Secure Network Coding for Distributed Secret Sharing with Low Communication Cost

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. 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 every participant. In this paper, we consider the problem of secret sharing when the dealer does not have direct communication links to all participants, and instead, they form a general network. We present an algorithm for secret sharing over networks that satisfy what we call the k-propagating-dealer condition. The algorithm is communication-efficient, distributed and deterministic. Interestingly, the solution constitutes an instance of a network coding problem admitting a distributed and deterministic solution, and furthermore, handles the case of nodal-eavesdropping, about which very little appears to be known in the literature.**

In the second part of the paper, we derive information-theoretic lower bounds on the communication complexity of secret sharing over any network, which may also be of independent interest. We show that for networks satisfying the k-propagating-dealer condition, the communication complexity of our algorithm is $\Theta(n)$, and furthermore, is always within a constant factor of the lower bound. We also show that, in contrast, existing solutions in the literature entail a communication-complexity that is super-linear for a wide class of networks, and is $\Theta(n^2)$ in the worst case. Our algorithm thus allows for efficient generalization of several cryptographic protocols to a large class of networks.

I. Introduction

Shamir's classical (n,k) secret sharing scheme [1] is an essential ingredient of several cryptographic protocols. The scheme considers a set of $(n+1)$ entities: a *dealer* and n *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 properties are satisfied:

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

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 multiparty-computation [2], secure key management [3], and secure archival storage [4]. For instance, under the celebrated Ben-Or-Goldwasser-Wigderson (BGW) protocol [2] for secure-multiparty function computation, the initialization step requires n instances of secret sharing and every multiplication operation requires $2n$ additional instances.

Most protocols including those listed above assume that the dealer has direct communication links to every participant. In this case, the dealer can compute the shares as per Shamir's scheme [1] and directly pass the shares to the respective participants. In several situations, the dealer may not have direct communication links with every participant; instead, the dealer and the participants may form a general network, e.g., as in Fig. 1. The network is described by a graph $\mathcal{G}$ with $(n+1)$ nodes. These $(n+1)$ nodes comprise the dealer and the n participants. An edge represents a secure communication link between its two end-points. We shall say a participant is 'directly connected to the dealer' if there exists an edge from the dealer to that participant. We shall use the terms 'edge' or 'link' to refer to a communication link.

Under a general network, all communication between the dealer and a participant who is not directly connected to it, must pass through other participants in the network. This poses the challenge of secret sharing over a network without leaking any additional information to any participant.

Previous solutions: The current practice is to perform separate secure transmissions across the network [5] from the dealer to each participant. Under this solution, the dealer first encodes s into n shares $\{t_\ell\}_{\ell=1}^{n}$ using Shamir's secret sharing scheme. To every node ℓ directly connected to the dealer, the dealer directly passes the share t_ℓ. To disseminate shares to the remaining nodes, the dealer performs the following actions, once separately for each remaining node $\ell' \in \{1,\ldots,n\}$. The dealer applies Shamir's secret sharing treating $t_{\ell'}$ as a secret. The resultant k shares are then passed to node ℓ' via k node-disjoint paths in the graph. Node ℓ' can decode its desired share $t_{\ell'}$, while the remaining nodes in the network do not obtain any information about s or $t_{\ell'}$.

Such a solution incurs a high communication cost, since the dealer needs to transmit shares across the network separately to every participant. Moreover, the requirement of setting up k node-disjoint paths to every participant requires knowledge of the global topology, and also requires significant coordination in the network. [1] Due to lack of a specific name, in the sequel, we shall refer this solution simply as the "previous solution".

In this paper, we consider the problem of efficient dissemination of the shares of a secret to participants forming a network. We provide an algorithm that performs this task over a wide class of networks in a communication-efficient and distributed manner. Our algorithm provides significant gains over the previous solution, with a communication complexity that is within a constant factor from the information-theoretic lower bounds, that are also derived in this paper.

