Secret Sharing Schemes for Threshold Graphs

Taylor Smith, William Herring and Zachary Frenette

August 10, 2016

1 Introduction

Secret sharing schemes are a family of methods used for distributing a secret among a group of users such that only select subsets of those users are able to reconstruct the secret. Introduced by Shamir [9] and Blakley [3] in 1979, secret sharing schemes have become a popular area of research in cryptography and security. Secret sharing has seen applications across computer science, with implementations being found in certificate issuing, database information storage, electronic voting, and e-commerce. Al Ebri et al. list further examples of applications in their paper [1]. The work of Beimel provides a good overview of various secret sharing schemes [2], and Stinson and Wei maintain an extensive bibliography of papers on secret sharing [11].

One of the most common types of secret sharing schemes is the so-called "threshold" scheme. In order for users to reconstruct a secret, they must work together by combining their shares in order to overcome some threshold value. More specifically, if we have a total of n users, then a (k, n)-threshold scheme requires k of the n users to reconstruct the secret. Many implementations of threshold schemes are possible. For example, Shamir achieves such a scheme through polynomial interpolation; not surprisingly, this method is known as Shamir's secret sharing scheme [9].

Generally, we are given an access structure that contains the qualified subsets of users. In this project, we examine the case where these qualified subsets are pairs of users. In this context, we can model an access structure as a graph. That is, we define a vertex for every user and we add an edge between two vertices if and only if the pair of users is included in the access structure. Many results for secret sharing schemes in the graph model are known. Stinson showed that, given a graph G of maximum degree d, there exists a secret sharing scheme for G whose information rate is bounded below by $\frac{2}{d+1}$ [10]. Later, van Dijk and Blundo et al. showed that this value is tight for certain families of d-regular graphs [7, 4]. Secret sharing results have also been extended to other families of graphs, such as trees, paths, and cycles [6].

Recently, secret sharing schemes have been studied in the online setting [6, 8]. In this setting, the set of users with whom the secret will be shared is not known in advance, and could potentially be infinite. In other words, the users arrive one at a time, with no indication of who will arrive next nor of whether any more users will arrive.

The remaining sections of this project are organized as follows. In section 2, we present some preliminary definitions. In section 3, we discuss some of the

results regarding online secret sharing from the work of Komargodski et al. [8]. In section 4, we present our own results for threshold graphs. Finally, we state some concluding remarks and mention avenues for future work in Section 5.

2 Preliminaries

We begin with some preliminary definitions which will be used throughout this project. Let $\mathcal{U}_n$ denote a set of n users and let $s \in \{0,1\}$ denote a secret bit. The notion of secret sharing relies upon the idea of an access structure. Essentially, an access structure defines which subsets of users are qualified to reconstruct s. More formally, given an access structure $\mathcal{A} \subseteq 2^{\mathcal{U}_n}$, if $X \in \mathcal{A}$, then the members of X can reconstruct s using their shares. On the other hand, if $X \notin \mathcal{A}$, then the members of X cannot learn anything about s.

Definition 1 (Access structure)**.** *Given a set of users $\mathcal{U}_n$, an access structure is a monotone collection of subsets $\mathcal{A} \subseteq 2^{\mathcal{U}_n}$. The subsets contained in $\mathcal{A}$ are called "qualified", while subsets not contained in $\mathcal{A}$ are called "unqualified".*

Next, we define the notion of an access structure in the online setting. We reuse the terminology from the paper of Komargodski et al. [8].

Definition 2 (Restriction of an access structure)**.** *Let $\mathcal{A}$ be an access structure over n users and let $m < n$ be a positive integer. Then the restriction of $\mathcal{A}$ to the first m users is defined as $\mathcal{A}|_m = \{X \in \mathcal{A} : \{m+1, \ldots, n\} \cap X = \emptyset\}$.*

Definition 3 (Evolving access structure)**.** *Let $\mathcal{A} = \{\mathcal{A}_t\}_{t \in \mathbb{N}}$ denote an infinite sequence of access structures. We say that $\mathcal{A}$ is an evolving access structure if, for every $t \in \mathbb{N}$, $\mathcal{A}_t$ is an access structure over t users and $\mathcal{A}_t|_{t-1} = \mathcal{A}_{t-1}$.*

In their paper, Komargodski et al. proved that there exists a secret sharing scheme for any evolving access structure. In particular, they show that the size of the share for the t^{th} arriving user is at most 2^{t-1} bits. For completeness, we state their result here.

