Secret sharing on trees: problem solved

László Csirmaz[*] Gábor Tardos[†]

Abstract

We determine the worst case information rate for all secret sharing schemes based on trees. It is the inverse of $2 - 1/c$, where c is the size of the maximal core in the tree. A *core* is a connected subset of the vertices so that every vertex in the core has a neighbor outside the core. The upper bound comes from an application of the entropy method [2, 3], while the lower bound is achieved by a construction using Stinson's decomposition theorem [7].

It is shown that $2 - 1/c$ is also the *fractional cover number* of the tree where the edges of the tree are covered by stars, and the vertex cover should be minimized, cf [5]. We also give an $O(n^2)$ algorithm which finds an optimal cover on any tree, and thus a perfect secret sharing scheme with optimal rate.

Keywords. Secret sharing scheme; information rate; graph; fractional packing and cover; entropy method.

1 Introduction

Secret sharing schemes has been investigated in several papers, for an extended bibliography see [8]. Such a scheme is a randomized distribution of a secret among participants so that the secret can be recovered by any *qualified* subset of participants, while the total information an *unqualified* subset receives is (statistically) independent of the secret itself. The scheme is *based on the graph* G if the participants are the vertices, and a subset of vertices is qualified just in case it contains an edge. Thus the minimal qualified subsets are the edges, the maximal unqualified subsets are the independent vertex sets (i.e. subgraphs which span the empty graph).

The *worst case efficiency* of a scheme is measured by the amount of information the most heavily loaded participant must remember for each bit in the secret. For a graph G the *information ratio of* G, denoted as $R(G)$, is the infimum of the efficiency of all schemes based on G. The *information rate*, usually denoted as $\rho(G)$, is just the inverse of this value.

The information rate has been investigated for graphs with at most six vertices in [4], where the authors conclude:

> *An open research problem is to study secret sharing schemes for the 18 connected graphs on six vertices for which the optimal worst-case information rate is yet unknown.*

While we did not advance on these sporadic problems, we determine the optimal rate for an important infinite family of graphs, namely that of the trees. To state our result we need the notion of the *core* of a graph G.

Definition 1.1 Let X be a connected subset of vertices in G. X is a *core* if for each $v \in X$ there is a neighbor of v not in X such that from X it its connected to v only, and these neighbors form an independent set.

When G is a tree we only need to require that all $v \in X$ have a neighbor outside X. These neighbors will automatically satisfy the remaining conditions of the definition.

[*] Central European University

[†] Rényi Institute of Mathematics

Theorem 1.2 *Let G be a (connected) tree, and let $c = c(G)$ be the size of the maximal core in G. The* information ratio *of G is $R(G) = 2 - 1/c$, i.e. the information rate is $\rho(G) = c/(2c-1)$. Let n be the number of vertices in G. The number $c(G)$ can be determined in G in $O(n^2)$ steps. Furthermore a perfect secret sharing scheme based on G can be constructed in $O(n^2)$ steps as well which has optimal worst case ratio.*

In section 2 we show that $2 - 1/c \leq R(G)$ using the entropy method, see [2, 3]. The method is based on the observation that the shares and the secret are random variables, and the ratio can be expressed as a function of the Shannon entropy of these variables.

The direction $R(G) \leq 2 - 1/c$ is dealt with in section 3 and uses two main tools. One is Stinson's decomposition theorem [7] which lets us compose simple schemes into a composite one. The other tool is a new fractional packing result.

A star is *packed* into G if the star's center goes to a node v of G, and the rays end at different neighbors of v. Of course, not necessarily all neighbors of v are part of the star. Different packed stars may share the same edge. In a *fractional packing* the packed stars have positive real weights. The *fractional vertex cover* is the least upper bound on the weight of the vertices under the condition that all edges have weight at least 1. For more about fractional packing and covering see [6].

Theorem 1.3 *Let G be a connected tree. The fractional vertex cover number of G, when the edges of G are covered by stars, is $2 - 1/c$ where $c = c(G)$. Moreover an optimal packing can be found in time $O(n^2)$.*

From the optimal fractional cover a perfect secret sharing scheme with the same ratio can be constructed using the aforementioned Stinson's decomposition method [7]. The complexity of this construction is linear in the size of the cover.

