On-line secret sharing*

László Csirmaz[†] Gábor Tardos[‡]

Abstract

In a perfect secret sharing scheme the dealer distributes shares to participants so that qualified subsets can recover the secret, while unqualified subsets have no information on the secret. In an on-line secret sharing scheme the dealer assigns shares in the order the participants show up, knowing only those qualified subsets whose all members she has seen. We often assume that the overall access structure (the set of minimal qualified subsets) is known and only the order of the participants is unknown. On-line secret sharing is a useful primitive when the set of participants grows in time, and redistributing the secret when a new participant shows up is too expensive. In this paper we start the investigation of unconditionally secure on-line secret sharing schemes.

The complexity of a secret sharing scheme is the size of the largest share a single participant can receive over the size of the secret. The infimum of this amount in the on-line or off-line setting is the on-line or off-line complexity of the access structure, respectively.

For paths on at most five vertices and cycles on at most six vertices the on-line and offline complexities are equal, while for other paths and cycles these values differ. We show that the gap between these values can be arbitrarily large even for graph based access structures.

We present a general on-line secret sharing scheme that we call first-fit. Its complexity is the maximal degree of the access structure. We show, however, that this on-line scheme is never optimal: the on-line complexity is always strictly less than the maximal degree. On the other hand, we give examples where the first-fit scheme is almost optimal, namely, the on-line complexity can be arbitrarily close to the maximal degree.

The performance ratio is the ratio of the on-line and off-line complexities of the same access structure. We show that for graphs the performance ratio is smaller than the number of vertices, and for an infinite family of graphs the performance ratio is at least constant times the square root of the number of vertices.

Keywords: secret sharing; online algorithm; complexity; entropy method; performance ratio.

1 Introduction

Secret sharing is an important cryptographic primitive. It is used, for example, in protocols when individual participants are either unreliable, or participating parties don't trust each other, while they together want to compute reliably and secretly some function of their private data. Such protocols are, among others, electronic voting, bidding, data base access and data base computations, distributed signatures, or joint encryptions. Search for (efficient) secret sharing schemes led to problems in several different branches of mathematics, and a rich theory has been developed. For an extended bibliography on secret sharing see [27].

Secret sharing is a method to hide a piece of information – the *secret* – by splitting it up into pieces, and distributing these shares among participants so that it can only be recovered from certain subsets of the shares. Usually it is a trusted outsider – the *dealer* – who produces the shares and communicates them privately to the participants. Thus to define a secret sharing scheme we need to describe what the dealer should do.

*This research was supported by the "Lendület Program" of the Hungarian Academy of Sciences.

†Central European University, Budapest. This research was partially supported by grant NKTH OM-00289/2008

‡School of Computing Science, Simon Fraser University and Rényi Institute, Budapest. This research was partially supported by NSERC Discovery grant and Hungarian OTKA grants T-046234, AT-048826 and NK-62321

As schemes can easily be scaled up by executing several instances independently, the usual way to measure the efficiency of a scheme is to look at the ratio between the size of the largest share any participant receives and the size of the secret. The size of the shares and that of the secret is measured by their entropy, which is roughly the minimal expected number of bits which are necessary to define the value uniquely. We write $\mathbf{H}(\xi)$ to denote the Shannon entropy of the random variable ξ [14].

Let P denote the set of participants. We assume that both the secret ξ_s and the share ξ_i assigned to a participant $i \in P$ are random variables distributed over a finite range and all these variables have a joint distribution. We further require that $\mathbf{H}(\xi_s) > 0$ to avoid trivialities. The dealer simply draws the secret and the shares randomly according to the given distribution, and then distributes the (random) values of the shares to the participants. The *complexity* (or worst case complexity) of the scheme $\mathcal{S}$, denoted by $\sigma(\mathcal{S})$ is simply the ratio between the size of the largest share and size of the secret:

$$\sigma(\mathcal{S}) = \frac{\max_{i \in P} \mathbf{H}(\xi_i)}{\mathbf{H}(\xi_s)}.$$

The inverse of the complexity is dubbed as the *rate of the scheme*, in a strong resemblance to the decoding rate of noisy channels.

We call a hypergraph Γ on the vertex set P an *access structure*. A subset of the participants is *qualified* if it contains a hyperedge and it is *unqualified* otherwise. We say that the secret sharing scheme $\mathcal{S}$ *realizes* Γ if the values of the shares of the participants in any qualified set uniquely determine the value of the secret, but the shares of a set of the participants in an unqualified subset are statistically independent of the secret. Clearly, the non-minimal hyperedges in Γ play no role in defining which sets are qualified, so we can and will assume that the hyperedges in Γ form a *Sperner system* [7], i.e., no hyperedge contains another hyperedge. We further assume that the empty set is not a hyperedge as otherwise no scheme would realize Γ.

The *complexity* of Γ is the infimum of the complexities of all schemes realizing Γ:

$$\sigma(\Gamma) = \inf\{\sigma(\mathcal{S}) \,:\, \mathcal{S} \text{ realizes } \Gamma\},$$

this notation was introduced in [20]. By the result of Ito *et al.* [18], every non-trivial Sperner system has a complexity, i.e., every access structure is realized by some scheme. The complexity of their construction is the maximal degree of Γ. The *degree* of a vertex in a hypergraph is the number of hyperedges containing it. The *maximal degree* of Γ, denoted by $d(\Gamma)$, is the maximum of the degrees of vertices of Γ. The complexity of the scheme realizing Γ can be reduced from $d = d(\Gamma)$ to $d - (d-1)/n$, where n is the number of participants. Another general construction for arbitrary access structure is given by Maurer [23]. It is, in a sense, a dual construction, and its complexity is the maximal number of maximal unqualified subsets a certain participant is *not* a member of. Both type of constructions show that the complexity of any access structure is at most exponential in the number n of participants. It is an open problem whether there exists an access structure with $\sigma(\Gamma) \geq n$.

A simple observation yields that $\sigma(\Gamma) \geq 1$ for all access structures Γ with at least one hyperedge, see, e.g., [9]. Access structures with complexity exactly 1 are called *ideal*. An intense research was conducted to characterize ideal access structures. For example, results in [20] connect the problem of characterizing ideal access structures to representability of certain matroids.

A widely studied special case is when all minimal qualified sets are pairs, that is, the access structure is a graph. Stinson [26] showed that the complexity of a graph G is at most $(d+1)/2$ where d is the maximal degree of the graph. This, together with the lower bound in [9] established the complexity of both the path and the cycle of length $n > 4$ to be $3/2$. Blundo *et al.* in [5] showed that the $(d+1)/2$ bound is tight for certain d-regular graph families. Lower and upper bounds on the complexity on graphs with a few nodes were investigated in [15]. The complexity of trees was determined in [13] to be $2 - 1/c$ where c is the size of the largest core[1] in the tree. In particular, the complexity of every tree is strictly less than 2.

[1] A *core* is a connected subtree such that each vertex in the core is connected to a vertex not in the core.

On the other hand, based on the result of Erdős and Pyber [16], Blundo *et al.* [4] show that the complexity of any graph G on n vertices is $O(n/\log n)$. So far, however, no graph has been found with complexity above $\Theta(\log n)$. The graph with the largest known complexity (as a function of the number of vertices) is from [11], namely the edge-graph of the d dimensional hypercube. This graph is d-regular, has 2^d vertices, and its complexity is $d/2$.

1.1 On-line secret sharing

In the model discussed so far the dealer generates all shares simultaneously, and communicates them to the corresponding participants. We call such schemes *off-line*. In the *on-line share distribution* participants form a queue, and they receive their shares in the order they appear. When a participant arrives the dealer is told all those qualified subsets which are formed by this and previously seen participants. We often assume that the dealer knows the entire access structure at the beginning but she doesn't know the order in which the participants arrive or the identity of the participant when he arrives. Still she has to assign a share to him and she cannot modify this share later. In this respect on-line secret sharing resembles on-line graph coloring: there the color of the next vertex should be decided knowing only that part of the graph which is spanned by this and previous vertices.

On-line secret sharing is a useful primitive when the set of participant is not fixed in advance and shares are assigned as participants show up. The usual way to handle such cases is by redistributing all shares every time a new participant shows up. Redistribution, however, has high cost, while using on-line secret sharing can be cheap and efficient.

