On Linear Secret Sharing for Connectivity in Directed Graphs

Amos Beimel[1] and Anat Paskin[2]

[1] Dept. of computer science, Ben-Gurion University, Beer Sheva, Israel.
[2] Dept. of computer science, Technion, Haifa, Israel.

Abstract. In this work we study linear secret sharing schemes for s-t connectivity in directed graphs. In such schemes the parties are edges of a complete directed graph, and a set of parties (i.e., edges) can reconstruct the secret if it contains a path from node s to node t. We prove that in every linear secret sharing scheme realizing the st-con function on a directed graph with n edges the total size of the shares is $\Omega(n^{1.5})$. This should be contrasted with s-t connectivity in undirected graphs, where there is a scheme with total share size n. Our result is actually a lower bound on the size monotone span programs for st-con, where a monotone span program is a linear-algebraic model of computation equivalent to linear secret sharing schemes. Our results imply the best known separation between the power of monotone and non-monotone span programs. Finally, our results imply the same lower bounds for matching.

1 Introduction

Secret sharing schemes, introduced by [11, 35, 26], are a method in which a dealer holding a secret can distribute shares to parties in a network such that only predefined authorized sets of parties can reconstruct the secret from their shares. These schemes, whose original motivation was secure storage, have found numerous applications as a building box in complex cryptographic schemes, e.g., Byzantine agreement [32], secure multiparty computations [8, 16, 17], threshold cryptography [20], access control [30], and attribute based encryption [25]. In most applications it is important that the scheme is linear, that is, the shares are a linear combination of the secret and some random elements. Linear secret sharing schemes are equivalent to monotone span programs, a computational model introduced by Karchmer and Wigderson [28].

In this work we study linear secret sharing schemes for a natural function: the parties are edges of a complete *directed* graph, and a set of parties (i.e., edges) is authorized if it contains a path from node s to node t. We prove that in every linear secret sharing scheme realizing the st-con function on a directed graph with n edges the total size of the shares is $\Omega(n^{1.5})$. Studying linear secret sharing for this function has both a cryptographic motivation and a computational complexity motivation. We first discuss the cryptographic motivation. Benaloh

and Rudich [10] (see also [4, 28]) showed that there exists a simple and very efficient linear secret sharing scheme for the analogous function where the graph is *undirected*. This scheme was used in [30] to design a protocol for reliable access control. The obvious open problem is if this scheme can be generalized to deal with directed graphs. The computational complexity motivation is separating the power of monotone and non-monotone span programs. Our results imply that over infinite fields and large finite fields non-monotone span programs are more efficient than monotone span programs by a multiplicative factor of $\Omega(n^{0.5})$. This is the best separation known to-date.

1.1 Previous Results

In this section we will give a short background on secret sharing schemes, linear secret sharing schemes, monotone span programs, and the equivalence of the latter two notions. Finally, we will discuss some known results on the s-t connectivity function.

Secret-sharing schemes were first introduced by Blakley [11] and Shamir [35] for the threshold case, that is, for the case where the subsets that can reconstruct the secret are all the sets whose cardinality is at least a certain threshold. Secret-sharing schemes for general access structures were introduced by Ito, Saito, and Nishizeki [26]. More efficient schemes were presented in, e.g., [9, 36, 14, 28, 37, 22]. Even with the more efficient schemes, the size of the shares for general access structures with n parties is $\ell 2^{O(n)}$, where the secret is an ℓ-bit string. Lower bounds for secret sharing schemes were proved in [29, 9, 15, 13, 21, 18, 19, 12, 31]. The best lower bound was proved by Csirmaz [18], proving that, for every n, there is an access structure with n parties such that sharing an ℓ-bit secrets requires shares of length $\Omega(\ell n/\log n)$. Still there is an exponential gap between the lower-bounds and the upper-bounds.

Span programs and monotone span programs, introduced by Karchmer and Wigderson [28], are linear-algebraic models of computation. More specifically, a monotone span program is presented as a matrix over some field, with rows labeled by variables. The span program accepts an input if the rows whose variables are satisfied by the input span a fixed nonzero vector. The size of a span program is its number of rows. A detailed definition is given in Section 2. Lower bounds for monotone span programs have been studied in several papers. Beimel, Gál, and Paterson [6] provided a technique for proving lower bounds for monotone span programs and proved a lower bound of $O(n^{2.5})$ for a function with n variables. Babai, Gál, and Wigderson [2], using the technique of [6], proved the first super-polynomial lower bound – they prove an $n^{\Omega(\log n/\log\log n)}$ lower bound for the size of monotone span programs for the clique problem. Gál [23] gave a characterization of span program size and improved the lower bound for the clique function to $n^{\Omega(\log n)}$. Proving exponential lower bounds for an explicit function is an open problem (it is known that such lower bound holds for most functions [34]). Gál and Pudlák [24] have shown limitations of current techniques for proving lower bounds for monotone span programs. Beimel and Weinreb [7] showed a separating of the power of monotone span programs over

different fields, for example, they showed that there are functions that have small monotone span program over the field GF(2), however, they require super polynomial monotone span programs over fields whose characteristic is not 2.

