Improving efficiency of fractional secret sharing schemes

Tom Suad, Advisor : Anat Paskin-Cherniavsky

Abstract

Secret sharing schemes allow to share a secret among set of parties so that any set $A \in \mathcal{A}$ reconstructs the secret and any other set learns nothing about the secret. Secret sharing is very useful. Fractional secret sharing schemes allow for more than 2 types of sets, with intermediate sets learning partial info.

Ishai et al. (STACS 13')[1] have shown that all fractional access structures can be implemented, however their general construction is inefficient. We optimise their approach for devising fractional secret sharing, obtaining improved share complexity in many cases.

Introduction

Secret sharing schemes are ideal for storing information that is highly sensitive and highly important. Secret sharing refers to methods for distributing a secret amongst a group of participants, each of whom is allocated a share of the secret. The secret can be reconstructed only when a sufficient set T of shares are combined together when certain set of participants shows up, while other sets learn nothing about the secret. A *fractional secret sharing* scheme realizes an access structure $f : 2^{[n]} \rightarrow \{0, m-1\}$ by guaranteeing that for each set T, the secret given their shares is uniformly distributed over a set of size $f(T)+1$ (assuming the dealer picks the secret unifonly out of the secret domain $[m]$). In particular, for each sufficient set of shares the secret is uniformly distributed over a set of 1 secret (the right one!), and for each insufficient set of shares the secret distributed over set of $m-1$ potential secrets. In general secret sharing schemes the share size is at least as the secret size.

Fractional secret sharing generalizes traditional secret sharing by allowing a fine-grained control over the amount of uncertainty about the secret. Fractional secret sharing scheme realizes a fractional access structure $f : 2^{[n]} \rightarrow \{0, ..., m-1\}$ by guaranteeing that for each set $T \subseteq [n]$ of parties, the secret is uniformly distributed over a set of $f(T)+1$ potential secrets. There are cases, in which we want for certain groups to partially learn the secret, in these cases we will use fractional secret sharing schemes. Additionally, there are cases, in which we do not mind that certain groups will partially learn the secret, so in some of these cases we can also use fractional secret sharing schemes, and use the more efficient share complexity. The motivation here comes from ramp schemes from the literature. Hao and Ronald [HR] show that with ramp schemes a share size size can be less than the secret size (unlike standard secret sharing schemes).

In [I+13] that introduced the concept, Ishai et al put forward a construction for symmetric access structures f ($f(T)$ depends only on $|T|$), where $f(T)$ depends only on the size of T, with the quite efficient share size of $n \cdot \lceil \log(max\{n,m\}) \rceil$ However, for other f, their secret complexity is $\Omega(N)$,such that $N = max_{i \in [m]} |\{A | f(A)+1 \leq i\}|$.

For many families of access structures $N = 2^{\Omega(n)}$. [I+13] use a concrete secret sharing scheme for each $f^{-1}(k)$, which is very inefficient.

In the proposed research we want a better understanding about the share complexity of fractional secret sharing for non-symmetric functions. For standard secret sharing scheme, more efficient constructions than $\Omega(N)$ are often known. We improve [I+13] construction, to get better share complexity for certain f's. [I+13] reduces fractional secret sharing scheme for f to making parallel application of the scheme to different m standard secret sharing scheme, one for each k in $[m]$, applied to access structure $\mathcal{A}_{\rangle} = \{A | f(A)+1 \leq k\}$. We replace the scheme used for each k with the best known one. Better schemes than currently used are known for many cases. [I+13] already do it for symmetric access structures.

Related work

[I+13]

They show that every monotone fractional access structure $f : 2^{[n]} \rightarrow \{0, ..., m-1\}$ can be realized by showing a general reduction of fractional secret sharing to lossy chain.

Lossy chain is a Markov chain $(X_0, ..., X_n)$, where X_0 is a random secret, and each step loses additional information about the secret. This loss is by a loss function $g : [n] \rightarrow [m]$, such that for each $1 \leq i \leq n$ and x_i in the support of X_i, the distribution of X_0 conditioned on $X_i = x_i$ is uniform over a set of size $g(i)$. In addition, the algorithm for generating the lossy chain iteratively generate subset S_i of size $g(i)$ for every $i \in [n]$ in decreasing order, where each subset S_i is a random cyclic interval in S_{i+1}. The cyclic interval also realizing loss function g.