The study of on-line schemes stems from mainly theoretical interest. It is a quite natural extension, and definitely more powerful than the traditional static schemes. If we can provide a more powerful tool without giving up too much from the efficiency, then why should we settle for less? On the other hand, if on-line schemes turn out to be prohibitively complicated, then we should discard them as interesting but unpractical. As our results show, for general access structures when each participant is in a relatively small number of qualified subsets (say, less than ten) which is a reasonable assumption, then independently of the total number of the participants, the complexity of the system, both on-line and off-line, has a very low complexity (below 10). The best common bounds for the complexity of on-line and off-line schemes are quite close. Thus, if the access structure has no special properties, the efficiency of an on-line scheme is not inferior to that of the more restricted off-line scheme. Of course, there are special structures – and we will present some of them – where the efficiencies are quite far away from each other.

In our work we took the analogy with graph coloring. There is a quite extensive literature for on-line graph coloring, see the bibliography in [24]. On-line graph coloring is interesting both for its theoretical and practical aspects. We hope the same is, or will be, true for the present investigation.

The *on-line secret sharing* of Cachin [8] and follow-up papers differ from our approach significantly. Cachin's model considers computationally secure schemes only, while our schemes are unconditionally secure. In addition, it requires other authentic (but not secret) publicly accessible information, which can (or should) be broadcast to the participants over a public channel. In our schemes only information possessed by the participants is necessary to recover the secret. We are mainly interested in proving lower and upper bounds on the complexity of such schemes compared to the complexity of unconditionally secure off-line schemes, which are not touched in [8] at all.

Dynamic access structures were investigated by Blundo *et al.* in [3]. Their model provides unconditional security, and the dealer is able to activate a particular access structure out of a given collection by sending an appropriate *broadcast* message to all participants. The dynamic is provided by the dealer's ability to choose from a range of possible access structures, while in our on-line schemes it is the unpredictability of the order participants appear in which makes the scheme dynamic.

1.2 Our contribution

On-line secret sharing appeared first in the conference presentation [12]. In this paper we give a precise definition of this notion and define the *on-line complexity* $o(\Gamma)$ of an access structure Γ as the infimum of the complexity of an on-line secret sharing scheme realizing it. We present a general on-line secret sharing scheme that can realize any access structure. We call our scheme the *first-fit on-line secret sharing scheme* on account of its similarity to the simplest on-line graph coloring strategy.

Theorem 1.1 *The on-line secret sharing scheme* first-fit *realizes any access structure* Γ *with complexity* $d = d(\Gamma)$. *In particular,* $o(\Gamma) \leq d$.

As usual, P_n denotes the path on n vertices, and C_n denotes the cycle on n vertices. It is well known that the complexity of P_n is $3/2$ for $n \geq 4$ and complexity of C_n is also $3/2$ for $n \geq 5$, see, e.g., [9]. The following theorem separates the on-line and off-line complexities.

Theorem 1.2 (i) *For paths* P_n *with* $n \leq 5$ *and for the cycles* C_n *with* $n \leq 6$ *the on-line and off-line complexity is the same.*
(ii) *For paths* P_n *with* $n \geq 6$ *and for cycles* C_n *with* $n \geq 7$ *the on-line complexity is strictly above the off-line complexity.*
(iii) *The on-line complexity of both* P_n *and* C_n *tends to* 2 *as* n *tends to infinity. In fact,*

$$2 - \frac{1}{4n} \geq o(C_{n+1}) \geq o(P_n) \geq 2 - \frac{4}{n}.$$

Recall that, by [13], the complexity of a tree is below 2.

Theorem 1.3 *The on-line complexities of trees is unbounded. In particular, there exists an* n*-vertex tree* T_n *with* $o(T_n) \geq \lfloor\sqrt{n}\rfloor/2$. *Consequently the gap between* $o(\Gamma)$ *and* $\sigma(\Gamma)$ *can be arbitrarily large.*

The *performance ratio* tells us how much worse the on-line scheme must be compared to the best off-line scheme. The *secret sharing performance ratio of* Γ is defined to be $o(\Gamma)/\sigma(\Gamma)$. The similarly defined quantity for on-line graph coloring is sublinear in the number of vertices [19], and it is at least $n/\log^2 n$ for certain graphs with n vertices [24]. Our upper bound on the secret sharing performance ratio of graphs comes from an upper bound of the on-line complexity and the trivial lower bound of 1 for the off-line complexity:

Theorem 1.4 (i) *Let* $d = d(G)$ *be the maximal degree of the graph* G *on* n *vertices. Then* $o(G)$, *and therefore the secret sharing performance ratio, is at most* $d - 1/(2dn)$.
(ii) *For some graphs on* n *vertices the performance ratio is at least* $\frac{1}{3}\sqrt{n}$.

Finally we show that the first-fit scheme is *never* the best on-line scheme. The gain, however, can be exponentially small in cases when minimal qualified subsets are big. Recall from Theorem 1.1 that the first-fit scheme has complexity $d(\Gamma)$.

Theorem 1.5 *Let* Γ *be an access structure,* $d = d(\Gamma)$ *be the maximal degree of* Γ, n *be the number of vertices in* Γ, *and* $r \geq 2$ *be an upper bound on the size of any hyperedge in* Γ *(thus* $r = 2$ *for graphs). There is an on-line secret sharing scheme realizing* Γ *with complexity at most*

$$d - \frac{1}{ndM + nd^2 + n}$$

where $M = \min(r \cdot n^{2r-3}, 3^{n-1})$.

1.3 Organization

The rest of the paper is organized as follows. In section 2 we give precise definition for the on-line secret sharing. In section 3 we describe variants of our general first-fit scheme and prove Theorem 1.1. Section 4 deals with the on-line complexity of paths and cycles and we prove there Theorem 1.2(i). In section 5 we exhibit graphs with the on-line complexity close to the maximal degree. These include the long paths and cycles proving Theorem 1.2(ii) and trees proving Theorem 1.3. Finally, in Section 6 we show that the first-fit scheme is never optimal proving Theorems 1.4 and 1.5.

2 On-line secret sharing schemes

Having defined off-line secret sharing schemes in the preceding section we define on-line secret sharing here. On-line secret sharing relates to the secret sharing in the same way as on-line graph coloring relates to graph coloring. Here the structure Γ is known in advance, and the participants receive their shares one by one and the assigned share cannot be changed later on. Participants appear according to an unknown permutation. When a participant p shows up, his identity (as a vertex of Γ) is not revealed, only those qualified subsets are shown to the dealer which p is the last member of (i.e., all other members arrived previously). Based only on the emerging hypergraph (on the participants who have arrived so far) the dealer assigns a share to the new participant. At the end the dealer will see a permuted version of the access structure Γ and the shares distributed must satisfy the usual properties: the collection of shares assigned to a qualified subset must determine the secret, and the collection of shares of an unqualified subset must be independent of the secret.

2.1 An example

Suppose we have three participants: a, b, c, and the minimal qualified subsets are $\{a,b\}$ and $\{b,c\}$. Thus this access structure is based on the path P_3 on three vertices. In Scheme 1 below we describe an on-line scheme realizing P_3. As can be checked readily, qualified subsets can always

The dealer chooses two independent random bits: r and t, and sets the secret to be $r \oplus t$ the modulo 2 sum of these values.
When the first participant (A) shows up, he could be any of a, b, and c. In any case he gets the share r. When the next participant shows up (B), the dealer also learns whether A and B together form a qualified set.

If* $\{A,B\}$ *is independent: *the last participant (who did not appear yet) is b, and then B will get the same share as A did (that is, r). The last participant will receive t.*

If* $\{A,B\}$ *is a qualified subset: *any of A and B can be the middle person b. Nevertheless, B gets the share t, thus $\{A,B\}$ can recover the secret. When the last person arrives, he is connected to either A or B, but not both. If he is connected to A, then he receives the same share as B did (that is t), and if he is connected to B, then he receives the same share as A did (that is s).*

Scheme 1: A sample on-line scheme

recover the secret, and unqualified subsets have no information on the secret. Every participant in the scheme receives a single bit no matter which order they arrived. The secret is a single bit, thus the complexity of this scheme is 1.

2.2 Formal definition

To formalize this concept, we assume all the shares that may be assigned to participants form a large (predetermined) finite collection of random variables $\{\xi_\alpha : \alpha \in \Omega\}$. As usual, these and the secret ξ_s are random variables with a finite range and with a joint distribution. We assume $\mathbf{H}(\xi_s) > 0$. The dealer assigns one of the variables ξ_α to each participant as soon as he shows up. The choice of the index α for a participant depends only on the emerging hypergraph, i.e., the set of hyperedges consisting of this and earlier participants. In particular, assuming there is no singleton hyperedge, the first participant always gets the same variable. Notice that the distribution process does not depend on the values of the random variables, in fact one can visualize the process as assigning variables to participants, and only after all assignments evaluating the variables according to their joint distribution.