As a secure network coding problem: The problem of secret share dissemination can also be cast as a specific instance of

The authors are with the Dept. of EECS, University of California, Berkeley. E-mail: {nihar, rashmikv, kannanr}@eecs.berkeley.edu

N. B. Shah was supported by a Berkeley Fellowship and K. V. Rashmi was supported by a Facebook Fellowship. This work was also supported in part by AFOSR grant FA9550-10-1-0567 and in part by NSF grant CCF-0964018.

[1] The communication efficiency of this solution can be improved if more than k node-disjoint paths are available, by employing *two-threshold secret sharing* over these node-disjoint paths. The analysis and comparisons performed subsequently in Section IV consider this version of the solution.

978-1-4799-0446-4/13/$31.00 ©2013 IEEE

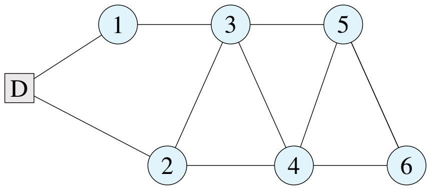

Fig. 1: A network formed by the dealer (D) and participants (1 to 6), as considered in Example 1.

a secure network coding problem, by connecting sinks to each of the $\binom{n}{k}$ subsets of k participants. It requires secrecy from an eavesdropper that can gain access to subsets of the *nodes* (in particular, to any subset of $(k-1)$ participants). However, to the best of our knowledge, only the setting where the eavesdropper can access subsets of *links* is well understood in the literature. [6], [7] consider the setting where a collection of subsets of the links is specified, and an eavesdropper may gain access to precisely one of these subsets. This work addresses the problem of node-compromise by treating it as a case of link-compromise by allowing the eavesdropper to gain access to all links that are incident upon the compromised nodes. However, this scheme requires the network to satisfy a certain condition, which is almost always violated in our problem. Moreover, the scheme is not explicit and requires the size of the finite field to be exponential in n. Communication-efficient algorithms to secure a network from an eavesdropper having access to a bounded number of links are provided in [8], [9]. These algorithms communicate a message of size equal to the difference between the largest message that can be sent in the absence of secrecy requirements and the bound on number of compromised links. In our problem, this difference is generally 0 or smaller (e.g., the difference is -2 in the network of Fig. 1), thus making these algorithms inapplicable here.

The algorithms currently found in the network coding literature, even for the setting where there are no secrecy requirements, are either random (thus not guaranteed) [10], or deterministic but centralized [11]. On the other hand, our algorithm is both distributed and deterministic (i.e., is successful with probability 1).

Illustrative example: The following toy example illustrates the previous solution and our new algorithm.

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

$$t_i = s + ir$$

where r is a value chosen by the dealer uniformly at random from $\mathbb{F}_7$. While the dealer can directly pass 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 3 by passing t_3 along the path 'D $\rightarrow$ 1 $\rightarrow$ 3', then 1 gains access to two shares, t_1 and t_3. Using these two shares, 1 can recover s, thus violating the $(k-1)$-collusion resistance requirement.

The solution previously proposed in the literature is to perform separate secure transmissions from the dealer to each participant [5]. In the example of Fig. 1, 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 'D $\rightarrow$ 1 $\rightarrow$ 3', and r_3 along 'D $\rightarrow$ 2 $\rightarrow$ 3'. Now, participant 3 can recover its share t_3, and no participant gains any additional information about s or t_3 in this process. In a similar manner, the dealer communicates t_i $(4 \leq i \leq 6)$ to participant i by passing $(t_i + r_i)$ and r_i through $k=2$ node-disjoint paths. Although this solution guarantees successful share dissemination, it is communication inefficient, and requires knowledge of the global topology, as well as considerable coordination in the network.

