, I, (s_i = 0, s_j = 1, (s_k)_{k \in Q \setminus \{i,j\}})\right) = 0,$$

  that is, the parties in the forbidden set $Q \setminus \{j\}$ when holding $(s_i = 0, (s_k)_{k \in Q \setminus \{i,j\}})$ can infer that the secret is 0, contradicting the perfect security of the scheme.

- If $\mathsf{Recover}\left(Q, I, (s_i = 0, s_j = 1, (s_k)_{k \in Q \setminus \{i,j\}})\right) = 1$, then,

  $$\mathsf{Recover}\left(Q, I, (s_i = 0, s_j = 1, (s_k)_{k \in Q \setminus \{i,j\}})\right) = \mathsf{Recover}\left(Q, I, (s_i = 1, s_j = 1, (s_k)_{k \in Q \setminus \{i,j\}})\right) = 1,$$

  that is, the parties in the forbidden set $Q \setminus \{i\}$ when holding $(s_j = 1, (s_k)_{k \in Q \setminus \{i,j\}})$ can infer that the secret is 1, contradicting the perfect security of the scheme.

In both cases we reach a contradiction, thus, there is no perfect secret-sharing scheme realizing $\mathcal{Q}$ with one bit shares. □

We next generalize Lemma 9.5 to computational secret-sharing schemes with public information. The first problem is that there might be an error in the reconstruction. This will be dealt by conditioning the sharing algorithm to output vectors of shares without reconstruction errors (since the proof Lemma 9.5 only considers the correctness for a small number of sets, by a union bound, this conditioning will not change the distribution of the shares drastically). The second problem is that we do not have perfect security and the security only holds against polynomial time adversaries. In the next lemma, we show how to construct an appropriate efficient adversary.

Lemma 9.6. *Let A be set of parties and $i \notin A$ be such that $A \notin \mathcal{Q}$ and $A \cup \{i\} \in \mathcal{Q}$. If in a perfectly correct secret-sharing scheme* $(\mathsf{Share}, \mathsf{Recover})$

$$\Pr\left[\begin{array}{l}\mathsf{Recover}\left(A \cup \{i\}, I, (s_i = 0, (s_k)_{k \in A})\right) \\ = \mathsf{Recover}\left(A \cup \{i\}, I, (s_i = 1, (s_k)_{k \in A})\right)\end{array}\middle| b \xleftarrow{\$} \{0,1\}, (I, s_1, \dots, s_n) \xleftarrow{\$} \mathsf{Share}(1^n, b)\right] \geq 0.06,$$

then there is a non-uniform polynomial time adversary that holds $I, (s_k)_{k \in A}$ and guesses the secret with probability at least 0.53.

Proof. For $a \in \{0,1\}$ we define an adversary $\mathcal{A}_a$ as follows: Given $I, (s_k)_{k \in A}$, if

$$\mathsf{Recover}\left(A \cup \{i\}, I, (s_i = 0, (s_k)_{k \in A})\right) = \mathsf{Recover}\left(A \cup \{i\}, I, (s_i = 1, (s_k)_{k \in A})\right) \tag{6}$$

output $\mathsf{Recover}\left(A \cup \{i\}, I, (s_i = 0, (s_k)_{k \in A})\right)$, else output a.

By the perfect correctness of the scheme, when (6) holds, the adversary always outputs the correct value of the secret. Thus, for at least one value of a the adversary $\mathcal{A}_a$ guesses the secret correctly with probability at least $0.06 + 0.94 \cdot 0.5 = 0.53$. □

Lemma 9.7. *Suppose $\mathcal{Q}$ is an n-party access structure such that $|Q \setminus F|$ is even for some minimal qualified set $Q \in \min(\mathcal{Q})$ and maximal forbidden set $F \in \max\left(\overline{\mathcal{Q}}\right)$. Then, for any secret-sharing scheme with public information, 1-bit shares, and reconstruction error of at most 0.01, there exists a non-uniform polynomial-time adversary that holds the public information and shares of a forbidden set and guesses the secret correctly with probability at least 0.51.*

Proof. First assume that the secret-sharing scheme is perfectly correct, we will get rid of this assumption in the end of the proof. By Lemma 9.4, we can assume that $Q \setminus F = \{i, j\}$ for some parties $i \neq j$. For a uniformly chosen secret $b \in \{0,1\}$, let $(I, s_1, \dots, s_n) \xleftarrow{\$} \mathsf{Share}(b)$. Since F is a maximal forbidden set, $F \cup \{i\}, F \cup \{j\} \in \mathcal{Q}$. If

$$\Pr[\mathsf{Recover}\left(F \cup \{i\}, I, (s_i = 0, (s_k)_{k \in F})\right) = \mathsf{Recover}\left(F \cup \{i\}, I, (s_i = 1, (s_k)_{k \in F})\right)] \geq 0.1 \tag{7}$$

(where the probability is the choice of $I, (s_k)_{k \in F}$ generated by $\mathsf{Share}(1^n, b)$ for a random b), then, by Lemma 9.6, there is an adversary that guesses the secret with probability at least 0.53. Thus, we can assume that

$$\Pr[\,\mathsf{Recover}\left(F \cup \{i\}, I, (s_i = 0, (s_k)_{k \in F})\right) \neq \mathsf{Recover}\left(F \cup \{i\}, I, (s_i = 1, (s_k)_{k \in F})\right)] > 0.9. \tag{8}$$

Similarly, we can assume that

$$\Pr[\,\mathsf{Recover}\left(F \cup \{j\}, I, (s_j = 0, (s_k)_{k \in F})\right) \neq \mathsf{Recover}\left(F \cup \{j\}, I, (s_j = 1, (s_k)_{k \in F})\right)] > 0.9. \tag{9}$$

By permuting the values of the share of party j we can assume that

$$\Pr[\,\mathsf{Recover}\,(F \cup \{i\}, I, (s_i = 0, (s_k)_{k \in F})) = \mathsf{Recover}\,(F \cup \{j\}, I, (s_j = 0, (s_k)_{k \in F}))\,] \geq 0.5. \tag{10}$$

We conclude with the following inequality.

$$\begin{aligned}
&\Pr\left[\begin{array}{l}\mathsf{Recover}\,(F \cup \{i\}, I, (s_i = 0, (s_k)_{k \in F})) = \mathsf{Recover}\,(F \cup \{j\}, I, (s_j = 0, (s_k)_{k \in F})) \\ \quad \neq \mathsf{Recover}\,(F \cup \{i\}, I, (s_i = 1, (s_k)_{k \in F})) = \mathsf{Recover}\,(F \cup \{j\}, I, (s_j = 1, (s_k)_{k \in F}))\end{array}\right] \\
&\quad \geq 1 - \Pr\,[\,\mathsf{Recover}\,(F \cup \{i\}, I, (s_i = 0, (s_k)_{k \in F})) \neq \mathsf{Recover}\,(F \cup \{j\}, I, (s_j = 0, (s_k)_{k \in F}))\,] \\
&\qquad - \Pr\,[\,\mathsf{Recover}\,(F \cup \{i\}, I, (s_i = 0, (s_k)_{k \in F})) = \mathsf{Recover}\,(F \cup \{i\}, I, (s_i = 1, (s_k)_{k \in F}))\,)] \\
&\qquad - \Pr\,[\,\mathsf{Recover}\,(F \cup \{j\}, I, (s_j = 0, (s_k)_{k \in F})) = \mathsf{Recover}\,(F \cup \{j\}, I, (s_j = 1, (s_k)_{k \in F}))\,)] \\
&\quad \geq 0.3.
\end{aligned} \tag{11}$$