An on-line secret sharing scheme realizes the access structure Γ if at the end of the process, provided that the emerging hypergraph is indeed a vertex-permuted copy of Γ, the shares of every qualified subset determine the secret and the shares of every unqualified subset are independent of the secret. Notice however that many sets of the random variables ξ_α get never assigned to participants simultaneously, and those collections do not have to satisfy any requirement.

The *complexity of the scheme* $\mathcal{S}$ is the size of the largest share divided by the size of the secret:

$$\sigma(\mathcal{S}) = \frac{\max\{\mathbf{H}(\xi_\alpha) : \alpha \in \Omega\}}{\mathbf{H}(\xi_s)}.$$

The *on-line complexity* $o(\Gamma)$ of an access structure Γ is the infimum of the complexities of all on-line schemes realizing Γ:

$$o(\Gamma) = \inf\{\sigma(\mathcal{S}) : \mathcal{S} \text{ is on-line and realizes } \Gamma\}.$$

By fixing the order of the participants, any on-line scheme can be downgraded to an off-line scheme. Consequently the on-line complexity cannot be smaller than the off-line one: $o(\Gamma) \geq \sigma(\Gamma)$ holds for any Γ.

3 First-fit on-line scheme

In this section we present a general on-line secret sharing scheme. We name it *first-fit scheme* because of the analogy to the first-fit on-line graph coloring algorithm [24]. The analogy even carries further. As first-fit on-line coloring is oblivious of the graph structure on the unseen vertices, similarly our first-fit scheme works without the knowledge of the "global" access structure. However, for our scheme to work the dealer must know the maximum degree d. For graphs we present a version of the scheme later where the maximum degree does not have to be known in advance. This modified scheme has complexity $d+1$ instead of d given by the first-fit scheme.

For on-line schemes we distinguish the hyperedges containing a participant p as *backward edges* and *forward edges* at v, with backward edges being those that are revealed when p arrives, and the forward edges being those that will be revealed later.

Let us assume that d is the maximal degree of an unknown access structure. The first-fit on-line secret sharing scheme works described in the box below.

As an example, we give a more detailed description in Scheme 3 for the access structure P_3. Here the maximal degree d is two.

Proof (Theorem 1.1) To check that the general first-fit scheme defined as Scheme 2 is indeed a correct secret sharing scheme realizing the access structure Γ, first we note that if the maximal degree is d then no participant runs out of unassigned bits.

Second, the complexity of the scheme is d as each participants receives exactly d bits and the secret is a single uniform bit. The participants in a hyperedge E can determine the secret by adding mod 2 the bits which were assigned to E. All bits received by an unqualified set together with the secret form a set of independent random bits. So the first-fit scheme realizes any access structure of maximal degree (at most) d. $\square$

The secret is a uniform random bit s.
When a participant p arrives do the following for each backward edge E containing p:

> *For each participant $q \in E$ different from p we select a previously unassigned (random) bit given to q previously, and assign it to the hyperedge E. We also give a bit to p which is also assigned to the hyperedge E. We choose this last bit in such a way that the mod 2 sum of all bits assigned to E be the secret s.*

Finally, if the number of backward edges at p is $m < d$, then we give $d-m$ fresh uniform random unassigned bits to p (in anticipation of the forward edges).

Scheme 2: General first-fit scheme

The secret is a uniform random bit s.
When the first participant A arrives, there is no backward edge, thus he gets two independent random bits A_1 and A_2.
When the second participant B arrives, then we distinguish two cases:

- $\{A, B\}$ ***is a backward edge:*** *the (first) unused random bit from participant A assigned to this edge will be A_1. Thus the first bit B receives is $A_1 \oplus s$ from which $\{A, B\}$ can recover the secret. Also B receives a new fresh random bit denoted as B_1. Either A or B can be the middle vertex of P_3. When the third participant C arrives then*
  - ***if*** $\{C, A\}$ ***is an edge:*** *C gets $A_2 \oplus s$ plus an extra random bit.*
  - ***if*** $\{C, B\}$ ***is an edge:*** *C gets $B_1 \oplus s$ plus an extra random bit.*
- $\{A, B\}$ ***is unqualified:*** *in this case there is no backward edge, thus B gets two fresh random bits: B_1 and B_2. A and B are the two endpoints of the path P_3.*

  When C arrives, there will be two backward edges: $\{C, A\}$, and $\{C, B\}$. The first backward edge is assigned the random bit A_1, and C receives $A_1 \oplus s$. The second backward edge $\{C, B\}$ gets B_1, and C also receives $B_1 \oplus s$, a total of two bits.

Scheme 3: Details of first-fit scheme for P_3

We remark that the bound given by this theorem matches the complexity of the general off-line secret sharing scheme of Ito *et al.* [18].

For graphs there is a modified version of the first-fit scheme 2 detailed in Scheme 4. The secret is still a uniform random bit s, but each participant receives a share whose size is only one more than the number of backward edges containing that vertex, i.e., edges which are revealed when the vertex arrives. Thus the maximum possible share size is $d+1$, slightly worse than the d above.

The secret is a uniform random bit s, the access structure is a graph G.
When the participant p arrives, the dealer gives him a new random bit r_p (independently from every other bits), and do the following for each backward edge $(q, p) \in G$:

> *take the random bit r_q assigned to q, and give $r_q \oplus s$ to p as well.*

Scheme 4: Special first-fit scheme for graphs

The advantage of this modified scheme is that the dealer needs not to know the maximum degree d in advance. It is easy to check that this scheme realizes any graph. It is interesting to note however, that we could not find any analogous scheme for general hypergraphs.

Yet another version of the first-fit scheme for graphs is when in Scheme 4 we simply do not give

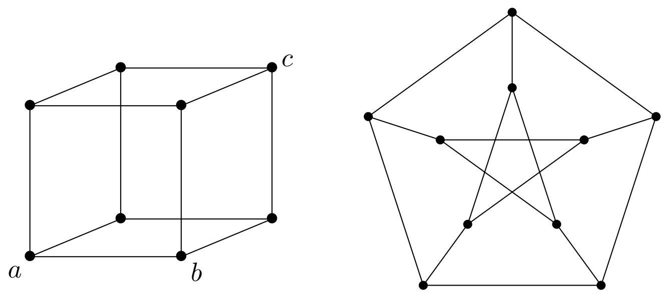

Figure 1: Edge graph of the 3d cube and the Petersen graph

the new random bit r_p to p whenever p has the maximum number d of backward edges. For this scheme to work we need to know d in advance. The advantage compared to the general first-fit scheme 2 is that most participants receive fewer than d bits, only participants with d or $d-1$ backward edges receive a d bit share.

4 Paths and cycles

There are cases when the on-line and off-line complexity coincide. The simplest ones are covered by the Claim 4.1. To state it we need some definition. Let Γ be a hypergraph and S be a subset of the vertices of Γ. The *sub-hypergraph of* Γ *induced* (or spanned) *by* S is the hypergraph with vertex set S and with those hyperedges of Γ that are contained in S. For simplicity we call induced sub-hypergraphs *substructures.* We call a hypergraph Γ *fully symmetric* if each isomorphism between two of its substructures can be extended to an automorphism of Γ.

An easy example for a fully symmetric hypergraph is the (n, k)-threshold structure. It consists of all k-element subsets of an n-element vertex set. As all permutations of the vertex set is an automorphism of the structure, any permutation of a subset can be extended to an automorphism of the whole structure.

The edge graph of the 3d cube depicted on Figure 1 is *not* fully transitive. To see this, the subgraph spanned on vertices a, b, and c has the automorphism which swaps a and b (and leaves c untouched). This automorphism cannot be extended to an automorphism of the whole graph.

A less trivial example for a fully symmetric hypergraph is the so-called *Petersen graph* depicted on Figure 1. This graph is a 3-regular graph on 10 vertices with lots of symmetries.

