On the Price of Proactivizing Round-Optimal Perfectly Secret Message Transmission *

Ravi Kishore † Ashutosh Kumar ‡ Chiranjeevi Vanarasa § Kannan Srinathan ¶

Abstract

In a network of n nodes (modelled as a digraph), the goal of a perfectly secret message transmission (PSMT) protocol is to replicate sender's message m at the receiver's end without revealing any information about m to a computationally unbounded adversary that eavesdrops on any t nodes. The adversary may be mobile too – that is, it may eavesdrop on a different set of t nodes in different rounds. We prove a necessary and sufficient condition on the synchronous network for the existence of r-round PSMT protocols, for any given $r > 0$; further, we show that round-optimality is achieved without trading-off the communication complexity; specifically, our protocols have an overall communication complexity of $O(n)$ elements of a finite field to perfectly transmit one field element. Apart from optimality/scalability, two interesting implications of our results are: (a) *adversarial mobility does not affect its tolerability:* PSMT tolerating a static t-adversary is possible *if and only if* PSMT tolerating mobile t-adversary is possible; and (b) *mobility does not affect the round optimality:* the fastest PSMT protocol tolerating a static t-adversary is *not faster* than the one tolerating a mobile t-adversary.

1 Introduction

We address the problem of Perfectly Secret Message Transmission (PSMT)[1] defined as follows: The sender S wishes to send a message m to the receiver R such that the adversary, that eavesdrops on no more than t out of the n nodes (in one time-period/round) in the network, learns nothing (except the information that the probability distribution on the message space reveals) about m. For fast protocols, the adversary may be assumed to be *static*, that is, the same set of nodes are corrupt (in every round) throughout the protocol execution. However for protocols that last long, a more suitable model is that of a *mobile* adversary which corrupts different set of t nodes in different rounds (catering to an equilibrium between (a) curing/replacing faulty machines and (b) breaking-in to new machines during the protocol execution). Evidently, protocols tolerating mobile

*This article was published in IEEE Transactions on Information Theory 2018 [15]. A preliminary version appeared in ICITS 2015 [14].

†IIIT Hyderabad. ravikishore.vasala@research.iiit.ac.in. Work supported by Tata Consultancy Services (TCS), India.

‡UCLA. a@ashutoshk.com. Work done while the author was a student at IIIT Hyderabad

§IIIT Hyderabad. chiranjeevi.v@research.iiit.ac.in

¶IIIT Hyderabad. srinathan@iiit.ac.in

[1]In this work, we interchangeably use PSMT to mean both Perfectly *Secret* Message Transmission as well as Perfectly *Secure* Message Transmission; the former when the adversary is passive and the latter when the adversary is Byzantine. At any rate, our technical contributions are only in the passive adversarial case.

t-adversary are likely to be far more cumbersome and complex than the ones tolerating static t-adversaries. Counter-intuitively, we show that protocols for perfectly secret message transmission can withstand adversarial mobility for *free*. Specifically, for PSMT in any directed graph influenced by a passive/eavesdropping adversary, we show that: (a) *adversarial mobility does not affect its tolerability:* PSMT tolerating a static t-adversary is possible *if and only if* PSMT tolerating mobile t-adversary is possible; (b) *mobility does not affect the round optimality:* the fastest PSMT protocol tolerating static t-adversary is *not faster* than the fastest one tolerating mobile t-adversary; and (c) *mobility does not affect communication complexity:* we design PSMT protocols that have linear communication complexity in both static as well as mobile adversarial settings.

Our inquiry includes: (a) *characterization*: under what conditions is a solution possible? (b) *feasibility*: is the characterization efficiently testable and is there an efficient protocol? (c) *round complexity*: what is the *fastest* solution? and (d) *communication complexity*: what is the *cheapest* solution? Intuitively, the above questions are in increasing order of difficulty. Consequently, question '(a)' has been answered in settings that are far more general than those where optimal solutions are known yet.

Although the literature on information theoretically secure message transmission is rich (e.g., [7, 2, 10, 36, 30, 28, 23]), there are settings where answers to none of the aforementioned four questions are known yet. For instance, we do not know of a necessary and sufficient condition on digraphs influenced by a Byzantine adversary corrupting up to any t nodes for the existence of protocols for perfectly secure message transmission from S to R, where S is the sender and R is the receiver [27]; not to mention, the design of optimal protocols for the same are still far-fetched. Researchers have therefore addressed the PSMT problem in scenarios that are not as general as mentioned above – the harder the inquiry, the more specific the chosen setting. Notwithstanding, researchers have also worked on interesting generalizations in some dimensions (while, of course, being more specific in other parameters so that the problem is tractable using contemporary techniques), including hypergraphs (e.g.,[32, 11]), non-threshold adversaries (e.g., [26, 13]), mobile faults (e.g.,[24, 34, 25]), mixed/hybrid faults (e.g., [9, 31, 2, 33, 3]), asynchronous networks (e.g., [28, 31, 4, 5, 1, 17, 20]), to name a few.

The PSMT problem was conceived and first solved by Dolev *et al.* [7]. They assumed that the graph is *undirected*. It is proved that PSMT from S to R tolerating t Byzantine faults is possible *if and only if* there are at least $(2t+1)$ vertex disjoint paths between S and R. Further, the protocols are efficient too. However, designing round optimal protocols for PSMT (even in undirected graphs) remains a hard open problem. Consequently, results are known only with further restrictions.

A setting where round-optimal protocols have been designed (on arbitrary digraphs) is when a small probability of error is permitted [35] (that is, perfectness is negligibly traded-off). However, the design of communication optimal solutions is still open as mentioned in [23].

A particular setting where communication optimum protocols for PSMT are designed is the following: applying Menger's theorem [22], the undirected graph can be abstracted as a collection of wires (vertex-disjoint paths) between S and R, up to t among which are corrupted by the adversary. In this setting, a two phase protocol for PSMT that is optimal in communication complexity is known [18]. While the notion of phase complexity has been studied in the works of [2, 18, 8], we stress that round complexity (e.g.,[34, 25]) is markedly different from phase complexity, even in the case of undirected networks (see Section 2.1).

Recently, restricting to passive adversaries, Renault *et al.* [27] characterized the digraphs that enable PSMT. In fact *Renault et al.* in [27] use a more general non-threshold adversary model,

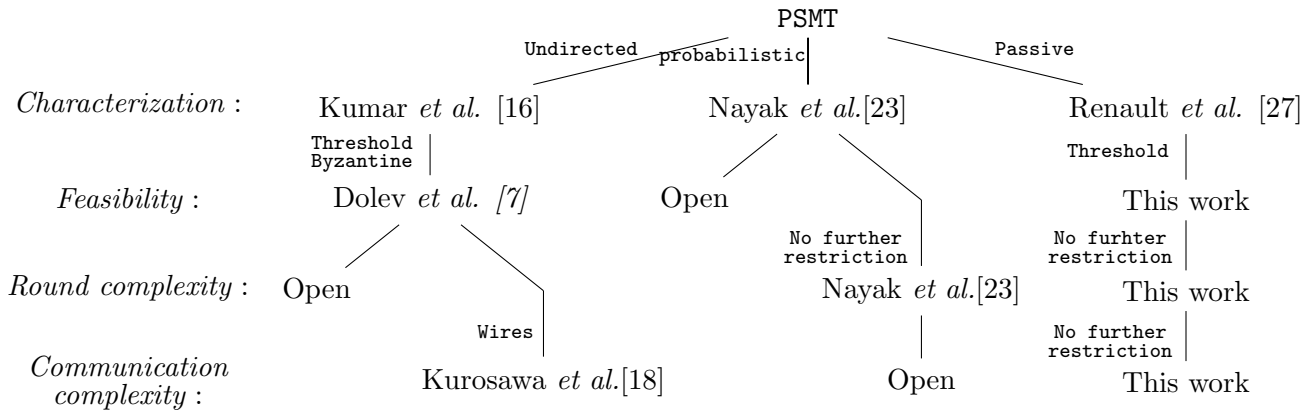

Figure 1: Restrictions based solutions.

characterized via an adversary structure, which is a collection of subsets of nodes in the graph, wherein the adversary may choose to corrupt (passively in this case) the nodes in any one subset from the collection. The protocols of [27] are, therefore, not always efficient (that is, may be super-polynomial in n) as discussed in [23].

In summary, as depicted in the Fig. 1, all the four questions in our inquiry, with respect to the problem of PSMT, have remained open in the general case of digraphs influenced by a Byzantine adversary characterized via an adversary structure. However, (im)possibility results are known if one restricts the setting to either undirected graphs [16] or passive adversary or security with error (e.g., [27, 23]). Nevertheless, efficient protocols are still elusive. To design efficient protocols using contemporary techniques, further restriction (apart from moving to undirected graphs) is required, namely, *threshold* adversary. For instance, Dolev *et al.* in [7] have given one such efficient protocol, which, however, is neither round optimal nor bit-optimal.

Round-optimal protocols are known only in the case of weaker (not perfect) security models like statistical [35] or computational security [6]. Bit-optimal protocols have been designed in the wires-based abstraction of the undirected graph in [18]. While a similar wires-based approach has been used for digraphs too in [36], it is known to be inadequate to capture all digraphs on which protocols exist as shown in [35].

2 Our Contributions

As depicted in Fig. 1, we ask: *does restricting to the setting of passive threshold adversaries lead to the design of efficient and round-optimal and/or communication optimal protocols?* (or, are further restrictions like wires-based abstractions still required?)