Furthermore, if there is a value $\sigma \in \{0,1\}$ such that

$$\Pr\left[s_i = \sigma \middle| b \xleftarrow{\$} \{0,1\}, (I, s_1, \ldots, s_n) \xleftarrow{\$} \mathsf{Share}(\mathbb{1}^n, b)\right] \geq 0.55,$$

then an adversary that controls F and holds shares $(s_k)_{k \in F}$ can output

$$\mathsf{Recover}\,(F \cup \{i\}, I, (s_i = \sigma, (s_k)_{k \in F}))$$

and succeed with probability at least 0.55. Thus, we assume that for every $\sigma \in \{0,1\}$

$$\begin{aligned}
\Pr\left[s_i = \sigma \middle| b \xleftarrow{\$} \{0,1\}, (I, s_1, \ldots, s_n) \xleftarrow{\$} \mathsf{Share}(\mathbb{1}^n, b)\right] \geq 0.45, \\
\Pr\left[s_j = \sigma \middle| b \xleftarrow{\$} \{0,1\}, (I, s_1, \ldots, s_n) \xleftarrow{\$} \mathsf{Share}(\mathbb{1}^n, b)\right] \geq 0.45.
\end{aligned} \tag{12}$$

Let $b_0, (s_k)_{k \in F \cup \{i,j\}} \in \{0,1\}$ be a secret and shares. By the perfect correctness and the fact that $Q \subseteq F \cup \{i,j\}$, if

$$\mathsf{Recover}\,(F \cup \{i\}, I, (s_i, (s_k)_{k \in F})) = b_0 \text{ and } \mathsf{Recover}\,(F \cup \{j\}, I, (s_j, (s_k)_{k \in F})) = b_0,$$

then $\mathsf{Recover}\,\big(Q, I, (s_i, s_j, (s_k)_{k \in Q \setminus \{i,j\}})\big) = b_0$.[9] Thus, by (11),

$$\Pr\left[\begin{array}{l}\mathsf{Recover}\,\big(Q, I, (s_i = 0, s_j = 0, (s_k)_{k \in Q \setminus \{i,j\}})\big) \\ \quad \neq \mathsf{Recover}\,\big(Q, I, (s_i = 1, s_j = 1, (s_k)_{k \in Q \setminus \{i,j\}})\big)\end{array}\right] \geq 0.3. \tag{13}$$

There are two cases for $\mathsf{Recover}\,\big(Q, I, (s_i = 0, s_j = 1, (s_k)_{k \in Q \setminus \{i,j\}})\big)$.

- The first option is that

$$\Pr\left[\begin{array}{l}\mathsf{Recover}\,\big(Q, I, (s_i = 0, s_j = 0, (s_k)_{k \in Q \setminus \{i,j\}})\big) \\ \quad = \mathsf{Recover}\,\big(Q, I, (s_i = 0, s_j = 1, (s_k)_{k \in Q \setminus \{i,j\}})\big)\end{array}\right] \geq 0.15, \tag{14}$$

[9] If the probability of the vector $I, ((s_k)_{k \in F \cup \{i,j\}})$ is zero, then this fact does not follow from the correctness; however, in this case we can define $\mathsf{Recover}\,(Q, I, (s_k)_{k \in Q})$ as we wish.

where the probability is over the choice of $I, (s_k)_{k\in Q\setminus\{i,j\}}$ generated by $\mathsf{Share}(\mathbb{1}^n, b)$ for a random b. In this case, by (12),

$$\Pr\left[\begin{array}{l}\mathsf{Recover}\left(Q, I, (s_i, s_j = 0, (s_k)_{k\in Q\setminus\{i,j\}})\right)\\ \quad = \mathsf{Recover}(Q, I, (s_i, s_j = 1, (s_k)_{k\in Q\setminus\{i,j\}}))\end{array}\right] \geq 0.15 \cdot 0.45 > 0.06,$$

where the probability is over the choice of $I, (s_k)_{k\in Q\setminus\{j\}}$ generated by $\mathsf{Share}(\mathbb{1}^n, b)$ for a random b (i.e., also over the choice of s_i). By Lemma 9.6 applied to the forbidden set $Q \setminus \{j\}$, there is an adversary that guesses the secret with probability at least 0.53.

- Otherwise, by (13) and (14)

$$\Pr\left[\begin{array}{l}\mathsf{Recover}\left(Q, I, (s_i = 0, s_j = 1, (s_k)_{k\in Q\setminus\{i,j\}})\right)\\ \quad = \mathsf{Recover}\left(Q, I, (s_i = 1, s_j = 1, (s_k)_{k\in Q\setminus\{i,j\}})\right)\end{array}\right] > 0.15.$$

  Similar to the previous case, by Lemma 9.6 applied to the forbidden set $Q \setminus \{i\}$, there is an adversary that guesses the secret with probability at least 0.53.

In all cases we construct a non-uniform adversary that guesses the secret with probability at least 0.53 (assuming that the scheme is perfectly correct).

Finally, we deal with the fact that the secret-sharing scheme might have (small) error in the reconstruction, that is, the correctness for every qualified set holds with probability at least 0.99. In the proof of the lemma we only consider the qualified sets $F \cup \{i\}, F \cup \{j\},$. By the union bound, with probability at least 0.97 these three sets correctly reconstruct the secret simultaneously. We apply the above argument to the modification of the scheme in which the sharing algorithm only deals vectors of shares in which these 3 sets do not err. In this modified secret-sharing scheme, there is a non-uniform adversary controlling a forbidden set that guesses the secret with probability at least 0.53. We apply the same adversary to the original secret-sharing scheme; the adversary guesses the secret with probability at least $0.97 \cdot 0.53 > 0.51$. □

9.1 Proof of Lemma 9.4

We next prove Lemma 9.4, that is, we prove that if there are a minimal qualified set Q and a maximal forbidden set F such that $|Q \setminus F|$ is even, then there are a minimal qualified set Q and a maximal forbidden set F such that $|Q \setminus F| = 2$.

Proof of Lemma 9.4. Choose $Q^* \in \min(\mathcal{Q})$ and $F^* \in \max\left(\overline{\mathcal{Q}}\right)$ such that $|Q^* \setminus F^*|$ is minimal amongst all pairs such that $|Q \setminus F|$ is even, that is, for any $Q \in \min(\mathcal{Q})$ and $F \in \max\left(\overline{\mathcal{Q}}\right)$ such that $|Q \setminus F|$ is even, $|Q \setminus F| \geq |Q^* \setminus F^*|$. Since $\mathcal{Q}$ is a monotone access structure, $F^* \not\subseteq Q^*$ and $|Q^* \setminus F^*| \geq 2$. If $Q^* \setminus F^*$ is of size 2, we are done; we will prove a contradiction assuming this size is greater than 2.