Claim 4.1 *For a fully symmetric access structure the on-line and off-line complexities are equal.*

Proof Suppose we have an off-line secret sharing scheme realizing a fully symmetric access structure Γ consisting of the shares ξ_p for vertices p of Γ and ξ_s for the secret. We can use the very same variables for an on-line secret sharing scheme as follows. We maintain an isomorphism α between the emerging hypergraph and a substructure of Γ and give the next participant q the share $\xi_{\alpha(q)}$. We keep ξ_s in its role as the secret. Before the first participant arrives α is empty. As Γ is fully symmetric, whenever a new participant arrives and the emerging hypergraph grows, we can extend α to this new vertex so that the value of α does not change on the older vertices and α remains to be an isomorphism between the emerging hypergraph and a substructure of Γ. At the end of the on-line process α becomes an isomorphism between the full access structure and Γ. As the off-line scheme realizes Γ, the constraints on qualified and unqualified subsets will hold in this on-line scheme as well. $\square$

As a toy example, let us consider the above procedure when Γ is the fully symmetrical structure C_4. Let the four vertices be a, b, c and d, and suppose that a perfect off-line secret sharing scheme assign the shares $\xi_a, \ldots, \xi_d$ to these vertices.

When the first participant arrives, we pretend him to be a, and give him the share ξ_a. When the second participant arrives, we learn whether he is connected to the first one or not. If they are connected, then we pretend him to be b and give him the share ξ_b, otherwise we think of him as C and assign him the share ξ_c. As the participants are indeed the vertices of some C_4, after all of them arrives, their "pretended" and their "real" roles form an isomorphism between these structures which established the correctness of the on-line scheme.

Note that the strong symmetry requirement of Claim 4.1 seems to be necessary. The weaker assumption that the automorphism group of Γ is *transitive* on the vertices and/or on the hyperedges is not enough. As a counterexample, consider C_n, the cycle on $n \geq 7$ vertices. Its automorphism group is transitive on both the edges and vertices, but it is not transitive on certain isomorphism classes of induced substructures. For example no automorphism brings a pair of second neighbors to a pair of third neighbors, despite the fact that they induce isomorphic (empty) subgraphs. The off-line complexity of C_n is $3/2$, but the on-line complexity is strictly larger than this value (and approaches 2 as n goes to infinity) by Theorem 1.2.

Let Γ' be a hypergraph obtained from Γ by replacing each vertex of Γ by a nonempty class of equivalent vertices, and replacing each hyperedge with the complete multipartite hypergraph on the corresponding classes. We call Γ' a *blowup* of Γ. Note that $\sigma(\Gamma') = \sigma(\Gamma)$ since one can assign the same random variable to all equivalent vertices in a class. We shall see later that the on-line complexity of the blowup can be larger than that of Γ. Indeed, Lemma 5.3 implies that the blowups of the simple graph G_0 with three vertices and a single edge have unbounded on-line complexity.

Claim 4.1 applies to the *threshold structures*, these are the complete uniform hypergraphs. Among graphs it applies to the complete graphs and it also applies to the complete multi-partite graphs with equal number of vertices in each class. All these access structures have complexity 1, so their on-line complexity is also 1. The same is true for arbitrary complete multi-partite graphs (the blowups of complete graphs) as they are induced subgraphs of some fully symmetric complete multipartite graph.

With these preliminaries, we turn to the complexity of paths and cycles. First we show that the on-line complexity of short paths and cycles are the same as their off-line complexity.

Proof (Theorem 1.2(i)) P_2, and C_3 are complete graphs, P_3 and C_4 are complete bipartite graphs, so their on-line and off-line complexity are the same and equal to 1. C_5 is neither complete, nor complete bipartite graph, but it is fully symmetric. So its on-line and off-line complexities agree by Claim 4.1. P_4 is not fully symmetric, still its on-line and off-line complexities are both $3/2$. To see this notice that P_4 is an induced subgraph of C_5, so we have $o(P_4) \leq o(C_5) = \sigma(C_5)$ and it is well known that $\sigma(P_4) = \sigma(C_5) = 3/2$, see e.g., [9]. A similar argument shows that $o(P_5) = \sigma(P_5) = o(C_6) = \sigma(C_6) = 3/2$ once we show the bound $o(C_6) \leq 3/2$. We show this by presenting an on-line secret sharing scheme of complexity $3/2$ realizing C_6.

Note that in Scheme 5 the size of the secret is $\mathbf{H}(\xi_s) = 2$, while the size of any share is 3, so the complexity of the scheme is $3/2$ as claimed.

All neighboring pairs in Σ and Π can determine the secret. For example, from ξ_7 and ξ_5 one can get $x = e + (e + x)$, then extract the value $b + c + d = (b + c + d + x) + x$, finally $y = (f + y) + (b + c + d) + a + e$. We leave it to the reader to verify that Scheme 5 works indeed for every permutation of the vertices. $\square$

5 The entropy method

In this section we prove lower bounds on the on-line complexity of access structures. We start with recalling the so-called entropy method discussed, among others, in [9, 10] as that seems to be the most powerful method for proving lower bounds for the off-line complexity. Then we extend it to the on-line model.

Let us consider a secret sharing scheme with the set of participants being P. For any subset A of P we define $f(A)$ as the joint entropy of the random variables (the shares) belonging the

This scheme uses random bits a, b, c, d, e, f and x, y, z whose joint distribution is uniform on the values satisfying $a+b+c+d+e+f=x+y+z=0$. Here and in the list below summation is understood modulo 2. The random variables representing the shares and the secret ξ_s are as follows

$$\begin{aligned}
\xi_1 &= (a, b+x, c),\\
\xi_2 &= (b, c+y, d),\\
\xi_3 &= (c, d+z, e),\\
\xi_4 &= (d, e+x, f),\\
\xi_5 &= (e, f+y, a),\\
\xi_6 &= (f, a+z, b),\\
\xi_7 &= (c, b+c+d+x, e+x),\\
\xi_8 &= (f+y, a+b+c+y, b),\\
\xi_s &= (x, y, z).
\end{aligned}$$

Let Σ be the cycle on the six vertices ξ_1, ξ_2, ξ_3, ξ_4, ξ_5 and ξ_6 in this cyclic order and Π be the cycle on the vertices ξ_1, ξ_2, ξ_7, ξ_5, ξ_4 and ξ_8 in this cyclic order. We assign the variables to participants such that at the end the assignment represents an isomorphism between the emerged access structure and either Σ or Π. Notice that if we succeed, then the conditions on qualified and unqualified subsets are satisfied as both cycles Σ and Π represent off-line secret sharing schemes realizing C_6.
We start with assigning shares to participants from the intersection of Σ and Π (that is, we assign one of ξ_1, ξ_2, ξ_4 or ξ_5 until we can). We choose the variables in such a way that at any time the assignment represents an isomorphism between the emerging graph and an induced subgraph of the intersection. We fail when either two adjacent edges appear in the emerging graph or three vertices form an independent set. At that point we commit to either Σ or Π and assign variables so that at the end we get an isomorphism to the selected cycle.

Scheme 5: Optimal on-line scheme for C_6

members of A, divided by the entropy of the secret:

$$f(A) = \frac{\mathbf{H}(\{\xi_i : i \in A\})}{\mathbf{H}(\xi_s)}. \tag{1}$$

The so-called Shannon inequalities for the entropy, see [14], can be translated to linear inequalities for f as follows.

a) $f(\emptyset) = 0$,

b) monotonicity: if $A \subseteq B$ then $f(B) \ge f(A)$,

c) submodularity: $f(A) + f(B) \ge f(A \cap B) + f(A \cup B)$.

Furthermore, if the scheme realizes an access structure Γ, then the conditions that qualified subsets determine the secret, while unqualified subsets are independent of it imply further inequalities:

d) strict monotonicity: if $A \subset B$, A is unqualified but B is qualified, then $f(B) \ge f(A) + 1$,

e) strict submodularity: if A and B are both qualified but $A \cap B$ is unqualified, then $f(A) + f(B) \ge f(A \cap B) + f(A \cup B) + 1$.

We call a real function f satisfying the conditions a)–e) above an *entropy function* for Γ. An entropy function f is *α-bounded* if $f(A) \leq \alpha$ for all singleton sets A. The entropy method can be summarized as the following claim:

Claim 5.1 *For any access structure Γ there exists a $\sigma(\Gamma)$-bounded entropy function for Γ.*

Proof Let us consider a secret sharing scheme realizing Γ. Equation (1) defines the function f and as discussed above it is an entropy function for Γ. By the definition of complexity it is α-bounded for the complexity α of the scheme. In case the complexity $\sigma(\Gamma)$ is not achieved as the complexity of a scheme realizing Γ we use a compactness argument to finish the proof, see [22]. $\square$

The power of the entropy method lies in the fact that finding the smallest α such that an α-bounded entropy function exists for a given Γ is a linear programming problem and it is tractable for small access structures. This minimal α, denoted by $\kappa(\Gamma)$ in [20], is a lower bound on the complexity $\sigma(\Gamma)$.

