Distributed Secret Dissemination Across a Network

Nihar B. Shah, K. V. Rashmi, and Kannan Ramchandran, Fellow, IEEE

Abstract—Shamir’s $(n, k)$ threshold secret sharing is an important component of several cryptographic protocols, such as those for secure multiparty-computation and key management. These protocols typically assume the presence of direct communication links from the dealer to all participants, in which case the dealer can directly pass the shares of the secret to each participant. In this paper, we consider the problem of secret sharing when the dealer does not have direct communication links to all the participants, and instead, the dealer and the participants form a general network. Existing methods are based on separate secure message transmissions from the dealer to each participant, requiring considerable coordination and communication in the network. We present a distributed algorithm for disseminating shares over a network, which we call the “SNEAK algorithm,” requiring each node to know only the identities of its one-hop neighbors. While SNEAK imposes a stronger condition on the network by requiring the dealer to be what we call $k$-connected rather than $k$-connected as required by the existing solutions, we show that in addition to being distributed, it achieves significant reduction in the amount of communication and the randomness required. We also derive information-theoretic lower bounds on the amount of communication for secret sharing over networks, which may be of independent interest.

Index Terms—Secret sharing, distributed algorithm, networks.

I. INTRODUCTION

SHAMIR’S classical $(n, k)$ secret sharing scheme [2] is an essential ingredient of several cryptographic protocols. The scheme considers $(n + 1)$ entities: a dealer and $n$ honest-but-curious participants. The dealer possesses a secret $s$ and wishes to pass functions (called shares) of this secret to the $n$ participants, such that the following two properties are satisfied:

- $k$-secret-recovery: the shares of any $k$ participants suffice to recover the secret,
- $(k - 1)$-collusion-resistance: the aggregate data gathered by any $(k - 1)$ participants reveals no knowledge (in the information-theoretic sense) about the secret.

Several cryptographic protocols in the literature require execution of one or more instances of secret sharing among all the participants. These include protocols for secure multi-party-computation [3]–[6], secure key management [7], [8], general Byzantine agreement between all participants [3], [9]–[11], proactive secret sharing [12], [13], and secure archival storage [14]. For instance, under the celebrated Ben-Or-Goldwasser-Wigderson (BGW) protocol [3] for secure-multiparty function computation among $n$ participants, the initialization step requires $n$ instances of secret sharing with all participants, and every multiplication operation requires $2n$ additional instances.

Most protocols including those listed above assume that the dealer has direct secure communication links to every participant. In this case, the dealer can compute the shares as per Shamir’s scheme [2] and directly pass the shares to the respective participants. Such a setting is depicted in Fig. 1(a) for the parameters $(n = 6, k = 2)$. In many situations, however, the dealer may not have direct communication links with every participant; instead, the dealer and the participants may form a general network. Fig. 1(b) depicts such a scenario. The network is described by a graph $G$ with $(n + 1)$ nodes. These $(n + 1)$ nodes comprise the dealer and the $n$ participants. An edge in this graph implies a communication link between its two end-points, while the absence of an edge denotes the non-existence of any direct communication link. We make the standard assumption that the communication links are secure. We will say that a participant is ‘directly connected to the dealer’ if there exists an edge from the dealer to that participant.

Under a general network $G$, all communication between the dealer and any participant who is not directly connected to it, must pass through other participants in the network. This poses
the challenge of not leaking any additional information to any participant while disseminating the shares over the network.

Existing methods use separate secure message transmissions (SMT) from the dealer to each participant across the network [15]. Under such a solution, in order to communicate the designated share to a participant, the dealer treats this share as a secret, employs Shamir's scheme to compute k shares of this secret, and communicates these k shares to the participant via k node-disjoint paths. (This solution is described in more detail in Section III-B.) This solution requires a significant coordination in the network in setting up the node-disjoint paths from the dealer to every participant. It also incurs a high communication cost since the dealer needs to transmit shares across the network separately for every participant.

In this paper, we present a distributed and communication-efficient algorithm for secret share dissemination across a network, which we call SNEAK. We analyze the performance of SNEAK and compare it to the state-of-the-art, i.e., the SMT-based solution. In addition to being distributed, SNEAK provides significant gains in terms of the amount of communication and the randomness required. On the other hand, while the SMT-based solution requires the graph to satisfy a certain “k-connected-dealer” condition (which is in fact necessary for the feasibility of secret sharing), SNEAK imposes a stronger condition on the network, which we call the k-propagating-dealer condition, that will be formalized later in the paper.

We now present a toy example illustrating the existing SMT-based solution and the SNEAK algorithm.

Example 1: Consider the network depicted in Fig. 1(b). Here \( n = 6 \) and \( k = 2 \), with the alphabet of operation as the finite field \( \mathbb{F}_7 \). Under Shamir's scheme of encoding the secret \( s \), the share \( t_i \) (\( 1 \leq i \leq 6 \)) for participant \( i \) is

\[
t_i = s + \alpha r_i,
\]

where \( r \) is a value chosen by the dealer uniformly at random from the alphabet \( \mathbb{F}_7 \). While the dealer can directly pass the shares \( t_1 \) and \( t_2 \) to participants 1 and 2 respectively, the difficulty arises in communicating shares to the remaining participants with whom the dealer does not have direct communication links. For instance, if the dealer tries to pass share \( t_3 \) to participant 3 by simply communicating \( t_3 \) along the path 'dealer \( \rightarrow \) 1 \( \rightarrow \) 3', then participant 1 gains access to two shares, \( t_1 \) and \( t_3 \). Using these two shares, participant 1 can recover the secret \( s \), thus violating the \((k-1)\)-collusion resistance requirement.

The SMT-based solution is illustrated in the sequence of steps depicted in Fig. 2(a). In order to pass the share \( t_3 \) to participant 3, the dealer chooses another random value \( r_3 \), passes \( t_3 + r_3 \) along the path 'dealer \( \rightarrow \) 1 \( \rightarrow \) 3', and \( r_3 \) along the path 'dealer \( \rightarrow \) 2 \( \rightarrow \) 3'. Now, participant 3 can recover \( t_3 \), and no participant gains any additional information about the secret \( s \) in this process. In a similar manner, the dealer can communicate \( t_i \) (\( 4 \leq i \leq 6 \)) to participant \( i \) through \( k - 2 \) node-disjoint paths as shown in the figure.

Observe that the SMT-based solution transmits data across several hops in the network in every step, but the data transmitted in any step is never used in subsequent steps of the protocol. Thus, in order to design an efficient and distributed algorithm, one may wish to propagate data in a manner that allows its subsequent reuse downstream. This is the key idea underlying SNEAK, which is illustrated in the sequence of steps in Fig. 2(b). Here, the dealer first draws two values \( r \) and \( r_n \) uniformly at random from \( \mathbb{F}_7 \). The dealer then passes the two values \( s + r \) and \( r + r_n \) to node 1, and the two values \( s + 2r \) and \( r + 2r_n \) to node 2. Upon receiving its data, each node passes a certain linear combination of its received data to each of its downstream neighbors. For instance, node 1 passes \( s + r + 3(r + r_n) \) to node 3, which can equivalently be written as \( s + 3r + (r + 3r_n) \). Node 2 passes \( s + 2r + j(r + 2r_n) = (s + jr + 2r + jr_n) \) to node \( j \in \{3,4\} \) respectively. Node 3 can thus recover the two values \( s + 3r \) and \( r + 3r_n \) from the data it receives. Similarly, as shown in the sequence of steps depicted in Fig. 2(b), every node \( i \in \{1, \ldots, 6\} \) can recover its requisite share \( s + \alpha r_i \), along with a random counterpart \( r + \alpha r_n \) which is used to disseminate shares further downstream. Note that in Fig. 2(b), the expression written above an edge is the linear combination that is transmitted, and the corresponding expression written in the parenthesis below that edge is a simple rewriting of the data transmitted.