We will consider the restriction of $\mathcal{Q}$ to the parties in $Q^* \cup F^*$, that is, the access structure $\mathcal{Q}' = \{Q \in \mathcal{Q} : Q \subseteq Q^* \cup F^*)\}$. If there exists $Q \in \min(\mathcal{Q}')$ and $F \in \max\left(\overline{\mathcal{Q}'}\right)$ such that $|Q \setminus F|$ is even and smaller than $|Q^* \setminus F^*|$, then $Q \in \min(\mathcal{Q})$ and F can be extended to $F' \in \max\left(\overline{\mathcal{Q}}\right)$ such that $|Q \setminus F| = |Q \setminus F'|$, a contradiction.

Observe that, for any $Q \in \min(\mathcal{Q}')$, either $Q \setminus F^* = Q^* \setminus F^*$ or $|Q \setminus F^*|$ is odd. The following definitions are central to the remainder of the proof.

Definition 9.8 (A core)**.** *Let $\mathcal{B} = \min(\mathcal{Q}') \setminus \{Q^*\}$. We say that a set $C \subseteq F^* \setminus Q^*$ is a core if $C \cap Q \neq \emptyset$ for all $Q \in \mathcal{B}$, and, for any $i \in C$, there exists $Q \in \mathcal{B}$ such that $Q \cap C = \{i\}$.*

Fix a core C; for $\ell \in Q^* \setminus F^*$, we say that a set C_ℓ is an ℓ-core if C_ℓ is a maximal subset of C such that for every $Q \in \mathcal{B}$,

$$\textit{if } Q \cap C \subseteq C_\ell, \textit{ then } \ell \in Q. \tag{15}$$

Finally, fix a core C and ℓ-cores $(C_\ell)_{\ell \in Q^* \setminus F^*}$; for every $i \in C$ define

$$E_i = \{\ell \in Q^* \setminus F^* \textit{ s.t. } i \in C_\ell\}.$$

We start with a simple claim that will be used a few times in the proof; this claim provides the motivation to the definition of an ℓ-core.

Claim 9.9. *Let $\ell \in Q^* \setminus F^*$. Then there exists a maximal forbidden set $F \in \max\left(\overline{\mathcal{Q}'}\right)$ such that $(Q^* \cup F^*) \setminus ((C \setminus C_\ell) \cup \{\ell\}) \subseteq F$ and $\ell \notin F$.*

Proof. Consider the set $A = (Q^* \cup F^*) \setminus ((C \setminus C_\ell) \cup \{\ell\})$. We claim that A does not contain any minimal qualified set $Q \in \min(\mathcal{Q}') = \mathcal{B} \cup \{Q^*\}$:

- $Q^* \nsubseteq A$, since $\ell \in Q^*$ and $\ell \notin A$.
- If $Q \in \mathcal{B}$ and $Q \cap (C \setminus C_\ell) \neq \emptyset$, then $Q \nsubseteq A$.
- If $Q \in \mathcal{B}$ and $Q \cap (C \setminus C_\ell) = \emptyset$, then $Q \cap C \subseteq C_\ell$ and, by the definition of an ℓ-core, $\ell \in Q$, hence $Q \nsubseteq A$.

Therefore, there exists $F \in \max\left(\overline{\mathcal{Q}'}\right)$ such that $A \subseteq F$. If $\ell \in F$, then $Q^* \subseteq F$ (since $C \subseteq F^* \setminus Q^*$), contradicting the fact that $F \in \max\left(\overline{\mathcal{Q}'}\right)$ and $Q^* \in \min(\mathcal{Q}')$. (of Claim 9.9) □

The structure of the rest of the proof is as follows. We first prove in Claim 9.10 that a core exists. Then we prove in Claim 9.11 that the assumption that $|Q^* \setminus F^*| > 2$ implies that $E_i = Q \setminus F^*$ for every $Q \in \mathcal{B}$ such that $Q \cap C = \{i\}$ (by the definition of a core, such a set exists). Thereafter, we prove in Claim 9.13 that the sets $(E_i)_{i \in C}$ are ordered by inclusion, in particular, there exists an ℓ such that $\ell \in E_i$ for all $i \in C$. We complete to proof by showing that this is not possible and the assumption that $|Q^* \setminus F^*| > 2$ is false.

Claim 9.10. *There exists a core C.*

Proof. For all $Q \in \mathcal{B}$, $Q \nsubseteq Q^*$ and $Q \subseteq Q^* \cup F^*$. Consequently, for all $Q \in \mathcal{B}$,

$$Q \cap (F^* \setminus Q^*) \neq \emptyset.$$

We construct a core $C \subseteq F^* \setminus Q^*$ in steps, maintaining the property that $Q \cap C \neq \emptyset$ for every $Q \in \mathcal{B}$. We start with $C = F^* \setminus Q^*$. If for every $i \in C$ there is some $Q \in \mathcal{B}$ such that $Q \cap C = \{i\}$, then C is a core. Otherwise, there exists $i \in C$ such that $Q \cap C \neq \{i\}$ for all $Q \in \mathcal{B}$, that is, $Q \cap (C \setminus \{i\}) \neq \emptyset$ for every $Q \in \mathcal{B}$ and we set $C = C \setminus \{i\}$. We repeat this step until the set C is a core. (of Claim 9.10) □

Claim 9.11. *For every $i \in C$ and every $Q \in \mathcal{B}$ such that $Q \cap C = \{i\}$,*

$$E_i = Q \setminus F^*.$$

Proof. We first prove that $E_i \subseteq Q \setminus F^*$. Let $\ell \in E_i$, i.e., $i \in C_\ell$ and $Q \cap C = \{i\} \subseteq C_\ell$. This implies that $\ell \in Q$ (since C_ℓ is an ℓ-core). We have shown that $E_i \subseteq Q$. By definition, $E_i \subseteq Q^* \setminus F^*$, thus, $E_i \subseteq Q \setminus F^*$.

Next we prove the "harder direction", namely, $Q \setminus F^* \subseteq E_i$. Let $\ell \in Q \setminus F^*$. Suppose towards a contradiction that $\ell \notin E_i$, i.e., $i \notin C_\ell$. Since C_ℓ is a maximal set satisfying (15), there exists $Q' \in \mathcal{B}$ such that

$$i \in Q', \ell \notin Q', \text{ and } Q' \cap C \subseteq C_\ell \cup \{i\}.$$

By Claim 9.9, there exists $F \in \max\left(\overline{\mathcal{Q}'}\right)$ such that $(Q^* \cup F^*) \setminus ((C \setminus C_\ell) \cup \{\ell\}) \subseteq F$ and $\ell \notin F$. If $i \in F$, then $Q' \subseteq F$, since $\ell \notin Q'$ and $Q' \cap C \subseteq C_\ell \cup \{i\}$; this is not possible as $Q' \in \min(\mathcal{Q}')$ and $F \in \max\left(\overline{\mathcal{Q}'}\right)$. We conclude that $i, \ell \notin F$. On the other hand, $Q \cap C = \{i\}$ and $\ell \in Q \setminus F^*$, thus, $\{i, \ell\} \subseteq Q$ and $\{i, \ell\} \subseteq Q \setminus F$. Furthermore, as $Q \cap C = \{i\}$,

$$Q \setminus F \subseteq Q \setminus ((Q^* \cup F^*) \setminus ((C \setminus C_\ell) \cup \{\ell\})) \subseteq Q \cap ((C \setminus C_\ell) \cup \{\ell\}) = \{i, \ell\}.$$

This contradicts our assumption that $Q^* \setminus F^*$ is the smallest set with even size, concluding the proof that $Q \setminus F^* \subseteq E_i$, implying that $E_i = Q \setminus F^*$. (of Claim 9.11) □

We will hereafter denote $Q^* \setminus F^*$ by E.