Interestingly, we design communication efficient and round optimal protocols, with no further restrictions beyond assuming that the adversary passively corrupt up to t nodes in the digraph. Incidentally, it turns out that our techniques for designing round-optimal protocols are orthogonal to those that entail linear communication complexity – therefore, when applied together, we obtain protocols that are *simultaneously* round optimal as well as communication optimal. Further, the *simplicity* of our protocol ensures the implementability of highly scalable perfectly secret message

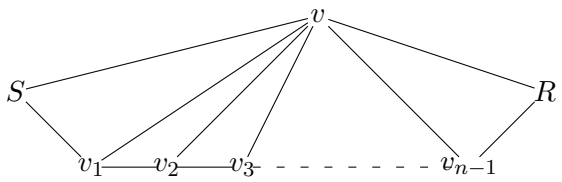

Figure 2: An undirected graph tolerating one passive fault.

transmission. Surprisingly, as proved in Section 7, it turns out that most of our protocols can be adapted to work for the mobile adversary case too. In a nutshell, we address the PSMT problem in such a way that all the four questions, namely, characterization, feasibility, communication and round optimality, are answered in one-shot. In the subsections below, we briefly describe our results and their significance.

2.1 Complete characterization of networks wherein an r-round secret communication protocol tolerating static adversary is (im)possible

In [7] Dolev et al. proved that $(t+1)$-vertex disjoint paths are necessary and sufficient for PSMT from S to R in undirected graphs to tolerate passive t-threshold static adversary. Consequently, as noted in [7] too, without loss of generality, any network (undirected graph) may be abstracted as a set of wires (vertex disjoint paths) between S and R. However, in the design of round optimal PSMT protocols, such an abstraction is inadequate even if the length of the wires is recorded. Specifically, using the edges connecting across these wires (or practically every edge in the network) it is possible to design *faster* protocols. For example, consider the graph in Fig. 2; The two wires corresponding to two vertex disjoint paths $\langle S, v, R\rangle$ and $\langle S(=v_0), v_1, v_2, v_3, \ldots v_{n-1}, R(=v_n)\rangle$ have lengths of *two* and n respectively. Following Dolev's protocol, S sends two points on a linear polynomial whose constant term is the secret m, individually through these two wires. The receiver R gets the two points and hence the message after n rounds. Does a faster protocol exist? Our answer: *Yes. In fact, a* 3*-round protocol exists irrespective of how large* n *is.* Perhaps it is not conspicuous at first glance and certainly not if we continue to use the wires-based abstraction of the network. As a corollary to our *Theorem* 7, we know that *three* rounds are necessary and sufficient for S to R PSMT in the graph given in Fig. 2. Thus, extant techniques are insufficient to design round optimal protocols and new techniques are necessary to design, and more importantly, prove round optimality. To summarize, the problem of characterizing round optimal protocols in directed networks is a non-trivial and interesting problem.

2.2 Linear Communication Complexity

Folklore suggests that optimizing the number of rounds for a distributed protocol typically increases the communication complexity. In rare cases, round optimality can co-exist with communication-optimality – PSMT is indeed one such case! Specifically, we prove that the number of edges used by our protocol can be brought down to linear in the number of nodes (see Section 6.1). We also ensure that an edge is used to send at most one field element (or in general, bits equivalent to the size of the message). Thus, we arrive at a surprising protocol for secret communication which is round optimal and at the same time has linear communication complexity. Even more interesting is the case when the shortest path from S to R has $\Omega(n)$ nodes. In such cases, *perfect secrecy is*

achieved for "free" because any (insecure routing) protocol would also take $\mathcal{O}(n)$ rounds and send $\mathcal{O}(n)$ messages for transmission – one message along each edge in the shortest path.

2.3 Efficient Discriminant Algorithms

Succinctly specifying the necessary and sufficient condition does not necessarily imply that there exists an efficient algorithm for checking the same. Indeed, the literature on possibility of PSMT protocols in directed graphs is replete with several problem specific characterizations, none of which are known to be efficiently testable. For instance, the possibility of reliable/secure message transmission in Byzantine adversarial setting in digraphs is characterized in [35, 23]. However, no efficient algorithms to test these conditions are known. In fact they may be NP-hard too as mentioned in [21] though no such study has been carried out. In contrast, for each of the results in this paper, we have a polynomial time algorithm for testing the same. *Algorithm 5.4* is a polynomial-time algorithm for testing the existence of an r-round secret communication protocol in a given network (and if yes, for obtaining a round optimal one).

2.4 Mobile adversary

Typically, mobile adversaries are notoriously difficult to withstand due to their dynamic movements across the network at a scorching pace. If the problem/protocol requires sustained long-distance collaboration for the task at hand, it is very easy for the mobile adversary to breach any kind of purported defences in-built in the protocol. And, we notice that in PSMT protocols it appears that the messages/packets need to travel across the network and therefore are easily susceptible to mobile adversarial attacks. A key ingredient in our solution tolerating mobile faults is the following: we address the problem by generating randomness across the network within a short span of time (say within one round) so that even a mobile adversary is bound to miss substantial part of the random coins used by the protocol. More importantly, if the random-coins are locally deleted by the respective generators *before* the adversary can spy on them, there is ample scope for the protocol to withstand adversarial mobility as easily as its static counterpart. The challenge here is: what can be accomplished by random-coins that are ephemeral and have a very short life-span? We show that the answer isn't nothing; in particular, PSMT protocols can be designed with such short-lived randomness. In Section 7 we show how to use ephemeral random-coins and modify our static protocol to tolerate mobile faults.

3 Notations and Definitions

3.1 Notations

1. The message space is a large enough finite field $\langle \mathbb{F}, +, \star \rangle$ and all the calculations are done in the field $\mathbb{F}$ only. By "a number r is chosen randomly" we mean that "r is chosen uniformly at random from the field $\mathbb{F}$".

2. Throughout this article, by a "faulty node" we mean that the node is "passively corrupted by the adversary" and by "secure" we mean "perfectly secret".

3. For brevity, by "PSMT is possible" we mean "PSMT tolerating t-threshold passive adversary is possible".

4. We use $[l, u]$ to denote the set $\{m \in \mathbb{Z} \mid l \leq m \leq u\}$.

3.2 Graph and Paths

Definition 1. *(**Underlying Undirected Graph**) The underlying undirected graph of a directed graph $G(V, E)$ is denoted by $G_u(V, E_u)$, where $E_u = \{(u, v) \mid (u, v) \in E \text{ or } (v, u) \in E\}$.*

Definition 2. ***(Path)** In a directed graph $G(V, E)$, a sequence $p : \langle v_0(= u), v_1, v_2, \ldots, v_k, v_{k+1}(= v)\rangle$ of nodes is a path from u to v, if and only if $(v_j, v_{j+1}) \in E$, $\forall j \in [0, k]$.*

Definition 3. ***(Weak Path)** In a directed graph $G(V, E)$, a sequence $p : \langle v_0(= u), v_1, v_2, \ldots, v_k, v_{k+1}(= v)\rangle$ of nodes is a weak path from u to v if and only if $\forall j \in [0, k]$, either $(v_j, v_{j+1}) \in E$ or $(v_{j+1}, v_j) \in E$.*

Definition 4. ***(Corresponding Path of a Weak Path)** We say that the path $p' : \langle v_0(= u), v_1, v_2, \ldots, v_k, v_{k+1}(= v)\rangle$ in G_u is the corresponding path of a weak path $p : \langle v_0(= u), v_1, v_2, \ldots, v_k, v_{k+1}(= v)\rangle$ in G.*

3.3 Network Model

Definition 5. *(**Network**) We model our communication network as a directed graph $G(V, E)$, where each edge is a private, authentic and reliable channel. We assume that every player (node) including the adversary completely knows the protocol specifications and the topology of the network.*

Definition 6. *(**Synchronous Network and Round**) ([19]) A network is synchronous if every node has access to a global clock and the communication proceeds in rounds (time-steps) according to this global clock. From the communication point of view, it takes exactly one round (one time-step) to transmit field element(s) along any link (edge) of the network. More formally, in any **round**, a player can execute commands in the following order :*

1. *Perform local computations.*
2. *Send messages to its out-neighbour(s).*
3. *Receive all the messages sent earlier in this round by its in-neighbour(s).*
4. *Perform local computations.*

Definition 7. *(**Round Complexity**) The round complexity of any synchronous protocol is defined as the total number of rounds required to execute the protocol before its termination.*

Definition 8. *(**Communication complexity**) The communication complexity of any protocol is defined as the total number of field elements communicated through all the links in the network during the execution of the protocol.*

3.4 Adversary

In this work we consider an adversary which can eavesdrop on the network by passively corrupting the nodes. We formally define this type of corruption below.

Definition 9. *(**Passive Corruption**) ([9]) A node P is said to be (passively) corrupted if the adversary has full access to the information and internal state of P. We note that in this corruption model the adversary has only read access to the corrupted node and cannot alter its protocol execution. As a result, the corrupted node P honestly follows the protocol.*

Definition 10. *(**Static Adversary**) We say that the adversary is t-static if it is allowed to corrupt only one fixed set of nodes of cardinality at most t throughout the protocol execution. In other words, if the adversary is static then once a node is corrupted it remains corrupted in each of the subsequent rounds of the protocol.*

Definition 11. *(**Mobile Adversary**) We say that the adversary is t-mobile if it is allowed to corrupt different sets of nodes (except S and R) of cardinality at most t in different rounds of the protocol. Formally, on a synchronous network $G(V, E)$ for any protocol $\Pi(G, S, R)$ with round complexity r, in each round $i \in [1, r]$, the mobile adversary is allowed to corrupt up to t nodes (except S and R) of its choice.*

Definition 12. *(**View of a node**) ([31]) In any digraph $G(V, E)$, we define the view of a node $v \in V$ at any point during the execution of a protocol Π, to be the information the computationally-unbounded node can compute from its local input (if any), all the messages that it had earlier sent or received, its random coins and the protocol specification and the topology of the network.*

Definition 13. *(**View of the adversary**) ([10, 36]) The view of the adversary at any point during the execution of a protocol Π is defined as all the information that the computationally-unbounded adversary can compute from the views of all the corrupted players.*

3.5 Message Transmission

The following definition is inspired from [10] and [36]. We use M to denote the random variable on the message space and $VIEW$ to denote the random variable on the set of all possible views of the adversary.