We can see that SNEAK is completely distributed and requires knowledge of only one-hop neighbors as opposed to the SMT-based solution which requires the knowledge of the global topology in order to set-up communication over node-disjoint paths. Also, SNEAK requires communication of only 12 values, as opposed to 24 under the SMT-based solution. The number of random values generated under SNEAK is only 2, whereas the SMT-based solution requires generation of 5 random values.

The remainder of the paper is organized as follows. Section II provides a formal description of the system model and summarizes the results of this paper. Section III reviews related literature. Section IV describes SNEAK. Section V presents a comparative analysis of the SMT-based solution, SNEAK, and the lower bounds in terms of the communication and the randomness requirements. Section VI concludes the paper.

II. SYSTEM MODEL AND SUMMARY OF RESULTS

A. Secret Sharing Across a General Network

The dealer possesses a secret \( s \) that is drawn from some alphabet \( \mathcal{A} \), and wishes to pass shares of this secret to \( n \) participants. The dealer and the participants form a communication network, denoted by graph \( G \). The graph \( G \) has \( n + 1 \) nodes comprising the dealer and the \( n \) participants, and an edge in the graph denotes a secure and private communication link between the two end-points. Thus, at times, we will also refer to a participant as a node of the graph. We will also use the terms ‘network’ and ‘graph’ interchangeably, and ‘link’ and ‘edge’ interchangeably. The problem is to design a protocol which will allow the dealer to pass shares (of the secret) to the \( n \) participants, meeting the requirements of \((k-1)\)-collision-resistance and \( k\)-secret-recovery (described in Section I). The participants are assumed to be honest-but-curious, i.e., they follow the protocol correctly, but may store any accessible data in order to
gain information about the secret. The edges in the graph $G$ are allowed to be directed or undirected: a directed edge implies existence of only one way communication link, while an undirected edge implies direct communication links both ways. The parameters $n$ and $k$ are assumed to satisfy $n > k > 1$, since $n \leq k - 1$ prohibits the secret from ever being recovered, while $k - 1$ degenerates the problem into a trivial case in which no secrecy is required.

We now discuss a condition that the graph $G$ must necessarily satisfy for any algorithm to successfully perform secret sharing on it.

**Definition 1 ($m$-connected-dealer):** A graph with $(n + 1)$ nodes (the dealer and $n$ participants) satisfies the $m$-connected-dealer condition for a positive integer $m$ if each of the $n$ participants in the graph either has an incoming edge directly from the dealer or has at least $m$ node-disjoint paths from the dealer to itself.

**Proposition 1 (Necessary Condition [15]):** For any graph $G$, a necessary condition for any algorithm to perform $[n, k]$ secret sharing is that $G$ satisfies the $k$-connected-dealer condition.

Thus no algorithm can operate successfully on all network topologies, and must require the graph $G$ to obey at least the $k$-connected-dealer condition.

**B. Class of Networks Considered**

We saw above in Proposition 1 that feasibility of secret sharing on a graph $G$ requires $G$ to satisfy the $k$-connected-dealer condition. SNEAK requires the communication network $G$ to satisfy a stronger condition, which we term the $k$-propagating-dealer condition.

**Definition 2 ($m$-propagating-dealer):** A graph with $(n + 1)$ nodes (the dealer and $n$ participants) satisfies the $m$-propagating-dealer condition for a positive integer $m$ if there exists an ordering of the $n$ participants in the graph such that every node either has an incoming edge directly from the dealer or has incoming edges from at least $m$ nodes preceding it in the ordering.

As an illustration of the $m$-propagating-dealer condition, consider the network of Example 1 (Fig. 1(b)). This network satisfies the 2-propagating-dealer condition, with the ordering 1, 2,
3, 4, 5, 6 (observe that this is also the order in which the participants receive their shares under SNEAK in Fig. 2(b)).

We enumerate below examples of a few classes of graphs that satisfy the \(m\)-propagating-dealer condition. Fig. 3 depicts three examples of graphs that satisfy the 3-propagating dealer condition. These examples can be generalized to the following classes of graphs. The first three classes consider undirected graphs and permit any of the nodes to be the dealer.

(a) Layered networks (Fig. 3(a)): Each layer contains at least \(m\) nodes, and each node is connected to all nodes in the neighboring layers. An ordering that satisfies the \(m\)-propagating-dealer condition is the ordering of the nodes with respect to the distance (number of hops) from the dealer.

(b) Networks with a backbone (Fig. 3(b)): A subset of nodes, termed the ‘backbone’, form a densely connected subgraph (that satisfies the \(m\)-propagating-dealer condition), and every node outside the backbone is connected directly to at least \(m\) nodes in the backbone. An ordering that satisfies the \(m\)-propagating-dealer condition is: the ordering under the backbone subgraph, followed by all remaining nodes not in the backbone in any order.

(c) \(m\)-connected one-dimensional geometric networks (Fig. 3(c)): A one-dimensional geometric network is formed by arranging the nodes (in an arbitrary fashion) along a line, and connecting a pair of nodes by an edge if the distance (number of hops) between them is smaller than a fixed threshold. A one-dimensional geometric network that satisfies the \(m\)-connected-dealer condition also satisfies the \(m\)-propagating dealer condition. In this case, an ordering that satisfies the \(m\)-propagating-dealer condition is the arrangement of the nodes in an ascending order of their Euclidean distance from the dealer.

(d) Directed acyclic graphs: Any directed acyclic graph that satisfies the \(m\)-connected-dealer condition also satisfies the \(m\)-propagating-dealer condition. The root must be the dealer, and any topological ordering of the DAG suffices as the requisite node-ordering. E.g., the directed version of the graph of Fig. 1(b), with the edges directed from the left to the right would fall in this class.

We note that while the necessity of the \(k\)-propagating-dealer condition under SNEAK requires the existence of some such ordering of the nodes, the execution of the algorithm is completely distributed and oblivious to the actual ordering. We also note that while SNEAK requires the graph to satisfy the \(k\)-propagating dealer condition, it is robust to the network topology, i.e., it satisfies the \([k − 1]\)-collusion-resistance property over any arbitrary network topology irrespective of whether it satisfies \(k\)-propagating dealer condition or not. This robustness property is important since the secrecy of the secret is paramount, and in practice, the structure of the network may not be known beforehand. Moreover, under a dynamic network, the graph structure may also vary with time (even during the execution of the algorithm), thus further motivating robustness.

Apart from the parameters \(n\) and \(k\), an additional parameter “\(d\)” is associated to SNEAK. We saw earlier that the \(d\)-propagating-dealer condition is satisfied if and only if all nodes receive data from at least \(d\) other nodes.

C. Summary of Results

This paper presents a distributed algorithm, called SNEAK, that takes parameters \(n\), \(k\), and \(d \geq k\) as input, and enables a dealer to disseminate shares of a secret to \(n\) participants forming a general network \(G\), such that the properties of (i) \(k\)-secret-recovery (when \(G\) satisfies the \(d\)-propagating-dealer condition), and (ii) \([k − 1]\)-collusion-resistance (irrespective of the network topology) are satisfied. The algorithm is completely distributed with each node required to know only the identities of its neighbors, is efficient with respect to the communication cost and the amount of randomness required, and is optimal with respect to the size of the shares.