In most applications of secret sharing schemes it is important that the scheme is linear, that is, the shares are a linear combination of the secret and some random elements. Linearity implies that if we sum shares distributed for two secrets, then we get shares of the sum of the secrets. This property is useful, for example, when designing secure multi-party protocols [8, 16, 17]. Karchmer and Wigderson [28] showed that monotone span programs imply linear secret sharing schemes (this result was implicitly proved also by Brickell [14]). More precisely, if there is a monotone span of size s computing a function f over a field $\mathbb{F}$ then there is a secret sharing scheme realizing f such that the domain of secrets is $\mathbb{F}$ and the total number of bits of the shares is $s \log |\mathbb{F}|$. In fact, monotone span programs and linear secret sharing schemes are equivalent [3]. Thus, proving lower bounds for monotone span programs implies the same lower bounds for linear secret sharing schemes.

In this work we prove lower bounds for the st-con function. This function is widely studied in complexity both for directed and undirected graphs. For example, st-con in directed graphs is NL-complete, while Reingold [33] has proved that st-con in undirected graphs is in deterministic log-space. Another example where undirected st-con is easier than directed st-con was given by Ajtai and Fagin [1]; they showed that while undirected st-con is definable in monadic second order logic, the directed case is not. We continue this line of research by proving that for monotone span programs undirected st-con is easier than directed st-con, although the gap we can prove is much smaller.

The circuit complexity of st-con has been studied as well. The directed (and undirected) st-con function has a polynomial-size monotone circuit of depth $O(\log n)$; this circuit has unbounded fan-in. This implies a monotone formula for st-con of size $n^{O(\log n)}$ and, using the construction of Benaloh and Leichter [9], there is a secret sharing scheme realizing the st-con function in which the size of the shares is $n^{O(\log n)}$. Karchmer and Wigderson [27] have proved that for monotone formulae this is optimal – every monotone formula computing undirected (and, hence, directed) st-con function has size $n^{\Omega(\log n)}$.

1.2 Our Results

In this work we prove that a monotone span program computing the st-con function on a *directed* graph with n edges has size $\Omega(n^{1.5})$. We supply two proofs of this lower bound. The first proof uses the characterization of span program size given by Gál [23]; this proof only holds for finite fields. The second proof uses the condition of Beimel, Gál, and Paterson [6]; this proof holds for every field. As monotone span program are equivalent to linear secret sharing schemes, our result implies that in every linear secret sharing scheme realizing the st-con function in directed graphs, the total size of the shares is $\Omega(n^{1.5})$.

Our lower bound has a few additional implications. First, it shows that, for monotone span programs and linear secret sharing, undirected st-con is easier

than directed st-con. This is true since there is a monotone span program realizing undirected st-con whose size is n [10, 28] (see Example 1 below).

Furthermore, our lower bound supplies the best known separation between the power of monotone and non-monotone span programs. Beimel and Gál [5] proved that over infinite fields and large finite fields the directed st-con function on graphs with n edges has a *non-monotone* span program of size $O(n)$. Thus, our result shows a separation of multiplicative factor of $\Omega(n^{0.5})$ between monotone and non monotone span programs for directed st-con. Separations between monotone and non-monotone models of computation is an important question in complexity, e.g., the exponential separation between the power of monotone and non-monotone circuits [38]. Separations between the power of monotone and non-monotone span programs is interesting since monotone span programs can be exponentially more powerful than monotone circuits [2].

Finally, our result implies the same lower bound for matching and bipartite matching. This follows from the projection reduction from directed st-con to bipartite matching. Babai, Gál, and Wigderson [2] constructed a non-monotone span program, over large enough fields, for matching whose size is n (where n is the number of edges in the graph). Thus, the same separation between monotone and non-monotone span programs holds for matching.

1.3 Organization

In Section 2 we define monotone span programs. In Section 3 we give our first proof of the lower bound and in Section 4 we give our second proof of the lower bound.

2 Preliminaries

2.1 Monotone Span Programs

We start with the definition of monotone span programs. As discussed above, monotone span programs are equivalent to linear secret sharing schemes; we use monotone span programs to prove lower bounds on linear secret sharing schemes.