Definition 14. *(**Perfectly Secret Message Transmission (PSMT)**) Let $G(V, E)$ be a synchronous network with the designated sender S and receiver R. A message transmission protocol (for transmitting the message m from S to R) is said to be perfectly secret tolerating the computationally unbounded adversary $\mathcal{A}$, if the following two properties hold:*

- ***Perfect Reliability**: At the end of the protocol the receiver R should receive the transmitted message m with probability 1.*
- ***Perfect Secrecy**: For any two messages m and m', it is impossible for the adversary $\mathcal{A}$ to distinguish whether the message being transmitted from S to R is m or m'. Formally, for every probability distribution on the message space, for every two messages m, m' and every possible view v of the adversary, $P[VIEW = v|M = m] = P[VIEW = v|M = m']$, where the probabilities are taken over the coin flips of the uncorrupted nodes/parties.*

4 PSMT in directed networks

In this section, we study about the design of efficient `PSMT` protocols in arbitrary directed graph setting. We notice that, in a directed graph $G(V, E)$, for a given node $v \in V$ if there is no path

from v to the receiver R, then v cannot convey any information to R in any message transmission protocol. Therefore, we assume that each node (in the graph) has at least one path to the receiver R. Then, in *Theorem* 3 we show that PSMT from S to R is possible in a directed graph G if and only if PSMT from S to R is possible in its underlying undirected graph G_u. To show the same, in Section 4.1.2 we present a communication efficient PSMT protocol $\mathbf{\Pi_{Eff}}$. Now, we move to the existing result for PSMT (im)possibility in undirected graphs, which is as follows.

Theorem 1. *(Dolev* et al. *[7]) In an undirected graph G_u,* PSMT *from S to R is possible tolerating up to t passive faults if and only if there exist $t+1$ vertex disjoint paths from S to R.*

Proof. **Necessity**: Suppose there exist at most t vertex disjoint paths from S to R. Then, we have from Menger's theorem [22], that there exists a vertex-cut of size t between S and R. Therefore, by corrupting every node in the vertex-cut, the adversary corrupts each of these t paths and gets the information identical to what the receiver would receive from the sender.
Sufficiency: The sufficiency is achieved using Shamir's secret sharing scheme. The sender S chooses a random degree-t polynomial $p(x)$ such that $p(0)$ is the message m. The sender S sends $p(i)$ to the receiver R along the i^{th} disjoint path. We know that, $t+1$ points on $p(x)$ are enough to reconstruct it whereas t or fewer points reveal nothing about its constant term $p(0)$ [29]. Therefore, the adversary learns nothing (additional) about the message m. □

4.1 Communication Efficient PSMT Protocol

This section contributes to the design of a communication efficient PSMT protocol $\mathbf{\Pi_{Eff}}$. In undirected graphs we have seen a simple protocol, where, each disjoint path carries exactly one point on degree-t polynomial. And, the uncorrupted path (no node of it is corrupted) guarantees the security of the protocol. In directed graphs, we achieve the same effect with the protocol $\mathbf{\Pi_{Eff}}$. The core of the protocol $\mathbf{\Pi_{Eff}}$ is the sub-protocol $\mathbf{\Pi_{Sim}}$, which simulates the corresponding path p' of a given weak path p. By simulation we mean, for any given weak path p, the protocol $\mathbf{\Pi_{Sim}}$ always *reliably* transmits the message m from S to R using each node of p, as if p were a path. Moreover, if no node of p is corrupted then the adversary learns nothing (additional) about the message being transmitted using p. Thus, executing $\mathbf{\Pi_{Sim}}$ on $t+1$ disjoint weak paths results in the PSMT protocol $\mathbf{\Pi_{Eff}}$. Before going into the technical details of the protocol $\mathbf{\Pi_{Sim}}$, we first show that such a simulation is possible. Let $p : \langle S(= u_0), u_1, \ldots, u_l, u_{l+1}(= R)\rangle$ be a weak path in G. Then, we have two cases:

1. Case (1): If p is a path in G, then the simulation is trivial – S simply sends the message to u_1 and u_1 forwards it to u_2, u_2 in turn forwards it to u_3 and so on until it reaches R. As no node of p is corrupted, the adversary learns nothing (additional) about the message, whereas R gets the message m.

2. Case (2): If p is not a path in G, then there exist at least one u_i such that the forward edge $(u_i, u_{i+1}) \notin E$. Let $\{u_{i_1}, u_{i_2}, \ldots, u_{i_k}\}$ be the set of all nodes on the weak path p such that $(u_{i_j}, u_{i_j+1}) \notin E$ for $j \in [1, k]$. Without Loss of Generality (W.L.G), we assume that $i_p < i_q$ for $p < q$ (see Fig. 3). Also, from the context it is clear that $u_{i_k} \neq R$; that is $i_k < l+1$. In Lemma 2, we prove that such a protocol/simulation exists for this case too. We use the following lemma to prove the correctness of Lemma 2.

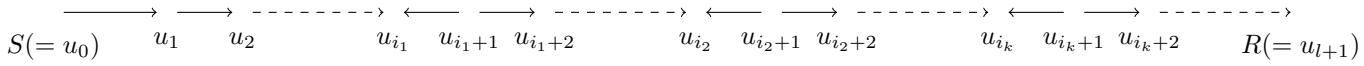

Figure 3: Weak path p.

Lemma 1. *In a directed graph G, let u, v, w be three uncorrupted nodes such that* PSMT *from w to u is possible and* PSMT *from w to v is possible. Then,* PSMT *from u to v is possible in G if there exists a path from u to v in G.*

Proof. Let m be the field element that u wants to secretly transmit to v. First, the node w chooses a random field element r and sends it to both u and v *secretly*, as PSMT is possible from w to both u and v. Now u masks the message m using the received number r as $m - r$ and sends it to the destination node v along a path from u to v, as there exists such a path. Finally, v obtains the message m by adding r to $m - r$. This protocol is perfectly secure even if the adversary corrupts the path from u to v, which carries $m-r$. Since, in a field $\langle \mathbb{F}, +, * \rangle$ for a given $x, z \in \mathbb{F}$, there exists a unique $y \in \mathbb{F}$ such that $x - y = z$. In other words, if the adversary corrupts the path from u to v then it learns $m - r$, which reveals nothing (additional) about m. □

Lemma 2. *In a directed graph G, let $p : \langle S(= u_0), u_1, \ldots, u_l, u_{l+1}(= R) \rangle$ be a weak path such that there exists a path from every node u_i (of the weak path p) to R. Then,* PSMT *from S to R is possible in G if no node of the weak path p is corrupted.*

Proof. Recall that, if p is a path in G then S simply sends the message to R along p. Therefore, PSMT from S to R is trivially possible in G (as no node of p is corrupted). If p is not a path in G, then recall that $\{u_{i_1}, u_{i_2}, \ldots, u_{i_k}\}$ is the set of all nodes that do not have a forward edge on the weak path p, where $i_k < l + 1$ (that is, $u_{i_k} \neq R$). As the node u_{i_k} is the last one satisfying $(u_{i_k}, u_{i_k+1}) \notin E$, there is a *secure* backward edge $(u_{i_k+1}, u_{i_k}) \in E$. For u_{i_k+1}, we have two cases:
Case (1): If $u_{i_k+1} = R$, then PSMT from u_{i_k+1} to R is trivially possible in G (as R can securely communicate with itself).
Case (2): If $u_{i_k+1} \neq R$ then (as no node of p is corrupted) there is a *secure path* from u_{i_k+1} to R along the nodes of the weak path p itself, which implies PSMT from u_{i_k+1} to R is possible in G.
Therefore, in any case, PSMT from u_{i_k+1} to R is possible in G. This implies, by applying the *Lemma* 1, we get that PSMT from u_{i_k} to R is possible in G. Now, we iteratively apply the above idea in reverse direction and show that PSMT from S to R is possible in G.

We notice that, for $j = k - 1, k - 2, \ldots, 1$:

1. We have a secure sub-path of p from u_{i_j+1} to $u_{i_{(j+1)}}$ in G (see Fig. 3). [a]
2. We have already shown that PSMT from $u_{i_{(j+1)}}$ to R is possible in G.
3. The above two steps (step 1 and 2) together ensure that PSMT from u_{i_j+1} to R is possible in G.
4. We have a secure backward edge $(u_{i_j+1}, u_{i_j}) \in E$.

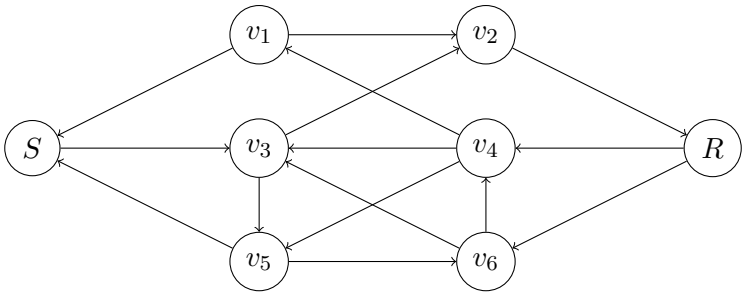

Figure 4: Graph G with three vertex-disjoint weak paths.

5. The above two steps together (step 3 and 4), on applying *Lemma* 1, ensure that PSMT from u_{i_j} to R is possible in G.