For any \(n, k\) and any graph \(G\) with \(n + 1\) nodes, we derive and compare (a) information-theoretic lower bounds on the total communication under any algorithm, (b) lower bounds on the communication under the SMT-based solution, and (c) communication under SNEAK. Using these results, we establish the communication efficiency of SNEAK. SNEAK has particularly useful implications on bounded degree graphs\(^2\): For networks with bounded degree and satisfying the \(k\)-propagating-dealer condition, the communication under SNEAK is within a constant (multiplicative) factor of the information-theoretic lower bound, and is \(\Theta(n)\) in the worst-case. In contrast, the communication under the SMT-based solution is lower bounded.

\(^2\)Bounded degree graphs have their maximum degree upper bounded by a constant that is independent of \(n\).
as $\Omega(n \log n)$, and is $\Omega(n^2)$ in the worst case. Moreover, the amount of randomness required under SNEAK is $\Theta(1)$, while that required under the SMT-based solution is $\Theta(n)$.

The extended version [16, Appendix C] of this paper on arXiv contains three heuristic techniques to extend SNEAK algorithm to networks where the $k$-propagating-dealer condition is not satisfied. [16, Appendix C] also contains an extension of the algorithm to handle active adversaries, and an extension to support two-threshold secret sharing.

D. Notational Conventions

A vector will be treated as a column vector by default. The transpose of a vector or a matrix will be denoted by a superscript $T$. For any integer $\ell \geq 1$, $[\ell]$ will represent the set $\{1, \ldots, \ell\}$. For any participant $j$ ($1 \leq j \leq n$), the set of its neighbors will be denoted by $\mathcal{N}(j)$. In case of a directed graph, $\mathcal{N}(j)$ will denote the set of nodes to which node $j$ has an outgoing edge. The dealer will be denoted as $D$, and the set of neighbors of the dealer by $\mathcal{N}(D)$. We will say that a node $j$ is directly connected to the dealer if $j \in \mathcal{N}(D)$.

III. RELATED LITERATURE

A. Shamir's Secret Sharing Protocol

We first give a brief review of Shamir's secret sharing protocol [2]. To this end, we assume that the dealer has a direct (secure) communication link with every participant (as in the example in Fig. 1(a)).

Assume that the secret $s$ is drawn from some finite field $\mathbb{F}_q$ of size $q (> n)$. The dealer chooses $(k - 1)$ values $\{r_i\}_{i=1}^{k-1}$ independently and uniformly at random from $\mathbb{F}_q$. Define a $k$-length vector $\mathbf{m}$ as

$$\mathbf{m} = \begin{bmatrix} s & r_1 & r_2 & \cdots & r_{k-1} \end{bmatrix}^T.$$  \hfill (1)

Next, define a set of $n$ vectors $\{\mathbf{\psi}_i\}_{i=1}^n$, each of length $k$, as

$$\mathbf{\psi}_i = \begin{bmatrix} 1 & i & i^2 & \cdots & i^{k-1} \end{bmatrix}^T.$$  \hfill (2)

The share $t_i$ of participant $i$ is simply the inner product

$$t_i = \mathbf{\psi}_i^T \mathbf{m}.$$  \hfill (3)

One can verify that the vectors $\{\mathbf{\psi}_i\}_{i \in \mathbb{Z}}$ are all linearly independent. It follows that for any set $\mathcal{I} \subseteq [n]$ of cardinality $k$, the secret $s$ can be recovered from the set of values $\{\mathbf{\psi}_i^T \mathbf{m}\}_{i \in \mathcal{I}}$. Furthermore, it can also be verified that for any set $\mathcal{I}' \subseteq [n]$ of cardinality smaller than $k$, the set $\{\mathbf{\psi}_i^T \mathbf{m}\}_{i \in \mathcal{I}'}$ provides no knowledge about $s$.

Under the assumption that the dealer has direct communication links with every participant $i \in [n]$, the dealer can simply pass $t_i$ to participant $i$.

Remark 2: To see the Shamir's secret sharing scheme in the conventional polynomial representation [2], note that each share $t_i$ can be seen as an evaluation of the $(k - 1)$ degree polynomial with the secret $s$ as its constant term and the remaining $(k - 1)$ coefficients chosen uniformly at random and independent of the secret $s$. Thus, $k$ evaluations are necessary and sufficient to recover the polynomial and hence the secret. This provides the $k$-secret-recovery and $(k - 1)$-collusion-resistance properties.

This completes the description of Shamir's secret sharing protocol. There are numerous other extensions and generalizations of Shamir's secret sharing protocol in the literature, and the reader is referred to [17], [18] and references therein for more details. Of particular historical interest is [19] which connects Shamir secret sharing to Reed-Solomon codes.

We now describe the SMT-based solution that addresses the situation when the dealer and participants form a general communication network.

B. The SMT-Based Solution

This section describes a scheme for secret sharing over a general communication network employing secure message transmissions (SMT) from the dealer to each participant [15].

Ref. 2(a) in Example 1 is an example of such a solution. Under this solution, the dealer first encodes the secret $s$ into $n$ shares $\{t_i\}_{i=1}^n$ using Shamir's secret sharing scheme (3). The size $q$ of the underlying finite field $\mathbb{F}_q$ is assumed to be greater than $n$. To every node $\ell$ directly connected to the dealer, the dealer directly passes its share $t_\ell$. To disseminate shares to the remaining nodes, the dealer performs the following actions. Let $\ell$ now denote a node that is not connected directly to the dealer. The dealer applies Shamir's secret sharing scheme treating $t_\ell$ as a secret, and computes $k$ shares $\{u_{\ell,j}\}_{j=1}^k$, as

$$u_{\ell,j} = \begin{bmatrix} 1 & j & j^2 & \cdots & j^{k-1} \end{bmatrix}^T,$$ \hfill (4)

where the values $\{r_{\ell,1}, \ldots, r_{\ell,k-1}\}$ are chosen independently and uniformly at random from $\mathbb{F}_q$. The dealer then finds $k$ node-disjoint paths (from itself) to node $\ell$, and passes $u_{\ell,j}$ along the $j^{th}$ path ($1 \leq j \leq k$). At the end of these transmissions, node $\ell$ receives $\{u_{\ell,j}\}_{j=1}^k$ from which it can recover its share $t_\ell$. Moreover, since each of the random values are independent, no participant can obtain any information about any other participant's share, or any additional information about the secret $s$. This process is repeated once for every node that is not connected directly to the dealer.

The solution described above requires transmission of data across $k$ node-disjoint paths once separately for every node that is not connected directly to the dealer. Thus this solution is not distributed, and furthermore is not efficient in terms of the communication and the randomness costs.

We note that the communication efficiency of this solution can be improved if more than $k$ node-disjoint paths are available, by employing two-threshold secret sharing [20] over these node-disjoint paths. Under this setting, for any $i \in [n]$, let us suppose there are $w_i$ ($\geq k$) node-disjoint paths from the dealer to node $i$. Here the dealer chooses a value $\omega \in \{w_i, \ldots, w_1\}$, encodes the share of participant $i$ into $w$ chunks in a manner [20] that satisfies $\omega$-secret-recovery and $(\omega - 1)$-collusion-resistance, and passes these chunks via the $\omega$ shortest node-disjoint paths.
to participant $i$. The dealer chooses $w$ such that the amount of communication in transmitting the share to participant $i$ is minimized; the special case of choosing $w = k$ for all participants is equivalent to the procedure described in the previous paragraphs. The analysis and comparisons performed subsequently in Section V will consider this more efficient two-threshold version of the SMT-based solution.