Definition 1 (Monotone Span Program [28]). *A* monotone span program *over a field $\mathbb{F}$ is a triplet $\widehat{M} = \langle M, \rho, \boldsymbol{v} \rangle$, where M is a matrix over $\mathbb{F}$, $\boldsymbol{v}$ is a nonzero row vector called the target vector (it has the same number of coordinates as the number of columns in M), and ρ is a labeling of the rows of M by variables from $\{x_1, \ldots, x_n\}$ (every row is labeled by one variables, and the same variable can label many rows).*

A monotone span program accepts or rejects an input by the following criterion. For every input $u \in \{0,1\}^n$ define the sub-matrix M_u of M consisting of those rows whose labels are satisfied by the assignment u. The monotone span program $\widehat{M}$ accepts *u if and only if $\boldsymbol{v} \in \text{span}(M_u)$, i.e., some linear combination of the rows of M_u gives the target vector $\boldsymbol{v}$. A monotone span program* computes

a Boolean function f if it accepts exactly those inputs u where $f(u) = 1$. The size *of $\widehat{M}$ is the number of rows in M.*[3]

Monotone span programs compute only monotone functions, and every monotone Boolean function can be computed by a monotone span program. The size of the smallest monotone span program over $\mathbb{F}$ that computes f is denoted by $\mathrm{mSP}_{\mathbb{F}}(f)$.

Example 1. Consider the undirected-st-con function, whose input is an undirected graph with two designated nodes s and t and its output is 1 iff the graph contains a path from s to t. Formally, we consider the following function: The input is an undirected graph with $m+2$ nodes; the variables of the function are the $n = \binom{m+2}{2}$ possible edges. Karchmer and Wigderson [28] construct a monotone span program of size n for this function, that is, each edge labels exactly one row in the program (a linear secret sharing scheme equivalent to this program was previously shown in [10]).

We next describe this span program. Assume the nodes of the input graph are $z_0, \ldots, z_{m+1}$, where $z_0 = s$ and $z_{m+1} = t$. The program has $m+2$ columns and n rows. For every edge (z_i, z_j), where $i < j$, there is a row in the program; in this row all entries in the row are zero, except for the ith entry which is 1 and the jth entry which is -1. The target vector is the same as the row labeled by (s,t), that is, $(1, 0, \ldots, 0, -1)$. It can be proven that over every field $\mathbb{F}$, this program computes the undirected-st-con function.

2.2 The st-con Function

In the rest of the paper we will refer to the st-con function in directed graphs as st-con. Formally, we consider the following function: The input is a directed graph with $m+2$ nodes. The graph contains two designated nodes s, t. The variables are the $n = m(m+1)$ possible edges in the graph. The function outputs 1 iff there is a directed path from node s to node t. Our main results are summarized below.

Theorem 1. *For every field $\mathbb{F}$*

$$\mathrm{mSP}_{\mathbb{F}}(\text{st-con}) = \Omega(n^{1.5}).$$

Theorem 2. *For every finite field $\mathbb{F}$ and every linear secret sharing scheme over $\mathbb{F}$ realizing* st-con *the total number of bits in the shares is*

$$\Omega(n^{1.5} \log |\mathbb{F}|).$$

[3] The choice of the fixed nonzero vector $\boldsymbol{v}$ does not affect the size of the span program. It is always possible to replace $\boldsymbol{v}$ by another nonzero vector $\boldsymbol{v}'$ via a change of basis without changing the function computed and the size of the span program. Most often $\boldsymbol{v}$ is chosen to be the $\mathbf{1}$ vector (with all entries equal 1).

3 First Proof

3.1 Proof outline

We use the following theorem of Gál [23] to prove our lower bound.

Theorem 3 ([23]). *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function. Let U denote the set of maxterms of f, and V denote the set minterms of f, and let $U' \subseteq U, V' \subseteq V$. If there exists a monotone span program of size s computing f over a field $\mathbb{F}$, then there exist matrices $F_1, \ldots, F_n$, each matrix has $|U'|$ rows and $|V'|$ columns (each row of the matrix is labeled by a $u \in U'$ and each column is labeled by a $v \in V'$) such that*

1. *$\sum_{i=1}^{n} F_i = \mathbf{1}$ (that is, the sum of the matrices over $\mathbb{F}$ is the all-one matrix).*
2. *The non-zero entries in F_i are only in rows labeled by a $u \in U'$ such that $u_i = 0$ and in columns labeled by a $v \in V'$ such that $v_i = 1$.*
3. *$\sum_{i=1}^{n} \mathrm{rank}_{\mathbb{F}}(F_i) \leq s$.*