2 Lower bound

In this section we show that the information ratio of an arbitrary graph is at least $2 - 1/c$ where c is the size of any core in G. This proves the $2 - 1/c \leq R(G)$ part of Theorem 1.2. We remark that while this bound is sharp for trees, in general it gives only a proper lower bound. The graph depicted on figure 1 has ratio $3/2 = 2 - 1/2$, see [2], while it has only one-element core.

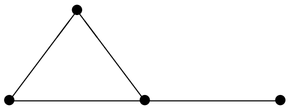

Figure 1: Graph with ratio 3/2 and maximal core size 1

The proof uses the *entropy method*, see, e.g. [2, 3]. For the sake of the reader we sketch the idea behind it. As shares are random variables, their size is measured by the Shannon entropy. There are well-known inequalities this entropy satisfies, and they are used to get a lower bound on the entropy – and thus on the size – of the share of some participant. Instead using the entropy directly, it is customary to scale it down so that the secret has value one, and then the method simply boils down to exhibit a participant who is assigned a value $\geq c$.

Thus we are given a function f which assigns real numbers to the subsets of participants – the relative entropy of that subset. Values assigned to the vertices correspond to the relative size of the assigned share. f satisfies the following inequalities derived from the corresponding properties of the entropy function:

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

(b) $f(A) \geq f(B)$ when $A \supseteq B$ (monotonicity)

(c) $f(A) + f(B) \geq f(AB) + f(A \cap B)$ (submodularity)

(d) $f(A) \geq f(B) + 1$ when $A \supseteq B$, A is qualified while B is not (strict monotonicity)

(e) $f(A) + f(B) \geq f(AB) + f(A \cap B) + 1$ when A, B are qualified while $A \cap B$ is not (strict submodularity)

Here, as usual, we write AB instead $A \cup B$. Then we have to prove that given any f satisfying (a)–(e), there is always a vertex v with $f(v) \geq c$.

Theorem 2.1 *Let $X \subseteq G$ be a core in the graph G, and let f be a real valued function defined on the subsets of the vertices of G satisfying properties* (a)–(e). *Then*

$$\sum_{v \in X} f(v) \geq 2|X| - 1,$$

i.e. for some vertex v in X we have $f(v) \geq 2 - 1/|X|$.

Proof First we show that if X is connected, then

$$\sum_{v \in X} f(v) \geq f(X) + |X| - 2. \tag{1}$$

If the elements of X are $v_1, \ldots, v_k$, then the assumptions imply the existence of vertices $w_1, \ldots, w_k$ such that $\{w_1, \ldots, w_k\}$ is independent, and w_i is connected to v_i only in X. Thus

$$f(X) \geq |X| + 1$$

by the lemma in [1, 3] dubbed as the “independent sequence lemma.” Thus we need to prove only (1). Observe that the elements of X can be enumerated so that all initial segments are also connected. Thus we can use induction on the number of the vertices in X.

If X has at most two elements, then (1) simplifies to

$$f(a) + f(b) \geq f(ab)$$

which holds by the submodularity property of f.

Now suppose X is connected, has at least two vertices, a is a vertex not in X connected to $b \in X$ (maybe among others). Let $Y = X - \{b\}$, then none of ab and $X = Yb$ are independent, thus property (e) gives

$$f(ab) + f(Yb) \geq f(Yab) + f(b) + 1.$$

Also, $f(a) + f(b) \geq f(ab)$, which yields

$$f(X) + f(a) \geq f(Xa) + 1.$$

This with the induction hypothesis proves (1). □

3 Upper bound

Theorem 3.1 *Let G be a tree, and suppose each core of G has size at most c. Then we can pack stars into G so that* (i) *all edges are covered exactly c times, and* (ii) *all vertices are covered at most $2c - 1$ times.*

Proof We start with assigning positive integers – weights – to each vertex. The weight of a subset is the sum of the weights of the vertices in that subset. We consider a weighting in which each vertex has the largest possible weight with the restriction that every core has weight $\leq c$. Such a weighting exists: as each vertex is an element of a core, its weight cannot be larger than c. On the other hand if all vertices have weight 1, then by the assumption all cores have weight $\leq c$. As we have finitely many possibilities, we can pick a maximal one.