First, they define what is fractional access structure - $P = \{p_1, ..., p_n\}$ is a finite set of parties and m is an integer. Function $f : 2^P \rightarrow \{0, ..., m-1\}$ is monotone if $B \subseteq C$ implies that $f(B) \geq f(C)$. A fractional access structure is a monotone function $f : 2^P \rightarrow \{0, ..., m-1\}$, with $f(\emptyset) = m-1$, and f is symmetric if $f(B)$ depends only on $|B|$.

After this, they define what is fractional secret sharing scheme, $f : 2^P \rightarrow \{0, ..., m-1\}$ is a fractional access structure and S is a finite secret domain and D is a randomized algorithm which output a uniformly random $s \in S$ together with a $n - tuple$ of shares $(S_1, ..., S_n)$. D is a fractional secret sharing scheme realizing f with secret domain S if there exists a positive integer k such that the following hold : For every $Q \subseteq P$ and any possible share vector s_Q of parties in Q, the distribution of s conditioned on the event that parties in Q receive the shares s_Q is uniform over a subset of S of size $f(Q) \cdot k + 1$. If this holds with $k = 1$, then they say that D strictly realizes f. Their notion of realizing a fractional access structure views the structure as only specifying a kind of ratio between the amount of uncertainty of different sets, without specifying the absolute amount of uncertainty or the size of secret domain.

In the first part they show that for any fractional access structure, there exists a fractional secret sharing scheme which strictly realizes f. First, they assume that $f(P) = 0$ and $f(\emptyset) = m-1$, and thay use $S = [m]$ as the secret domain. They define different values in increasing order - $\alpha_0, ..., \alpha_l$ in the range of f, and by their assumption they get that $\alpha_0 = 0$ and $\alpha_l = m-1$. They also define a loss function $g : [l] \rightarrow [m]$ such that $g(i) = \alpha_i + 1$, and a lossy chain $\overline{X}$ realizing g. They use the share generation algorithm D that first sample values $(X_0, ..., X_l)$ from $\overline{X}$ and define $s = x_0$, and second, for every set of parties $Q \subseteq P$ define $f(Q) = \alpha_j + 1$. By using traditional $|Q| - out - of - |Q|$ secret sharing scheme, the algorithm share x_j and give every party in Q its share accordingly.

Then, they show that D is a fractional secret sharing scheme that strictly realizing f. They set $Q \subseteq P$ be a subset of parties, and by the properties of the underlying $|Q| - out - of - |Q|$ scheme, the information available to the parties in Q is equivalent to learning all values x_j such that $f(Q') = \alpha_j + 1$ for some $Q' \subseteq Q$. By the monotonicity of f this means that the parties in Q learn x_i, where i is the index such that $f(Q) = \alpha_j + 1$, and possibly additional values x_j for $j > i$. And by the Markov property of a lossy chain, the distribution of the secret s conditioned on the above value x_i and x_j is uniform over a set of size $g(i) = \alpha_i + 1 = f(Q)$.

We noticed that for every subset of parties $Q \subseteq P$, the parties can reconstruct all the values out of $x_0, ..., x_n$ that were shared in a threshold requiring $|Q|$ or less parties, and we noticed that it is good, because in the definition of fractional access structure we can see that f is monotone and if $B \subseteq C$ it implies that $f(B) \geq f(C)$, and $f(B)$ depends only on $|B|$.

In the second part, they show that for symmetric fractional access structure $f : 2^P \rightarrow \{0, ..., m-1\}$ with $f(\emptyset) = m-1$, there exists a fractional secret sharing scheme D which strictly realizes f with secret domain $[m]$, where the bit-length of each share is at most $n \cdot \lceil \log(max\{n, m\}) \rceil$. As before, they use $\alpha_1, ..., \alpha_l$ to be all the different values in the range of f in increasing order and define $g : [l] \rightarrow [m]$ such that $g(i) = \alpha_i + 1$. After that, they define fractional secret sharing scheme D, at first D generate values $\overline{x} = (x_0, ..., x_l)$ for the cyclic intervals lossy chain realizing g, and set $s = x_0$ in addition they define $a_1, ..., a_{l-1}$ as the starting values of the cyclic intervals defining $\overline{x}$. Secondly, for every $i \in [n]$, α_j is the value such that for any subset of parties $Q \subseteq P$ of size i - $f(Q) = \alpha_j + 1$. They use Shamir's $i - out - of - n$ threshold secret sharing to create share of a_j and the algorithm give one share to each of the parties in P. In this way, each party received n different shares, one from each invocation of the threshold secret sharing algorithm done by D. The secrets shared are $a_1, ..., a_l$ and all of them are values picked from at most m values. When they used Shamir's threshold secret sharing scheme, each of the values is shared with shares of size $\lceil \log(max\{n, m\}) \rceil$, and for each party, this amounts to the share size of at most $n \cdot \lceil \log(max\{n, m\}) \rceil$.