C. Network Coding and Distributed Storage

The problem of secret sharing over a general communication network can be cast as a specific instance of a network coding problem [21], requiring security from eavesdropping on the nodes. This casting can be performed in the following manner. The dealer is the source node, and the secret $s$ is the message. The network graph in the network coding problem is identical to that in the secret sharing problem, but with a set of $\binom{n}{k}$ additional nodes that act as the sinks. Each of the $\binom{n}{k}$ sinks is connected to a distinct subset of $k$ participants, and has one directed link of infinite capacity coming in from each of the corresponding $k$ participants. Each sink must recover the entire message: this requirement corresponds to the condition of $k$-secret-recovery. To satisfy the $(k - 1)$-collusion-resistance property, a compromise of up to $(k - 1)$ arbitrary nodes (excluding the source and the sinks) to a passive eavesdropper should reveal no information about the message. In this manner, the secret sharing problem is equivalent to a network coding problem requiring secrecy from an eavesdropper that can gain access to a subset of the nodes. However, with respect to this setting, very little appears to be known in the network coding literature.

To the best of our knowledge, the literature on secure network coding (e.g., [22]–[25]) considers the setting where the eavesdropper gains access to a subset of the links. The problem of node compromise is typically treated as a case of link compromise by allowing the eavesdropper to gain access to all links that are incident upon the compromised nodes. In [24], [25], authors consider the setting wherein a collection of subsets of the links is specified, and an eavesdropper may gain access to precisely one of these subsets. However, the scheme requires the knowledge of the network topology, and is computationally expensive. Moreover, the scheme requires the graph to satisfy a particular condition, which is almost always violated in our problem setting. Communication-efficient algorithms to secure a network from an eavesdropper having access to a bounded number of links are provided in [22], [23]. Given the network topology, the actions to be performed at the nodes can be derived in a computationally efficient manner. However, these algorithms communicate a message of size equal to the difference between the largest message that can be sent without secrecy requirements and the bound on number of compromised links. Under our problem setting, this difference is generally zero or smaller (e.g., the difference is $-2$ in the network of Fig. 1(b)), thus rendering these algorithms inapplicable.

The SNEAK algorithm constructed in the present paper thus turns out to be an instance of a secure network coding problem that admits an explicit solution that is distributed, communication efficient, and provides deterministic (probability 1) guarantees. Furthermore, the solution handles the case of nodal eavesdropping, about which very little appears to be known in the literature.

SNEAK is based on a variant of the Product-Matrix codes [26] which were originally constructed for distributed storage systems. These codes possess useful properties that SNEAK exploits in the present context. The product-matrix codes are a practical realization of the concept of ‘regenerating codes’ [27] proposed for distributed storage. To date, apart from the MDS codes of [28], these are the only known constructions of regenerating codes that are scalable (i.e., other parameters of the system impose no constraints on the total number of nodes in the system). It turns out that this scalability property is an essential ingredient for our problem. Secure versions of the product-matrix codes were constructed in [29], [30]. The reader familiar with the literature on regenerating codes for distributed storage may recognize later in the paper that we employ the minimum-bandwidth (MBR) version, and not the minimum-storage (MSR) version, of the product-matrix codes [26]. We make this choice to guarantee secrecy from honest-but-curious participants, who may store all the data that they receive, a characteristic of the MBR point on the storage-bandwidth tradeoff [27], [31].

D. Computational Secret Sharing

In this paper, we consider the information-theoretic notion of secrecy where the adversary is allowed to possess unbounded computational power. Another popular notion is that of ‘computational’ secrecy [32], [33] that considers adversaries bounded in their computational power. The basic idea here is to encrypt the secret using a key and distribute the encrypted secret among the agents by encoding it using a Reed-Solomon code. Shamir’s secret sharing is used only for creating shares for the key used in the encryption process. Each agent’s share constitutes a part of the encrypted and encoded secret and the Shamir’s share of the encryption key. Under the computational notion of secrecy, in the networked setting, the dealer can communicate the shares to the agents using public-key-cryptography.

IV. THE SNEAK ALGORITHM

Consider a network $\mathcal{G}$ that obeys the $d$-propagating-dealer condition for some parameter $d \geq k$. The secret $s$ belongs to the alphabet $\mathcal{A}$, and we assume that $\mathcal{A} = \mathbb{F}^{d-k+1}_q$, for some arbitrary field size $q > n$. Thus we can equivalently denote the secret as a vector $s = [s_1 s_2 \ldots s_d]_{k+1}^{T}$ with each element of this vector belonging to the finite field $\mathbb{F}_q$.

A. Initial Setting up by the Dealer

The dealer first constructs an $[n \times d]$ Vandermonde matrix $\Psi$, with the $i^{th}$ ($1 \leq i \leq n$) row of $\Psi$ being

\[
\psi_i = [1 \ i \ i^2 \ \cdots \ i^{d-1}]^T.
\]

The vector $\psi_i$ is termed the encoding vector of node $i$. The dealer first constructs an $[n \times d]$ Vandermonde matrix $\Psi$, with the $i^{th}$ ($1 \leq i \leq n$) row of $\Psi$ being

\[
\psi_i = [1 \ i \ i^2 \ \cdots \ i^{d-1}]^T.
\]
Next, the dealer constructs a \((d \times d)\) symmetric matrix \(M\) comprising the secret \(s\) and a collection of randomly generated values as

\[
M = \begin{bmatrix}
  s_A & r_a^T & s_B^T \\
  r_a & R_b & R_c \\
  s_B & R_c & 0 \\
  1 & k-1 & \delta-k \\
\end{bmatrix},
\]

(6)

where the depicted sub-matrices of \(M\) are

- \(s_A\) is a scalar,
- \(s_B = [s_1 \cdots s_{d-k}]^T\) is a vector of length \((d - k)\),
- \(r_a\) is a vector of length \((k - 1)\) with its entries populated by random values,
- \(R_b\) is a \(((k-1) \times (k-1))\) symmetric matrix with its \(k(k-1)/2\) distinct entries populated by random values,
- \(R_c\) is a \(((d-k) \times (k-1))\) matrix with its \((k-1)(d-k)\) entries populated by random values.

These random values are all picked independently and uniformly from \(\mathcal{F}_q\). One can compute that the total number of random values \(R\) in matrix \(M\) is

\[
R = (k-1)d - \frac{k-1}{2}.
\]

(7)

The entire secret is contained in the components \(s_A\) and \(s_B\) as

\[
s^T = [s_1 \cdots s_{d-k+1}] = [s_B^T s_A].
\]

Observe that the structure of \(M\) as described in (6), and the symmetry of matrix \(R_b\) make the matrix \(M\) symmetric.

The share \(t_j\) for participant \(j\) \((1 \leq j \leq n)\) is a vector of length \((d - k + 1)\):

\[
t_j^T = \psi_j^T \begin{bmatrix}
  s_A & s_B^T \\
  r_a & R_b \\
  s_B & R_c \\
\end{bmatrix}.
\]

(8)

We will show subsequently in Theorem 3 that any \(k\) of these shares suffice to recover the entire secret.