Theorem 1 (Theorem 3.1, [8])**.** *Let $\mathcal{A}$ be an evolving access structure. Then there exists a secret sharing scheme for $\mathcal{A}$ where the t^{th} user receives at most 2^{t-1} bits.*

Next, we define some graph-theoretic terms and notation. Consider a graph $G = (V, E)$. For any subset $W \subseteq V$, we define $G[W]$ to be the subgraph of G induced by W. Moreover, for any subgraph H of G, we define $G \setminus H$ to be the graph edge difference; that is, the graph obtained by deleting all of the edges of H from G. We will also need the following classes of graphs.

Definition 4 (Complete graph)**.** *A graph G is a complete graph if every pair of vertices is connected by an edge.*

Definition 5 (Threshold graph)**.** *A graph G is a threshold graph if it can be constructed from the empty graph by repeatedly adding either an isolated vertex connected to no other vertices or a dominating vertex connected to all existing vertices. The sequence of vertex insertions used to construct G is called the insertion sequence.*

Note that a complete graph is also a threshold graph, where every vertex in the corresponding insertion sequence is a dominating vertex.

In this project, we will consider access structures that induce graphs. That is, every subset in the access structure has size two. To obtain a graph from the given structure, we define a vertex for every user and we add an edge between two vertices if and only if the pair of users is in the access structure. In this way, the access structures considered by Komargodski et al. are structures that induce complete graphs [8]. In our project, we will consider access structures that induce threshold graphs. To tie everything together, we require the following definition.

Definition 6 ($\mathfrak{G}$-consistent)**.** *Let $\mathfrak{G}$ be an infinite family of graphs and let $\mathcal{A} = \{\mathcal{A}_t\}_{t\in\mathbb{N}}$ be an evolving access structure. We say that $\mathcal{A}$ is $\mathfrak{G}$-consistent if for every $t \in \mathbb{N}$, $\mathcal{A}_t$ induces some graph $G \in \mathfrak{G}$.*

We will use $\mathfrak{K}$ to denote the family of complete graphs and $\mathfrak{T}$ to denote the family of threshold graphs.

3 A Scheme for $\mathfrak{K}$-Consistent Access Structures

In this section, we describe the construction of a secret sharing scheme for a $\mathfrak{K}$-consistent access structure [8]. In particular, we show that size of the share for the t^{th} arriving user is at most $\log t + O(\log\log t)$ bits. The main result used to prove Theorem 2 is given in the following lemma.

Lemma 1 (Lemma 4.2, [8])**.** *Suppose that there exists a secret sharing scheme for a $\mathfrak{K}$-consistent access structure where the t^{th} user receives $\sigma(t)$ bits. Then there exists a secret sharing scheme for the same access structure where the t^{th} user receives $\log t + \sigma(\log t + 1)$ bits.*

Proof. Let Π denote a secret sharing scheme for a $\mathfrak{K}$-consistent access structure where the t^{th} user receives $\sigma(t)$ bits. Using Π, we construct a secret sharing scheme Π' for the same access structure, where the t^{th} user receives $\log t + \sigma(\log t + 1)$ bits. From here on, we refer to each user as a vertex.