In this section, we prove the result for GF(2), but the proof easily generalizes to other finite fields. The skeleton of the proof is as follows. We appropriately choose subsets U', V' of the maxterms and minterms of st-con. We show that for any matrices $F_1, \ldots, F_n$ satisfying (1) and (2) in Theorem 3, there exist "many" ($\Omega(n)$) matrices F_e, such that a large fraction ($\Omega(1)$) of the entries of F_e are zero entries. Also, every F_e has some "singleton" 1 entries at fixed positions, which are "well-spread" over the matrix. We then prove that every matrix F_e with "many" zero entries has rank $\Omega(n^{0.5})$, this proof uses the partial knowledge on the distribution of singletons, and the large number of zeros. By Theorem 3, this implies that the size of every monotone span program computing st-con over GF(2) has at least $\Omega(n^{0.5} \cdot n) = \Omega(n^{1.5})$ rows.

3.2 Details

To apply Theorem 3 we need to understand the minterms and maxterms of st-con. Every minterm of st-con is a simple directed paths from s to t. Every maxterm can be specified by a partition $S \cup T$ of V with $s \in S, t \in T$ where the edges in $S \times T$ are *excluded* and all other edges are included in the maxterm (that is, the maxterm contains all edges in $S \times S$, $T \times T$, and $T \times S$).

Defining U', V': Let $w = m/d$, where d is some constant to be fixed later.[4] We arrange the nodes of the graph in layers $L_0, L_1, \ldots, L_{d+1}$, where $L_0 = \{s\}, L_{d+1} = \{t\}$, and all other layers contain w nodes. We consider the restriction st-con$'$ of the st-con function to directed graphs that contain only edges directed from layer L_i to layer L_{i+1}. Note that the number of edges in the restricted function st-con$'$ is a constant fraction of the number of edges in the function st-con, so every lower bound for st-con$'$ implies the same lower bound for st-con (up to a constant factor). We define the subsets U', V' as follows. Let V' be all the s-t

[4] As we see later, $d = 4$ will do.

paths, that is, paths $s, v_1, \ldots, v_d, t$, where $v_i \in L_i$. Let U' be the set of all s-t cuts where $1/2$ of the nodes in each layer L_i, where $1 < i < d$, are in S (and the other half is in T). Additionally, $\{s\} \cup L_1 \subset S$ and $\{t\} \cup L_d \subset T$. Note that $|V'| = w^d$ and $|U'| = \binom{w}{w/2}^{d-2}$.

Assume there is a monotone span program over $\mathbb{F}$ computing st-con$'$ and let $F_1, \ldots, F_n$ be the matrices guaranteed by Theorem 3.[5] For an edge $e = (x, y)$, let R_e denote the restriction of F_e to rows labeled by a cut $u \in U'$ such that $u_e = 0$ (that is, the maxterm does not contain the edge (x, y)) and to columns labeled by a path $v \in V'$ such that $v_e = 1$ (that is, the path contains the edge (x, y)). Note that R_e has $w^{d-2} = |V'|/w^2$ columns and $0.25\binom{w}{w/2}^{d-2} = |U'|/4$ rows (as we consider cuts such that $x \in S$ and $y \in T$).[6] By (2) in Theorem 3, $\text{rank}_{\mathbb{F}}(R_e) = \text{rank}_{\mathbb{F}}(F_e)$. We say that R_e covers (u, v) if $u_e = 0$ and $v_e = 1$. Denote the set of edges e such that R_e covers (u, v) by $S(u, v)$.

We start with a few simple observations. Obseration 1 and Obseration 2 follow directly from (1) and (2) in Theorem 3 and the definition of the R_e's.

Observation 1. If $|S(u, v)| = 1$, then $F_e(u, v) = R_e(u, v) = 1$ for the edge $e \in S(u, v)$. We refer to such entries (u, v) as "singletons".

Observation 2. If $|S(u, v)|$ is even, then $R_e(u, v) = 0$ for some $e \in S(u, v)$.

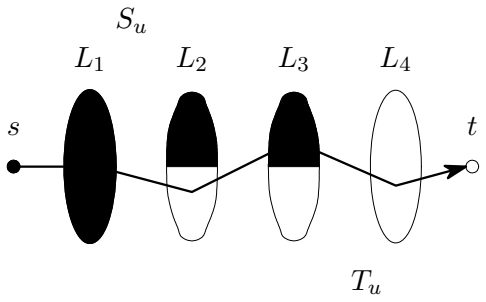

Fig. 1. An illustration of a path and a cut for which $|S(u, v)|$ is even. The vertices in S_u are black and the vertices in T_u are white. The edges in $S(u, v)$ are the edge between L_1 and L_2 and the edge between L_3 and L_4.