[a] In case if $u_{i_j+1} = u_{i_{(j+1)}}$, then we trivially assume that there is a path from u_{i_j+1} to $u_{i_{(j+1)}}$ in G (as u_{i_j+1} can (securely) communicate with itself).

In particular, when $j = 1$, PSMT from u_{i_1} to R is possible in G. And, we have a secure sub-path of p from S to u_{i_1}, therefore, PSMT from S to R is possible in G. □

4.1.1 Communication Efficient Simulation

We apply the same idea (used in the Lemma 2) to design the protocol $\mathbf{\Pi_{Sim}}$ which simulates the corresponding path p' of a given weak path $p : \langle S(= u_0), u_1, \ldots, u_l, u_{l+1}(= R)\rangle$. Recall that, $\{u_{i_1}, u_{i_2}, \ldots, u_{i_k}\}$ (where $u_{i_k} \neq R$) is the set of all nodes on the weak path p such that $(u_{i_j}, u_{i_j+1}) \notin E$, for each $j \in [1, k]$. This implies, there exists (i) a backward edge $(u_{i_j+1}, u_{i_j}) \in E$ and (ii) a sub-path of p, say p_{i_j+1}, from u_{i_j+1} to $u_{i_{(j+1)}}$ in G, where W.L.G we assume that $u_{i_{(k+1)}}$ is R. Moreover, in case if $u_{i_j+1} = u_{i_{(j+1)}}$, then the path p_{i_j+1} is nothing but a path from u_{i_j+1} to u_{i_j+1} (which we assume trivially exists as u_{i_j+1} can (securely) communicate with itself). The Protocol $\mathbf{\Pi_{Sim}}$ is given below.

The Protocol $\mathbf{\Pi_{Sim}}$

1. For $j = 1, 2, 3, \ldots, k$: The node u_{i_j+1} chooses a random number r_{i_j+1} and sends it to the node $u_{i_{(j+1)}}$ (along the path p_{i_j+1}) and to the node u_{i_j} (along the edge (u_{i_j+1}, u_{i_j})).
2. For $j = 2, 3, 4, \ldots, k$: The node u_{i_j} calculates $r_{i_{(j-1)}+1} - r_{i_j+1}$ and sends it to R along a path from u_{i_j} to R.
3. The sender S sends the message m to the node u_{i_1} along the path $\langle S(= u_0), u_1, u_2, \ldots, u_{i_1}\rangle$.
4. The node u_{i_1} calculates $m - r_{i_1+1}$ and sends it to R along a path from u_{i_1} to R.
5. For $j = k-1, k-2, \ldots, 1$: R computes $r_{i_j+1} = (r_{i_j+1} - r_{i_{(j+1)}+1}) + r_{i_{(j+1)}+1}$.
6. Once R gets r_{i_1+1} for $j = 1$, it finally computes $m = (m - r_{i_1+1}) + r_{i_1+1}$.

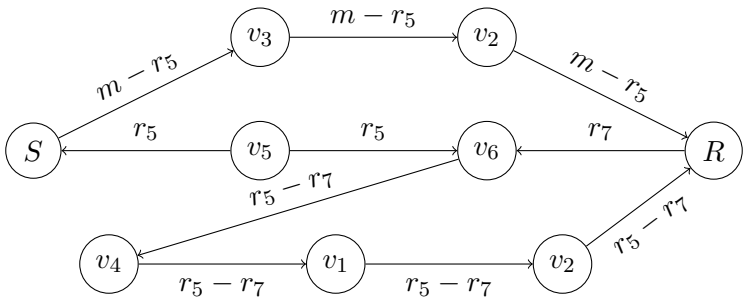

Figure 5: Simulation of the corresponding path p_3'.

Now with an example, we illustrate the execution of the protocol $\mathbf{\Pi_{Sim}}$. We consider the graph G given in Fig. 4 which has a maximum of three vertex disjoint weak paths. Therefore, this graph can tolerate up to two faulty nodes. Let three weak paths be $p_1 : \langle S, v_1, v_2, R\rangle$, $p_2 : \langle S, v_3, v_4, R\rangle$ and $p_3 : \langle S, v_5, v_6, R\rangle$. The simulation of the corresponding path of the weak path p_3 is shown in Fig. 5 and works as follows:

An execution of the protocol $\mathbf{\Pi_{Sim}}$ for the weak path $\mathbf{p_3 : \langle S, v_5, v_6, R\rangle}$

1. R chooses a random number r_7 and sends it to v_6.
2. v_5 chooses a random number r_5 and sends it to both S and v_6.
3. v_6 masks r_5 with r_7 as $r_5 - r_7$ and sends it to R along the path $\langle v_6, v_4, v_1, v_2, R\rangle$.
4. S masks the message m as $m - r_5$ and sends it to R along the path $\langle S, v_3, v_2, R\rangle$.
5. R first unmasks r_5 by adding r_7 to $r_5 - r_7$ then unmasks m by adding r_5 to $m - r_5$.

The correctness of the protocol $\mathbf{\Pi_{Sim}}$ is proved in the following theorem.

Theorem 2. *Let $G(V, E)$ be a directed graph in which S and R are two special nodes and $p : \langle S(= u_0), u_1, \ldots, u_l, u_{l+1}(= R)\rangle$ be a weak path such that there exists a path from every node u_i (of the weak path p) to R. Then, the protocol $\mathbf{\Pi_{Sim}}$ secretly transmits the message m from S to R in G if no node of the weak path p is corrupted.*

Proof. Let p be the path as given in the theorem statement and m be the message being transmitted by the protocol $\mathbf{\Pi_{Sim}}$. We know that the adversary cannot eavesdrop on any of these nodes as no node u_{i_j} is corrupted. However, for each $j \in [1, k]$, node u_{i_j} sends $r_{i_{(j-1)}+1} - r_{i_j+1}$ to R, where $r_{i_0+1} = m$. In the worst case, the adversary may intercept each of these values, in which case the view of the adversary is $\{r_{i_{(j-1)}+1} - r_{i_j+1} | j \in [1, k]\}$. We show that the view of the adversary is independent of the message being transmitted. In other words, we show that, for each view v of the adversary, there is exactly one valid execution of the protocol for every message m', and all these executions are equally likely.

Consider the following *valid* execution of the protocol $\mathbf{\Pi_{Sim}}$. Let m' be a message that is different from m, and define $r = m' - m$. Suppose each node u_{i_j+1} actually generates the random number $r_{i_j+1} + r$, for $j \in [1, k]$. Then, as per the protocol code, for each $j \in [1, k]$, node u_{i_j} sends

$(r_{i_{(j-1)}+1} + r) - (r_{i_j+1} + r)$ to R. This implies, the view of the adversary is $\{(r_{i_{(j-1)}+1} + r) - (r_{i_j+1} + r) | j \in [1,k]\}$, which is nothing but $\{r_{i_{(j-1)}+1} - r_{i_j+1} | j \in [1,k]\}$. This shows that, the view of the adversary when the sender's message is m is the same as the view of the adversary when the sender's message is m', albeit for a different set of random coins of uncorrupted players. As m' is independent of m, the adversary's view is independent of the message being transmitted.

To prove the same mathematically, we individually compute $P[VIEW = v|M = m]$ and $P[VIEW = v|M = m']$ and show that these two probabilities are same.

Let m be the message being transmitted and $v = \{v_1, v_2, \ldots, v_k\}$ be the view of the adversary. Then, for each $j \in [1,k]$, $v_j = r_{i_{(j-1)}+1} - r_{i_j+1}$ if r_{i_j+1} is the random number generated by u_{i_j+1} for each $j \in [1,k]$, and $r_{i_0+1} = m$. This implies:

$$\begin{aligned}
& P\big[VIEW = v | M = m\big] \\
= \; & P\big[(v_1 = r_{i_0+1} - r_{i_1+1}) \text{ and } \ldots \text{ and } (v_k = r_{i_{(k-1)}+1} - r_{i_k+1}) \big| r_{i_0+1} = m\big] \\
= \; & P\big[(v_1 = m - r_{i_1+1}) \text{ and } \ldots \text{ and } (v_k = r_{i_{(k-1)}+1} - r_{i_k+1})\big] \\
= \; & P\big[(r_{i_1+1} = m - v_1) \text{ and } \ldots \text{ and } (r_{i_k+1} = r_{i_{(k-1)}+1} - v_k)\big] \\
= \; & \frac{1}{|\mathbb{F}|^k}
\end{aligned}$$

where the last step is because of k independent events, each one is occurring with probability of $\frac{1}{|\mathbb{F}|}$.

Similarly, let m' be the message being transmitted and $v = \{v_1, v_2, \ldots, v_k\}$ be the view of the adversary. Then, for each $j \in [1,k]$, $v_j = \mu_{i_{(j-1)}+1} - \mu_{i_j+1}$ if μ_{i_j+1} is the random number generated by u_{i_j+1} for each $j \in [1,k]$, and $\mu_{i_0+1} = m'$. This implies:

$$\begin{aligned}
& P\big[VIEW = v | M = m'\big] \\
= \; & P\big[(v_1 = \mu_{i_0+1} - \mu_{i_1+1}) \text{ and } \ldots \text{ and } (v_k = \mu_{i_{(k-1)}+1} - \mu_{i_k+1}) \big| \mu_{i_0+1} = m'\big] \\
= \; & P\big[(v_1 = m' - \mu_{i_1+1}) \text{ and } \ldots \text{ and } (v_k = \mu_{i_{(k-1)}+1} - \mu_{i_k+1})\big] \\
= \; & P\big[(\mu_{i_1+1} = m' - v_1) \text{ and } \ldots \text{ and } (\mu_{i_k+1} = \mu_{i_{(k-1)}+1} - v_k)\big] \\
= \; & \frac{1}{|\mathbb{F}|^k}
\end{aligned}$$

In other words, for every probability distribution on the message space, for every two distinct messages m, m' and every possible view v of the adversary, $P[VIEW = v|M = m] = P[VIEW = v|M = m']$. Therefore the protocol $\mathbf{\Pi_{Sim}}$ is perfectly secure. □

4.1.2 Efficient Protocol

We now present a communication efficient PSMT protocol $\mathbf{\Pi_{Eff}}$ in G if and whenever one exists. Recall that, in *Theorem* 1 Dolev *et al.* [7] have shown that, PSMT from S to R is possible *only if* there exist $(t+1)$ vertex disjoint paths between S and R in G_u. This implies, $t+1$ vertex disjoint weak paths from S to R are necessary for PSMT in G as well. Accordingly, let us assume that there exist $t+1$ vertex disjoint weak paths in G, namely p_i for each $i \in [1, t+1]$. Then, the protocol $\mathbf{\Pi_{Eff}}$ is as follows.