Remark 3: To see these shares in the conventional polynomial representation of Shamir’s secret sharing scheme, recall that the vector \(\psi_j^T\) is drawn from a Vandermonde matrix. Thus each entry of \(t_j\) in (8) can be seen as the evaluation of a polynomial at value \(j\). Thus there is one polynomial for each secret value \(s_i\) \((1 \leq i \leq d - k + 1)\), with the corresponding secret symbol as its constant term.

Example 2: Consider the setting of Example 1 (Fig. 2(b)) where \(n = 6, k = 2, d = 2\). Here

\[
M = \begin{bmatrix}
  s & r & r_a \\
  r & r & r_a \\
\end{bmatrix},
\]

and for every \(j\) \((1 \leq j \leq n)\), \(\psi_j^T = [s_j \ r \ r_a]\) and the share for participant \(j\) is \(t_j^T = [s_j + jr]\).

B. Communication Across the Network

Algorithm 1 describes the communication protocol to securely transmit the shares \(\{t_j\}_{j=1}^n\) to the \(n\) participants.

Algorithm 1 Communication Protocol

**Dealer:** For every \(j \in \mathcal{N}(D)\), compute and pass the \(d\)-length vector \(\psi_j^T M\) to participant \(j\).

**Participant \(\ell \in \mathcal{N}(D)\):** Wait until receipt of data \(\psi_j^T M\) from the dealer. Upon receipt, perform the following actions. For every \(j \in \mathcal{N}(\ell)\), compute the inner product of the data \(\psi_j^T M\) with the encoding vector \(\psi_j\) of participant \(j\). Transmit the resulting value \(\psi_j^T M\psi_j\) to participant \(j\).

**Participant \(\ell \notin \mathcal{N}(D)\):** Wait until receipt of data each from any \(d\) neighbors, and then perform the following actions (if more than \(d\) neighbors pass data, retain data from some arbitrary \(d\) of these nodes). Denote this set of \(d\) neighbors as \(\{i_1, \ldots, i_d\}\), and the data received from them as \(\{\sigma_1, \ldots, \sigma_d\}\) respectively. Compute the vector

\[
v = \begin{bmatrix}
  \psi_{i_1}^T \\
  \vdots \\
  \psi_{i_d}^T \\
\end{bmatrix}^{-1} \begin{bmatrix}
  \sigma_1 \\
  \vdots \\
  \sigma_d \\
\end{bmatrix}.
\]

For every neighbor \(i \in \mathcal{N}(\ell)\) from whom you did not receive data, compute and pass the inner product \(v^T \psi_i\) to participant \(i\).

Remark 4: In order to reduce the amount of communication, one would like to ensure that a participant receives data from no more than \(d\) of its neighbors. This can be ensured via a simple handshaking protocol between neighbors, wherein a participant who is ready to transmit data to its neighbors, queries the neighbors for the requirement of the respective transmissions, prior to actually sending the data.

Example 3: Consider the setting of Example 1 (Fig. 2(b)), wherein \(n = 6, k = 2, d = 2\). The values of \(M\), \(\psi_j\) and \(t_j\) \((1 \leq j \leq n)\) under this setting are specified in Example 2. For the given network, we have \(\mathcal{N}(D) = \{1, 2\}\). As per Algorithm 1, participant \(j \in \{1, 2\}\) receives \(\psi_j^T M = \{s + jr\}\) directly from the dealer. Now let us focus on participant 3. Since participant 3 is a neighbor to participants 1 and 2, following Algorithm 1, participant \(j \in \{1, 2\}\) passes \(\psi_j^T M\psi_{i_1} = (s + jr + 3(r + jr_{a})\) to participant 3. Participant 3 thus receives the two values \(\sigma_1 = (s + r) + 3(r + ra)\) and \(\sigma_2 = (s + 2r) + 3(r + 2ra)\) from neighbors \(i_1 = 1\) and \(i_2 = 2\). Using the fact that \(\psi_{i_1}^T = [1 \ 1]\) and \(\psi_{i_2}^T = [1 \ 2]\), it computes

\[
v = \begin{bmatrix}
  1 & 1 \\
  1 & 2 \\
\end{bmatrix}^{-1} \begin{bmatrix}
  (s + r) + 3(r + ra) \\
  (s + 2r) + 3(r + 2ra) \\
\end{bmatrix} = \begin{bmatrix}
  s + 3r \\
  r + 3ra \\
\end{bmatrix}.
\]

A similar procedure is executed at participants 4, 5 and 6 as well.

C. Correctness of the Algorithm

The following theorems show that each participant indeed receives its intended data (8), and the algorithm satisfies the properties of \(k\)-secret-recovery, and \((k-1)\)-collusion-resistance, and that the \((k-1)\)-collusion-resistance property is also robust to network topology.
Theorem 2 (Successful Share Dissemination): Under SNEAK, every participant $\ell \in [n]$ can recover $\psi^T \mathcal{M}$, and hence obtains its intended share $s_{\ell}^T$.

**Proof:** Recall that the graph satisfies the $d$-dealer propagation condition. Let us assume without loss of generality that the ordering of nodes satisfying this condition is $1, \ldots, n$. Then the first $d$ nodes in this ordering must be connected directly to the dealer. The proof proceeds via induction. The induction hypothesis is as follows: every participant $\ell$ can recover the data $\psi^T \mathcal{M}$, and if $\ell$ passes any data to any other node $j \in N(\ell)$ then this data is precisely the value $s_j^T \mathcal{M}$.

Consider the base case of node 1. Since this node is directly connected to the dealer, it receives the data $\psi^T \mathcal{M}$ from the dealer, according to the communication protocol in Algorithm 1. Moreover, following the communication protocol, it passes $\psi^T \mathcal{M}$ to each neighbor $j \in N(1)$. Let us now assume that the hypothesis holds true for the first $(\ell - 1)$ nodes in the ordering. If node $\ell$ is directly connected to the dealer, then the hypothesis is satisfied for this node by an argument identical to the case of node 1. Suppose $\ell$ is not directly connected to the dealer. It follows that node $\ell$ must be connected to at least $d$ other nodes preceding it in the ordering, and furthermore, must receive data from at least $d$ of these nodes (say, nodes $\{j_1, \ldots, j_d\} \subseteq \{\ell - 1\}$). By our hypothesis, these $d$ nodes pass the $d$ values $\{\psi_{j_1}^T \mathcal{M}, \ldots, \psi_{j_d}^T \mathcal{M}\}$. The algorithm running at node $\ell$ thus operates on the input

$$T_i^T = \psi_i^T \begin{bmatrix} s_{A} & s_{B}^T & r_a & \mathbf{0} \end{bmatrix}.$$

By construction, the matrix in (9) with $\psi_1^T, \ldots, \psi_d^T$ as its rows is a $(d \times d)$ Vandermonde matrix, and is hence invertible. Thus, the computation of $\psi$ as described in Algorithm 1 can be performed efficiently using standard Reed-Solomon decoding algorithms [19], [34], [35]. It further follows that $\psi = \mathcal{M} \psi^T$, and since $\mathcal{M}$ is a symmetric matrix, we get $\psi^T = \mathcal{M}^T = \psi^T \mathcal{M}$. Finally, the data passed by node $\ell$ to any other node $i \in N(\ell)$, according to the protocol, is $\psi_i^T \psi^T = \psi_i^T \mathcal{M}$. This proves the hypothesis for node $\ell$.

Due to the specific structure (6) of $\mathcal{M}$, the desired share $T_i$ is a subset of the elements of the vector $\psi_i^T \mathcal{M}$. Thus, every participant obtains its intended share.