Replace each edge of G by c *directed lines.* A covering star will have center at some vertex and its rays will be among the outgoing lines Considering all of these stars, each edge of G will be covered by exactly c of them. The number of stars with center at v is the maximal number of *outgoing* lines along the edges starting at v; furthermore v is at the end of a ray of a covering star for each *incoming* line along the edges at v. Thus the cover number for v is the total number of all incoming lines, plus the maximal number of outgoing lines along an edge. As there are exactly c lines along each edge, this cover number is c plus the total number of incoming lines *except the smallest number* of incoming lines along any edge.

Thus it suffices to show that we can direct the lines so that this latter sum is at most $c-1$.

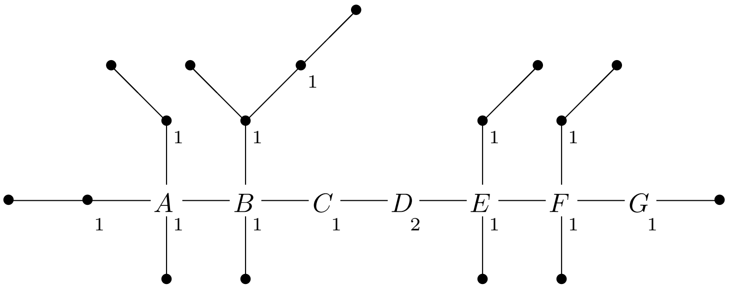

Figure 2: A tree with weights and maximal core size $c = 7$

Now let v_1v_2 be an edge. If either v_1 of v_2 is a leaf, then direct all lines between them toward the leaf. (If both v_1 and v_2 are leaves, then G is a single edge, and there is nothing to prove.)

If neither v_1 nor v_2 is a leaf, then removing the edge v_1v_2 splits the tree into two disjoint subtrees, G_1 and G_2 where G_i contains v_i. Let C_i be a core of maximal weight in G_i which contains v_i; let this maximal weight be c_i. As $C_1 \cup C_2$ is a core of weight $c_1 + c_2$, consequently $c_1 + c_2 \le c$. Among the c lines between v_1 and v_2 direct c_1 from v_1 towards v_2, and c_2 from v_2 towards v_1. If $c_1 + c_2 < c$ then direct the rest of the lines arbitrarily.

The tree depicted on figure 2 has maximal core size $c = 7$, and the numbers show a maximal weighting. Each edge is replaced by seven directed ones, and the numbers the above procedure gives are

$$\begin{array}{cccccc}
A \to B & A \leftarrow B & B \to C & B \leftarrow C & C \to D & C \leftarrow D \\
3 & 4 & 6 & 1 & 1 & 2 \\
D \to E & D \leftarrow E & E \to F & E \leftarrow F & F \to G & F \leftarrow G \\
2 & 5 & 4 & 3 & 6 & 1
\end{array}$$

For example, when the edge CD is deleted, D is contained in a core which consists of itself only, thus it has weight 2. This gives the value 2 to $C \leftarrow D$.

We claim that construction satisfies the above requirement. Indeed, if w is a leaf, then it has exactly c incoming lines and no outgoing line. Otherwise let w be a non-leaf vertex, and C be a core of maximal weight containing w. By the maximality of weighting, C has weight c (otherwise the weight of w could be increased). Let $v_1, v_2, \ldots, v_s$ be all neighbors of w in C, and let C_i be the connected part of C containing v_i and not containing w. Then

$$c = \text{weight}(C) = \text{weight}(w) + \text{weight}(C_1) + \ldots + \text{weight}(C_s).$$

Both C_i and $C - C_i$ are cores. Thus, by the construction, among the c lines between v_i and w at least $\text{weight}(C_i)$ goes from v_i to w, and at least $\text{weight}(C - C_i) = c - \text{weight}(C_i)$ goes from w to v_i. As their sum is exactly c, these are the exact values. Thus the total number of *incoming* lines into w from vertices in C is

$$\text{weight}(C_1) + \ldots + \text{weight}(C_s) = k - \text{weight}(w) \le c - 1.$$

We have two cases: either w has a leaf neighbor, or it has none. In the first case all non-leaf neighbors of w are in C, as C was chosen to be maximal. There are no incoming lines from leaves, thus in this case we are done.