The Protocol $\mathbf{\Pi_{Eff}}$

1. S chooses a random degree-t polynomial $p(x)$ such that the constant term $p(0)$ is the message m being transmitted to R.
2. S sends $p(i)$ to R by simulating the corresponding path p'_i of the weak path p_i using the protocol $\mathbf{\Pi_{Sim}}$, for each $i \in [1, t+1]$.
3. R reconstructs $p(x)$ once it receives all $t+1$ points and gets the message m.

Corollary 1. *The protocol $\mathbf{\Pi_{Eff}}$ is perfectly reliable.*

Proof. The perfect reliability of the protocol $\mathbf{\Pi_{Sim}}$ assures that the receiver gets $t+1$ points on $p(x)$. And, we know that these $t+1$ points are enough to reconstruct $p(x)$ and the message m. □

Corollary 2. *The protocol $\mathbf{\Pi_{Eff}}$ is perfectly secure.*

Proof. We have $t+1$ vertex disjoint weak paths and the adversary can corrupt at most t nodes. Therefore, there exist some $i \in [1, t+1]$ such that no node of the weak path p_i is corrupted. This guarantees (from *Theorem* 2) that the receiver R *reliably* receives the point $p(i)$, whereas the adversary learns nothing about $p(i)$. This implies, in the worst case, the adversary learns at most t points on $p(x)$. And, the rest of the proof directly follows from the Shamir's secret sharing scheme [29]. □

The communication complexity of the protocol $\mathbf{\Pi_{Eff}}$ is $\mathcal{O}(|V|^2)$. This follows from the fact that, $t+1$ weak paths together may contain all the $|V|$ nodes and each of these nodes may need to send a masked value to the receiver R along some path, which in turn may contain $\mathcal{O}(|V|)$ nodes.

Theorem 3. *Let $G(V, E)$ be a directed graph in which S and R are two special nodes and there exists a path from every node to R. Then,* `PSMT` *from S to R is possible in G if and only if* `PSMT` *from S to R is possible in G_u.*

Proof. **Necessity**: If `PSMT` from S to R is not possible in G_u, then clearly `PSMT` from S to R is not possible in G as G is a subgraph of G_u.
Sufficiency: If `PSMT` from S to R is possible in G_u, then the protocol $\mathbf{\Pi_{Eff}}$ guarantees that `PSMT` from S to R is possible in G. □

4.2 Polynomial time algorithm to check if PSMT from S to R is possible in G

In this section, we give a simple (efficient) algorithm to check if `PSMT` from S to R is possible in a given directed graph G tolerating t faults. We know that (from Theorem 3), in G `PSMT` from S to R is possible only if there exist $t+1$ vertex-disjoint weak paths from S to R such that each node of these weak paths has a path to R. Accordingly, we first construct a subgraph G' (of G) by removing each node of G which do not have a path to R in G. Then, we run the max-flow algorithm to check if $t+1$ vertex-disjoint weak paths exist or not from S to R in G', which in turn answers whether `PSMT` from S to R is possible or not in G.

1. `If` either edge $(R, S) \in E$ and there is a path from S to R in G or edge $(S, R) \in E$, then return *true*.

2. `Else`:

   (a) create a (induced) subgraph $G'(V', E')$ of $G(V, E)$, where $V' = V \setminus \{v \in V \mid$ there is no path from v to R in $G\}$ and $E' = \{(u, v) \in E \mid u, v \in V'\}$.

   (b) create an auxiliary graph $G^{aux}(V^{aux}, E^{aux})$ of G' as follows:

      i. Split each vertex $v_i \in V' \setminus \{S, R\}$ into two vertices v_{i1} and v_{i2} and add an edge from v_{i1} to v_{i2}.

      ii. $V^{aux} = \{S, R\} \cup_{v_i \in V' \setminus \{S,R\}} \{v_{i1}, v_{i2}\}$.

      iii. Point all incoming edges of v_i to v_{i1} as incoming edges of v_{i1}.

      iv. Point all out going edges of v_i as out going edges of v_{i2}.

      v. For every edge, add uniform edge capacity of 1.

   (c) In G^{aux} run the *Max-flow* algorithm to find the maximum flow, say f, from S to R.

   (d) If $f \geq t + 1$, then return *true* else return *false*.

This is a polynomial time algorithm as the construction of graph G' requires $\mathcal{O}(|V|^2)$ time and *Max-flow* runs in $\mathcal{O}(|V|^3)$ time (see [12]).

5 Round optimality

This section contributes to the design of a round optimal protocol for perfectly secret message transmission. At first, it appears that the longest among the $t + 1$ disjoint paths from S to R would act as a lower bound for the round complexity of `PSMT`. This is mainly because, to execute a protocol like $\mathbf{\Pi_{Sim}}$, each node needs to wait for the simulation to iteratively reach it, so that it can securely communicate a random number to R. However, recall the Fig. 2 where it is noted that the length of the $(t + 1)^{th}$ shortest path is not necessarily related to the minimum number of rounds required for `PSMT`. Intriguingly, constant round protocols can sometimes exist in very large sparse graphs. This is because the (intermediate) nodes that need to send data to R, need not wait ($\mathbf{\Pi_{Sim}}$-like protocols) to iteratively simulate a secure channel to R – as what is being sent by them is just a random number. Specifically, in $\mathbf{\Pi_{Sim}}$, the receiver R receives the message masked by another random number, which yet again is masked by another random number and so on. R also receives securely (and iteratively) all these random numbers to successively unmask the message. Note that the message can be kept secret as long as none of these secondary/tertiary masks are unmasked. Therefore, all the randomness required for unmasking need not reach R in plain – in fact, it would suffice if (some sort of) a linear combination of them reaches R. This is exactly what we achieve through our protocol $\mathbf{\Pi_{Rnd\_Eff\_Sim}}$ in Section 5.1.

Note that once the bottleneck-of-iteration is circumvented, it is easy to apply the protocol $\mathbf{\Pi_{Rnd\_Eff\_Sim}}$ to obtain a round-efficient `PSMT` protocol $\mathbf{\Pi^{Static}_{Rnd\_Eff}}$ (see Section 5.2) in a manner exactly analogous to how the protocol $\mathbf{\Pi_{Eff}}$ designed using $t + 1$ instances of $\mathbf{\Pi_{Sim}}$.

We remark that our round-efficient protocol is perhaps improvable further; thus the question of round-optimal protocols for `PSMT` is still yet to be fully addressed with the ideas discussed so far and new ideas are needed. Towards that end, we introduce in Section 5.3, the notion of a round

evolution graph, a subgraph of G which evolves as the number of rounds increases. That is, the round evolution graph of order i is a subgraph of the round evolution graph of order $i+1$. Further, the full graph G evolves (in the worst case) when the order number is $|V|$.

Crucially, we prove in *Theorem* 6 that for any round evolution graph of order i, say H_i, if at all any protocol for PSMT exists in H_i then our round efficient protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Eff}}$ is an i-round PSMT protocol in H_i. Thus the smallest i for which our protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Eff}}$ succeeds in securely transmitting the message in H_i is a *round optimal* PSMT protocol. We show that the search for such an i can be easily accomplished via the standard binary-search method. Note that a linear-search would also suffice for our purpose. However, we highlight that the setting is tailor-made for the much faster binary search method. We illustrate our round optimal protocol for the ongoing example.

5.1 Round Efficient Simulation Protocol $\mathbf{\Pi_{Rnd\_Eff\_Sim}}$

The protocol $\mathbf{\Pi_{Rnd\_Eff\_Sim}}$ simulates the corresponding path p' of a weak path p in the least possible number of rounds as each node starts its computation and/or communication from the first round itself; and, if it needs to send anything to R then it sends directly along a shortest path (so that it conveys the required information to R in the least possible number of rounds). Technical details are as follows. Let $p : \langle S(=u_0), u_1, \ldots, u_l, u_{l+1}(=R)\rangle$ be a weak path in G and m be the message that S wishes to send to R along the corresponding path p'. Moreover, let p_{u_i} be a shortest path from u_i to R.

The Protocol $\mathbf{\Pi_{Rnd\_Eff\_Sim}}$
First round:

1. For each $i \in [1, l+1]$: node u_i chooses a random number r_i.
2. $S(=u_0)$ initializes $r_0 = m$ as well as $Left[u_0] = m$.
3. For each $i \in [0, l]$:
   (a) *if* $(u_i, u_{i+1}) \in E$ then:
      i. u_i sends r_i to u_{i+1} and initializes $Right[u_i] = r_i$.
      ii. u_{i+1} receives r_i from u_i sent earlier in this round and initializes $Left[u_{i+1}] = r_i$.
   (b) *else if* $(u_i, u_{i+1}) \notin E$,[a] then:
      i. u_{i+1} sends r_{i+1} to u_i and initializes $Left[u_{i+1}] = r_{i+1}$.
      ii. u_i receives r_{i+1} from u_{i+1} sent earlier in this round and initializes $Right[u_i] = r_{i+1}$.
4. For each $i \in [0, l+1]$: node u_i calculates its *value*, $Val[u_i] = Left[u_i] - Right[u_i]$.