Observe that the solution described above transmits data across several hops in the network in every step, which is however, never used subsequently in the protocol. Thus, in order to design efficient algorithms, one may wish to propagate data in a manner that allows its subsequent reuse downstream, thus reducing the overall communication in the network, as typical of solutions based on network coding. Under our algorithm, the dealer first draws two values r and r_a uniformly at random from $\mathbb{F}_7$. The dealer then passes the two values $(s+r)$ and $(r+r_a)$ to node 1, and the two values $(s+2r)$ and $(r+2r_a)$ to node 2. Now, node 1 passes $(s+r)+j(r+r_a)$ to its neighbouring node $j=3$, and this expression can equivalently be written as $(s+jr)+(r+jr_a)$. Similarly, node 2 passes $(s+2r)+j(r+2r_a)$ $(=(s+jr)+2(r+jr_a))$ to its neighbours $j \in \{3,4\}$. Node 3 thus receives $(s+3r)+(r+3r_a)$ and $(s+3r)+2(r+3r_a)$ from which it recovers the two values $(s+3r)$ and $(r+3r_a)$. Node 3 now passes $(s+3r)+j(r+3r_a)$ $(=(s+jr)+3(r+jr_a))$ to its other neighbours $j \in \{4,5\}$. Node 4 thus receives $(s+4r)+2(r+4r_a)$ and $(s+4r)+3(r+4r_a)$ from nodes 2 and 3 respectively, from which it recovers $(s+4r)$ and $(r+4r_a)$. In general, in this network, every node $i \in \{1,\ldots,6\}$ recovers $(r+ir_a)$ and $(s+ir)$, and can thus recover its requisite share $(s+ir)$, along with a random counterpart $(r+ir_a)$ which is used to disseminate shares further downstream.

One can see that the new algorithm presented in this paper requires a communication of only 12 values, as opposed to 24 in the previous solution. Furthermore, this algorithm requires knowledge of only the local topology, whereas the previous solution requires the knowledge of the global topology to set-up communication over node-disjoint paths.

Summary of results: We first present an algorithm that enables a dealer to disseminate shares of a secret to n participants in network $\mathcal{G}$, such that the properties of

- k-secret-recovery (when $\mathcal{G}$ satisfies a condition, which we term the k-propagating-dealer condition)
- $(k-1)$-collusion-resistance (for any $\mathcal{G}$)

are satisfied. The algorithm is completely distributed, and each node needs to know only the identities of its neighbours. The algorithm is explicit, works with any finite field of size n or higher, and requires computations consisting only of encoding and decoding one instance of a Reed-Solomon code at every node. Thus, this algorithm allows for efficient generalization of various cryptographic protocols, that previously assumed direct communication links from the dealer to every participant, to a large class of networks.

For any (n,k), and $\mathcal{G}$ with $(n+1)$ nodes, we also derive

- Information-theoretic lower bounds on the total communication complexity under *any* algorithm.

- Communication complexity under our algorithm
- Lower bounds on the communication complexity under the previous solution.

Using these results, we establish that when the k-propagating dealer property is satisfied, the communication complexity of our algorithm is $\Theta(n)$, and is always within a constant factor from the lower bounds. On the other hand, the communication complexity of the previous solution grows super-linearly for a large class of graphs, and is $\Theta(n^2)$ in the worst case.

Additional results in the extended version [12]: Heuristic methods, based on this algorithm, addressing the case when the k-propagating-dealer condition is not satisfied are provided. Extensions incorporating active adversaries, efficient addition of new participants in the absence of trusted entities, and two-threshold secret sharing are also presented. Bounds on the amount of randomness required are derived: the amount of randomness required under our algorithm is independent of n, which is not the case with the previous solution.

Our algorithm is based on a variant of the *Product-Matrix codes* [13] which were originally constructed for distributed storage systems.

Organization of the paper: Section II describes the system model. Section III presents the main algorithm. Section IV presents an analysis of the communication-complexity of the algorithm, lower bounds for the problem, and comparisons with the performance of the previous solution. Finally, Section V presents conclusions and discusses open problems.

II. System Model

A. Secret Sharing in a 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 network, denoted by graph $\mathcal{G}$. The graph $\mathcal{G}$ 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.[2] 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 $(k-1)$-*collusion-resistance* and *k-secret-recovery*. All the participants are assumed to be honest-but-curious, i.e., they follow the protocol correctly, but may store any accessible data to gain information about the secret. [3] The edges in $\mathcal{G}$ can be directed or undirected: a directed edge implies existence of only a one way communication link and an undirected edge implies direct communication links both ways. n and k are assumed to satisfy $n \geq k > 1$, since $n \leq k-1$ prohibits the secret from ever being recovered, while $k=1$ degenerates the problem to the case wherein no security is required.