**Theorem 3 ($k$-Secret-Recovery):** Any $k$ participants suffice to recover the secret.

**Proof:** Let $I \subseteq [n]$ denote the set of the $k$ participants attempting to recover the secret. Let $\Psi_I$ be a $(k \times d)$ matrix with its $k$ rows comprising $\{\psi_i^T\}_{i \in I}$. Further, let $\tilde{\Psi}_I$ denote the $(k \times k)$ submatrix of $\Psi_I$ comprising the first $k$ columns of $\Psi_I$.

Consider the last $(d - k)$ columns of this data, i.e.,

$$\Psi_I \begin{bmatrix} s_A & s_B^T & r_a & \mathbf{0} \end{bmatrix} = \tilde{\Psi}_I \begin{bmatrix} s_A^T \left( \psi_i^T \mathcal{M} \right) \end{bmatrix}.$$  

Since $\Psi_I$ is a $(k \times d)$ Vandermonde matrix, it follows that $\tilde{\Psi}_I$ is a $(k \times k)$ Vandermonde matrix. Thus, $\tilde{\Psi}$ is invertible. This allows for the decoding of $s_B$ (via algorithms [19], [34], [35] identical to those for decoding under Shamir's original secret sharing scheme). It now remains to recover $s_A$, and to this end consider the first column of the data, i.e.,

$$\Psi_I \begin{bmatrix} s_A \left( \psi_i^T \mathcal{M} \right) \end{bmatrix} = \tilde{\Psi}_I \begin{bmatrix} s_A^T \left( \psi_i^T \mathcal{M} \right) \end{bmatrix}.$$  

Since $\tilde{\Psi}_I$ is invertible, the value of $s_A$ can be decoded from this data.

**Theorem 4 ((k - 1)-Collusion-Resistance and Robustness):** Any set of $(k - 1)$ or fewer colluding participants can gain no information about the secret. This guarantee is robust to network topology, i.e., holds for arbitrary graphs that may or may not satisfy the $d$-propagating dealer condition.

The proof of Theorem 4 is provided in the extended version [16, Appendix A].

V. COMPLEXITY ANALYSIS AND BOUNDS

In this section, we provide an analysis and comparison of the complexity of SNEAK, the SMT-based solution, and lower bounds for any secret-sharing scheme.

Recall that $D$ denotes the dealer and $N(D)$ denotes the set of neighbors of the dealer (or, in case of directed edges, the set of nodes with edges coming in from the dealer). Let $N(D)$ denote the size of this set. We assume without loss of generality that the units of data are normalized with one unit defined to be equal to the size of the secret. We will use the notation $\Gamma(\cdot)$ to denote amount of communication, and $\rho(\cdot)$ to denote amount of randomness required. The proofs of all the results presented in this section are available in the extended version [16, Appendix B].

Before moving on to the details of the communication and randomness costs, we note that the size of each share $\mathcal{S}$ under
SNEAK equals the size of the secret, which is known to be the minimum possible [Theorem 1, [36]].

A. Communication and Randomness Required

Theorem 5: For an \((n, k)\) secret sharing problem on any graph \(G\) with \((n + 1)\) nodes satisfying the \(d\)-propagating-dealer condition for some (known) \(d\), SNEAK:

(a) requires every node to receive \(d/(d - k + 1)\) units of data, and hence requires a total communication of

\[
\Gamma_{\text{SNEAK}}(G) = n \frac{d}{d - k + 1} \tag{10}
\]

units of data, and

(b) requires an amount of randomness given by

\[
\rho_{\text{SNEAK}}(G) = \frac{(k - 1)(2d - k)}{2(d - k + 1)} \tag{11}
\]

Theorem 6: For an \((n, k)\) secret sharing problem on any graph \(G\) with \((n + 1)\) nodes, the SMT-based solution

(a) requires a total communication of

\[
\Gamma_{\text{SMT}}(G) = |N(D)| + \sum_{i \in N(D)} \min_{w \geq k} \left[ \frac{w}{w - k + 1} \times \ell_w(D \rightarrow i) \right] \tag{12}
\]

units of data, where \(\ell_w(D \rightarrow i)\) is the average of the path lengths of the \(w\) shortest node-disjoint paths from the dealer to node \(i\) (with \(\ell_w(D \rightarrow i) = \infty\) if there do not exist \(w\) node-disjoint paths from \(D\) to \(i\)), and

(b) requires an amount of randomness lower bounded by

\[
\rho_{\text{SMT}}(G) \geq k - 1 + \sum_{i \in N(D)} \frac{k - 1}{w_{\text{max}}(i) - k + 1} \tag{13}
\]

\[
\geq \left( n - |N(D)| \right) \frac{k - 1}{|N(D)| - (k - 1)} \tag{14}
\]

where \(w_{\text{max}}(i)\) is the maximum number of node-disjoint paths from the dealer to node \(i\).

Remark 5: The lower bound on the randomness requirement of the SMT-based solution provided in (13) is achievable, however, at the cost of increased communication (the communication will be higher than that specified in (12)), wherein the optimal \(w\) chosen for every term inside the summation would be replaced by \(w_{\text{max}}(i)\).

From the two theorems stated above, we can see that SNEAK provides the greatest gains over the SMT-based solution when the distance in the graph between the dealer and the participants is large on an average.

The following theorem provides information-theoretic lower bounds applicable to any scheme, which serve as a benchmark to compare SNEAK and the SMT-based solution.

Theorem 7: For an \((n, k)\) secret sharing problem on any graph \(G\) with \((n + 1)\) nodes under any algorithm, \(a\) any node \(\ell \in [n]\) must receive at least

\[
\Gamma_{\text{every}}(\ell) \geq \begin{cases} 
\frac{\deg(\ell)}{\deg(\ell) - k + 1} & \text{if } \ell \notin N(D) \text{ and } \deg(\ell) \geq k \\
1 & \text{if } \ell \in N(D) \\
\infty & \text{if } \ell \notin N(D) \text{ and } \deg(\ell) < k
\end{cases} \tag{15}
\]

units of data, where \(\deg(\ell)\) denotes the number of incoming edges at node \(\ell\). Further, this bound is the best possible, given only the identities of the neighbors of node \(\ell\). Hence the total communication under any algorithm is lower bounded by

\[
\Gamma_{\text{any}}(G) \geq |N(D)| + \sum_{i \in N(D)} \frac{\deg(i)}{\deg(i) - k + 1} \tag{16}
\]

\[
g \geq n. \tag{17}
\]

(b) the amount of randomness required under any algorithm is lower bounded [37] by

\[
\rho_{\text{any}}(G) \geq k - 1. \tag{18}
\]

Remark 6: The lower bound (17) can be deduced alternatively from the fact that the share of each participant must be at least the size of the secret [Theorem 1, [36]].

B. Implications for the Case of Bounded Degree Graphs

The SNEAK algorithm has particularly striking implications on secret sharing on graphs whose maximum degree is bounded (independent of \(n\)), for example, in the graph depicted in Fig. 3(a) with the nodes partitioned into ‘layers’ of three nodes each.

As discussed earlier, for any secret sharing algorithm to succeed, the graph must satisfy the \(d\)-connected-dealer condition for some \(d\). Now, if a graph satisfies the \(d\)-connected-dealer (or the stronger \(d\)-propagating-dealer) condition, the value of \(d\) must necessarily be upper bounded by the maximum degree of the graph. It follows that the parameters \(k\) and \(d\) are upper bounded by the maximum degree of the graph, and are therefore independent of \(n\). We first present a series of lemmas and finally present the main results for this setting in Theorem 12 and Theorem 13.