Lemma 1. *Let $c = 1/4$. For $d = 4$ there are at least $c|U'||V'|$ pairs (u, v) such that $|S(u, v)|$ is even.*[7]

[5] For an edge $e = (s, x)$ or $e = (x, t)$, the matrix $F_e = \mathbf{0}$ (by the definition of the maxterms). We, therefore, ignore such matrices).

[6] This is true if $x \in L_j$ for $2 \leq j \leq d - 1$; the number of rows for $x \in L_1$ or $x \in L_{d-1}$ is $0.5\binom{w}{w/2}^{d-2} = |U'|/2$.

[7] For $d = 5$, the constant c is 0.5 and for any sufficiently large d, this constant approaches $1/2$.

Proof. From the definition of U', V', and (2) in Theorem 3, it follows that $S(u,v)$ is precisely the set of edges in v that belong to $S_u \times T_u$ (where the partition $S_u \cup T_u$ specifies the maxterm u). Fix some cut $u \in U'$. For a path v the size of $S(u,v)$ is even if the path has an even number of edges going from S_u to T_u. For $d = 4$ this is true if the vertex in L_2 is in T_u and the vertex in L_3 is in S_u, that is, the edges in $S(u,v)$ are the edge between L_1 and L_2 and the edge between L_3 and L_4. See Fig. 3.2 for a description. Since half of the vertices of L_2 are in T_u and half of the vertices of L_3 are in S_u, for a quarter of the paths $v \in V'$, the size of $S(u,v)$ is even. □

We now move to our two main lemmas.

Lemma 2. *There exist at least cw^2 edges e such that R_e contains at least a $\frac{c}{d}$ fraction of zeros, where c is the constant from Lemma 1.*

Proof. We construct a set of edges as required, proceeding in iterations. By Lemma 1, for all (u,v) the set

$$B = \{(u,v) \,:\, |S(u,v)| \text{ is even}\}$$

must satisfy $R_e(u,v) = 0$ for some edge $e \in S(u,v)$. That is, we need to "cover" the set B by edges in this sense, where e covers $(u,v) \in B$ if $R_e(u,v) = 0$.

Denote by B_i the set of entries uncovered at the beginning of iteration i. In particular, $B_1 = B$. By Lemma 1, $|B_1| = c|U'||V'|$ for some constant c. We start an iteration i if $|B_i| \geq c|U'||V'|/2$. Since there are at most $w^2(d-1) - i \leq w^2 d$ edges to choose from, at least one of them should cover at least $\frac{c|U'||V'|/2}{w^2 d}$ uncovered entries (by the pigeon hole principle). We pick any of those e's and add it to the list. Note that the rectangle R_e has at most $|U'|/2 \cdot |V'|/w^2$ entries[8], thus a fraction of at least c/d of the entries of R_e are 0. Each selected edge e covers at most $|U'||V'|/2w^2$ uncovered entries (the number of entries in R_e). Since we halt only when at least $c|U'||V'|/2$ pairs in B have been covered, at least cw^2 iterations are needed. □

To complete the proof, it remains to show that every rectangle R_e with "many" zeros, as in Lemma 2, has high degree.

Lemma 3. *Let R_e, for $e = (x,y)$, be a rectangle with a fraction of at least c/d zero entries. Then* $\text{rank}_{\text{GF}(2)}(R_e) = \Omega(n^{0.5})$.

Proof. In the following proof we restrict our attention only to the rows and columns of R_e. First note that a fraction of at least $c/2d$ of the rows contain at a fraction of at least $c/2d$ zero entries (otherwise the fraction of zero entries in R_e is less than $c/2d \cdot 1 + (1 - c/2d) \cdot c/2d < c/d$). Thus, the number of rows with at least $c|V'|/(2w^2 d)$ zero entries is at least $c|U'|/(8d)$. We will show that these rows contain many distinct rows, which will imply that R_e has rank $\Omega(n^{0.5})$.

[8] This is the number of entries if $x \in L_1$, otherwise this number is $|U'|/4 \cdot |V'|/w^2$.

Fix any row u_0 of R_e with at least $c|V'|/(2w^2d)$ zero entries. We show that the row u_0 can only appear in R_e a small number of times (labeled by different u's). Let M be the set of columns in which this row has zero entries; the size of M is at least $c|V'|/(4d)$. Let $e = (x, y)$, where x belongs to layer L_j for some j and $y \in L_{j+1}$.