Let $s \in \{0, 1\}$ denote the secret to be shared. Each vertex, when it arrives, is assigned to a generation. The generations are growing in size, where the g^{th} generation begins with the arrival of the 2^g-th vertex. Therefore, the t^{th} vertex is in generation $g = \lfloor \log t \rfloor$, while the number of vertices in generation g is $\textsc{size}(g) = 2^g$. At the start of a generation, the dealer will prepare the shares for every vertex in that generation, which is accomplished as follows:

1. Create the shares $u_1^{(g)}, \ldots, u_{\textsc{size}(g)}^{(g)}$ using Shamir's $(2, \textsc{size}(g))$-threshold scheme [9].
2. Generate a single share using Π given the previous shares $\{w^{(i)}\}_{i\in\{0,\ldots g-1\}}$. We denote this new share by $w^{(g)}$.
3. Assign the pair $(u_j^{(g)}, w^{(g)})$ to the j^{th} vertex of generation g.

Correctness and security. Let v and v' denote two distinct vertices. We will argue that the value of s can be reconstructed from their shares. There are two cases to consider. First, suppose that v and v' are part of the same generation

g. In this case, they can reconstruct s using the reconstruction procedure of the $(2, \text{SIZE}(g))$-threshold scheme and their corresponding shares $u_v^{(g)}$ and $u_{v'}^{(g)}$. Otherwise, suppose that v and v' are part of generations g and g'. In that case, they can reconstruct s using the reconstruction procedure of Π and their corresponding shares $w^{(g)}$ and $w^{(g')}$. For the security of the scheme, we will argue that no single vertex can determine any information about s. Since the $(2, \text{SIZE}(g))$-threshold scheme and Π are both secure, and since both parts of the share $(u_v^{(g)}, w^{(g)})$ are generated independently, the shares cannot be used to learn anything about s.

Share size analysis. Let v denote the t^{th} arriving vertex and suppose that v is the j^{th} vertex of generation g.

- Since $u_j^{(g)}$ is generated using Shamir's $(2, \text{SIZE}(g))$-threshold scheme, we know that $|u_j^{(g)}| \leq \log(\text{SIZE}(g)) = \log(2^g) = g = \lfloor \log t \rfloor$.
- When generation g began, a total of g shares were already generated by Π for the previous generations. Therefore, $|w^{(g)}| = \sigma(g+1) = \sigma(\lfloor \log t \rfloor + 1)$.

Hence, v receives at most $\log t + \sigma(\log t + 1)$ bits, as required. □

To prove the main result, it suffices to repetitively apply Lemma 1.

Theorem 2 (Theorem 4.1, [8])**.** *Let $\mathcal{A}$ be a $\mathfrak{K}$-consistent access structure. Then there exists a secret sharing scheme for $\mathcal{A}$ where the t^{th} user receives at most $\log t + O(\log \log t)$ bits.*

Proof. Let $\Pi^{(0)}$ denote the secret sharing scheme described in Theorem 1, where the t^{th} arriving user receives $\sigma^{(0)}(t) = 2^{t-1}$ bits. Applying Lemma 1 gives a new secret sharing scheme $\Pi^{(1)}$ in which the share size of the t^{th} arriving user is bounded by

$$\begin{aligned}\sigma^{(1)}(t) &= \log t + \sigma^{(0)}(\log t + 1) \\ &= \log t + 2^{\log t} \\ &= t + \log t.\end{aligned}$$

Applying Lemma 1 again gives a secret sharing scheme $\Pi^{(2)}$ in which the share size of the t^{th} arriving user is bounded by

$$\begin{aligned}\sigma^{(2)}(t) &= \log t + \sigma^{(1)}(\log t + 1) \\ &= \log t + (\log t + 1) + \log(\log t + 1) \\ &\leq 2 \log t + \log \log t + 2.\end{aligned}$$

Finally, applying Lemma 1 one last time gives a secret sharing scheme $\Pi^{(3)}$ in which the share size of the t^{th} arriving user is bounded by

$$\begin{aligned}\sigma^{(3)}(t) &= \log t + \sigma^{(2)}(\log t + 1) \\ &= \log t + 2 \log(\log t + 1) + \log \log(\log t + 1) + 2 \\ &\leq \log t + 2 \log \log t + \log \log \log t + 4.\end{aligned}$$

This proves the theorem. □

4 A Scheme for $\mathfrak{T}$-Consistent Access Structures

In this section, we explore how the techniques used in the work of Komargodski et al. can be adapted to construct an efficient secret sharing scheme for $\mathfrak{T}$-consistent access structures. We begin by providing a secret sharing scheme for the offline version of our problem, which we then use as a building block in the construction of our online secret sharing scheme.