Best known existing secret sharing schemes

Most secret sharing schemes from the literature, are linear seret sharing shcemes. In linear sharing algorithm we have specific structure, in this structure we defined a finite field F_p, the secret s is variable in F_p, and there are random element $r_1, ..., r_t$ that they are also variables in F_p. First we set t that will be the number of random elements, and second we choose the random elements randomly and independently. Every party p_i get some fixed linear function $\ell(s, r_1, ..., r_h)$, those functions are the shares of p_i. The secret sharing scheme defined by matrix, every row in the matrix is linear function and every row belongs to a party accordingly. The secret can be learned if and only if the linear solution of the vector which have been adjusted to the parties, is vector such that there is 1 at the beginning where the secret s was originally, and there are t zeros where the random elements was originally, so the linear solution is $(1, \overbrace{0, ..., 0}^{t})$.

In particular, we notice that [BL88] (Benaloh and Leichter)[3], is also linear secret sharing scheme, only it is private case in which F_2.

[I+13] uses simple DNF BL for each f, DNF formula can be bigger than the most efficient monotone formula. Therefore, monotone formula BL, such as [BL88] describe (as explained in our improvement), is more efficient than DNF BL, in fact this is the most efficient BL. Additionally, Karchmer and Wigderson [KW93] introduced and studied the complexity theoretic model of Monotone Span Programs (MSP) which is equivalent to linear secret sharing schemes. MSP can be more efficient than monotone formula BL.

$ODDFACTOR_n$[4] is related to disjointness functions, it is an example for functions with efficient linear scheme but exponencial monotone formula. Using this function L'aszl'o et al. prove lower bound for MSP, which is - $n^{\Omega(\frac{log(n)}{log(log(n))})}$, this shows that there exists scheme which are more efficient then BL.

Notable special cases

One useful access structure is (k, n)-threshold. Here, already Shamir 79' which introduced the concept of secret sharing, put forward an essentially optimal scheme with share complexity of $n \log(n)$ for 1-bit secrets (and 1 bit share per party per secret bit for long enough secrets). Recently, a CDS- based scheme [5] improved over the best known for k-regular access structures, where all minterms are of the same size k. They provided a scheme of share complexity $2^{\tilde{O}(\sqrt{n})}$. We refer to this scheme as AA.

Our improvement

Our main contribution, is improving the general share complexity for fractional secret sharing, as summrized by comparing the best known comlpexity (Theorem 1) vs. the share complexity of our scheme.

Theorem 1 [I+13]: Let $f : 2^{[n]} \rightarrow \{0, ..., m-1\}$ denote a fractional access structure, and for each $i \in [m]$, $F_i = \{A[n] | f(A) + 1 \leq i\}$. Then there exists a fractional secret sharing scheme for f with share complexity at most

$$m \cdot log(m) \cdot max_{i \in [m-1]}(\text{smallest monotone DNF formula for } F_i)$$

.

Theorem 2: Let $f : 2^{[n]} \rightarrow \{0, ..., m-1\}$ denote a fractional access structure, and for each $i \in [m]$, $F_i = \{A[n] | f(A)+1 \leq i\}$. Then there exists a fractional secret sharing scheme for f with share complexity at most

$$m \cdot log(m) \cdot max_{i \in [m-1]}(\text{best share complexity of a secret sharing scheme for } F_i \text{ for 1-bit secrets})$$

The improvement, boils down to replacing the DNF scheme used in [I+13] by the best known scheme for the relevant access structure. In the following, we provide a few natural examples where this leads to a large improvement.

Finding 2 st-connectivity in directed graph

We set graph G, directed and complete, with n vertices and s, t are unique vertices.

Theorem 3:

Observe the access structure $\mathcal{A}_G$, in which the players are the edges in graph G, and the minimal qualified set is the edges that contain 2 s-t directed paths in length $\leq n = |V|$ each, not necessarily a simple path. Therefore a secret sharing scheme is exists for $\mathcal{A}_G$ with share complexity $n^{O(log(n))}$

Proof :

We use a BL[3] construction in which we get share complexity as the number of leaves in the monotone formula for $\mathcal{A}_G$ (as we refer to $\mathcal{A}_G$, as a monotone function). Brief of the construction, let $f : P[m] \to \{0,1\}$ such that m is the number of players, and we map all the subsets from $[m]$ to $\{0,1\}$, such that 0 is for unqualified set , 1 is for qualified set and f is a monotone function. Since f is a monotone function it can be realized by a monotone formula. The monotone formula will be construct from AND and OR logic gates.