Second round onwards:

1. For each $i \in [0, l]$: If $Val[u_i]$ is non-zero (i.e. $Left[u_i] \neq Right[u_i]$), then in the second round, u_i sends $Val[u_i]$ to its out-neighbour of the shortest path p_{u_i}. In turn, in the third

round, the out-neighbour of u_i forwards $Val[u_i]$ to its out-neighbour of p_{u_i}. This process continues till the the receiver receives $Val[u_i]$ from its in-neighbour of p_{u_i}.

2. In the last round, the receiver R computes $m = (\sum_{i=0}^{l} Val[u_i]) + Left[u_{l+1}]$.

[a] On any weak path, if u and v are two adjacent vertices such that $(u, v) \notin E$ then by definition $(v, u) \in E$.

5.2 Round Efficient Protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Eff}}$

We now present a round efficient PSMT protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Eff}}$ in G if and whenever one exists. We have already seen that, in a directed graph G, PSMT from S to R is possible *only if* there exist $t+1$ vertex disjoint weak paths from S to R in G. Accordingly, let us assume that there are $t+1$ vertex disjoint weak paths, namely p_i, for each $i \in [1, t+1]$. Then the protocol is as follows.

The protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Eff}}$

1. S chooses a random degree-t polynomial $p(x)$ and replaces the constant term $p(0)$ with the message m.
2. S sends $p(i)$ to R by simulating the corresponding path p'_i of the weak path p_i using the protocol $\mathbf{\Pi}_{\mathbf{Rnd\_Eff\_Sim}}$, for each $i \in [1, t+1]$.
3. R reconstructs $p(x)$ once it receives all $t+1$ points and gets the message m.

This protocol terminates in at most $|V|$ rounds. This is because, after sharing random numbers with their neighbours in the first round as per the protocol code, each node u sends $Val[u]$ (if it is non-zero) to R along the shortest path p_u. In any graph, as the length of every shortest path is trivially bounded by $|V| - 1$, overall our protocol can take up to $|V|$ rounds.

Now we prove the correctness of the protocols $\mathbf{\Pi}_{\mathbf{Rnd\_Eff\_Sim}}$ and $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Eff}}$.

Theorem 4. *The protocol $\mathbf{\Pi}_{\mathbf{Rnd\_Eff\_Sim}}$ for sending message m from S to R is perfectly reliable.*

Proof. By our protocol design, we have $Right[u_i] = Left[u_{i+1}]$ for each node u_i (except R) on the weak path p. As R finally computes the $Sum = (\sum_{i=0}^{l} Val[u_i]) + Left[u_{l+1}]$, we show that the Sum is nothing but m, which ensures perfect reliability.

$$\begin{aligned} Sum &= (\sum_{i=0}^{l} (Left[u_i] - Right[u_i])) + Left[u_{l+1}] \\ &= (\sum_{i=0}^{l} (Left[u_i] - Left[u_{i+1}])) + Left[u_{l+1}] \\ &= Left[u_0] - Left[u_{l+1}] + Left[u_{l+1}] = Left[u_0] = m \end{aligned}$$

□

Corollary 3. *The protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Eff}}$ for sending message m from S to R is perfectly reliable.*

Proof. The perfect reliability of the protocol $\mathbf{\Pi_{Rnd\_Eff\_Sim}}$ ensures that R gets $t+1$ points on degree-t polynomial $p(x)$. And, we know that these $t+1$ points on $p(x)$ are enough to get the message m [29]. □

Theorem 5. *The protocol $\mathbf{\Pi_{Rnd\_Eff\_Sim}}$ for simulating the corresponding path p' of a weak path $p : \langle S(=u_0), u_1, \ldots, u_l, u_{l+1}(=R)\rangle$, secretly transmits the message m from S to R if no node of p is corrupted.*

Proof. Proof is analogous to the proof given in *Theorem* 2. We notice that, other than R, each node u_i on the weak path p sends $Val[u_i]$ (if it is non-zero) to the receiver R along the shortest path p_{u_i}. In the worst case, the adversary may learn $Val[u_i]$, for each $i \in [0, l]$. In this case too, we show that the adversary learns nothing (additional) about m by showing that the view of the adversary is independent of the message being transmitted.

In the execution of the protocol $\mathbf{\Pi_{Rnd\_Eff\_Sim}}$ for the sender's message m, the view of the adversary is $\{Val[u_i] | i \in [0, l]\}$, where $Left[u_0] = m$ and $Val[u_i] = Left[u_i] - Right[u_i] = Left[u_i] - Left[u_{i+1}]$. Let us denote $Left[u_i] = r'_i$ for each $i \in [0, l+1]$, thus the view of the adversary is $\{r'_i - r'_{i+1} | i \in [0, l]\}$.

Consider the following *valid* execution of the protocol $\mathbf{\Pi^{Static}_{Rnd\_Eff}}$. Let m' be a message that is different from m, and define $r = m' - m$. Suppose the node u_i actually generates the random number $r_i + r$, for each $i \in [1, l+1]$. Then, as per the protocol code, for each $i \in [0, l]$, node u_i sends $(r'_i + r) - (r'_{i+1} + r)$ to R. This implies, the view of the adversary is $\{(r'_i + r) - (r'_{i+1} + r) | i \in [0, l]\}$, which is nothing but $\{r'_i - r'_{i+1} | i \in [0, l]\}$. The rest of the proof follows exactly as in the proof of the *Theorem* 2. Therefore, the protocol $\mathbf{\Pi_{Rnd\_Eff\_Sim}}$ is perfectly secure. □

Corollary 4. *The protocol $\mathbf{\Pi^{Static}_{Rnd\_Eff}}$ for sending message m from S to R is perfectly secure.*

Proof. As the adversary can corrupt at most t nodes, there exists $i \in [1, t+1]$, such that no node of the weak path p_i is corrupted. And, the protocol $\mathbf{\Pi_{Rnd\_Eff\_Sim}}$ assures that $p(i)$ is perfectly secure. We have from Shamir's secret sharing scheme that t or fewer points on a degree-t polynomial reveal nothing about the constant term, which is the message. □

5.3 PSMT in Round Evolution Graphs

Graphs have been used as a very powerful abstraction of the network by modelling the physical link from one player to another as a directed edge between the corresponding vertices of the graph. However, in this kind of modelling of the network, the edges of the graph only indicate the link between two spatial locations. It does not contain any temporal information. To incorporate the notion of time (rounds) in our graph, we propose a representation named *round evolution graph* that contains both spatial and temporal information.

Definition 15. *Let $G(V, E)$ be a directed graph in which R is a special node such that there exists a path from every node to R. Then, given a round number r, the round evolution graph $G^{(r)}(V, E^{(r)})$ of order r is a subgraph of G, defined as (edge set) $E^{(r)} = E \setminus \{(u, v) \in E \mid d_v \geq r\}$, where d_v denotes the length of the shortest path from v to R. In other words, remove those edges from which R can not receive any information in r rounds.*

Theorem 6. *Let $G(V, E)$ be a directed graph in which S and R are two special nodes and there exists a path from every node to R. Then,* `PSMT` *from S to R is possible in $G^{(r)}$ if and only if an r-round* `PSMT` *protocol (from S to R) exists in $G^{(r)}$.*

Proof. **Sufficiency**: If an r-round PSMT protocol (from S to R) exists in $G^{(r)}$, then PSMT from S to R is trivially possible in $G^{(r)}$.
Necessity: Suppose PSMT from S to R is possible in $G^{(r)}$, then we show that the *round efficient protocol* $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Eff}}$ given in Section 5.2 achieves PSMT (from S to R) in r rounds. As the protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Eff}}$ is nothing but executing $t+1$ times the protocol $\mathbf{\Pi}_{\mathbf{Rnd\_Eff\_Sim}}$, it is enough to show that the protocol $\mathbf{\Pi}_{\mathbf{Rnd\_Eff\_Sim}}$ succeeds in r-rounds. In other words, it is enough to show that every node u_i can send the required information to R in r-rounds (which implies, R can reconstruct the message in r-rounds).

We observe that, each node u_i on the weak path $p : \langle S(= u_0), u_1, \ldots, u_l, u_{l+1}(= R)\rangle$, (if required) sends the chosen random number r_i to its neighbour(s) in the first round as per the protocol $\mathbf{\Pi}_{\mathbf{Rnd\_Eff\_Sim}}$. We have three cases for each node u_i of the weak path p:

1. If $(u_{i-1}, u_i) \in E^{(r)}$, then by our construction of $G^{(r)}$ we have $d_{u_i} \leq r-1$. Therefore, even if u_i takes one round (entire first round) to receive random numbers from its neighbour(s), it can send $Val[u_i]$ to R in a total of r-rounds.

2. If $(u_i, u_{i+1}) \notin E^{(r)}$, then by definition $(u_{i+1}, u_i) \in E^{(r)}$. Moreover, by our construction of $G^{(r)}$ we have $d_{u_i} \leq r-1$. The rest follows as in previous case.

3. If $(u_{i-1}, u_i) \notin E^{(r)}$ but $(u_i, u_{i+1}) \in E^{(r)}$, then $Val[u_i] = Left[u_i] - Right[u_i] = r_i - r_i = 0$. This implies, u_i is not required to send its *value* to the receiver R as per the protocol code.