We shall now discuss a condition that the graph $\mathcal{G}$ 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 property 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.

Lemma 1 (Necessary condition): For any graph $\mathcal{G}$, a necessary condition for any algorithm to perform (n,k) secret sharing is that $\mathcal{G}$ satisfies the k-connected-dealer property.

Proof: The proof is straightforward. Suppose $\mathcal{G}$ does not satisfy the k-connected-dealer property. Then there exists some node (say, node i) that is not directly connected to the dealer, and has at most $(k-1)$ node-disjoint paths from the dealer to itself. Since every path from the dealer to node i must necessarily pass through at least one of these $(k-1)$ nodes, they can together recover the entire share of participant i. Putting in their own $(k-1)$ shares, these $(k-1)$ participants can together recover s, thus violating the $(k-1)$ collusion resistance requirement. ■

Thus no algorithm can operate successfully on all network topologies, and must require the graph $\mathcal{G}$ to obey at least the k-connected-dealer condition. In this regard, we remark that the algorithm presented in this paper is robust to the network topology, i.e., the $(k-1)$-collusion-resistance property is satisfied irrespective of the topology of the network.

Our algorithm successfully disseminates secret shares on a large class of networks. This class is described below.

B. Class of Networks Considered

The algorithm presented in this paper requires the communication network $\mathcal{G}$ to satisfy an additional condition, the k-propagating-dealer condition, as discussed below.

Definition 2 (m-propagating-dealer): A graph with $(n+1)$ nodes (the dealer and n participants) satisfies the m-propagating-dealer property 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 this condition, consider the network of Example 1 (Fig. 1). 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 our algorithm in Example 1). Examples of other classes of graphs that satisfy this condition include layered networks, one-dimensional geometric graphs, backbone networks. In each of these graphs, the k-propagating-dealer condition is satisfied for any node as the dealer. In addition, any directed acyclic graph (DAG) that satisfies the m-connected-dealer condition automatically satisfies the m-propagating-dealer condition (any topological ordering of the DAG suffices as the requisite node-ordering).

Our algorithm successfully performs secret share dissemination to all participants if the graph satisfies the k-propagating-dealer property. We note that while our algorithm requires *existence* of some ordering of the nodes satisfying the k-propagating-dealer property, the algorithm itself is completely distributed and oblivious of this ordering.

Apart from the parameters n and k, an additional parameter d is associated to our algorithm. We saw earlier that the k-

[2] 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.

[3] An extension to handling active adversaries is presented in the extended version [12] of this paper.

connected-dealer condition is necessary for *any* secret sharing algorithm, and our algorithm requires the k-propagating-dealer condition to be satisfied. Now, assuming that these necessary conditions have been met, one would intuitively expect the efficiency of the algorithm to be higher if the graph has a greater connectivity. The parameter d is used to capture this intuition: our algorithm takes the parameter d $(\geq k)$ as input, and under the assumption that the graph satisfies the *d-propagating-dealer* condition, achieves a greater communication efficiency. Note that in the scenario that one does not have an estimate of d, one can execute the algorithm by simply setting d to be equal to the secret sharing parameter k.

C. Notational Conventions

For any node j, the set of its neighbouring nodes is denoted as $\mathcal{N}(j)$. In case of a directed graph, $\mathcal{N}(j)$ denotes the set of nodes that have an incoming edge from node j. The dealer is denoted by D. We say that a node j is directly connected to the dealer if $j \in \mathcal{N}(D)$. Transpose of a vector or matrix is denoted by a superscript T. For any integer $\ell \geq 1$, $[\ell]$ represents the set $\{1,\ldots,\ell\}$.