Claim 9.12. *For any $i_1, i_2 \in C$, if $E_{i_1} \setminus E_{i_2}$ and $E_{i_2} \setminus E_{i_1}$ are non-empty, then there exists a $Q_0 \in \mathcal{B}$ such that*

$$Q_0 \cap E = E \setminus (E_{i_1} \triangle E_{i_2}). \tag{16}$$

Proof. As a first step, we show that there exists $Q_0 \in \mathcal{B}$ such that

$$Q_0 \cap C = \{i_1, i_2\}. \tag{17}$$

Suppose that this is not true. Let $\ell_1 \in E_{i_1} \setminus E_{i_2}$ and $\ell_2 \in E_{i_2} \setminus E_{i_1}$. Take $Q_1, Q_2 \in \mathcal{B}$ such that $C \cap Q_1 = \{i_1\}$ and $C \cap Q_2 = \{i_2\}$ (by definition of the core C, such Q_1, Q_2 exist); by Claim 9.11, $E_{i_1} = Q_1 \setminus F^*$ and $E_{i_2} = Q_2 \setminus F^*$. Since $\ell_1 \in E_{i_1}$ and $\ell_2 \notin E_{i_1}$, we have $i_1 \in C_{\ell_1}$ and $i_1 \notin C_{\ell_2}$. As $\ell_1 \in E_{i_1} = Q_1 \setminus F^*$, it must be that $\ell_1 \in Q_1$. As $\ell_2 \notin E_{i_1} = Q_1 \setminus F^* \subseteq Q^* \setminus F^*$ and $\ell_2 \in Q^* \setminus F^*$, it must be that $\ell_2 \notin Q_1$. We claim that

$$A = (Q^* \cup F^*) \setminus ((C \setminus \{i_1, i_2\}) \cup \{\ell_1, \ell_2\})$$

is a forbidden set in $\mathcal{Q}'$, i.e., it does not contain any minimal qualified sets in $Q \in \min(\mathcal{Q}') = \mathcal{B} \cup \{Q^*\}$:

- $\ell_1, \ell_2 \in Q^*$ and $\ell_1, \ell_2 \notin A$, thus $Q^* \nsubseteq A$.
- If $Q \in \mathcal{B}$ and $Q \cap C \nsubseteq \{i_1, i_2\}$, then $Q \nsubseteq A$.
- By our assumption, there is no $Q \in \mathcal{B}$ such that $Q \cap C = \{i_1, i_2\}$.
- If $Q \in \mathcal{B}$ and $Q \cap C = \{i_1\} \subseteq C_{\ell_1}$, then, by the definition of an ℓ_1-core, $\ell_1 \in Q$ and $Q \nsubseteq A$.
- If $Q \in \mathcal{B}$ and $Q \cap C = \{i_2\} \subseteq C_{\ell_2}$, then, by the definition of an ℓ_2-core, $\ell_2 \in Q$ and $Q \nsubseteq A$.

Let $F \in \max\left(\overline{\mathcal{Q}'}\right)$ be a maximal forbidden set containing A. If $\ell_1 \in F$, then $Q_1 \subseteq F$, a contradiction to the definition of sets in $\min(\mathcal{Q}')$, $\max\left(\overline{\mathcal{Q}'}\right)$. Similarly (by considering Q_2), $\ell_2 \notin F$ and $\{\ell_1, \ell_2\} \subseteq Q^* \setminus F$. Furthermore, $Q^* \setminus F \subseteq Q^* \setminus A \subseteq \{\ell_1, \ell_2\}$ (since $Q^* \cap C = \emptyset$). Thus, $Q^* \setminus F = \{\ell_1, \ell_2\}$, contradicting the choice of Q^*, F^*.

Let $Q_0 \in \mathcal{B}$ be a set satisfying (17), i.e., $Q_0 \cap C = \{i_1, i_2\}$; we argue that $(E_{i_1} \triangle E_{i_2}) \cap Q_0 = \emptyset$. Otherwise, there exists ℓ_1 such that

$$\ell_1 \in (E_{i_1} \triangle E_{i_2}) \cap Q_0. \tag{18}$$

Without loss of generality, let $\ell_1 \in E_{i_1} \setminus E_{i_2}$. Then, $i_1 \in C_{\ell_1}$ and $i_2 \notin C_{\ell_1}$. By Claim 9.9, there is a maximal forbidden set $F \in \max\left(\overline{\mathcal{Q}}\right)$ such that $(Q^* \cup F^*) \setminus ((C \setminus C_{\ell_1}) \cup \{\ell_1\}) \subseteq F$ and $\ell_1 \notin F$. Let $Q_2 \in \mathcal{B}$ such that $C \cap Q_2 = \{i_2\}$ (Q_2 exists by the definition of the core C); by Claim 9.11, $E_{i_2} = Q_2 \setminus F^*$. Since $\ell_1 \notin E_{i_2} = Q_2 \setminus F^* \subseteq Q^* \setminus F^*$ and $E_{i_2} \subseteq Q^* \setminus F^*$, we deduce that $\ell_1 \notin Q_2$. This implies that $i_2 \notin F$ (otherwise $Q_2 \subseteq F$). On the other hand, by (17) and (18), $\{i_2, \ell_1\} \subseteq Q_0$, i.e., $\{i_2, \ell_1\} \subseteq Q_0 \setminus F$. Furthermore,

$$Q_0 \setminus F \subseteq Q_0 \setminus ((Q^* \cup F^*) \setminus ((C \setminus C_{\ell_1}) \cup \{\ell_1\})) = (Q_0 \cap (C \setminus C_{\ell_1})) \cup \{\ell_1\} = \{i_2, \ell_1\}.$$

This implies that $Q_0 \setminus F = \{i_2, \ell_1\}$, a contradiction.

This proves that there exists $Q_0 \in \mathcal{B}$ such that $Q_0 \cap C = \{i_1, i_2\}$ and $(E_{i_1} \triangle E_{i_2}) \cap Q_0 = \emptyset$. We will show that Q_0 satisfies the requirements of the claim, namely, $Q_0 \cap E = E \setminus (E_{i_1} \triangle E_{i_2})$. We prove the equality by double inclusion. The inclusion $Q_0 \cap E \subseteq E \setminus (E_{i_1} \triangle E_{i_2})$ is implied by the facts that $Q \subseteq E$ and $(E_{i_1} \triangle E_{i_2}) \cap Q_0 = \emptyset$.