□

Theorem 7. *Let $G(V, E)$ be a directed graph in which S and R are two special nodes and there exists a path from every node to R. Then, an r-round* PSMT *protocol (from S to R) exists in G if and only if* PSMT *from S to R is possible in the* round evolution graph $G^{(r)}$ *of order r.*

Proof. **Sufficiency**: If PSMT from S to R is possible in $G^{(r)}$, then, the theorem directly follows from *Theorem* 6 as $G^{(r)}$ is a subgraph of G.
Necessity: Assume that an r-round PSMT protocol Π exists in G. We show that for the same protocol Π, the *extra* edges which are present in E but not in $E^{(r)}$ never convey *any* information to R. This implies, at the end of the protocol Π, the view of the receiver R remains the same whether these edges are present or not. Therefore, any such r-round protocol Π achieves PSMT in $G^{(r)}$.

Let (u, v) be an edge in E but not in $E^{(r)}$. This implies, by definition of $E^{(r)}$, $d_v \geq r$. As the shortest distance from v to R is at least r, any message sent by v takes at least r rounds to reach R. Also we know that, if u sends a message m to v along the edge (u, v) then by definition one round is required for m to reach v. Therefore, a total of at least $r+1$ rounds are required for any message to reach R from u via edge (u, v). Therefore, these edges are of no use in any r-round protocol. This concludes the proof. □

Corollary 5. *Let $G(V, E)$ be a directed graph in which S and R are two special nodes and there exists a path from every node to R. Then, an r-round* PSMT *protocol (from S to R) exists in G if and only if an r-round* PSMT *protocol (from S to R) exists in $G^{(r)}$.*

Proof. Directly follows from *Theorem* 6 and *Theorem* 7. □

5.4 Polynomial time algorithm for identifying the optimal number of rounds

We have from *Corollary* 5 that the optimal number of rounds required for PSMT from S to R in G is nothing but the least r for which PSMT from S to R is possible in $G^{(r)}$. Also, it is easy to see that if PSMT from S to R is possible in $G^{(r)}$, then PSMT from S to R is trivially possible in $G^{(r+1)}$ (as $G^{(r)}$ is a subgraph of $G^{(r+1)}$). Combining these two together, we get the *optimal* r for which PSMT is possible in G if we perform standard binary search over $r \in [1, |V|]$ while we check for PSMT possibility from S to R in $G^{(r)}$. Since the overhead of binary search is logarithmic, we just need to show that each iteration of binary search takes only polynomial time. In each iteration, we are constructing the subgraph $G^{(r)}$ of G and checking if PSMT from S to R is possible in $G^{(r)}$. Constructing a subgraph $G^{(r)}$ of G requires only quadratic (polynomial) time. And, in Section 4.2, we have already shown that we can efficiently check if PSMT from S to R is possible in any given graph.

5.5 An Example of Round Optimal Protocol

In this section, we illustrate the execution protocol $\mathbf{\Pi^{Static}_{Rnd\_Eff}}$ with an example. Let us consider the graph G given in Fig. 4. We have already seen that, in G PSMT from S to R is possible tolerating two faults but not three.

We need a shortest path from each node to R to execute our round optimal protocol. Also, we have to find the least r for which PSMT from S to R is possible in $G^{(r)}$ tolerating two faults. For quick reference, the shortest distance and a shortest path from each node to R is shown in Fig. 6. And, *round evolution graphs* $G^{(3)}$ of order three and $G^{(4)}$ of order four are depicted in Fig. 7 and Fig. 8 respectively.

We notice that the shortest distance from S to R is three. Therefore, any protocol will take at least three rounds. However, in $G^{(3)}_u$ there is only one vertex disjoint path from S to R. Thus, it fails to meet the necessary conditions of *Theorem* 3 tolerating two faults. This implies that PSMT from S to R is impossible in $G^{(3)}$. On the other hand, there are three vertex disjoint paths from S to R in $G^{(4)}_u$, and every node on these paths has a path to R in G. Thus, PSMT from S to R is possible in $G^{(4)}$ tolerating two faults. Therefore, the minimum number of rounds required for PSMT in G is four.

Now we execute our protocol $\mathbf{\Pi^{Static}_{Rnd\_Eff}}$ in $G^{(4)}$, which achieves PSMT tolerating two faults in four rounds as follows.

An execution of four round protocol in $\mathbf{G^{(4)}}$

First round:

1. The sender S chooses a random degree-2 polynomial $p(x)$ and replaces the constant term $p(0)$ with the message m.
2. Every node v, except S and R, chooses a random number r_v.
3. R chooses three random numbers $r_{R_1}, r_{R_2}, r_{R_3}$.

Node	Shortest path to R	Shortest distance
S	$p_S : \langle S, v_3, v_2, R\rangle$	3
v_1	$p_{v_1} : \langle v_1, v_2, R\rangle$	2
v_2	$p_{v_2} : \langle v_2, R\rangle$	1
v_3	$p_{v_3} : \langle v_3, v_2, R\rangle$	2
v_4	$p_{v_4} : \langle v_4, v_3, v_2, R\rangle$	3
v_5	$p_{v_5} : \langle v_5, v_6, v_3, v_2, R\rangle$	4
v_6	$p_{v_6} : \langle v_6, v_3, v_2, R\rangle$	3

Figure 6: Shortest paths from each node to the receiver R for the graph given in Fig. 4.

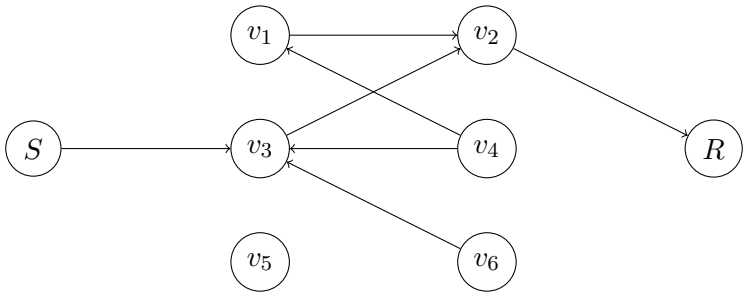

Figure 7: Round Evolution Graph $G^{(3)}$ of order three for the graph given in Fig. 4.

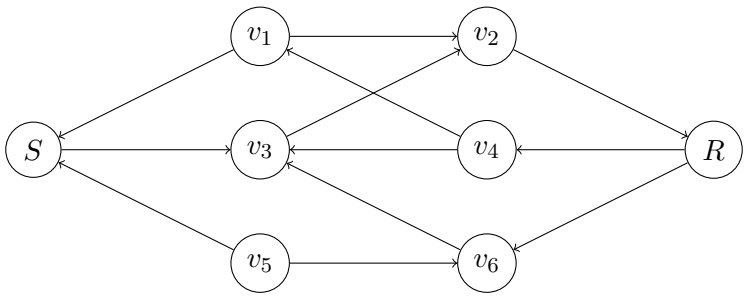

Figure 8: Round Evolution Graph $G^{(4)}$ of order four for the graph given in Fig. 4.

4. The node v_1 sends r_{v_1} to the node v_2 as well as the sender $S(= S_1)$ and both the nodes v_2 and $S(= S_1)$ receive r_{v_1}.

5. The node v_2 sends r_{v_2} to the receiver R and R receives r_{v_2}.

6. The node $S(= S_2)$ sends $p(2)$ to the node v_3 and v_3 receives $p(2)$.

7. The node v_4 sends r_{v_4} to the node v_3 and v_3 receives r_{v_4}.

8. The receiver R sends r_{R_2} to the node v_4 and v_4 receives r_{R_2}.

9. The node v_5 sends r_{v_5} to the node v_6 as well as the sender $S(= S_3)$ and both the nodes v_6 and $S(= S_3)$ receive r_{v_5}.

10. The receiver R sends r_{R_3} to the node v_6 and v_6 receives r_{R_3}.

11. Every node v, except R, calculates its *value*, $Val[v]$:

    (a) $Val[S_1] = p(1) - r_{v_1}$, $Val[v_1] = r_{v_1} - r_{v_1}$, $Val[v_2] = r_{v_1} - r_{v_2}$
    (b) $Val[S_2] = p(2) - p(2)$, $Val[v_3] = p(2) - r_{v_4}$, $Val[v_4] = r_{v_4} - r_{R_2}$
    (c) $Val[S_3] = p(3) - r_{v_5}$, $Val[v_5] = r_{v_5} - r_{v_5}$, $Val[v_6] = r_{v_5} - r_{R_3}$

Second round:

1. The node S_1 sends its *value* $Val[S_1]$ to the node v_3 and v_3 receives $Val[S_1]$.

2. The node S_3 sends its *value* $Val[S_3]$ to the node v_3 and v_3 receives $Val[S_3]$.

3. The node v_2 sends its *value* $Val[v_2]$ to the receiver R and R receives $Val[v_2]$.

4. The node v_3 sends its *value* $Val[v_3]$ to the node v_2 and v_2 receives $Val[v_3]$.

5. The node v_4 sends its *value* $Val[v_4]$ to the node v_3 and v_3 receives $Val[v_4]$.

6. The node v_6 sends its *value* $Val[v_6]$ to the node v_3 and v_3 receives $Val[v_6]$.

Third round:

1. The node v_2 sends $Val[v_3]$ to the receiver R and R receives $Val[v_3]$.

2. The node v_3 sends $Val[S_1]$, $Val[S_3]$, $Val[v_4]$ and $Val[v_6]$ to the node v_2 and v_2 receives $Val[S_1]$, $Val[S_3]$, $Val[v_4]$ and $Val[v_6]$.

Fourth round:

1. The node v_2 sends $Val[S_1]$, $Val[S_3]$, $Val[v_4]$ and $Val[v_6]$ to the receiver R and R receives $Val[S_1]$, $Val[S_3]$, $Val[v_4]$ and $Val[v_6]$.