We first prove that M contains a subset M' of paths of size $\epsilon \cdot w$ for some sufficiently small constant ϵ (to be fixed later) such that every two paths in M' have no nodes in common except for x, y, s, t. Similarly to the proof of Lemma 2, we construct this set iteratively. In the first iteration, we add to M' some arbitrary path in M. We continue adding paths until there are ϵw paths. In iteration $i + 1$, we have added i paths to M'. We prove that another path can be added so that all the paths in M' satisfy the invariant of being disjoint up to including s, x, y, t. Any path using one of the $w - i$ unused nodes in every layer L_k, where $k \neq j, j + 1$, can be used here. The number of all columns of R_e with this property is at least $(w - i)^{d-2} \geq (w(1 - \epsilon))^{d-2}$, thus the number of columns in R_e violating this property is at most

$$w^{d-2} - (w(1-\epsilon))^{d-2} = w^{d-2}(1-(1-\epsilon)^{d-2}) \approx w^{d-2}\epsilon(d-2) = |V'|\epsilon(d-2) < |V'|\epsilon d.$$

(for a sufficiently small constant ϵ). Taking $\epsilon \leq c/(4d^2)$, there are at least $c|V'|/(4d) - |V'|\epsilon d > 1$ paths in M satisfying this property.

Having proved $M' = \{v_1, \ldots, v_{\epsilon|M|}\}$ as above exists, we consider the set of rows

$$B' = \{u : e \notin u \text{ and } |S(u, v)| > 1 \text{ for every } v \in M'\}.$$

Notice that for every $u \notin B'$, where $e \notin u$, there exists a $v \in M'$ such that $|S(u, v)| = 1$, thus, by Obseration 1, $R_e(u, v) = 1$, however, $R_e(u_0, v) = 0$ since $v \in M$ (where M is the set of columns with zero entries in the row u_0). Thus, B' is the set of rows in R_e that can be equal to the row u_0. Recall that $e = (x, y) \in S(u, v)$ for every row u of R_e and every column v of R_e (by the definition of R_e). Thus, $S(u, v) > 1$ if the cut u does not contain at least one edge on the path v in addition to the edge (x, y).

We next show that B' is of negligible size. We do this by calculating the probability that a cut chosen with uniform distribution is B'. We choose a random cut $u = (S, T)$ by first choosing for each node in v_1 if its in S or in T, then the nodes corresponding to v_2, and so on, where the inclusion in S or T is selected at random according to the proportion of the remaining colors for that layer (conditioned on the choices for the previous v_i's). The cut u forms a singleton with a given v_i, selected in iterations i, if the node in v_i from L_k for $j' \leq j$ are in S, and the rest of the nodes in v_i are in T. This happens with probability at least $(1/2 - \epsilon)^{d-2} \stackrel{\text{def}}{=} 1 - f$. Thus, with probability at most f the cut u does not form a singleton with a given v_i. Note f is some constant. Therefore, $|S(u, v)| > 1$ for every $v \in M'$ with probability at most

$$f^{|M'|} = f^{\epsilon w} = 2^{-\theta(w)}.$$

This implies that the size of B' is at most $2^{-\theta(w)}|U'|/2$.

Since there are at least $c|U'|/(4d)$ rows with a fraction of at least $c/(2d)$ zeros, and each such row can appear at most $2^{-\theta(w)}|U'|/2$ in R_e, the number of distinct rows in R_e is at least

$$\frac{c|U'|/(4d)}{2^{-\theta(w)}|U'|/2} = 2^{\theta(w)}.$$

This implies that $\text{rank}_{\text{GF}(2)}(F_e) = \text{rank}_{\text{GF}(2)}(R_e) = \log(2^{\theta(w)}) = \theta(w) = \theta(n^{0.5})$. □

By Lemma 2 and Lemma 3, there are $\Omega(n)$ matrices F_e such that

$$\text{rank}_{\text{GF}(2)}(F_e) = \theta(n^{0.5}).$$

Thus, by Theorem 3, every monotone span program computing st-con has size $\Omega(n^{1.5})$.

4 Second Proof

In this proof we use a technique of [6] to prove lower bounds for monotone span programs. They prove that if the set of minterms of f contains a "big" set of self-avoiding minterms as defined below, then for every field $\mathbb{F}$ the size of every monotone span program over $\mathbb{F}$ computing f is "big".

Definition 2 (Self-Avoiding Minterms). *Let f be a monotone Boolean function and V be the set of all of its minterms. Let $V' \subseteq V$ be a subset of the minterms of f. We say that V' is* self avoiding for f, *if every $v \in V'$ contains a set $\text{C}(v) \subseteq v$, called the core of v, such that the following three conditions are satisfied.*

1. $|\,\text{C}(v)| \geq 2$.
2. *The set $\text{C}(v)$ uniquely determines v in V'. That is, no other minterm in V' contains $\text{C}(v)$.*
3. *For any subset $Y \subseteq \text{C}(v)$, the set*

$$S_Y = \bigcup_{A \in V', A \cap Y \neq \emptyset} A \setminus Y$$

does not contain any minterm in V.