Our next theorem gives the on-line version of the entropy method. It naturally extends to on-line complexities of *classes of access structures*, a natural concept to consider, but we restrict our attention to single access structures in this paper. Let us denote the family of substructures of an access structure Γ by $S(\Gamma)$.

Theorem 5.2 (i) *For every access structure Γ there exists a system $\{F_\Delta : \Delta \in S(\Gamma)\}$ such that F_Δ is a non-empty collection of $o(\Gamma)$-bounded entropy functions for Δ and they satisfy the following extension property: if μ is an isomorphism from $\Delta_1 \in S(\Gamma)$ to a substructure of $\Delta_2 \in S(\Gamma)$ and $f_1 \in F_{\Delta_1}$, then there exists a function $f_2 \in F_{\Delta_2}$ with $f_2(\mu(A)) = f_1(A)$ for any subset A of the vertices in Δ_1.*
(ii) *For an arbitrary substructure Δ of Γ one has an $o(\Gamma)$-bounded entropy function f for Γ that is symmetric on Δ, that is, for any automorphism μ of Δ one has $f(\mu(A)) = f(A)$ for all sets A of the vertices of Δ.*

Proof For (i) let us consider an on-line secret sharing scheme of complexity α realizing Γ. For $\Delta \in S(\Gamma)$ we consider all permutations of the vertices of Δ and the shares assigned to them when they arrive in that order. Each assignment yields an α-bounded entropy function for Δ through equation (1). We let F_Δ be the set of these functions.

To show that the extension property holds assume μ is an isomorphism between $\Delta_1 \in S(\Gamma)$ and a substructure of $\Delta_2 \in S(\Gamma)$, furthermore $f_1 \in F_{\Delta_1}$. Consider the permutation $v_1, \ldots, v_k$ of the vertices of Δ_1 yielding the entropy function f_1 and let f_2 be the entropy function for Δ_2 obtained from a permutation of its vertices starting with $\mu(v_1), \ldots, \mu(v_k)$ followed by the rest in an arbitrary order. After the arrival of the first k vertices the situation for the dealer is the same as when the vertices of Δ_1 arrived in the given order, so it distributes the same shares. After that she distributes further shares, but by the definition in (1) this will not effect the required equality $f_2(\mu(A)) = f_1(A)$ if A is a set of vertices of Δ_1.

This finishes the proof of part (i) in case there is an on-line secret sharing scheme of complexity $o(\Gamma)$ for Γ. If no such scheme exists we should use compactness again.

For part (ii) consider the sets F_Δ and F_Γ guaranteed by part (i) and pick an arbitrary entropy function $f_0 \in F_\Delta$. Any automorphism μ of Δ is an isomorphism between Δ and a substructure (namely Δ) of Γ, so we have an extension $f_\mu \in F_\Gamma$ with $f_\mu(\mu(A)) = f_0(A)$ for all sets A of vertices in Δ. Let f be the average of these functions f_μ for the automorphisms μ of Δ. It is easy to see that the linear constraints defining an entropy function are preserved under taking averages, so f is also an entropy function for Γ and it is also $o(\Gamma)$-bounded like all the functions f_μ. To see that f is symmetric on Δ consider an automorphism μ_0 of Δ and a set A of vertices of Δ and notice that $f(A)$ is the average of $f_\mu(A) = f_0(\mu^{-1}(A))$, while $f(\mu_0(A))$ is the average of the same values $f_\mu(\mu_0(A)) = f_0(\mu^{-1}\mu_0(A))$. $\square$

Note that making an entropy function for Γ symmetric on Γ is possible for off-line secret sharing schemes as well. But using Theorem 5.2(ii) one can make the entropy function symmetric on a well chosen substructure of Γ that may have much more automorphisms than Γ itself.

Theorem 5.3 *Let the graph G consist of a star with $d \geq 2$ edges and m isolated vertices. Then*

$$o(G) \geq d - \frac{d^3 - d^2}{2m + 2 + d^2 + d} > d - \frac{d^3}{2m}.$$

Proof Let H be the (empty) subgraph of G spanned by all vertices but the degree d vertex v, the center of the star. Let f be the $o(G)$-bounded entropy function for G that is symmetric on H, the existence of which is claimed by Theorem 5.2(ii). Note that, by symmetry, $f(A)$ is determined by $|A|$ for sets $v \notin A$, so for such a set of size k let us have $f(A) = c_k$.

Let $v_1, \ldots, v_d$ be the neighbors of v and $V_i = \{v_1, \ldots, v_i\}$. Let H be an arbitrary set of isolated vertices. By strict submodularity (rule e) for $2 \leq i \leq d$ we have

$$f(H \cup V_{i-1} \cup \{v\}) + f(H \cup \{v_i, v\}) \geq f(H \cup V_i \cup \{v\}) + f(H \cup \{v\}) + 1.$$

By submodularity (rule c) for $1 \leq i \leq d$ we have

$$f(H \cup \{v_i\} + f(H \cup \{v\}) \geq f(H \cup \{v_i, v\}) + f(H).$$

By rules a and c we have

$$f(H) + f(\{v\}) \geq f(H \cup \{v\}),$$

and finally by strict monotonicity (rule d) we have

$$f(H \cup V_d \cup \{v\}) \geq f(H \cup V_d) + 1.$$

Adding all these $2d + 1$ inequalities one obtains

$$\sum_{i=1}^{d} f(H \cup \{v_i\}) + f(\{v\}) \geq (d-1)f(H) + f(H \cup V_d) + d.$$

All terms except $f(\{v\})$ involve subsets of H, so the formula simplifies to

$$dc_{k+1} \geq (d-1)c_k + c_{k+d} + v - f(\{v\}),$$

where $k = |H|$. Introducing $\delta_i = c_{i+1} - c_i$ we can rewrite our inequality as

$$(d-1)\delta_k \leq \delta_{k+1} + \delta_{k+2} + \cdots + \delta_{k+d-1} + d - f(\{v\}).$$

Here $k = |H|$ is arbitrary in the range $0 \leq k \leq m$. When we add the $m + 1$ corresponding inequalities most δ_i cancel. Using the bounds $0 \leq \delta_i \leq c_1$ (coming from monotonicity and submodularity) on the remaining terms δ_i we obtain

$$\binom{d}{2} c_1 \geq (m+1)(d - f(\{v\})).$$

Finally as f is $o(G)$-bounded we have $c_1 \leq o(G)$ and $f(\{v\}) \leq o(G)$ yielding the bound on $o(G)$ stated. □

We use this Theorem to prove Theorems 1.2(iii) and 1.3.

Proof (Theorem 1.2(ii) and (iii)) For part (iii) notice that the graph G consisting a P_3 component and $\lceil n/2 \rceil - 2$ isolated vertices is an induced subgraph of P_n, which is also an induced subgraph of C_{n+1}. Thus we have $o(C_{n+1}) \geq o(P_n) \geq 2 - 4/n$, where the last inequality comes from Theorem 5.3. The upper bound on the on-line complexity of cycles comes from our general observation that first-fit is never optimal, as stated in Theorem 1.4(i). The proof of this latter statement is postponed to Section 6.

The lower bound proved in general establishes $o(P_n) > 3/2 = \sigma(P_n)$ for $n \geq 9$. To find the exact threshold as claimed in part (ii) it is enough to prove that $o(P_6) > 3/2$ as the longer paths and cycles contain P_6 as an induced subgraph. For this we use Theorem 5.2(ii) with the subgraph H of P_6 induced by the first, second, fourth and fifth vertex of the path. Notice that the automorphism group of H has order 8. Linear programming shows that there is no α-bounded entropy function on P_6 that is symmetric on H with $\alpha < 7/4$, thus the theorem tells us that $o(P_6) \geq 7/4$. In the Appendix we give a direct proof of this fact. □

Proof (Theorem 1.3) Consider the graph G consisting of a d-edge star and m isolated vertices and the tree T obtained by adding a vertex to G and connecting it to the center of the star and to the isolated vertices. Clearly $o(T) \geq o(G)$. Choosing $d = \lfloor \sqrt{n} \rfloor$ and $m = n - d - 2$ the tree $T = T_n$ has n vertices and Theorem 5.3 gives the claimed lower bound on its on-line complexity. $\square$

6 Not so tight bounds on on-line complexity

Stinson proved in [26] that the (worst case) complexity of any graph is at most $(d+1)/2$ where d is the maximal degree. This bound was proved to be almost sharp by van Dijk [15] where for each positive ε he constructed a graph with complexity at least $(d+1)/2 - \varepsilon$. Later Blundo *et al.* [5] constructed, for each $d \geq 2$, an infinite family of d-regular graphs with exact complexity $(d+1)/2$.