III. Main Algorithm

Consider a network $\mathcal{G}$ that obeys the *d-propagating-dealer condition* for some parameter d $(\geq k)$. Assume secret s belongs to the alphabet $\mathbb{F}_q^{d-k+1}$, for some $q > n$. Thus we can equivalently denote the secret as a vector $\mathrm{s} = [s_1\ s_2\ \cdots\ s_{d-k+1}]^T$ 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 Ψ, with the i^{th} $(1 \leq i \leq n)$ row of Ψ being

$$\boldsymbol{\psi}_i^T = [1\ i\ i^2\ \cdots\ i^{d-1}]\ . \tag{1}$$

The vector $\boldsymbol{\psi}_i$ is termed the *encoding vector* of node i.

Next, the dealer constructs a $(d \times d)$ *symmetric* matrix M comprising the secret s and a collection of random values as:

$$M \;=\; \begin{array}{c}\left[\begin{array}{ccc} s_A & \boldsymbol{r_a}^T & \boldsymbol{s_B}^T \\ \boldsymbol{r_a} & R_b & R_c^T \\ \boldsymbol{s_B} & R_c & 0 \end{array}\right] \\ \underbrace{\hphantom{s_A}}_{1}\;\underbrace{\hphantom{R_bR}}_{k-1}\;\underbrace{\hphantom{R_cR}}_{d-k}\end{array} \tag{2}$$

where the depicted sub-matrices of M are

- $s_A = s_{d-k+1}$ is a scalar,
- $\boldsymbol{s_B} = [s_1 \cdots s_{d-k}]^T$ is a vector of length $(d-k)$,
- $\boldsymbol{r_a}$ is a random vector of length $(k-1)$,
- R_b is a $((k-1) \times (k-1))$ *symmetric* matrix with its $\frac{k(k-1)}{2}$ 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 $\mathbb{F}_q$. Note that the total number of random values R in matrix M is

$$R = (k-1) + \frac{k(k-1)}{2} + (k-1)(d-k) = (k-1)d - \binom{k-1}{2}$$

The entire secret is contained in the components s_A and $\boldsymbol{s_B}$ as $\mathrm{s}^T = [s_1 \cdots s_{d-k+1}] = [\boldsymbol{s_B}^T\ \ s_A]$. Observe that the structure of M as described in (2), along with the symmetry of matrix R_b, makes the matrix M *symmetric*.

The share $\mathbf{t}_j$ for participant j $(1 \leq j \leq n)$ is a vector of length $(d-k+1)$:

$$\boldsymbol{t}_j^T = \boldsymbol{\psi}_j^T \left[\begin{array}{cc} s_A & \boldsymbol{s_B}^T \\ \boldsymbol{r_a} & R_c^T \\ \boldsymbol{s_B} & 0 \end{array}\right]\ . \tag{3}$$

We shall show subsequently in Theorem 3 that any k of these shares suffice to recover the entire secret.

B. Communication across the Network

Algorithm 1 describes the communication protocol to securely transmit the shares $\{\boldsymbol{t}_j\}_{j=1}^n$ to the n participants.

Algorithm 1 Communication Protocol

```
Dealer: For every $j \in \mathcal{N}(D)$, compute and pass the $d$-length
    vector $\boldsymbol{\psi}_j^T M$ to participant $j$.
Participant $\ell \in \mathcal{N}(D)$: Wait until receipt of data $\boldsymbol{\psi}_\ell^T M$
    from the dealer. Then, for every $j \in \mathcal{N}(\ell)$, compute inner
    product of the data $\boldsymbol{\psi}_\ell^T M$ with the encoding vector $\boldsymbol{\psi}_j$ of
    participant $j$. Pass the resultant value $\boldsymbol{\psi}_\ell^T M \boldsymbol{\psi}_j$ to participant
    $j$.
Participant $\ell \notin \mathcal{N}(D)$: Wait until receipt of one value each
    from any $d$ neighbours. Then, denote this set of $d$ neigh-
    bours as $\{i_1,\ldots,i_d\}$, and the values received from them as
    $\{\sigma_1,\ldots,\sigma_d\}$ respectively. Compute the vector
        $\boldsymbol{v}^T = [\sigma_1 \cdots \sigma_d]^T [\boldsymbol{\psi}_{i_1} \cdots \boldsymbol{\psi}_{i_d}]^{-1}$.
    For every neighbour $i \in \mathcal{N}(\ell)$ from whom you did not receive
    data, pass the inner product $\boldsymbol{v}^T \boldsymbol{\psi}_i$ to participant $i$. [4]
```

C. Correctness of the Algorithm

The proofs of the following theorems are available in [12].