Note that (3) requires that S_Y contains no minterm from f, not just none from V'.

Theorem 4. *Let f be a monotone Boolean function, and let V' be a self-avoiding subset of minterms for f. Then for every field $\mathbb{F}$,*

$$\text{mSP}_{\mathbb{F}}(f) \geq |V'| \ .$$

As in the first proof, we consider a graph with $m+2$ nodes, and let $w = m/4$. We arrange the nodes of the graph in layers $L_0, L_1, \ldots, L_5$, where $L_0 = \{s\}, L_5 = \{t\}$, and all other layers contain w nodes. We denote the nodes in layer L_j, where $1 \leq j \leq 4$ by $v_{j,1}, \ldots, v_{j,w}$. We consider the restriction st-con$'$ of the st-con function to directed graphs that contain only edges directed from layer L_i to layer L_{i+1}. We prove that every monotone span program for st-con$'$ has size $\Omega(w^3) = \Omega(n^{1.5})$. The proof is by exhibiting a self-avoiding set of minterms as defined in Definition 2.

The self-avoiding set for st-con$'$. For every $a, b, c \in \{1, \ldots, w\}$ there is a path $P_{a,b,c}$ in the set:

$$s, v_{1,a}, v_{2,b}, v_{3,c}, v_{4,a}, t.$$

That is, the indices of the nodes from L_1 and L_4 are equal. The core $\mathrm{C}(P_{a,b,c})$ is $\{(v_{1,a}, v_{2,b}), (v_{3,c}, v_{4,a})\}$. Clearly, the core determines the path $P_{a,b,c}$.

We have to show that for every $Y \subseteq \mathrm{C}(P)$ the set S_Y does not contain a path from s to t. If $|Y| = 1$ then S_Y does not contain an edge from one layer. E.g., if $Y = \{(v_{1,a}, v_{2,b})\}$ then S_Y does not contain any edges going from V_1 to V_2.

We next consider the somewhat more complex case when $|Y| = 2$. In this case S_Y is composed of the following edges:

1. $(s, v_{1,a})$ from the first level.
2. $(v_{1,a}, v_{2,b'})$ for every $b' \neq b$ from the second level.
3. $(v_{2,b}, v_{3,c'})$ for every c', and $(v_{2,b'}, v_{3,c})$ for every b' from the third level.
4. $(v_{3,c'}, v_{4,a})$ for every $c' \neq c$ from the fourth level.
5. $(v_{4,a}, t)$ from the fifth level.

Assume S_Y contains a path from s to t. Since $v_{2,b}$ does not have any incoming edges then the path has to pass through $v_{2,b'}$ for some $b' \neq b$. Thus it must pass through $v_{3,c}$. But $v_{3,c}$ has no outgoing edges in S_Y, contradiction.

To conclude, we have proved that st-con$'$ has a self-avoiding set of size $w^3 = O(n^{1.5})$ and Theorem 4 implies our main result – Theorem 1.

Acknowledgment. We would like to thank Eyal Kushilevitz for helpful discussions.

References

1. M. Ajtai and R. Fagin. Reachability is harder for directed than for undirected finite graphs. *J. Symb. Log.*, 55(1), 1990.
2. L. Babai, A. Gál, and A. Wigderson. Superpolynomial lower bounds for monotone span programs. *Combinatorica*, 19(3):301–319, 1999.
3. A. Beimel. *Secure Schemes for Secret Sharing and Key Distribution.* PhD thesis, Technion, 1996. www.cs.bgu.ac.il/ beimel/pub.html.
4. A. Beimel and B. Chor. Universally ideal secret sharing schemes. *IEEE Trans. on Information Theory*, 40(3):786–794, 1994.