Theorem 1.1 claims that the on-line complexity is at most d for a degree d graph, and from Theorem 5.3 it follows that this bound is *almost tight*, namely, for each positive ε there is a d-regular graph with on-line complexity at least $d - \varepsilon$. In fact, the graph family defined in [5] works here as well, as these d-regular graphs have no triangles and have arbitrarily large independent subsets. These graphs also show that the on-line and off-line complexity can be far away, which is the conclusion of Theorem 1.3.

In this section we show that the bound d is never sharp for on-line complexity. In other words, the on-line complexity of any access structure is always strictly less than the maximal degree. We prove this result for graph-based structures, and only indicate how the proof can be modified for arbitrary access structures.

The idea is that during the secret distribution we maintain some tiny fraction of joint information among any pair of the participants. This joint information then can be used to reduce the number of bits the most heavily loaded participant should receive. We shall use a technique extending Stinson's decomposition construction from [26].

A *star k-cover* of G is a collection $\mathcal{S}$ of (not necessarily distinct) stars $\mathcal{S} = \{S_\alpha\}$ such that every edge of G is contained in at least k of the stars. The *weight of the cover* $\mathcal{S}$, denoted as $w(\mathcal{S})$, is the maximal number a vertex of G is included in some star (either as a center or as a leaf):

$$w(S) = \max_{v \in G} |\{S_\alpha \in \mathcal{S} \, : \, v \in V(S_\alpha)\}|.$$

Lemma 6.1 (Stinson, [26]**)** *Suppose $\mathcal{S}$ is a star k-cover of G. Then the complexity of G is at most $w(\mathcal{S})/k$.*

We present the proof here because our construction will be based on it.

Proof Let $\mathbb{F}$ be a large enough finite field. We describe a secret sharing construction in which the secret is a k-tuple of elements of $\mathbb{F}$, and each share is a collection of at most $w(\mathcal{S})$ elements from $\mathbb{F}$. Let V be the k-dimensional vector space over $\mathbb{F}$. Pick the vector $\mathbf{v}_\alpha \in V$ for each $S_\alpha \in \mathcal{S}$ so that any k of these vectors span the whole V. (This can be done if the field $\mathbb{F}$ has at least $|\mathcal{S}|$ non-zero elements.) The set of vectors together with their indices will be public information, and they do not constitute part of the secret. The secret is a uniform random vector $\mathbf{s} \in V$. For each star S_α in the cover the dealer chooses a random element $r_\alpha \in \mathbb{F}$, and tells r_α (with its index) to the leaves of S_α, and she tells $\langle \mathbf{s}, \mathbf{v}_\alpha \rangle - r_\alpha$ to the center of S_α where $\langle \mathbf{s}, \mathbf{v}_\alpha \rangle$ denotes the inner product of these vectors.

Obviously, in this scheme every participant receives at most $w(\mathcal{S})$ field elements. The secret consists of k independent field elements thus the complexity of the system is $w(\mathcal{S})/k$, as was claimed.

It is clear that the vertices of an edge can recover the secret: as the edge is covered by at least k stars, the two endpoints can recover the inner products $\langle \mathbf{s}, \mathbf{v}_\alpha \rangle$ for k distinct α. As these $\mathbf{v}_\alpha$ vectors span the whole space V, from these inner products they can recover $\mathbf{s}$ as well. On the other hand, any unqualified subset of the vertices receives field elements that (after removing repetitions) are independent from each other and from $\mathbf{s}$. $\square$

Let G be a graph with maximal degree $d \geq 2$. The first on-line secret sharing scheme for G we describe has complexity d but assigns smaller shares for most vertices. This is similar to the modified Scheme 4 presented at the end of Section 3. In that version of first-fit all vertices receive shares of size strictly less than the maximum of d times the size of the secret except for the vertices with d or $d-1$ backward edges. In the scheme we present here only vertices with d backward edges receive maximal size shares. Recall that when a vertex v appears we categorize the incident edges as *backward* or *forward* depending on whether the edge is revealed at that time (if it connects v to a vertex that appeared earlier) or will be revealed later.

The simplest way to apply Lemma 6.1 is to consider the collection of stars $\{S_v : v \in V(G)\}$, where the center of S_v is v and its leaves are the neighbors of v. Clearly, every edge of G appears in exactly two of these stars and the weight of this cover is $d+1$, so applying Lemma 6.1 one obtains $\sigma(G) \leq (d+1)/2$.

Our construction can be considered as an on-line implementation of the scheme in the proof of Lemma 6.1. For it to work we construct another double cover of the edges of G with stars and the corresponding shares as we go. We will maintain that each edge appears in two stars and will have at most d stars with center at the same vertex. Before we start we fix a finite field $\mathbb{F}$ (any field with more than dn elements will do, where n is the number of vertices), and let V be a two dimensional vector space over $\mathbb{F}$ and choose dn linearly independent vectors $\mathbf{v}_\alpha \in V$. The secret $\mathbf{s}$ is a uniform random vector from V. As we go we assign "leaves" and "centers" to the

When a vertex v appears we see its backward degree and all of its backward neighbors, but don't necessarily know its forward degree. Let m be the number of backward edges at v and let $m' = \max(1, m)$. We assign m' centers center_v^i for $1 \leq i \leq m'$ and d leaves leaf_v^i for $1 \leq i \leq d$ to v. For all the centers we choose a corresponding new vector $\mathbf{v}_\alpha$. For each backward edge vw we select a distinct center assigned to v and connect it to an unused leaf at w. This determines the value associated to the center selected as the value at the leaf is already decided. We also select a distinct leaf at v for each backward edge vw and connect it to center_w^1. Here, too, this determines the value associated to selected leaf at v. The remaining $d-m$ leaves at v and also the one remaining center in case $m=0$ is assigned to v in anticipation of the forward edges and are not connected to anything at this point. We select the associated values independently and uniformly at random for each remaining leaf or center.

Scheme 6: On-line star packing

vertices with corresponding field elements in such a way that elements assigned to distinct stars are independent from each other and from the secret, all leaves of the same star receive identical elements and the elements corresponding to the center and a leaf of a star together determine $\langle \mathbf{s}, \mathbf{v}_\alpha \rangle$ as their sum for a distinct $\mathbf{v}_\alpha$ for each distinct star. The share of a vertex consists of the values associated to all leaves and centers assigned to this vertex. Clearly, if we maintain these properties, then we obtain an on-line scheme for G. Details are in Scheme 6.

In Figure 2 we illustrate the process for the case when the maximal degree is $d = 3$, and vertices x, y, z and t arrived previously in this order. Shares given to participants appear as blobs on the vertical lines. The assigned centers are solid dots, and the leaves are hollow ones. When t arrived, he had two backward edges going to x and y. Thus t got two centers $\mathbf{c}_1$ and $\mathbf{c}_2$ (that is, two indices from the pool of vectors $\mathbf{v}_\alpha$), and $d = 3$ leaves. Two of the leaves were connected to the first centers at x and at y respectively (and t received the shares accordingly). Then $\mathbf{c}_1$ was connected to a free leaf of y – the corresponding share was the $\langle \mathbf{s}, \mathbf{v}_{\alpha_1} \rangle - r_{y,3}$ difference where $r_{y,3}$ is the random field value assigned to y's third leaf. Similarly, and $\mathbf{c}_2$ was connected to the second free leaf of x, so the corresponding share was $\langle \mathbf{s}, \mathbf{v}_{\alpha_2} \rangle - r_{x,2}$. The third leaf of t was free and had not been assigned to any star, thus t received a fresh random element from the field as a share.