Lemma 8: For any given \((n, k)\), and for any given \(d\) (\(k \leq d < n\)), consider any undirected graph with \((n + 1)\) nodes such that (a) every non-neighbor of \(D\) has a degree of \(d\), and (b) the graph satisfies the \(d\)-propagating-dealer condition. Under SNEAK, the amount of data received by any node \(\ell \notin N(D)\) meets the lower bound (15) with equality. Furthermore, under SNEAK, the amount of data received by any node \(\ell \in N(D)\) is independent of \(n\).

Lemma 9: For any given \((n, k)\), and for any given \(d\) (\(k < d \leq n\)), there exists a class of graphs such that the communication on graphs in this class is lower bounded by

\[
\Gamma_{\text{any}}(G) \geq n \frac{d}{d - k + 1} - (k - 1) \frac{d}{d - k + 1}.
\]

Thus, for the class of graphs considered in Lemma 10, the amount of communication (10) under SNEAK is within a constant (additive) factor of the lower bound.

The next two lemmas quantify the performance of the SMT-based solution. Lemma 10 is more general than for bounded degree graphs considered in this section: the lemma also applies to graphs whose maximum degree may grow with \(n\).

Lemma 10: On graphs with the maximum outgoing degree \(O((\log n)^{(1/2) - \epsilon})\) for some \(\epsilon > 0\), the SMT-based solution requires super-linear communication. Furthermore, for graphs
with degree bounded by a constant independent of \( n \), the SMT-based solution requires \( \Omega(n \log n) \) communication.

**Lemma 11:** For any given \((n, k)\), and for any given \( d \) \((k \leq d \leq n)\), there exists a class of graphs with \((n + 1)\) nodes such that each graph in this class satisfies the \( d \)-propagating-dealer condition, and \((n, k)\) secret sharing on any graph \( \mathcal{G} \) in this class using the SMT-based solution requires communication lower bounded by
\[
I_{\text{SMT}}(\mathcal{G}) \geq \frac{n(n + 1)}{4d}.
\]

Finally, we compare the SMT-based solution, SNEAK and the lower bounds.

**Theorem 12:** Consider graphs that satisfy the \( k \)-propagating-dealer condition and have their maximum degree upper bounded by a constant. The amount of communication under SNEAK is within a constant (multiplicative) factor of the information-theoretic lower bound, and is \( \Theta(n) \) in the worst-case. On the other hand, the amount of communication under SMT-based solution is \( \Omega(n \log n) \), and is \( \Omega(n^2) \) in the worst case.

**Theorem 13:** Consider graphs that satisfy the \( k \)-propagating-dealer condition and have their maximum degree upper bounded by a constant. The amount of randomness required under the SMT-based solution is \( \Theta(n) \), whereas the amount of randomness required under SNEAK is \( \Theta(1) \).

C. Computation Cost

For every node, both the SMT-based solution and SNEAK require the encoding and decoding of a Reed-Solomon code for which several efficient, polynomial time algorithms are known \([34, 35]\). We note that the SMT-based solution entails an additional computational overhead of finding node-disjoint paths from the dealer to every node in the graph; this is not required under SNEAK due to its distributed nature.

VI. DISCUSSION

Many cryptographic protocols in the literature require execution of one or more instances of secret sharing among all the participants. Most of these protocols assume that the dealer has direct communication links to all the participants. This paper presents SNEAK, a distributed and efficient algorithm for secret sharing in a setting where the dealer and the participants form a general communication network. While SNEAK requires the network to satisfy the stronger \( k \)-propagating-dealer condition as opposed to the \( k \)-connected-dealer condition required by the existing methods, it provides significant reduction in the amount of communication and randomness required, in addition to being distributed. The paper also presents information-theoretic lower bounds on the communication for secret sharing in general networks, which may be of independent interest.

The upper and lower bounds on the communication for secret sharing presented in this paper are shown to be tight for certain classes of networks. However, obtaining (tight) bounds on the communication for general networks still remains open. SNEAK requires the network to satisfy the \( k \)-propagating dealer condition, and only heuristics are known \([16, \text{Appendix C}]\) to address networks that satisfy the \( k \)-connected-dealer but not the \( k \)-propagating-dealer condition. Designing more efficient algorithms and quantifying the communication requirements for such settings remain open.

Finally, the results of this paper turn out to be an instance of a network coding problem that interestingly admits an explicit solution that is distributed, communication efficient, and provides probability 1 guarantees. Moreover, the solution handles the case of nodal eavesdropping, about which very little appears to be known in the literature. It remains to be seen if any of the ideas from this specific case of secure network coding carry over to more general network coding problems.

ACKNOWLEDGMENT

The authors would like to thank Prakash Ishwar, Piyush Srivastava, Anindya De, and Matthieu Finiasz for helpful discussions.

REFERENCES

[1] N. B. Shah, K. V. Rashmi, and K. Ramchandran, “Secure network coding for distributed secret sharing with low communication cost,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, Jul. 2013.
[2] A. Shamir, “How to share a secret,” *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.
[3] M. Ben-Or, S. Goldwasser, and A. Wigderson, “Completeness theorems for non-cryptographic fault-tolerant distributed computation,” in *Proc. 20th Annu. ACM Symp. Theory Comput.*, 1988, pp. 1–10.
[4] D. Chaum, C. Crépeau, and I. Damgard, “Multiparty unconditionally secure protocols,” in *Proc. 20th Annu. ACM Symp. Theory Comput.*, 1988, pp. 11–19.
[5] R. Cramer, M. Franklin, B. Schoenmakers, and M. Yung, “Multi-authority secret-ballot elections with linear work,” in *Proc. Adv. Cryptology (EUROCRYPT)*, 1996, pp. 72–83.
[6] I. Damgård, Y. Ishai, and M. Kroigaard, “Perfectly secure multiparty computation and the computational overhead of cryptography,” *Proc. Adv. Cryptology (EUROCRYPT)*, pp. 445–465, 2010.
[7] T. Pedersen, “A threshold cryptosystem without a trusted party,” in *Proc. Adv. Cryptology (EUROCRYPT)*, 1991, pp. 522–526.
[8] M. Marsh and F. Schneider, “CODEX: A robust and secure secret distribution system,” *IEEE Trans. Depend. Secure Comput.*, vol. 1, no. 1, pp. 34–47, Jan.-Mar. 2004.
[9] M. Rabin, “Randomized Byzantine generals,” in *Proc. 24th Annu. Symp. Foundat. Comput. Sci.*, 1983, pp. 403–409.
[10] P. Feldman and S. Micali, “Optimal algorithms for Byzantine agreement,” in *Proc. 30th Annu. ACM Symp. Theory Comput.*, 1988, pp. 148–161.
[11] I. Ingemarsson and G. Simmons, “A protocol to set up shared secret schemes without the assistance of a mutually trusted party,” in *Proc. Adv. Cryptol. (EUROCRYPT)*, 1990, pp. 266–282.
[12] R. Ostrovsky and M. Yung, “How to withstand mobile virus attacks,” in *Proc. 10th Annu. ACM Symp. Principles Distrib. Comput.*, 1991, pp. 51–59.
[13] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, “Proactive secret sharing or: How to cope with perpetual leakage,” in *Proc. Adv. Cryptol. (EUROCRYPT)*, 1995, pp. 339–352.
[14] M. Storer, K. Greenan, E. Miller, and K. Voruganti, “Posthards: A secure, recoverable, long-term archival storage system,” *ACM Trans. Storage*, vol. 5, no. 2, p. 2, 2009.
[15] D. Dolev, C. Dwork, O. Waarts, and M. Yung, “Perfectly secret message transmission,” *J. ACM*, vol. 40, no. 1, pp. 17–47, 1993.
[16] N. B. Shah, K. V. Rashmi, and K. Ramchandran, “Distributed secret dissemination across a network,” *arXiv: 1207.0120 [Online]. Available: arXiv:1207.0120*
[17] D. R. Stinson, “An explication of secret sharing schemes,” in *Adv. Cryptology. (EUROCRYPT)*, 1988, pp. 11–19.
[18] A. Beimel, “Secret-sharing schemes: A survey,” in *Coding and Cryptology. New York, NY, USA: Springer*, 2011, pp. 11–46.
[19] R. McEliece and D. Sarwate, “On sharing secrets and Reed-Solomon codes,” *Commun. ACM*, vol. 24, no. 9, pp. 583–584, 1981.