We next prove that $E \setminus (E_{i_1} \triangle E_{i_2}) \subseteq Q_0 \cap E$. Note that $E \setminus (E_{i_1} \triangle E_{i_2}) = (E_1 \cap E_2) \cup (E \setminus (E_1 \cup E_2))$. If $\ell \in E_{i_1} \cap E_{i_2}$, then $i_1, i_2 \in C_\ell$. As $Q_0 \cap C = \{i_1, i_2\} \subseteq C_\ell$, the definition of an ℓ-core implies that $\ell \in Q_0$. Hence, it suffices to show that if $\ell \in E \setminus (E_{i_1} \cup E_{i_2})$, then $\ell \in Q_0$. Suppose that this is not true, i.e., there is an $\ell \in E \setminus (E_{i_1} \cup E_{i_2})$ such that $\ell \notin Q_0$. By Claim 9.9, there is a maximal forbidden set $F \in \max\left(\overline{\mathcal{Q}'}\right)$ such that $(Q^* \cup F^*) \setminus ((C \setminus C_\ell) \cup \{\ell\}) \subseteq F$. Let $Q_{i_1}, Q_{i_2} \in \mathcal{B}$ such that $Q_{i_1} \cap C = \{i_1\}$ and $Q_{i_2} \cap C = \{i_2\}$. As argued above, since $\ell \notin E_{i_1} = Q_0 \setminus F^*$, it must be that $\ell \notin Q_0$ and, by similar arguments, $\ell \notin Q_2$. If $i_1 \in F$ (resp., $i_2 \in F$), then $Q_0 \subseteq F$ (resp., $Q_2 \subseteq F$), a contradiction. We proved that $\{i_1, i_2\} \subseteq Q_0 \setminus F$. On the other hand,

$$Q_0 \setminus F \subseteq Q_0 \setminus ((Q^* \cup F^*) \setminus ((C \setminus C_\ell) \cup \{\ell\})) \subseteq Q_0 \cap C = \{i_1, i_2\}.$$

But then, $Q_0 \setminus F = \{i_1, i_2\}$; a contradiction. (of Claim 9.12) □

Claim 9.13. *For any $i_1, i_2 \in C$, either $E_{i_1} \subseteq E_{i_2}$ or $E_{i_2} \subseteq E_{i_1}$.*

Proof. Towards a contradiction, suppose $E_{i_1} \setminus E_{i_2}$ and $E_{i_2} \setminus E_{i_1}$ are non-empty. Recall that the size of the set $E = Q^* \setminus F^*$ is even. Let Q_0 be a minimal authorized set in $\mathcal{B}$ guaranteed by Claim 9.12; observe that

$$|Q_0 \cap E| = |E \setminus (E_{i_1} \triangle E_{i_2})| = |E| - |E_{i_1} \triangle E_{i_2}| = |E| - |E_{i_1}| - |E_{i_2}| + 2(|E_{i_1} \cap E_{i_2}|). \tag{19}$$

Since $E_{i_1} \setminus E_{i_2}$ and $E_{i_2} \setminus E_{i_1}$ are non-empty, $E_{i_1} \neq E \neq E_{i_2}$. By Claim 9.11, $E_{i_1} = Q \setminus F^*$ for some $Q \in \mathcal{B}$. Since $E_{i_1} \nsubseteq E = Q^* \setminus F^*$, the size of E_{i_1} is an odd positive number; the same holds for $|E_{i_2}|$. But then, by (19) and the fact that $Q_0 \cap E \neq E$ (since $|E_{i_1} \triangle E_{i_2}| > 0$), the size of $Q_0 \cap E$ is an even number less that $|E| = |Q^* \setminus F^*|$. However, $Q_0 \cap E = Q_0 \setminus F^*$, a contradiction to the choice of Q^*, F^*. (of Claim 9.13) □

We complete the proof of lemma using Claim 9.13. As observed above, $E_i = Q \setminus F^*$ for some $Q \in \mathcal{B}$ such that $Q \cap C = \{i\}$. Since $Q \not\subseteq F^*$ and $Q \subseteq (Q^* \cup F^*)$, it must be that $Q \setminus F^* \neq \emptyset$. Hence, $|E_i|$ is a positive number for each $i \in C$. Since $(E_i)_{i \in C}$ is a monotone family of non-empty sets, there exists $\ell \in (Q^* \setminus F^*)$ such that $\ell \in E_i$ for all $i \in C$, , i.e., $C_\ell = C$. We will show that is impossible; i.e., we will show that for each $\ell \in Q^* \setminus F^*$ there exist $i \in C$ such that $i \notin C_\ell$. Choose $\ell' \in Q^* \setminus F^*$ such that $\ell' \neq \ell$; such ℓ' exists because $Q^* \setminus F^*$ is non-singleton. Choose $Q \in \mathcal{B}$ such that $Q \setminus F^* = \{\ell'\}$ (such Q' exists since F^* is a maximal forbidden set, i.e., $F^* \cup \{\ell'\}$ is qualified); then, Q belongs to $\mathcal{B}$, and $\ell \notin Q$. But then, $C_\ell \neq C$; otherwise, $Q \cap C \subseteq C_\ell$ but $\ell \notin Q$, contradicting (15). We conclude that there exists $i \notin C_\ell$. This proves of the lemma. □

Acknowledgements. We thank Uriel Feige, Sam Hopkins, Prasad Raghavendra, Daniel Reichman, and Alon Rosen for helpful discussions and literature pointers. Damiano Abram was supported by a GSNS travel grant from Aarhus University and by the Aarhus University Research Foundation (AUFF). Amos Beimel was supported by ERC Project NTSC (742754) and ISF grant 391/21. Yuval Ishai and Varun Narayanan were supported by ERC Project NTSC (742754), BSF grant 2018393, and ISF grant 2774/20. Work of Varun Narayanan was done while working at Technion, Israel Institute of Technology. Eyal Kushilevitz was supported by BSF grant 2018393 and ISF grant 2774/20.

References

[AAK+07] Noga Alon, Alexandr Andoni, Tali Kaufman, Kevin Matulef, Ronitt Rubinfeld, and Ning Xie. Testing k-wise and almost k-wise independence. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 496–505. ACM Press, June 2007.

[ABdR+18] Albert Atserias, Ilario Bonacina, Susanna F. de Rezende, Massimo Lauria, Jakob Nordström, and Alexander A. Razborov. Clique is hard on average for regular resolution. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 866–877. ACM Press, June 2018.

[ABI+23] Benny Applebaum, Amos Beimel, Yuval Ishai, Eyal Kushilevitz, Tianren Liu, and Vinod Vaikuntanathan. Succinct computational secret sharing. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023*, 2023.

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In Leonard J. Schulman, editor, *42nd ACM STOC*, pages 171–180. ACM Press, June 2010.

[AHMS18] Benny Applebaum, Thomas Holenstein, Manoj Mishra, and Ofer Shayevitz. The communication complexity of private simultaneous messages, revisited. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 261–286. Springer, Heidelberg, April / May 2018.

[AKS98] Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a Large Hidden Clique in a Random Graph. In *Random Structures and Algorithms 13*, 1998.

[ALM^{+}92] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof Verifiaction and Hardness of Approximation Problems. In *Proceedings of the 33rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 1992*, 1992.

[AOS23] Damiano Abram, Maciej Obremski, and Peter Scholl. On the (Im)possibility of Distributed Samplers: Lower Bounds and Party-Dynamic Constructions. Cryptology ePrint Archive, 2023, 2023.

[AS92] Sanjeev Arora and Shmuel Safra. Approximating Clique is NP Complete. In *Proceedings of the 33rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 1992*, 1992.