When v arrives his backward degree turns out to be 3. Thus v will be assigned three centers and three leaves. The leaves are connected to the first centers at x, z, and t correspondingly, and

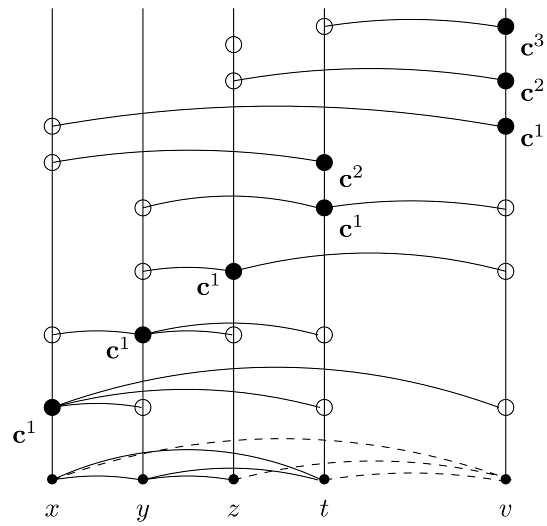

Figure 2: Intermediate stage: next vertex v is connected to x, z, and t

the centers are connected to free leaves at those participants. The shares are generated as above. Observe that each edge of the original graph is covered exactly twice. Thus all qualified subsets can recover at least two inner products $\rangle \mathbf{v}_\alpha, \mathbf{s}\rangle$. As the vector space has dimension two, it means that they can also recover the secret $\mathbf{s}$.

Note that in Scheme 6 the share of participant v consists of $m' + d$ field elements, so its size is $(m' + d)/2$ times the size of the secret. Here $(m' + d)/2 \le d - 1/2$ if $m < d$. But the complexity of the scheme is still d as vertices with d backward edges receive $2d$ field elements.

To apply this scheme one doesn't have to know the structure of G, it is enough to know the size $n = |V(G)|$ and the maximal degree d. The same is true for the more complicated scheme to be described below.

To push the complexity strictly below d we need to decrease the information given to vertices of backward degree d at the expense of adding further information to all other vertices.

Theorem 6.2 *Let G be a graph on n vertices with maximal degree $d \ge 2$. The on-line complexity of G is at most $d - 1/(2dn)$.*

Proof We modify the above construction to achieve the lower complexity. Let k be a large integer to be chosen later. We execute in parallel k independent copies of the secret distributing procedure above. Namely, the secret is k independent uniform vectors $\mathbf{s}^1, \dots, \mathbf{s}^k$ from the two dimensional vector space V and $\mathbf{s}^i$ is determined by the shares of a qualified subset in copy i of the game above. This process multiplies the size of the secret as well as the size of the shares by k so it does not alter the complexity of the scheme. Now we modify this combined scheme as follows.

For each pair $\{v, w\}$ of vertices if neither of them has backward degree d (but regardless whether they form an edge or not) we assign d special leaves with identical random field elements independent of each other and of all other choices. We do this by assigning $d(n-1)$ field elements to each vertex with backward degree less than d out of which values we choose $d - d$ from unused values assigned to each earlier such vertex and we select the rest uniformly at random. We treat these values as unused leaves.

Suppose v is a vertex of backward degree $m < d$. Then we assign kd leaves and km', $m' = \max(m, 1)$ centers to v in each copy and handle them exactly as in scheme 6. In addition we also assign $d(n-1)$ special leaves to v whose corresponding field elements are shared by some other special leaf. Namely, if we have t earlier vertices with backward degree less than d, then we select d unused special leaves from each such vertex and make the same random value correspond to the first dt special leaves at v. For the remaining special leaves we select uniform random values.

In total v is assigned km' centers, kd normal leaves and $d(n-1)$ special leaves for a total of $km' + kd + d(n-1)$ field elements in the share.

Next suppose v is a vertex with backward degree exactly d. In this case we modify just one copy of scheme 6 as follows. In the other $k-1$ copies of the original scheme we assign d centers and d leaves to v each, but in the modified copy we assign d leaves but only $d-1$ centers, one of which we call "special." All leaves and centers are handled as in the original scheme except the special center is connected to two leaves to take care of two backward edges. Let x and y be any two neighbors of v. We select two special leaves, one at x another at y that are not participating in a star yet but which correspond to the same random field element. We connect these special leaves to the special center at v to form a two edge star and this determines the value corresponding to the special center. Note that this is only possible because we have leaves at x and y sharing the corresponding field element as all the leaves of any star should share the same value. As the maximal degree is d, the vertex pair $\{x, y\}$ can occur at most d times in this process, thus there will always be a new pair of special leaves to choose from. We do not assign any special leaves to v so the share of v consists of $2dk - 1$ elements of $\mathbb{F}$.

It is clear that the scheme described is an on-line secret sharing scheme for G. The secret can be written as $2k$ independent field elements. A vertex with less than d backward edges receives at most $d(n-1) + (2d-1)k$ field elements, and a vertex with exactly d backward edges receives $2dk - 1$ field elements. Thus the complexity of the scheme is

$$\frac{2dk-1}{2k} = d - \frac{1}{2k}$$

if $d(n-1) + (2d-1)k \leq 2dk - 1$, which is the case when $k = dn$. This proves the theorem. $\square$

As the complexity of any nontrivial access structure is at least 1, from Theorem 6.2 it follows immediately that the performance ratio is at most $d - 1/(2dn)$ for any graph-based structure with maximal degree d. This was claimed as part (i) of Theorem 1.4.

A generalization of Theorem 6.2 for arbitrary access structure was stated as Theorem 1.5. In the construction we will use a bound on the number of elements in minimal qualified subsets. When Γ is graph based, this bound is 2, but in general it can be any number $r \leq n$. As usual, d denotes the maximal degree of Γ.

Stinson's Lemma 6.1 easily generalizes for hypergraphs as follows.

Lemma 6.3 *Consider a hypergraph Γ and a system $(S_\alpha, v_\alpha)_{\alpha \in A}$, where S_α is a subset of the hyperedges of Γ and v_α is a vertex of Γ. Assume each hyperedge in Γ appears in at least k of the sets S_α and define the weight of a vertex x as*

$$w(x) = \sum_{x = v_\alpha} 1 + \sum_{x \neq v_\alpha} |\{H \in S_\alpha : x \in H\}|,$$

where the summations are for $\alpha \in A$. Then $\sigma(\Gamma) \leq \max_x w(x)/k$. $\square$

Here we consider S_α as generalized stars with center v_α. If all hyperedges of a hypergraph Γ have size at most r, then one can consider the collection $(S_v, v)_{v \in V(G)}$, where S_v is the set of hyperedges containing v together with singleton S_α (with an arbitrary center) so that each hyperedge appears exactly r times. The lemma above applied to this system gives $\sigma(\Gamma) \leq d - (d-1)/r$, where $d = d(\Gamma)$ is the maximal degree. As in the case of graphs, the direct approach to turn this scheme into an on-line scheme increases the complexity to d but only vertices with d backward hyperedges receive maximal size shares.

We only need to lower the load on participants with d backward hyperedges. Let v be such a participant, and A be a minimal qualified set v is in. Then v gets a field element so that the sum of this and other elements preassigned to other participants in A yields the secret value. Now v's load can be lowered if he can receive the same field element for two different minimal qualified subsets A_1 and A_2. Thus we need randomly assigned numbers to $A_1 - \{v\}$ and to $A_2 - \{v\}$ so that their sum be equal. Such a thing can be found if for all disjoint subsets U and V of the participants with $|U| < r$, $|V| < r$ we maintain d such sums, plus d further random values to be used in $(A_1 \cap A_2) - \{v\}$. These random field elements will be assigned (with appropriate labels) to members of U and V.

Let M be the number of the (U, V) pairs a particular participant is in either U or V. An easy calculation shows that $M \le \min(r \cdot n^{2r-3}, 3^{n-1})$. Then each participant, except for those with backward degree d, will receive $d(M+d)$ extra field elements. If we execute k copies of the on-line scheme in parallel, then participants with less than d backward degree receive at most $k \cdot (dn-1) + d \cdot (M+d)$ field elements; those with exactly d backward degree receive $k \cdot (dn) - 1$ field elements. The secret in this case will be kn field elements, thus the complexity of the scheme is $d - 1/(kn)$ if

$$k \cdot (dn-1) + d \cdot (M+d) \le k \cdot (dn) - 1$$
$$d \cdot (M+d) + 1 \le k.$$

Choosing the smallest possible value for k gives the complexity in Theorem 1.5.

7 Conclusion

In this paper we defined the notion of on-line secret sharing scheme, as an extension of the classical, off-line schemes. Given a set P of participants, the dealer meets the participants one by one, and learns only the partial structure generated by participants who show up so far. In spite of this, final and irrevocable shares should be assigned to each participant. The question we investigated was how much worse does an on-line scheme perform compared to the best off-line one?