4.1 An Offline Scheme

Let $G \in \mathfrak{T}$ be an n-vertex threshold graph and let $V = (v_1, \ldots, v_n)$ denote the corresponding insertion sequence. Without loss of generality, assume that v_1 is an isolated vertex and that v_n is a dominating vertex. Next, we divide V into blocks $B_1, \ldots, B_l$ as follows. Let $B_1 = \{v_1, \ldots, v_{i_1}\}$, where i_1 is the largest index such that, for all $j \leq i_1$, v_j is an isolated vertex and v_{i_1+1} is a dominating vertex. For $k > 1$, we define B_k in a similar manner. Let $B_k = \{v_{i_{k-1}+1}, \ldots, v_{i_k}\}$, where i_k is the largest index such that, for all $i_{k-1} + 1 \leq j \leq i_k$:

- If B_{k-1} is a block of isolated vertices, then every v_j is a dominating vertex while v_{i_k+1} is an isolated vertex.
- If B_{k-1} is a block of dominating vertices, then every v_j is an isolated vertex while v_{i_k+1} is a dominating vertex.

Therefore, $B_1, \ldots, B_l$ divides V into maximal alternating blocks of isolated and dominating vertices. Finally, by setting $q = \frac{l}{2}$, we define $B_I = \bigcup_{k=0}^{q-1} B_{2k+1}$ and $B_D = \bigcup_{k=1}^{q} B_{2k}$ to be the sets of isolated and dominating vertices respectively.

In the construction of our secret sharing scheme, we will make use of the following observations.

Lemma 2. *The subgraph $G[B_D]$ is a clique.*

Proof. Let $v \in B_D$ be a dominating vertex. By definition, v is connected to every (dominating) vertex that occurs before it in the insertion sequence for G. Likewise, every dominating vertex that occurs after v in the insertion sequence is, by definition, connected to v. Therefore, $G[B_D]$ is a clique, as required. □

Lemma 3. *Let $G' = G \setminus G[B_D]$. If $q = 1$, then G' is a complete bipartite graph with parts (B_I, B_D).*

Proof. Since $q = 1$ and $l = 2$, the insertion sequence for G consists of exactly two blocks: B_1 and B_2. As a result, we know that $B_I = B_1$ and $B_D = B_2$. Since G' does not contain edges between two dominating vertices, B_I and B_D are both independent sets in G'. Furthermore, all of the dominating vertices occur after the isolated vertices. Hence, each $v \in B_D$ is connected to every vertex in B_I, which means that G' is a complete bipartite graph, as required. □

Lemma 4. *Let $G' = G \setminus G[B_D]$ and let $0 \leq c < r < d \leq q$ be integers, for some $q \geq 2$. If we define $B'_I = \bigcup_{k=c}^{r-1} B_{2k+1}$ and $B'_D = \bigcup_{k=r+1}^{d} B_{2k}$, then the subgraph $G'[B'_I \bigcup B'_D]$ is a complete bipartite graph with parts (B'_I, B'_D).*

Proof. First, observe that G' does not contain edges between two dominating vertices. Therefore, B'_I and B'_D are both independent sets in G'. As a result, to prove that $G'[B'_I \bigcup B'_D]$ is a complete bipartite graph, it suffices to show that any vertex $v \in B'_D$ is connected to every vertex in B'_I. Observe that the earliest block that v can occur in is B_{2r+2}. On the other hand, the vertices of B'_i occur in blocks no later than B_{2r-1}. Therefore, in the insertion sequence for G, v occurs after every vertex in B'_I, and since v is dominating, it must be connected to every vertex in B'_I. □

We will also make use of the following result from class.