[ASY22] Damiano Abram, Peter Scholl, and Sophia Yakoubov. Distributed (correlation) samplers: How to remove a trusted dealer in one round. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part I*, volume 13275 of *LNCS*, pages 790–820. Springer, Heidelberg, May / June 2022.

[AV11] Brendan Ames and Stephen Vavasis. Nuclear Norm Minimization for the Planted Clique and Biclique Problems. In *Mathematical Programming*, 2011.

[BB19] Matthew Brennan and Guy Bresler. Optimal Average-Case Reductions to Sparse PCA: From Weak Assumptions to Strong Hardness. In *Proceedings of 32nd Conference on Learning Theory*, 2019.

[BB20] Matthew Brennan and Guy Bresler. Reducibility and Statistical-Computational Gaps from Secret Leakage. In *Proceedings of 33rd Conference on Learning Theory*, 2020.

[BBB19] Enric Boix-Adserà, Matthew Brennan, and Guy Bresler. The average-case complexity of counting cliques in Erdős-Rényi hypergraphs. In David Zuckerman, editor, *60th FOCS*, pages 1256–1280. IEEE Computer Society Press, November 2019.

[BBH18] Matthew Brennan, Guy Bresler, and Wasim Huleihel. Reducibility and Computational Lower Bounds for Problems with Planted Sparse Structure. In *Proceedings of 31st Conference on Learning Theory*, 2018.

[BBH19] Matthew Brennan, Guy Bresler, and Wasim Huleihel. Universality of Computational Lower Bounds for Submatrix Detection. In *Proceedings of 32nd Conference on Learning Theory*, 2019.

[BC93] Amos Beimel and Benny Chor. Universally ideal secret sharing schemes (preliminary version). In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 183–195. Springer, Heidelberg, August 1993.

[BE76] Béla Bollobás and Paul Erdős. Cliques in Random Graph. In *Mathematical Proceedings of the Cambridge Philosophical Society*, 1976.

[BF07] Amos Beimel and Matthew K. Franklin. Weakly-private secret sharing schemes. In Salil P. Vadhan, editor, *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*, pages 253–272. Springer, 2007.

[BGIK22] Elette Boyle, Niv Gilboa, Yuval Ishai, and Victor I. Kolobov. Programmable distributed point functions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part IV*, volume 13510 of *LNCS*, pages 121–151. Springer, Heidelberg, August 2022.

[BGLR93] Mihir Bellare, Shafi Goldwasser, Carsten Lund, and Alexander Russell. Efficient probabilistic Checkable Proofs and Application to Approximation. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing, STOC 1993*, 1993.

[BGS95] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free Bits, PCPs and Non-Approximability: Towards Tight Results. In *Proceedings of the 36th IEEE Annual Symposium on Foundations of Computer Science, FOCS 1995*, 1995.

[BHK$^+$16] Boaz Barak, Samuel B. Hopkins, Jonathan A. Kelner, Pravesh Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. In Irit Dinur, editor, *57th FOCS*, pages 428–437. IEEE Computer Society Press, October 2016.

[BIKK14] Amos Beimel, Yuval Ishai, Ranjit Kumaresan, and Eyal Kushilevitz. On the cryptographic complexity of the worst functions. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 317–342. Springer, Heidelberg, February 2014.

[BKR23] Andrej Bogdanov, Pravesh Kothari, and Alon Rosen. Public-key encryption, local pseudorandom generators, and the low-degree method. *IACR Cryptol. ePrint Arch. 2023/1049*, 2023. To appear in TCC 2023.

[BLVW19] Zvika Brakerski, Vadim Lyubashevsky, Vinod Vaikuntanathan, and Daniel Wichs. Worst-case hardness for LPN and cryptographic hashing via code smoothing. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 619–635. Springer, Heidelberg, May 2019.

[BR13] Quentin Berthet and Philippe Rigollet. Complexity Theoretic Lower Bounds for Sparse Principal Component Detection. In *The 26th Annual Conference on Learning Theory, COLT 2013*, 2013.

[BRSV18] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Proofs of work from worst-case assumptions. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 789–819. Springer, Heidelberg, August 2018.

[BS94] Mihir Bellare and Madhu Sudan. Improved Non-Approximability Results. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing, STOC 1994*, 1994.

[CCX13] Ignacio Cascudo, Ronald Cramer, and Chaoping Xing. Bounds on the Threshold Gap in Secret Sharing and its Applications. In *IEEE Transactions on Information Theory*, 2013.

[Che15] Yudong Chen. Incoherence-Optimal Matrix Completion. In *IEEE Transactions on Information Theory*, 2015.

[CLR17] Tony Cai, Tengyuan Liang, and Alexander Rakhlin. Computational and Statistical Boundaries for Submatrix Localization in a Large Noisy Matrix. In *The Annals of Statistics*, 2017.

[COE15] Amin Coja-Oghlan and Charilaos Efthymiou. On Independent Sets in Random Graphs. In *Random Structures and Algorithms*, 2015.

[CX16] Yudong Chen and Jaiming Xu. Statistical-Computational Tradeoffs in Planted Problems and Submatrix Localization with a Growing Number of Clusters and Submatrices. In *Journal of Machine Learning Research*, 2016.

[DGGP14] Yael Dekel, Ori Gurel-Gurevich, and Yuval Peres. Finding Hidden Cliques in Linear Time with High Probability. In *Combinatorics, Probability and Computing*, 2014.

[DM15a] Yash Deshpande and Andrea Montanari. Finding Hidden Cliques of Size $\sqrt{N/e}$ in Nearly Linear Time. In *Foundations of Computational Mathematics*, 2015.

[DM15b] Yash Deshpande and Andrea Montanari. Improved Sum-of-Squares Lower Bounds for Hidden Clique and Hidden Submatrix Problems. In *Proceedings of 28th Conference on Learning Theory*, 2015.

[ERSY22] Reyad Abed Elrazik, Robert Robere, Assaf Schuster, and Gal Yehuda. Pseudorandom Self-Reductions for NP-Complete Problems. In *ITCS 2022*, 2022.

[FGL$^+$95] Uriel Feige, Shafi Goldwasser, Laszlo Lovasz, Shmuel Safra, and Mario Szegedy. Interactive Proofs and the Hardness of Approximating Cliques. In *Journal of the ACM*, 1995.

[FGN$^+$20] Uriel Feige, David Gamarnik, Joe Neeman, Miklós Rácz, and Prasad Tetali. Finding Cliques using few Probes. In *Random Structures and Algorithms*, 2020.

[FGR$^+$13] Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 655–664. ACM Press, June 2013.

[FK00] Uriel Feige and Robert Krauthgamer. Finding and Certifying a Large Hidden Clique in a Semirandom Graph. In *Random Structures Algorithms*, 2000.

[FK03] Uriel Feige and Robert Krauthgamer. The Probable Value of the Lovász–Schrijver Relaxations for Maximum Independent Set. In *SIAM Journal of Computing*, 2003.

[FKN94] Uriel Feige, Joe Kilian, and Moni Naor. A minimal model for secure computation (extended abstract). In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, STOC 1994*, pages 554–563, 1994.

[FR10] Uriel Feige and Dorit Ron. Finding Hidden Cliques in Linear Time. In *21st International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms*, 2010.