We defined a universal on-line scheme which we called *first-fit* in strong resemblance to the first-fit on-line graph coloring algorithm. Its complexity is the maximal degree of the realized access structure, thus its efficiency is comparable to the most efficient known general off-line schemes.

We looked at several graph-based access structures, and found that quite often the on-line and off-line complexities were not only close to each other, but actually they were equal. We could separate these complexities by showing that for paths on at most 5 vertices, and cycles on at most 6 vertices these complexities are, in fact, equal. For other paths and cycles the on-line complexity is strictly greater than the off-line (Theorem 1.2. Nevertheless the ratio between the complexities is always less than 4/3.

For trees this *performance ratio* can be much larger. In fact, there is a tree T_n on n vertices where this ratio is at least $\sqrt{n}/4$, as proved in Theorem 1.3. If the *maximal degree* of the access structure is constant (say at most 10), then the situation is much better. By Theorem 1.4 in this case on-line schemes are at most 10 times more expensive than off-line ones (but very probably much less). This result follows from the performance of the first-fit scheme.

In the last section of this paper we showed that the first-fit scheme never optimal. For general access structure we could improve it by an exponentially small amount only. We pose it as an open question how much this bound can be lowered.

On-line schemes can be generated from off-line ones when the access structure is *fully symmetrical*, see Claim 4.1. As we have remarked, threshold structures and complete multipartite graphs are induced subgraphs of fully symmetrical structures, thus for them the on-line and off-line complexities are the same. We also know that C_5, the cycle on 5 points is fully symmetrical as well as the Petersen graph on Figure 1. An independent research problem is to determine all fully symmetrical graphs.

Finally, we would be very much interested in computing the exact on-line complexity for any other graph or structure.

Acknowledgment

The first author would like to thank Carles Padró and Ronald Cramer for their hospitality, support, and the invitation to the RISC@CWI Conference on Combinatorics in Secret Sharing, where the half-cooked ideas of on-line secret sharing were first presented [12].

References

[1] G. R. Blakley: Safeguarding cryptographic keys, *AFIPS Conference Proceedings* vol 48 (1979) pp 313–317

[2] A. Beimel, N. Livne, C. Padró: Matroids can be far from ideal secret sharing, *Proceedings of TCC'08*, *LNCS*, Vol 4948 (2008), pp. 194–212

[3] C. Blundo, A. Cresti, A. De Santis, U. Vaccaro: Fully dynamic secret sharing schemes, *in Advances in Cryptology – Crypto'93*, D. R. stinson, ed., vol 773 of Lecture Notes in Computer Science, Springer-Verlag, 1994, pp. 110–125

[4] C. Blundo, A. De Santis, L. . Gargano, U. Vaccaro: On the information rate of secret sharing schemes, *Theoret. Comp. Sci.*, vol 154 (1996), pp. 283–306

[5] C. Blundo, A. De Santis, R. De Simone, U. Vaccaro: Tight Bounds on the Information Rate of Secret Sharing Schemes, *Des. Codes Cryptography*, vol 11(2) (1997), pp. 107–110

[6] C. Blundo, A. De Santis, D. R. Stinson, U. Vaccaro: Graph decomposition and secret sharing schemes, *J. Cryptology*, vol 8(2) (1995), pp. 39–64

[7] B. Bollobas: *Combinatorics - Set Systems, Hypergraphs, Families Of Vectors And Probabilistic Combinatorics*, Cambridge University Press, 1986

[8] C. Cachin: On-line Secret Sharing, in *Proc. of the 5th IMA Conf. on Cryptography and Coding*, 1995, pp. 190-198

[9] R. M. Capocelli, A. De Santis, L. Gargano, U. Vaccaro: On the size of shares of secret sharing schemes, *J. Cryptology*, vol 6(3) (1993), pp. 157–168

[10] L. Csirmaz: Secret sharing schemes on graphs, *Studia Sci. Math. Hungar.*, vol 44(2007) pp. 297–306 – available as IACR preprint `http://eprint.iacr.org/2005/059`

[11] L. Csirmaz: Secret sharing on the d-dimensional cube available as IACR preprint `http://eprint.iacr.org/2005/177`

[12] L. Csirmaz: Online secret sharing, Presentation at the conference *Combinatorics in secret sharing*, March 2010, Amsterdam available as `http://eprints.renyi.hu/39`

[13] L. Csirmaz, G. Tardos: Secret sharing on trees: problem solved – available as IACR preprint `http://eprint.iacr.org/2009/071`

[14] I. Csiszár and J. Körner: *Information Theory. Coding Theorems for Discrete Memoryless Systems*, Academic Press, New York, 1981.

[15] M. van Dijk: On the information rate of perfect secret sharing schemes, *Des. Codes Cryptography*, vol 6 (1995), pp. 143–169

[16] P. Erdős, L. Pyber: Covering a graph by complete bipartite graphs, *Discrete mathematics* Vol 170 (1997) pp. 249–251.

[17] A. Gyárfás, J. Lehel: First fit and on-line chromatic number of families of graphs, *Ars Combinatorica* 29C (1990) pp. 168–176

[18] M. Ito, A. Saito, T. Nishizeki: Secret sharing scheme realizing general access structure, *Electronics and Communications in Japan*, vol 72(9) (1989) pp. 56–64

[19] L. Lovász, M. Saks, W. T. Trotter: An on-line graph coloring algorithm with sublinear performance ratio *Discrete Mathematics*, vol 75(1–3) (1989) pp. 319–325

[20] J. Martí-Farré, C. Padró: On secret sharing schemes, matroids and polymatroids, *Fourth IACR Theory of Cryptography Conference TCC 2007*, Lecture Notes in Computer Science 4392 (2007), pp. 273-290.

[21] J. Martí-Farré, C. Padró: Ideal secret sharing schemes whose minimal qualified subsets have at most three participants, *Des. Codes Cryptography* vol 52(1) (2009), pp. 1–14

[22] F. Matuš: Two constructions on limits of entropy functions, *IEEE Trans. on Inf. Theory*, vol 53(1) (2007) pp. 320–330

[23] U. Maurer: Secure multi-party computation made simple, *Discrete Appl. Math.* vol 154 (2006), pp. 370–381

[24] A. Miller: Online graph colouring, Canadian Undergraduate Mathematics Conference (2004) `http://www.cumc.math.ca/2005/papers/miller.pdf`

[25] A. Shamir: How to share a secret, *Commun. of the ACM*, vol 22 (1979) pp. 612–613

[26] D. R. Stinson: Decomposition Constructions for Secret-Sharing Schemes *IEEE Transactions on Information Theory*, vol 40(1) (1994) pp. 118–125

[27] D. R. Stinson, R. Wei: Bibliography on Secret Sharing Schemes, available at `http://www.cacr.math.uwaterloo.ca/~dstinson/ssbib.html`

Appendix

We give a direct proof of the fact that the on-line complexity of the path P_6 is at least $7/4$. We are using the technique discussed in Section 5. Let us denote the vertices along the path P_6 by a, b, x, a', b', and y, and let the subgraph H be induced by the vertices a, b, a' and b'. H is a matching consisting of two edges and its automorphism group has order 8. Let f be a H-symmetric entropy function for P_6. By Theorem 5.2(ii) it is enough to show that f takes a value at least $7/4$ on some singleton.

Our starting point is the inequality

$$f(aa'b') - f(a) \geq 3. \tag{2}$$

This is well-known generalization of the inequality from [4], and follows from the fact that a is not connected to any vertex of the spanned path $xa'b'y$.

Strict submodularity and strict monotonicity yields

$$\begin{gathered} f(bx) + f(xa') \geq f(ba'x) + f(x) + 1 \\ f(ba'x) \geq f(ba') + 1. \end{gathered}$$

Using these together with $f(b) + f(x) \geq f(bx)$, $f(x) + f(a') \geq f(xa')$ we get

$$f(b) + f(a') + f(x) \geq f(ba') + 2. \tag{3}$$

As f is H-symmetric, $f(aa') = f(ab') = f(ba')$, and $f(a) = f(b) = f(a') = f(b')$, furthermore, by submodularity and by (2),

$$f(aa') + f(ab') \geq f(a) + f(aa'b') \geq f(a) + (f(a) + 3).$$

Plugging this into (3), we get

$$f(b) + f(a') + f(x) \geq 2 + \frac{2f(a) + 3}{2},$$

from where $f(a) + f(x) \geq 7/2$. Therefore either $f(a)$ or $f(x)$ is at least $7/4$, as was required.