Theorem 2 (Successful share dissemination): Under the algorithm presented, every participant $\ell \in [n]$ can recover $\boldsymbol{\psi}_\ell^T M$, and hence obtain its intended share (3).

Theorem 3 (k-secret-recovery): Any k shares suffice to recover the secret.

Theorem 4 ((k − 1)-collusion-resistance): Any set of $(k-1)$ or fewer colluding participants can gain no information about the secret. This holds for any graph, irrespective of whether it satisfies the required conditions.

This algorithm is also robust to any run-time changes in the network topology (e.g., removal or addition of new links).

IV. Complexity Analysis and Lower Bounds

In this section we provide an analysis and comparison of the communication complexity of our algorithm, the previous solution, and lower bounds for any scheme. Let $|\mathcal{N}(D)|$ denote the size of the set $\mathcal{N}(D)$. In the analysis, the parameters k and d are treated as constants, however, most part of the analysis considers finite n, k, and d, and hence holds even when these parameters depend on n. All proofs are available in [12].

We define, without loss of generality, one unit of data to be the size of the secret. We shall use the notation $\Gamma(.)$ to denote communication complexity. The following theorem provides a comparison of the communication complexity of our algorithm with lower bounds and with the previous solution.

Theorem 5: For any (n,k) and any $\mathcal{G}$ satisfying the k-propagating-dealer condition, the communication complexity of our algorithm is $\Theta(n)$, and is always within a constant (multiplicative) factor of the lower bound. The previous solution entails a super-linear communication-complexity for a wide class of networks, and there exists a class of graphs for which its communication complexity is $\Theta(n^2)$.

These claims are made more precise via the following results, which may also be of independent interest.

1) Our Algorithm

Theorem 6: For any (n,k), and any $\mathcal{G}$ with $(n+1)$ nodes satisfying the d-propagating-dealer condition for some (known) d, our algorithm entails a communication complexity

$$\Gamma_{\text{our}}(\mathcal{G}) = n\frac{d}{d-k+1} \ .$$

2) Information-theoretic Lower Bounds

The following theorem provides an information-theoretic lower bound to the amount of download at any node in the network under any scheme.

Theorem 7: For an (n,k) secret sharing problem on any graph $\mathcal{G}$ with $(n+1)$ nodes, any node $\ell \in [n]$ must download

$$\Gamma_{\text{any}}(\ell) \quad \geq \quad \begin{cases} \frac{\deg(\ell)}{\deg(\ell)-k+1} & \text{if } \ell \notin \mathcal{N}(D) \text{ and } \deg(\ell) \geq k \\ 1 & \text{if } \ell \in \mathcal{N}(D) \\ \infty & \text{if } \ell \notin \mathcal{N}(D) \text{ and } \deg(\ell) < k \end{cases}$$

where $\deg(\ell)$ denotes the number of incoming edges at node ℓ. Furthermore, this bound is the best possible, given only the identities of the neighbours of node ℓ.

Corollary 8: For an (n,k) secret sharing problem on any graph $\mathcal{G}$ with $(n+1)$ nodes, the total communication complexity is lower bounded by

$$\Gamma_{\text{any}}(\mathcal{G}) \geq |\mathcal{N}(D)| + \sum_{i \notin \mathcal{N}(D)} \frac{\deg(i)}{\deg(i)-k+1} \geq n \ .$$

Thus the communication complexity of our algorithm is a constant multiplicative factor away from the lower bound.

Corollary 9: For any (n,k) and any d-regular graph with $(n+1)$ nodes satisfying the d-propagating-dealer condition, under our algorithm, the amount of data downloaded by any node $\ell \notin \mathcal{N}(D)$ is the minimum possible. Furthermore, the amount of data downloaded by any node $\ell \in \mathcal{N}(D)$ is independent of n.