Claim 1:

There exists a secret sharing scheme for a access structure $\mathcal{A}_G$ with share complexity $\#l(F)$ such that F is a monotone formula for $\mathcal{A}_G$ and $\#l$ is the number of leaves in F.

The following will describe a formula F_G for $\mathcal{A}_G$ with $\#l(F_G) = n^{O(log(n))}$ and achieves a secret sharing scheme as required. We will describe the formula and block the number of leaves in it.

First, we describe the formula with number of predicates :

- $P^2(u,v)$ - this predicate indicates that there exists 2 u-v paths in length $\leq n = |V|$, not necessarily simple, such that

$$P^2(u,v) = \bigvee_{i=1}^{n} P^2(u,v,m) \bigvee_{1 \leq m_1 \leq m_2 \leq n} (P^1(u,v,m_1) \bigwedge P^1(u,v,m_2))$$

- $P^1(u,v,m)$- this predicate indicates that there exists 1 u-v path in length m, not necessarily simple, such that

$$P^1(u,v,m) = \bigvee_{w \in V} (P^1(u,w,\lceil \tfrac{m}{2} \rceil) \bigwedge P^1(w,v,\lfloor \tfrac{m}{2} \rfloor))$$

- $P^2(u,v,m)$ - this predicate indicates that there exists 2 u-v paths in length m, not necessarily simple, such that

$$P^2(u,v,m) = \bigvee_{w_1 \neq w_2, w_1, w_2 \in V} (P^1(u,w_1,\lceil \tfrac{m}{2} \rceil) \bigwedge P^1(w_1,v,\lfloor \tfrac{m}{2} \rfloor) \bigwedge P^1(u,w_2,\lceil \tfrac{m}{2} \rceil) \bigwedge P^1(w_2,v,\lfloor \tfrac{m}{2} \rfloor))$$
$$\bigvee_{w \in V} (P^2(u,w,\lceil \tfrac{m}{2} \rceil) \bigwedge P^1(w,v,\lfloor \tfrac{m}{2} \rfloor)) \bigvee_{w \in V} (P^1(u,w,\lceil \tfrac{m}{2} \rceil) \bigwedge P^2(w,v,\lfloor \tfrac{m}{2} \rfloor))$$

We notice that $F_G = P^2(s,t)$.

We also notice the tree that describe the formula:

For the first step we develop the predicate $P^2(u,v)$:

(1) For the part - $\bigvee_{i=1}^{n} P^2(u,v,m)$ we get n leaves.

(2) For the part - $\bigvee_{1 \leq m_1 \leq m_2 \leq n} (P^1(u,v,m_1) \bigwedge P^1(u,v,m_2))$ we get $\binom{n}{2}$ leaves.

Thus, for the first step we get $n + \binom{n}{2}$ leaves, such that each leave is monotone sub-formula.

For the second step we develop 2 predicates :

- For the predicate $P^1(u,v,m)$ we get $2n$ leaves.
- For the predicate $P^2(u,v,m)$ we have two parts -

(1) For the part - $\bigvee_{w_1 \neq w_2, w_1, w_2 \in V} (P^1(u,w_1,\lceil \tfrac{m}{2} \rceil) \bigwedge P^1(w_1,v,\lfloor \tfrac{m}{2} \rfloor) \bigwedge P^1(u,w_2,\lceil \tfrac{m}{2} \rceil) \bigwedge P^1(w_2,v,\lfloor \tfrac{m}{2} \rfloor))$ we get $4 \cdot \binom{n}{2}$.

(2) For the part - $\bigvee_{w \in V} (P^2(u,w,\lceil \tfrac{m}{2} \rceil) \bigwedge P^1(w,v,\lfloor \tfrac{m}{2} \rfloor)) \bigvee_{w \in V} (P^1(u,w,\lceil \tfrac{m}{2} \rceil) \bigwedge P^2(w,v,\lfloor \tfrac{m}{2} \rfloor))$ we get $4n$.

Thus, for $P^2(u,v,m)$ we get - $4 \cdot \binom{n}{2} + 4n$. We notice that for big enough m, $m \geq 2$, we get that $3n^2 \geq 4 \cdot \binom{n}{2} + 4n$, and denote $3n^2$ as the upper bound for the split coefficient for each level.