AUTHOR INFORMATION

*Corresponding Author*: N. B. Shah

**Address**: Department of Computer Science, School of Engineering and Science, Australian National University, Canberra, ACT 0200, Australia, and Department of Computer Science, University of California, Santa Barbara, CA 93106, USA

**Email**: n.shah@anu.edu.au

**Phone**: +61 2 6125 4520

**Fax**: +61 2 6125 4529

**Website**: http://www.cs.anu.edu.au/"nshah/
[20] M. Franklin and M. Yung, “Communication complexity of secure computation,” in Proc. 24th Annu. ACM Symp. Theory Comput., 1992, pp. 699–710.

[21] R. Ahlswede, N. Cai, S. Li, and R. Yeung, “Network information flow,” IEEE Trans. Inf. Theory, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[22] H. Yao, D. Silva, S. Jaggi, and M. Langberg, “Network codes resilient to jamming and eavesdropping,” in Proc. IEEE Int. Symp. Netw. Coding (NetCod), 2010, pp. 1–6.

[23] C. Ngai and R. Yeung, “Secure error-correcting (sec) network codes,” in Proc. IEEE Netw. Coding, Theory, Applicat. (NetCod), 2009, pp. 98–103.

[24] N. Cai and R. Yeung, “Secure network coding on a wiretap network,” IEEE Trans. Inf. Theory, vol. 56, no. 4, pp. 1567–1578, Jul. 2010.

[25] J. Feldman, T. Malkin, C. Stein, and R. Servedio, “On the capacity of secure network coding,” in Proc. 42nd Annu. Allerton Conf. Commun., Control, Comput., 2004.

[26] K. V. Rashmi, N. B. Shah, and P. V. Kumar, “Optimal exact-rege nerating codes for the MSR and MBR points via a product-matrix construction,” IEEE Trans. Inf. Theory, vol. 57, no. 8, pp. 5227–5239, Aug. 2011.

[27] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, “Network coding for distributed storage systems,” IEEE Trans. Inf. Theory, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.

[28] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, “Explicit construction of optimal exact regenerating codes for distributed storage,” in Proc. 47th Annu. Allerton Conf. Commun., Control, Comput., Urbana-Champaign, IL, USA, Sep. 2009, pp. 1243–1249.

[29] N. B. Shah, K. V. Rashmi, and P. V. Kumar, “Information-theoretically secure regenerating codes for distributed storage,” in Proc. Globecom, Houston, TX, USA, Dec. 2011.

[30] K. V. Rashmi, N. B. Shah, K. Ramchandran, and P. Kumar, “Regenerating codes for errors and erasures in distributed storage,” in Proc. Int. Symp. Inf. Theory, Jul. 2012, pp. 1202–1206.

[31] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, “Distributed storage codes with repair-by-transfer and non-achievability of interior points on the storage-bandwidth tradeoff,” IEEE Trans. Inf. Theory, vol. 58, no. 3, pp. 1837–1852, Mar. 2012.

[32] H. Krawczyk, “Secret sharing made short,” in Proc. Adv. in Cryptol. -CRYPTO’93, 1994, pp. 136–146.

[33] P. Rogaway and M. Bellare, “Robust computational secret sharing and a unified account of classical secret-sharing goals,” in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, pp. 172–184.

[34] R. Blahut, Theory and Practice of Error Control Codes. Boston, MA, USA: Addison-Wesley, 1983.

[35] I. Gohberg and V. Olshevsky, “Fast algorithms with preprocessing for matrix-vector multiplication problems,” J. Complexity, vol. 10, no. 4, pp. 411–427, 1994.

[36] E. D. Karin, J. Greene, and M. E. Hellman, “On secret sharing systems,” IEEE Trans. Inf. Theory, vol. 29, no. 1, pp. 35–41, Jan. 1983.

[37] C. Blundo, A. De Santis, and U. Vaccaro, “Randomness in distribution protocols,” Inf. Comput., vol. 131, no. 2, pp. 111–139, 1996.

Nihar B. Shah is a Ph.D. candidate in the Department of Electrical Engineering and Computer Science at UC Berkeley. He is a recipient of the Facebook Fellowship 2012–13, the Microsoft Research Ph.D. Fellowship 2013–15, and the IEEE Data Storage Best Paper and Best Student Paper awards for the years 2011/2012. His research interests lie in statistical learning theory, information theory, game theory and crowdsourcing.

K. V. Rashmi is a Ph.D. candidate in the Department of Electrical Engineering and Computer Science at UC Berkeley. He has been since 1999. He was on the faculty at UIUC from 1993 to 1999, and with AT&T Bell Labs from 1984 to 1990. Prof. Rashmi is a Fellow of the IEEE. He has published extensively in his field, holds over a dozen patents, and has received several awards for his research and teaching including an IEEE Information Theory Society and Communication Society Joint Best Paper award for 2012, an IEEE Communication Society Data Storage Best Paper award in 2010, two Best Paper awards from the IEEE Signal Processing Society in 1993 and 1999, an Okawa Foundation Prize for outstanding research at Berkeley in 2001, and an Outstanding Teaching Award at Berkeley in 2009, and a Hank Magnuski Scholar award at Illinois in 1998.

His research interests are broadly in the area of distributed systems theory and algorithms intersecting the fields of signal processing, communications, coding theory, and networking. His current research interests are on distributed storage systems, sparse signal processing and codes on graphs, collaborative content distribution, interactive synchronization, privacy, and crowdsourcing.

Kannan Ramchandran (Ph.D.: Columbia University, 1993) is a Professor of Electrical Engineering and Computer Science at UC Berkeley, where he has been since 1994. He was on the faculty at UIUC from 1993 to 1999, and at AT&T Bell Labs from 1984 to 1990. Prof. Ramchandran is a Fellow of the IEEE.

He has published extensively in his field, holds over a dozen patents, and has received several awards for his research and teaching including an IEEE Information Theory Society and Communication Society Joint Best Paper award for 2012, an IEEE Communication Society Data Storage Best Paper award in 2010, two Best Paper awards from the IEEE Signal Processing Society in 1993 and 1999, an Okawa Foundation Prize for outstanding research at Berkeley in 2001, and an Outstanding Teaching Award at Berkeley in 2009, and a Hank Magnuski Scholar award at Illinois in 1998.

His research interests are broadly in the area of distributed systems theory and algorithms intersecting the fields of signal processing, communications, coding theory, and networking. His current research interests are on distributed storage systems, sparse signal processing and codes on graphs, collaborative content distribution, interactive synchronization, privacy, and crowdsourcing.