Lemma 5. *Let G be a complete bipartite graph on n vertices. Then there exists a secret sharing scheme for G where the size of each share is exactly 1 bit.*

Given a complete bipartite graph with parts (U, V), this can be accomplished by assigning a random bit b to every vertex $u \in U$ and assigning $b \oplus s$ to every vertex $v \in V$.

Our secret sharing scheme will incorporate a decomposition construction by partitioning the edges of G into cliques and complete bipartite graphs. Shares for these partitions can then easily be distributed. We formalize this notion with the following theorem.

Theorem 3. *Let G be an n-vertex threshold graph. Then there exists a secret sharing scheme for G where the size of each share is at most $O(\log n)$ bits.*

Proof. Let $s \in \{0, 1\}$ denote the secret to be shared. First, we partition G into $G[B_D]$ and $G' = G \setminus G[B_D]$. In particular, $G[B_D]$ contains the edges connecting two dominating vertices, while G' contains the edges connecting a dominating vertex to an isolated vertex. We will consider both types of edges separately.

First, we consider the edges contained in $G[B_D]$, which by Lemma 2, forms a clique. Therefore, we can use Shamir's $(2, |B_D|)$-threshold scheme to assign a share u_v to every vertex $v \in B_D$ [9].

For the edges in G', we use a divide-and-conquer approach. If $q = 1$, then by Lemma 3, G' is a complete bipartite graph with parts (B_I, B_D). Therefore, we can apply Lemma 5 to assign 1 bit shares, denoted by $w_{(1,v)}$, to every vertex v in G'. For $q \geq 2$, let $r = \lfloor \frac{q}{2} \rfloor$. Initially, we set $c = 0$ and $d = q$. Then we define B'_I and B'_D as in Lemma 4, which implies that $G'[B'_I \bigcup B'_D]$ is a complete bipartite graph with parts (B'_I, B'_D). As a result, we apply Lemma 5 to assign 1 bit shares, denoted by $w_{(q,v)}$, to every vertex $v \in B'_I \bigcup B'_D$. This leaves the edges in $G'' = G' \setminus G'[B'_I \bigcup B'_D]$. However, observe that G'' has of two connected components. Hence we can assign the remaining shares by recursively applying Lemma 4 on both components, with the correct choice of parameters.

Correctness and security. Let v and v' denote two vertices such that (v, v') is an authorized pair. We will argue that the value of s can be reconstructed from their shares. First, suppose that v and v' are both dominating vertices. In this case, v and v' can use the shares u_v and $u_{v'}$ to reconstruct s. This follows immediately from the correctness of Shamir's $(2, |B_D|)$-threshold scheme. Now suppose that v is an isolated vertex while v' is a dominating vertex. Therefore, v must occur before v' in the corresponding insertion sequence. Hence by Lemma 4, there exists a complete bipartite graph and an index q such that the shares $w_{(q,v)}$ and $w_{(q,v')}$ can be used to reconstruct s.

For the security of the scheme, we will argue that no independent set F of vertices can determine any information about s. Note that we only assign related shares to subgraphs that are cliques or complete bipartite graphs. Therefore, the shares held by the vertices of F are all generated independently from one another, and hence, F cannot learn any information about s.

Share size analysis. Consider the shares assigned for the edges in $G[B_D]$. Since these shares are generated from Shamir's $(2, |B_D|)$-threshold scheme, we know that $|u_v| = O(\log |B_D|) = O(\log n)$. Now consider the shares assigned for the edges in G'. By Lemma 5, each vertex in $G'[B'_I \bigcup B'_D]$ receives exactly 1 bit. By setting $r = \lfloor \frac{q}{2} \rfloor$, both connected components of G'' have roughly the same size. Therefore, we have a logarithmic number of recursive calls before reaching the case when $q = 1$. As a result, each vertex in G' receives at most $O(\log n)$ bits. □

4.2 An Online Scheme

We now extend our results for threshold graphs to the online setting.