We advance recursively on the second step until we reach $m \leq 2$, eventually we will get that so far the tree height is $O(log(n))$.

For $m \leq 2$ we build an explicit DNF formula from the form :

- $P^1(u,v,m) = \bigvee\limits_{\overline{p_1}} \bigwedge\limits_{e_1 \in \overline{p_1}} X_{e_1}$, such that $\overline{p_1}$ is uv path in length $\leq m$
- $P^2(u,v,m) = \bigvee\limits_{\overline{p_1},\overline{p_2}} \bigwedge\limits_{e_1 \in \overline{p_1}} X_{e_1} \bigwedge\limits_{e_2 \in \overline{p_2}} X_{e_2}$, such that $\overline{p_1}, \overline{p_2}$ are 2 different uv paths in length $\leq m$

We notice to the size of each one:
- For $P^1(u,v,m)$ and $m \leq 2$ we get $\binom{n}{m} \leq n^2$.
- For $P^2(u,v,m)$ and $m \leq 2$ we get $\binom{n}{m}^2 \leq n^4$.
As before, we denote n^4 as the upper bound for the split coefficient.

Eventually, we get that the total number of leaves is $3n^2 \cdot n^4 = 3n^6$. In conclusion, we get that the size of each share is $(3n^6)^{O(log(n))} \sim n^{O(log(n))}$. Therefore. the share complexity is $n^{O(log(n))}$, as required.

Finding k st-connectivity in directed graph

In a similar way, we extend the claim for $k \in \mathbb{N}$ paths.
The players are the edges in graph G, and the minimal qualified set is the edges that contain k s-t directed paths in length $\leq n = |V|$ each, not necessarily simple. Then, there exists secret sharing scheme for $\mathcal{A}_G$ with share complexity $n^{O(log(n))}$.
We will describe the formula and block the number of leaves in it.
First, we describe the formula with number of predicates :

- $P^k(u,v)$ - this predicate indicates that there exists k u-v paths in length $\leq n = |V|$, not necessarily simple, such that

  $P^k(u,v) = \bigvee\limits_{H \subseteq \mathbb{N}^+ \times \mathbb{N}^+} (\bigwedge\limits_{j=1}^{l} P^{i_j}(u,v,h_j))$, such that $|H| = k$, $H = \{(h_1,i_1),...,(h_l,i_l)\}$,

  $\forall j \in [l]$ $\sum\limits_{j=1}^{l} i_j = k$, $h_j \leq n$, $i_j \geq 1$, $\forall i,j \in [l]$ $h_i \neq h_j$, i_j is the amount of paths in length h_j and h_j is the length of paths.

- $P^{k'}(u,v,m)$- this predicate indicates that there exists k' u-v path in length m, not necessarily simple, such that

  $P^{k'}(u,v,m) = \bigvee\limits_{W \subseteq V \times \mathbb{N}} (\bigwedge\limits_{j=1}^{l} P^{i_j}(u,v,w_j,m)))$, such that $|W| = k'$, $W = \{(w_1,i_1),...,(w_l,i_l)\}$,

  $\forall j \in [l]$ $\sum\limits_{j=1}^{l} i_j = k'$, $i_j \geq 1$, $\forall i,j \in [l]$ $w_i \neq w_j$, i_j is the amount of paths that pass through w_j and w_j is the middle vertex that the path pass through.

- $P^{k'}(u,v,w,m)$ - this predicate indicates that there exists k' u-v paths in length m, not necessarily simple, that pass through vertex w, such that

  $P^{k'}(u,v,w,m) = \bigvee\limits_{(k_1,k_2)} (P^{k_1}(u,w,\lceil \frac{m}{2} \rceil) \bigwedge P^{k_2}(w,v,\lfloor \frac{m}{2} \rfloor))$, such that $k_1 \cdot k_2 \geq k$ and this multiplition is minimal, in a way that k_1, k_2 can not be reduces and still get multiplition $\geq k$.

We noticed that $F_{G_k} = P^k(s,t)$.
We also notice the tree that describe the formula :

(1) For the first step we develop the predicate $P^k(u,v)$ - we would like to divide k identical paths between n different lengths, therefore we get $\binom{n-1+k}{k}$. Thus, the split coefficient will be $O(n^k)$.