In the other case no neighbor of w is a leaf. Then all but one of them must be in C, let v^* be the exceptional neighbor. Now $C - C_i$ is a core in the graph $G - \{wv^*\}$ which contains w but does not contain v^*, thus at least weight$(C - C_i) = k -$ weight(C_i) lines are directed from w toward v^*. It means that that the number of *incoming* lines from v^* cannot be more than weight(C_i), i.e. the number of incoming lines from v_i. It shows that the smallest number of incoming lines comes from v^*, and the total number of incoming lines from the other neighbors is at most $c - 1$, which was to be shown. □

4 Main results

We turn to the proof of theorems announced in section 1. Let G be a connected tree. The size of the maximal core in G can be found as follows. For each node v in G let $C(v)$ be the size of the maximal core containing v. We will determine all $C(v)$, and then take the maximum of these values. Let v_1, …, v_s be all neighbors of v. Delete v and all edges vv_i from the graph, and determine for each v_i the size of the maximal core containing v_i *in the component of* v_i. If v has a leaf neighbor, then $C(v)$ is one more than the sum of the values $C'(v_i)$ for non-leaf neighbors of v. If v has no leaf neighbor, then from the values $C'(v_i)$ discard the minimal one, add up all the others and add one to get $C(v)$.

For each node this algorithm uses $O(n)$ steps, thus determining $C(G)$ requires $O(n^2)$ steps, proving the statement in Theorem 1.2.

The construction in the proof of Theorem 3.1 needs a maximal weighting of the vertices of G. We claim that this weighting can also be found in $O(n^2)$ steps. Indeed, enumerate all vertices of G as v_1, …, v_n, and assign weight 1 to each vertex initially. At the i-th step only the weight of v_i may increase. Following the procedure outlined before and using the present vertex weights, determine the maximal weight core containing v_i. If this weight is c then do nothing; if the weight is smaller than c, increase the weight of v_i so that it becomes c. By this step no core weight can exceed c, and the weight of v_i will be maximal. Handling each vertex requires $O(n)$ steps, thus $O(n^2)$ steps are enough to find a maximal weighting.

From a maximal weighting covering stars can be generated by going through each edge and finding a maximal weight core on both sides of the edge. As G is a tree, it has $n - 1$ edges, and for each edge the values can be found in $O(n)$ steps, which totals to $O(n^2)$ steps again.

In the packing guaranteed by Theorem 3.1 we weight each star by $1/c$. This fractional packing will cover each edge, and the weight of each node is at most $(2c - 1)/c$ which proves the second part of Theorem 1.3.

On the other hand, finding the fraction cover number r of a graph G when the edges are covered by stars is an LP problem, thus it has an optimal solution where all weights are rational numbers. Multiplying everything by the lcm of the denominators of these rationals, we see that, for some integer p, stars can be packed into G so that each edge is covered at least p times, and every vertex is covered at most $r \cdot p$ times. As each star admits a perfect secret sharing scheme with rate (and ratio) 1, Using Stinson's decomposition method from [7], we can create a perfect secret sharing scheme on G which hides p bits of secret, and assigns at most $r \cdot p$ bits of share to each node. In other words, there is a perfect secret sharing scheme on G with ratio $r \cdot p/p = r$.

Theorem 2.1 shows that r is at least $2 - 1/c$ when G has a core of size c, thus the cover number for a tree is at least $2 - 1/c$ with $c = c(G)$. This proves Theorem 1.3, and the construction yields Theorem 1.2 as well.

References

[1] C. Blundo, A. G. Gaggia, D. R. Stinson: On the Dealer's Randomness Required in Secret Sharing Schemes *Designs, Codes and Cryptography*, Vol 11(3), pp.235–260 (1997)

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

[3] Secret sharing schemes on graphs, *Studia Mathematica*, vol 44(3), pp 297–306, 2007

[4] M. van Dijk, T. Kevenaar, G. Schrijen, P. Tuyls: Improved constructions of secret sharing schemes by applying (λ,ω)-decompositions, *Inf. Process. Lett.* vol 99(4), 2006, pp.154–157

[5] Serge A. Plotkin, David B. Shmoys, Eva Tardos: Fast Approximation Algorithms for Fractional Packing and Covering Problems *Math. Oper. Res.,* Vol 20, pp 257–301, 1995

[6] Edward R. Scheinerman, Daniel H. Ullman: *Fractional Graph Theory: A Rational Approach to the Theory of Graphs* Wiley-Interscience, (1997)

[7] D. R. Stinson: Decomposition constructions for secret sharing schemes, *IEEE Trans. Inform. Theory* IT-40 (1994) pp 118–125.

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