Theorem 4. *Let $\mathcal{A}$ be a $\mathfrak{T}$-consistent access structure. Then there exists a secret sharing scheme for $\mathcal{A}$ where the t^{th} user receives at most $O(\log t)$ bits.*

Proof. Let Π denote the secret sharing scheme from Theorem 3, let $\Pi_{\mathfrak{K}}$ denote the secret sharing scheme for $\mathfrak{K}$-consistent access structures from Theorem 2, and let $s \in \{0, 1\}$ denote the secret to be shared. Using Π and $\Pi_{\mathfrak{K}}$, we construct a secret sharing scheme $\Pi_{\mathfrak{T}}$ for $\mathfrak{T}$-consistent access structures, where the t^{th} user receives $O(\log t)$ bits. From here on, we refer to each user as a vertex.

Each vertex, when it arrives, is assigned to a generation. The generations are growing in size, where the g^{th} generation begins with the arrival of the 2^g-th vertex. Therefore, the t^{th} vertex is in generation $g = \lfloor \log t \rfloor$, while the number of vertices in generation g is $\text{SIZE}(g) = 2^g$. At the start of a generation, the dealer will prepare the shares for every vertex in that generation, which is accomplished as follows:

1. Assuming an insertion sequence of alternating isolated and dominating vertices, create the shares $y_1^{(g)}, \ldots, y_{\text{SIZE}(g)}^{(g)}$ using Π. We note that not all of these shares may be used.
2. Generate a single share using $\Pi_{\mathfrak{K}}$ given the previous shares $\{w^{(i)}\}_{i \in \{0, \ldots g-1\}}$. We denote this new share by $w^{(g)}$.
3. Generate a random bit $b^{(g)} \in \{0, 1\}$ and maintain a counter $l \leftarrow 1$.

When assigning shares to an arriving vertex, we have several cases to consider. Let v denote the arriving vertex.

- Suppose that v is an isolated vertex. If the previous vertex was also an isolated vertex, then v receives the shares $(y_l^{(g)}, b^{(g)})$. Otherwise, if the previous vertex was a dominating vertex, then v receives the shares $(y_{l+1}^{(g)}, b^{(g)})$, and l is incremented to $l \leftarrow l + 1$. In the case where there is no previous vertex in the current generation, v receives the shares $(y_1^{(g)}, b^{(g)})$.

- Suppose that v is a dominating vertex and let $S = \{b^{(i)} \oplus s\}_{i \in \{0,\ldots,g-1\}}$. If the previous vertex was also a dominating vertex, then v receives the shares $(y_l^{(g)}, w^{(g)}, S)$. Otherwise, if the previous vertex was an isolated vertex, then v receives the shares $(y_{l+1}^{(g)}, w^{(g)}, S)$, and l is incremented to $l \leftarrow l + 1$. In the case where there is no previous vertex in the current generation, v receives the shares $(y_1^{(g)}, w^{(g)}, S)$.

Correctness and security. Let v and v' denote two vertices such that (v, v') is an authorized pair. We will argue that the value of s can be reconstructed from their shares. First, suppose that v and v' are part of the same generation g. In this case, they can reconstruct s using the reconstruction procedure of Π and their corresponding shares $y_v^{(g)}$ and $y_{v'}^{(g)}$. Now, suppose that v and v' are part of generations g and g'. If v and v' are both dominating vertices, then they can reconstruct s using the reconstruction procedure of $\Pi_{\mathfrak{K}}$ and their corresponding shares $w^{(g)}$ and $w^{(g')}$. This follows from the fact that the set of dominating vertices in a threshold graph forms a clique. Now suppose that v is an isolated vertex while v' is a dominating vertex. Since v has to occur before v' in the insertion sequence, we know that $g < g'$. Therefore, they can use the shares $b^{(g)}$ and $b^{(g)} \oplus s$ respectively to reconstruct s.

For the security of the scheme, we will argue that no independent set F of vertices can determine any information about s. Since F is an unauthorized set of vertices, it contains at most one dominating vertex. First, if F contains only isolated vertices, then F cannot learn any information about s since the corresponding shares are computed independently of s. If F does contain a dominating vertex v having generation g, then the isolated vertices in F occur after v in the insertion sequence. In particular, if v' is an isolated vertex in F from generation g', then $g' \geq g$. Therefore, v does not have any shares of the form $b^{(g')} \oplus s$, and hence, F cannot learn any information about s.