(2) For the second step we develop the predicate $P^{k'} = (u,v,m)$ - first we notice that the worst case is $k' = k$, that is why in this case we would like to divide k identical paths between $|V| = n$ middle vertices that the paths pass through them, therefore also here we get $\binom{n-1+k}{k}$.Thus, the split coefficient will be $O(n^k)$.

(3) For the third step we develop the predicate $P^{k'} = (u,v,w,m)$ - we would like to count the number of k_1, k_2 pairs that fulfill the condition, and the number of this pairs is $2\sqrt{k}$. Thus, the split

coefficient will be $2\sqrt{k}$, this is a constant.

We can unite between the second and the third steps to a combined step. In the second step we got split coefficient $O(n^k)$, and in the third step we got constant number, thus the split coefficient of the combined step will be $O(n^k)$.

We advance recursively on the combined step until we reach $m \leq 2$ (as we got for 2 paths), eventually we will get that so far the tree height is $O(log(n))$.

For $m \leq 2$ we build an explicit DNF formula as we build for 2 paths, only here we generalize for k' :

$$P^{k'}(u, v, m) = \bigvee_{\overline{p_1}, ..., \overline{p_{k'}}} \bigwedge_{\forall i \in [k], e_i \in \overline{p_i}} X_{e_i}, \text{ such that } \overline{p_1}, ..., \overline{p_{k'}} \text{ are different uv paths in length} \leq m.$$

We notice that for $m \leq 2$ the split coefficient will be $\binom{n}{m}^{k'} = \binom{n}{2}^{k'} \leq n^{2k'}$, therefore $O(n^k)$, as we consider the k' worst case which is $k' = k$.

In conclusion, we get that the upper bound for the split coefficient is $O(n^k)$, and we get that the size of each share is $n^{k \cdot O(log(n))}$, because k is a constant number, the share complexity is $n^{O(log(n))}$, as required.

Hypergraph based fractional secret sharing

A hypergraph is generalization of a graph in which an edge can join any number of vertices - here we focus on regular hypergraphs where this number k is the same for all edges. In an interesting fractional variant of the access structure with k-hypergraph, the minimal qualified sets are edges of a fixed k-regular hypergraph $G(V, E)$. We define, for instance, $f(e) = m - 1$ for each $e \in E_0 \setminus E_1$. Also, we define a set $E_t \subset E_{t-1} \subset \ldots E_0 = E$ of edges, where the sets are E_i become more qualified as i increases, mapping to smaller and smaller values, with $f(e) = 0$ for $e \in E_t$.

Using AA's scheme, we conclude that there exist non-trivial fractional secret sharing schemes for the above fractional secret sharing.

Observation 1: There exists a fractional secret sharing scheme for hypergraph -based fractional secret sharing as above with share complexity of $2^{\tilde{O}(\sqrt{n})}$.

The observation follows directly from the fact that any subgraph of a k-regular hypergraph is also k-regular.

That can be very useful at the level of application, for example in offices we have certain groups of workers with different classification, and we can define each group with the same classification as edge in the hypergraph (E_i, V).

Failed approaches

While we searched for interesting natural examples in which our approach achieves more efficient solution than approach of [I+13], we found s-t connectivity as a nice example. We find it interesting to solve a variant of st-connectivity with simple paths. However we didn't succeeded to find a formula enough efficient in the literature, neither did we find other approaches (such as general MSP schemes).

Conclusion

We showed simple idea, which is instead of using any secret sharing scheme as one of the components, we can use the most efficient secret sharing scheme which is known, and indeed in many useful cases we know more efficient secret sharing scheme then what [I+13] used,we showed some natural examples through this paper.

Future work

In the section of Failed approaches we presented an idea for a solution with simple paths. We did not succeeded to find efficient scheme, but that does not mean it does not exist, that is why we leave this as an open question. Additionally, it could be interesting to find more natrualy examples of fractional access structures with good share complexity (instantiating our approach with some known secret sharing schemes).

References

[1] Lossy Chains and Fractional Secret Sharing by
Yuval Ishai, Eyal Kushilevitz and Omer Strulovich

[2] Algebraic Geometric Secret Sharing Schemes and Secure Multi-Party Computations over Small Fields by
Hao Chen and Ronald Cramer

[3] Generalized Secret Sharing and Monotone Functions by
Josh Benaloh and Jerry Leichter

[4] Superpolynomial Lower Bounds for Monotone Span Programs by
L'aszl'o Babai, Anna G'al and Avi Wigderson

[5] On the Power of Amortization in Secret Sharing: d-Uniform Secret Sharing and CDS with Constant Information Rate by
Benny Applebaum and Barak Arkis