[GKVZ22] Shafi Goldwasser, Michael Kim, Vinod Vaikuntanathan, and Or Zamir. Planting Undetectable Backdoors in Machine Learning Models. In *Proceedings of the 63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022*, 2022.

[GM75] Geoffrey Grimmett and Colin McDiarmid. On Colouring Random Graphs. In *Mathematical Proceedings of the Cambridge Philosophical Society*, 1975.

[GNW11] Oded Goldreich, Noam Nisan, and Avi Wigderson. On yao's xor-lemma. In Oded Goldreich, editor, *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation - In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, volume 6650 of *Lecture Notes in Computer Science*, pages 273–301. Springer, 2011.

[Gol98] Jovan Dj. Golic. On matroid characterization of ideal secret sharing schemes. *J. Cryptol.*, 11(2):75–86, 1998.

[GS14] David Gamarnik and Madhu Sudan. Limits of local algorithms over sparse random graphs. In Moni Naor, editor, *ITCS 2014*, pages 369–376. ACM, January 2014.

[Hås96a] Johan Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *37th FOCS*, pages 627–636. IEEE Computer Society Press, October 1996.

[Hås96b] Johan Håstad. Testing of the long code and hardness for clique. In *28th ACM STOC*, pages 11–19. ACM Press, May 1996.

[HJK+16] Dennis Hofheinz, Tibor Jager, Dakshita Khurana, Amit Sahai, Brent Waters, and Mark Zhandry. How to generate and use universal samplers. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 715–744. Springer, Heidelberg, December 2016.

[HK11] Elad Hazan and Robert Krauthgamer. How Hard is it to Approximate the Best Nash Equilibrium? In *SIAM Journal on Computing*, 2011.

[HKP+18] Samuel Hopkins, Pravesh Kothari, Aaron Potechin, Prasad Raghavendra, and Tselil Schramm. On the Integrality Gap of Degree-4 Sum of Squares for Planted Clique. In *ACM Transactions on Algorithm*, volume 14, Issue 3, Article No.: 28, pages 1—31, 2018.

[Hop18] Samuel Hopkins. *Statistical Inference and the Sum of Squares Method.* Phd thesis, Cornell University, 2018.

[HW21] Justin Holmgren and Alexander S. Wein. Counterexamples to the low-degree conjecture. In Lee [Lee21], pages 75:1–75:9.

[HWX15] Bruce Hajek, Yihong Wu, and Jiaming Xu. Computational Lower Bounds for Community Detection on Random Graphs. In *The 28th Annual Conference on Learning Theory, COLT 2015*, 2015.

[IK97] Yuval Ishai and Eyal Kushilevitz. Private simultaneous messages protocols with applications. In *Fifth Israel Symposium on Theory of Computing and Systems, ISTCS 1997, Ramat-Gan, Israel, June 17-19, 1997, Proceedings*, pages 174–184, 1997.

[IKM^{+}13] Yuval Ishai, Eyal Kushilevitz, Sigurd Meldgaard, Claudio Orlandi, and Anat Paskin-Cherniavsky. On the power of correlated randomness in secure computation. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 600–620. Springer, Heidelberg, March 2013.

[Jer92] Mark Jerrum. Large Cliques Elude the Metropolis Process. In *Random Structures and Algorithms*, 1992.

[JP00] Ari Juels and Marcus Peinado. Hiding Cliques for Cryptographic Security. In *Designs, Codes and Cryptography*, 2000.

[Kar72] Richard Karp. Reducibility among Combinatorial Problems. In *The Complexity of Computer Computations*. Plenum Press, 1972.

[Kar76] Richard Karp. Probabilistic Analysis of some Combinatorial Search Problems. In *Algorithms and Complexity: New Directions and Recent Results*, 1976.

[KN90] Joe Kilian and Noam Nisan. Private communication, 1990.

[Kuč95] Luděk Kučera. Expected Complexity of Graph Partitioning Problems. In *Discrete Applied Mathematics 57*, 1995.

[KZ14] Pascal Koiran and Anastasios Zouzias. Hidden Cliques and the Certification of the Restricted Isometry Property. In *IEEE Transactions on Information Theory*, 2014.

[Lee21] James R. Lee, editor. *ITCS 2021*, volume 185. LIPIcs, January 2021.

[LVW17] Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Conditional disclosure of secrets via non-linear reconstruction. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 758–790. Springer, Heidelberg, August 2017.

[Mat95] Frantisek Matúš. Probabilistic conditional independence structures and matroid theory: Background. *Int. J. of General Systems*, 22:185–196, 1995.

[McD74] Colin McDiarmid. Colouring Random Graphs. In *Annals of Operations Research 1.3*, 1974.

[McS01] Frank McSherry. Spectral partitioning of random graphs. In *42nd FOCS*, pages 529–537. IEEE Computer Society Press, October 2001.

[Mer78] Ralph Merkle. Secure Communications over Insecure Channels. In *Communications of the ACM*, 1978.

[MPW15] Raghu Meka, Aaron Potechin, and Avi Wigderson. Sum-of-squares lower bounds for planted clique. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th ACM STOC*, pages 87–96. ACM Press, June 2015.

[MRS21] Pasin Manurangsi, Aviad Rubinstein, and Tselil Schramm. The strongish planted clique hypothesis and its consequences. In Lee [Lee21], pages 10:1–10:21.

[MW15] Zongming Ma and Yihong Wu. Computational Barriers in Minimax Submatrix Detection. In *The Annals of Statistics*, 2015.

[Pit82] B Pittel. On the Probable Behaviour of some Algorithms for Finding the Stability Number of a Graph. In *Mathematical Proceedings of the Cambridge Philosophical Society*, 1982.

[Ros08] Benjamin Rossman. On the constant-depth complexity of k-clique. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 721–730. ACM Press, May 2008.

[Ros10] Benjamin Rossman. The monotone complexity of k-clique on random graphs. In *51st FOCS*, pages 193–201. IEEE Computer Society Press, October 2010.

[RV17] Mustazee Rahman and Balint Virag. Local Algorithms for Independent Sets are Half-Optimal. In *The Annals of Probability*, 2017.

[SBW19] Nihar Shah, Sivaraman Balakrishnan, and Martin Wainwright. Feeling the bern: Adaptive Estimators for Bernoulli Probabilities of Pairwise Comparisons. In *IEEE Transactions on Information Theory*, 2019.

[Sha79] Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979.

[SS97] Hung-Min Sun and Shiuh-Pyng Shieh. Secret Sharing in Graph-Based Prohibited Structures. In *INFOCOM'97*, 1997.

[WBP16] Tengyao Wang, Quentin Berthet, and Yaniv Plan. Average-Case Hardness of RIP Certification. In *Advances in Neural Information Processing Systems*, 2016.

[Yao82] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd FOCS*, pages 80–91. IEEE Computer Society Press, November 1982.

A Share-Size of Statistical Threshold Secret-Sharing Schemes

Kilian and Nisan [KN90] proved a lower bound of $\log n$ on the share size of information-theoretic 2-out-of-n secret-sharing schemes. Another proof of the lower bound was given in [CCX13]. We next generalize this lower bound to schemes in which the security is only statistical, the correctness is not perfect, and public information is allowed. We first prove the result for secret-sharing schemes without public information. For the next lemma, we say that a secret-sharing scheme has total correctness error δ if the sum of the probabilities of reconstruction errors over all minimal authorized sets and the two secrets $b \in \{0,1\}$ is at most δ.