Share size analysis. Let v denote the t^{th} arriving vertex and suppose that v is the j^{th} vertex of generation g.

- Since $y_j^{(g)}$ is generated using Π, we know that $|y_j^{(g)}| \leq O(\log(\textsc{size}(g))) = O(\log t)$.
- When generation g began, a total of g shares were already generated by $\Pi_{\mathfrak{K}}$ for the previous generations. Therefore, $|w^{(g)}| = O(\log t)$. Moreover, S contains a total of g bits, and thus $|S| = O(\log t)$.

Hence, v receives at most $O(\log t)$ bits, as required. □

5 Conclusions and Future Work

In this project, we investigated secret sharing over particular families of graphs in both the offline and online settings. Inspired by the work of Komargodski et al. [8], we presented a secret sharing scheme for the family of threshold graphs. In particular, by exploiting the properties of threshold graphs, we showed that there exists a secret sharing scheme for any n-vertex threshold graph in which the shares are logarithmic in size. In an unpublished proof by Kilian and Nisan,

later published by Cascudo Pueyo et al. [5], it was shown that, for a $(2, n)$-threshold access structure, $O(\log(n))$ bits are required to share a 1-bit secret among n users. Since complete graphs are also threshold graphs, it implies that our result is also asymptotically optimal. We also extended our results on threshold graphs from the offline setting to the online setting, while still maintaining a logarithmic bound on the share size.

Throughout the course of this project, we encountered a number of avenues for potential future work. First, we focused on the family of threshold graphs. This raises the question of whether we can attain a logarithmic bound on the share size for any other families of graphs. For example, interval graphs appear to be a natural extension of our work, since threshold graphs are also interval graphs. Second, if other families of graphs can indeed attain such bounds, then adapting these secret sharing schemes to the online setting would also be of great interest. Finally, another interesting avenue could be to see if one can obtain similar types of results for other forms of secret sharing, such as verifiable secret sharing, robust secret sharing, or visual secret sharing.

References

[1] Noura Al Ebri, Joonsang Baek, and Chan Yeob Yeun. Study on secret sharing schemes (SSS) and their applications. In *International Conference for Internet Technology and Secured Transactions (ICITST 2011)*, pages 40–45. IEEE, 2011.

[2] Amos Beimel. Secret-sharing schemes: A survey. In *Coding and cryptology*, pages 11–46. Springer, 2011.

[3] George R. Blakley. Safeguarding cryptographic keys. In *AFIPS Conference Proceedings: 1979 National Computer Conference*, volume 48, pages 313–317. AFIPS Press, 1979.

[4] Carlo Blundo, Alfredo De Santis, Roberto De Simone, and Ugo Vaccaro. Tight bounds on the information rate of secret sharing schemes. *Designs, Codes and Cryptography*, 11(2):107–122, 1997.

[5] Ignacio Cascudo Pueyo, Ronald Cramer, and Chaoping Xing. Bounds on the threshold gap in secret sharing and its applications. *IEEE Transactions on Information Theory*, 59(9):5600–5612, 2013.

[6] László Csirmaz and Gábor Tardos. On-line secret sharing. *Designs, Codes and Cryptography*, 63(1):127–147, 2012.

[7] Marten van Dijk. On the information rate of perfect secret sharing schemes. *Design, Codes and Cryptography*, 6:143–169, 1995.

[8] Ilan Komargodski, Moni Naor, and Eylon Yogev. How to share a secret, infinitely, 2016.

[9] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[10] Douglas R. Stinson. Decomposition constructions for secret sharing schemes. 40:118–125, 1994.

[11] Douglas R. Stinson and Ruizhong Wei. Bibliography on secret sharing schemes. http://cacr.uwaterloo.ca/∼dstinson/ssbib.html.