5. A. Beimel and A. Gál. On arithmetic branching programs. *J. of Computer and System Sciences*, 59:195–220, 1999.
6. A. Beimel, A. Gál, and M. Paterson. Lower bounds for monotone span programs. *Computational Complexity*, 6(1):29–45, 1997. Conference version: FOCS '95.
7. A. Beimel and E. Weinreb. Separating the power of monotone span programs over different fields. *SIAM J. on Computing*, 34(5):1196–1215, 2005.
8. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computations. In *Proc. of the 20th ACM Symp. on the Theory of Computing*, pages 1–10, 1988.
9. J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In S. Goldwasser, editor, *Advances in Cryptology – CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 27–35. Springer-Verlag, 1990.
10. J. Benaloh and S. Rudich. Private communication, 1989.
11. G. R. Blakley. Safeguarding cryptographic keys. In R. E. Merwin, J. T. Zanca, and M. Smith, editors, *Proc. of the 1979 AFIPS National Computer Conference*, volume 48 of *AFIPS Conference proceedings*, pages 313–317. AFIPS Press, 1979.
12. C. Blundo, A. De Santis, R. de Simone, and U. Vaccaro. Tight bounds on the information rate of secret sharing schemes. *Designs, Codes and Cryptography*, 11(2):107–122, 1997.
13. C. Blundo, A. De Santis, A. Giorgio Gaggia, and U. Vaccaro. New bounds on the information rate of secret sharing schemes. *IEEE Trans. on Information Theory*, 41(2):549–553, 1995.
14. E. F. Brickell. Some ideal secret sharing schemes. *Journal of Combin. Math. and Combin. Comput.*, 6:105–113, 1989.
15. R. M. Capocelli, A. De Santis, L. Gargano, and U. Vaccaro. On the size of shares for secret sharing schemes. *J. of Cryptology*, 6(3):157–168, 1993.
16. D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *Proc. of the 20th ACM Symp. on the Theory of Computing*, pages 11–19, 1988.
17. R. Cramer, I. Damgård, and U. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In B. Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 316–334. Springer-Verlag, 2000.
18. L. Csirmaz. The size of a share must be large. In A. De Santis, editor, *Advances in Cryptology – EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 13–22. Springer-Verlag, 1995. Journal version in: *J. of Cryptology*, 10(4):223–231, 1997.
19. L. Csirmaz. The dealer's random bits in perfect secret sharing schemes. *Studia Sci. Math. Hungar.*, 32(3–4):429–437, 1996.
20. Y. Desmedt and Y. Frankel. Shared generation of authenticators and signatures. In J. Feigenbaum, editor, *Advances in Cryptology – CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 457–469. Springer-Verlag, 1992.
21. M. van Dijk. On the information rate of perfect secret sharing schemes. *Designs, Codes and Cryptography*, 6:143–169, 1995.
22. M. van Dijk. A linear construction of secret sharing schemes. *Designs, Codes and Cryptography*, 12(2):161–201, 1997.
23. A. Gál. A characterization of span program size and improved lower bounds for monotone span programs. *Computational Complexity*, 10(4):277–296, 2002.
24. A. Gál and P. Pudlák. Monotone complexity and the rank of matrices. *Inform. Process. Lett.*, 87:321–326, 2003.

25. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proc. of the 13th ACM conference on Computer and communications security*, pages 89–98, 2006.
26. M. Ito, A. Saito, and T. Nishizeki. Secret sharing schemes realizing general access structure. In *Proc. of the IEEE Global Telecommunication Conf., Globecom 87*, pages 99–102, 1987. Journal version: Multiple assignment scheme for sharing secret. *J. of Cryptology*, 6(1):15-20, 1993.
27. M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. on Discrete Mathematics*, 3(2):255–265, 1990.
28. M. Karchmer and A. Wigderson. On span programs. In *Proc. of the 8th IEEE Structure in Complexity Theory*, pages 102–111, 1993.
29. E. D. Karnin, J. W. Greene, and M. E. Hellman. On secret sharing systems. *IEEE Trans. on Information Theory*, 29(1):35–41, 1983.
30. M. Naor and A. Wool. Access control and signatures via quorum secret sharing. *IEEE Transactions on Parallel and Distributed Systems*, 9(1):909–922, 1998.
31. C. Padró and G. Sáez. Secret sharing schemes with bipartite access structure. *IEEE Trans. on Information Theory*, 46:2596–2605, 2000.
32. M. O. Rabin. Randomized Byzantine generals. In *Proc. of the 24th IEEE Symp. on Foundations of Computer Science*, pages 403–409, 1983.
33. O. Reingold. Undirected ST-connectivity in log-space. In *Proc. of the 37th ACM Symp. on the Theory of Computing*, pages 376–385, 2005.
34. L. Rónyai, L. Babai, and M. K. Ganapathy. On the number of zero-patterns of a sequence of polynomials. *Journal of the AMS*, 14(3):717–735, 2001.
35. A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.
36. G. J. Simmons, W. Jackson, and K. M. Martin. The geometry of shared secret schemes. *Bulletin of the ICA*, 1:71–88, 1991.
37. D. R. Stinson. Decomposition construction for secret sharing schemes. *IEEE Trans. on Information Theory*, 40(1):118–125, 1994.
38. E. Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988.