Corollary 10: For any (n,k), and any d $(k \leq d < n)$, there exists a class of graphs with $(n+1)$ nodes such that each graph in this class satisfies the d-propagating dealer property, and the communication complexity for (n,k) secret sharing on any graph $\mathcal{G}$ in this class is lower bounded by

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

Thus, the complexity of our algorithm is a constant additive factor away from the lower bound for this class of graphs.

3) Previous Solution

Theorem 11: For any (n,k) and $\mathcal{G}$, the communication complexity of the previous solution is

$$\Gamma_{\text{prev}}(\mathcal{G}) = |\mathcal{N}(D)| + \sum_{i \notin \mathcal{N}(D)} \min_{w \geq k} \left[\frac{w}{w-k+1} \times \ell_w(D \to i) \right]$$

where $\ell_w(D \to i)$ is the average of the path lengths of the w shortest node-disjoint paths from D to i (with $\ell_w(D \to i) = \infty$ if there do not exist w node-disjoint paths from D to i).

Corollary 12: For any sequence of graphs of increasing size with the maximum outgoing degree being $O((\log n)^{\frac{1}{2}-\epsilon})$ for some $\epsilon > 0$, the previous solution requires a super-linear communication complexity.

Corollary 13: For any (n,k) and any d $(k \leq d < n)$, there exists a class of graphs with $(n+1)$ nodes such that each graph in this class satisfies the d-propagating dealer property, and (n,k) secret sharing on any graph $\mathcal{G}$ in this class using the previous solution requires a communication complexity

$$\Gamma_{\text{prev}}(\mathcal{G}) \geq \frac{n(n+1)}{4d} \ .$$

Thus, on a sequence of such classes of graphs, our algorithm requires $\Theta(n)$ communication complexity, as compared to $\Theta(n^2)$ required under the previous solution.

V. Conclusions and Open Problems

The problem of secret sharing in a network arises fundamentally in several problems for security and cryptography. By means of an explicit algorithm and information theoretic bounds, this paper provides upper and lower bounds on the communication complexity required for this problem. However, obtaining the precise complexity still remains open. The algorithm presented here requires the network to satisfy the k-propagating dealer condition, and heuristics to address general networks are presented in the extended version [12]. However, the guarantees achieved by the algorithm in the general case are not known. Finally, it remains to see if any of the ideas from this specific case of secure network coding carry over to more general network coding problems.

References

[1] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[2] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *ACM STOC*, 1988.

[3] T. Pedersen, "A threshold cryptosystem without a trusted party," in *Advances in Cryptology–EUROCRYPT*, 1991, pp. 522–526.

[4] M. Storer, K. Greenan, E. Miller, and K. Voruganti, "Potshards: A secure, recoverable, long-term archival storage system," *ACM Trans. on Storage*, 2009.

[5] D. Dolev, C. Dwork, O. Waarts, and M. Yung, "Perfectly secure message transmission," *Journal of the ACM*, vol. 40, no. 1, pp. 17–47, 1993.

[6] N. Cai and R. Yeung, "Secure network coding on a wiretap network," *IEEE Trans. on Inf. Th.*, Jan. 2011.

[7] J. Feldman, T. Malkin, C. Stein, and R. Servedio, "On the capacity of secure network coding," in *Allerton Conf.*, 2004.

[8] H. Yao, D. Silva, S. Jaggi, and M. Langberg, "Network codes resilient to jamming and eavesdropping," in *NetCod*, 2010.

[9] C. Ngai and R. Yeung, "Secure error-correcting (sec) network codes," in *NetCod*, 2009.

[10] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Th.*, Oct. 2006.

[11] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Th.*, Jun. 2005.

[12] N. B. Shah, K. V. Rashmi, and K. Ramchandran, "Secure network coding for distributed secret sharing with low communication cost." [Online]. Available: arXiv:1207.0120

[13] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for the MSR and MBR points via a product-matrix construction," *IEEE Trans. Inf. Th.*, Aug. 2011.