Lemma A.1. *The share size of any information-theoretic* 2*-out-of-*n *secret-sharing scheme without public information and with* $1/3$*-indistinguishability and total reconstruction error* $1/2$ *is at least* $\log n - 5$.

Proof. For $\varepsilon > 0$, consider an ε-secure 2-out-of-n secret-sharing scheme. Fix $i \in [n]$; let D be the domain of shares. For $b \in \{0,1\}$, let $p_b(x)$ be the distribution induced by the i-th share over D when the secret is b. By the ε-security of the scheme,

$$\sum_{x \in D : p_1(x) \geq p_0(x)} (p_1(x) - p_0(x)) \leq \varepsilon. \tag{20}$$

Let X_0 and X_1 be two independent random variables distributed according to distributions $p_0(x)$ and $p_1(x)$, respectively and let $G = \{x \in D : p_0(x) \geq \varepsilon p_1(x)\}$. Then,

$$\Pr[X_0 = X_1] = \sum_{x \in D} (p_1(x)p_0(x)) = \sum_{x \in D \setminus G} p_1(x)p_0(x) + \sum_{x \in G} p_1(x)p_0(x) \geq \sum_{x \in G} \varepsilon p_1^2(x). \tag{21}$$

By (20),

$$\varepsilon \geq \sum_{x \in D \setminus G} (p_1(x) - p_0(x)) \geq \sum_{x \in D \setminus G} p_1(x)(1-\varepsilon) \implies \sum_{x \in G} p_1(x) \geq \frac{1-2\varepsilon}{1-\varepsilon}. \tag{22}$$

Next, by Cauchy-Schwarz,

$$\left(\frac{1-2\varepsilon}{1-\varepsilon}\right)^2 \leq \left(\sum_{x \in G} p_1(x) \cdot 1\right)^2 \leq \left(\sum_{x \in G} p_1^2(x)\right)\left(\sum_{x \in \mathcal{G}} 1^2\right) \leq |D| \sum_{x \in G} p_1^2(x). \tag{23}$$

By (21) and (23),

$$\Pr[X_0 = X_1] \geq \frac{\varepsilon}{|D|}\left(\frac{1-2\varepsilon}{1-\varepsilon}\right)^2. \tag{24}$$

We next use an argument from the unpublished result by Kilian and Nisan [KN90]. Let $(S_1, \ldots, S_n)$ denote the joint distribution of the n shares induced by the scheme conditioned on the secret being 0. Let $(S'_1, \ldots, S'_n)$ denote the joint distribution of the n shares conditioned on the secret being 1, drawn independently of $(S_1, \ldots, S_n)$. If $S_i = S'_i$ and $S_j = S'_j$ occur simultaneously for $i \neq j$, then the reconstruction errs either for the secret 0 or for the secret 1. This implies that

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \Pr[S_i = S'_i, S_j = S'_j] \leq 1/2. \tag{25}$$

Hence,

$$\begin{aligned} 1 \geq \Pr[\exists j \text{ s.t. } S_j = S'_j] &\geq \sum_{i=1}^{n} \Pr[S_i = S'_i] - \sum_{i=1}^{n} \sum_{j=i+1}^{n} \Pr[S_i = S'_i, S_j = S'_j] \\ &\geq \frac{n\varepsilon}{|D|}\left(\frac{1-2\varepsilon}{1-\varepsilon}\right)^2 - \frac{1}{2}, \end{aligned}$$

where the last inequality uses (24) and (25). Setting $\varepsilon = 1/3$ in the above inequality, we get $|D| \geq n/18$ and the share size, i.e., $\log|D|$, is at least $\log n - 5$. □

We next prove the lower bounds for schemes with public information. To eliminate the public information in secret-sharing schemes with perfect correctness and security, we can fix any value I of the public information that has positive probability and share the secret conditioned on the public information being I. For statistical secret-sharing schemes this is more delicate; it can be done for 2-out-of-n secret-sharing schemes since the number of minimal authorized sets and the number of unauthorized sets is small. We note that removing the public information for 2-out-of-n secret sharing schemes and forbidden graph secret-sharing schemes is easy if we start with a scheme with $1/12n$-indistinguishability and reconstruction error $1/2n^4$; we get a secret-sharing scheme with the same share size and without public information that has 1/3-indistinguishability and reconstruction error. This is done by using the Markov inequality and the union bound. As we do not want to lose too much in the indistinguishability and $1/n^2$ reconstruction error, we use a somewhat more complicated argument that results in a scheme with $n/2$ parties.

Theorem A.2. *The share size of any information-theoretic* 2*-out-of-n secret-sharing scheme with public information and with* 1/12*-indistinguishability and reconstruction error* $1/(4n^2)$ *is at least* $\log n - 6$*.*

Proof. Let Π be an information-theoretic 2-out-of-n secret-sharing scheme with public information and with 1/6-indistinguishability and correctness error $1/(4n^2)$. Let I be a possible value of the public information in Π and Π_I be the scheme Π conditioned on the public information being I. Note that Π_I is a secret-sharing scheme without public information. We show that there exists an I such that Π_I implies a secret-sharing scheme without public information and with 1/6-indistinguishability and total correctness error less than 1/2. We then apply Lemma A.1 to deduce the theorem.

Fix a party i, and let S_i and S'_i be random variables denoting the share of party i with secret 1 and with the secret 0 respectively and I be the random variable denoting the public information. By the security of Π, the statistical distance between I, S_i and I, S'_i is at most 1/12. This implies that the expected value over I of the statistical distance between S_i and S'_i conditioned on I is at most 1/12. By the Markov inequality, if we sample I as in Π, then with probability at most 1/4, the statistical distance between S_i and S'_i conditioned on I exceeds 1/3. Next, by the correctness of Π, the sum of the probabilities of the reconstruction errors over all sets $\{i,j\}_{i \neq j}$ and the two secrets $b \in \{0,1\}$ is at most $1/2n$; the probabilities are taken over the public information and the share of party i. By the Markov inequality, if we sample I as in Π, then with probability at most 1/4, the sum of the reconstruction errors conditioned on I exceeds $2/n$.

To conclude, for the fixed party i, with probability at least half over I sampled as in Π, both the statistical distance is at most 1/3 and the sum of the probabilities of error is at most $2/n$. This implies that there exists a value I of the public information such that for at least half of the parties, the distance in the distributions of the share of the parties given the secret 0 and 1 respectively is at most half 1/3 and the sum of the error probabilities is at most $1/n$. We fix a set of $n/2$ such parties and consider the secret-sharing Π_I restricted to the parties in this set; this is a 2-out-of-$n/2$ secret-sharing scheme with 1/3-indistinguishability and total reconstruction error of at most $1/2 \cdot n/2 \cdot 2/n = 1/2$ (where we sum the errors over all $n/2$ parties counting each set twice, ignoring the fact that the error per party was computed with respect to all n parties). By Lemma A.1, the share size in the resulting scheme (hence also in Π) is at least $\log(n/2) - 5 = \log n - 6$. □