2. R computes:

(a) $p(1) = Val[S_1] + Val[v_1] + Val[v_2] + r_{v_2} = p(1) - r_{v_1} + 0 + r_{v_1} - r_{v_2} + r_{v_2} = p(1)$.

(b) $p(2) = Val[S_2] + Val[v_3] + Val[v_4] + r_{R_2} = 0 + p(2) - r_{v_4} + r_{v_4} - r_{R_2} + r_{R_2} = p(2)$.

(c) $p(3) = Val[S_3] + Val[v_5] + Val[v_6] + r_{R_3} = p(3) - r_{v_5} + 0 + r_{v_5} - r_{R_3} + r_{R_3} = p(3)$.

3. Finally R reconstructs the polynomial and hence gets the message.

The shortest distance from each node (except v_5) to the receiver R is less than or equal to three. And, each node in G may have to receive random number(s) from its neighbour(s) in the first round. Therefore, each node (except v_5) can send its *value* to the receiver R in at most four rounds. As per the protocol code, v_5 does not send anything to R since the *value* of the node v_5 is zero. Therefore, this protocol terminates in four rounds.

6 Linear communication complexity

This section contributes to the design of a round optimal PSMT protocol, whose communication complexity is *linear* in the number of *vertices* of the graph. In Section. 4.1.2, we have seen that the communication complexity of the protocol $\mathbf{\Pi_{Eff}}$ is $\mathcal{O}(|V|^2)$ due to the following reason. Once sharing (of random numbers) is done in the first round, each node sends its *value* to the receiver along a shortest path. In particular, each of these shortest paths may contain $\mathcal{O}(|V|)$ nodes, leading to quadratic complexity. However, we notice that, many of these shortest paths may have several edges in common. And, each such edge has to carry k field elements if it is part of k shortest paths. We make sure that such edges carry only one field element, leading us to the design of a linear-communication protocol. More details are as follows. We construct a subgraph H of G such that PSMT from S to R is possible in G *iff* PSMT from S to R is possible in H. And, H contains only $\mathcal{O}(|V|)$ edges. Therefore, if we design a protocol in H such that each edge in H carries *at most one* field element then trivially we get a linear communication protocol in G. As this technique can be adapted to any graph, we work with the round evaluation graph $G^{(r)}$ of order r, where r is the optimal number of rounds required for PSMT; to get a round optimal protocol with linear communication complexity. The graph H holds the following properties:

1. H contains only $\mathcal{O}(|V|)$ edges – The edge set of H is the union of two sets of edges. First one is the set of edges of $t+1$ vertex disjoint weak paths. And, the other one is the set of edges of a tree with R as its root. More elaborately, suppose the shortest distance from a node u to R is d. Then, to send any information to R possibly in least number of rounds, the node u must use another node whose shortest distance to R is $d-1$. We realize this by constructing a tree T of G with R as its root such that each node has exactly one path to R in the tree T. That is, a node in the i^{th} level connected to only one of its parent which is in the $(i-1)^{th}$ level, assuming R is at 0^{th} level. We already know that, no tree can have more than $|V|-1$ edges. Therefore, H contain only $\mathcal{O}(|V|)$ edges.

2. PSMT from S to R is possible in H whenever PSMT from S to R is possible in G – If PSMT from S to R is possible in G then G contains $t+1$ disjoint weak paths. As H contains every edge of these $t+1$ disjoint weak paths, nodes can share their random numbers with neighbours in the first round as per the protocol $\mathbf{\Pi^{Static}_{Rnd\_Eff}}$. Also, each node can send its *value* to R as it has a *shortest* path to R in the tree.

3. Each edge in H carries at most one field element – Instead of working with random degree-t polynomial, the sender randomly chooses $t+1$ field elements, say m_i for $i \in [1, t+1]$, such that their sum is the message m. To get the message m, it is *not necessary* for R to know each individual m_i but it is enough if R gets the corresponding sum. Therefore, instead of pushing the calculation to R at the end, each node locally adds all the *values* it received from its children with its *value* and sends as a single field element to its parent in the tree T. In other words, if an edge is part of multiple shortest paths, then instead of carrying multiple messages, it carries only one field element which is the sum of the corresponding multiple messages.

Now we are ready to formally introduce required definitions.

Definition 16. *Let $G(V, E)$ be a directed graph in which R is a special node such that there exists a path from every node to R. Then, a **Reverse Directed Rooted Tree** of G rooted at R is a digraph $T_G(V, E_T; R)$ such that a node u is at the i^{th} level (R is at level 0) if and only if the shortest distance from u to R is exactly i in G.*

A Note on Reverse Directed Rooted Tree: Every node in the tree has exactly one parent, else we would get cycles in tree T_G. Moreover, as there are no cycles, the maximum number of edges present in tree T_G is $|V| - 1$.

Definition 17. *Let $G(V, E)$ be a directed graph in which S and R are two special nodes and there exist k vertex disjoint weak paths from S to R, namely p_i for each $i \in [1, k]$ such that every node in these k weak paths has at least one path to R. Moreover, let $T_G(V, E_T; R)$ be a Reverse Directed Rooted Tree of G. Then, a communication graph of the digraph $G(V, E)$ of order k is denoted by $\mathcal{G}^k(V, \mathcal{E})$ and defined as $\mathcal{E} = E_p \cup E_T$, where $E_p = \bigcup_{i=1}^{k} E(p_i)$ and $E(p_i)$ is the set of all edges in the weak path p_i.*

Theorem 8. *Let $G(V, E)$ be a directed graph in which S and R are two special nodes and there exists a path from every node to R. Then,* PSMT *from S to R is possible in G if and only if* PSMT *from S to R is possible in communication graph $\mathcal{G}^{(t+1)}$ of order $t+1$.*

Proof. **Sufficiency**: Suppose PSMT from S to R is possible in G. Then from *Theorem* 3, we know that there exist at least $t+1$ vertex disjoint weak paths from S to R such that every node on these weak paths has a path to R in G. Observe that, by definition of $\mathcal{G}^{(t+1)}$, every edge of these $t+1$ weak paths is present in $\mathcal{G}^{(t+1)}$. Therefore, along these edges, nodes can share their random numbers with their neighbours in the first round as per the protocol $\mathbf{\Pi^{Static}_{Rnd\_Eff}}$. Also, by the definition of Reverse Directed Rooted Tree of G, each node on these $t+1$ weak paths has a (unique) shortest path to R in T_G. Therefore, each node on these $t+1$ weak paths can send its *value* to R as per protocol $\mathbf{\Pi^{Static}_{Rnd\_Eff}}$.
Necessity: If PSMT from S to R is impossible in G then trivially PSMT from S to R is impossible in the subgraph $\mathcal{G}^{(t+1)}$ as well. □

6.1 Round optimal protocol with linear communication complexity

In this section, we present a round optimal linear communication protocol $\mathbf{\Pi^{Static}_{Rnd\_Opt\_Lin}}$ if PSMT from S to R is possible in G. As we discussed earlier, the protocol $\mathbf{\Pi^{Static}_{Rnd\_Opt\_Lin}}$ is same as the

protocol $\mathbf{\Pi^{Static}_{Rnd\_Eff}}$ except that (1) we run the protocol in $\mathcal{G}^{(t+1)}$ (2) for each $i \in [1, t+1]$, $p(i)$ is replaced with m_i, where the sum of these m_i's is the message m and (3) if an edge (u, v) carries more than one field element then u adds corresponding field elements and sends to its parent v in T as a single field element. Now, we move to the technical details of the protocol.

Let r be the optimal number of rounds required for PSMT possibility from S to R in G tolerating t-threshold static adversary. Then, we have from *Corollary* 5 that, PSMT from S to R is possible in $G^{(r)}$. This implies, combing with *Theorem* 8, PSMT from S to R is possible in communication graph $\mathcal{G}^{(t+1)}$ of the digraph $G^{(r)}(V, E)$. Therefore, $(t+1)$ vertex disjoint weak paths from S to R exist in $\mathcal{G}^{(t+1)}$, namely $p_i : \langle u_{i0}(= S), u_{i1}, \ldots, u_{ik_i}, u_{i(k_i+1)}(= R)\rangle$, for each $i \in [1, t+1]$. Moreover, let the height of a Reverse Directed Rooted Tree $T_{G^{(r)}}$ of $G^{(r)}$ be h with root R is at the 0^{th} level. Here we notice that, each u_{i0} is S and each $u_{i(k_i+1)}$ is R.

The Protocol $\mathbf{\Pi^{Static}_{Rnd\_Opt\_Lin}}$

First round:

1. For each $i \in [1, t+1]$ and $j \in [0, k_i+1]$: node u_{ij}, except $u_{(t+1)0}$, picks a random number $r_{ij} \in \mathbb{F}$.
2. The sender $S(= u_{(t+1)0})$ computes $r_{(t+1)0} = m - \sum_{i=1}^{t} r_{i0}$.
3. For each $i \in [1, t+1]$: $S(= u_{i0})$ initializes $Left[u_{i0}] = r_{i0}$.
4. For each $i \in [1, t+1]$ and $j \in [0, k_i]$:
   (a) if $(u_{ij}, u_{i(j+1)}) \in E^{(r)}$, then:
      i. u_{ij} sends r_{ij} to $u_{i(j+1)}$ and initializes $Right[u_{ij}] = r_{ij}$.
      ii. $u_{i(j+1)}$ receives r_{ij} from u_{ij} sent earlier in this round and initializes $Left[u_{i(j+1)}] = r_{ij}$.
   (b) if $(u_{ij}, u_{i(j+1)}) \notin E^{(r)}$, then:
      i. $u_{i(j+1)}$ sends $r_{i(j+1)}$ to u_{ij} and initializes $Left[u_{i(j+1)}] = r_{i(j+1)}$.
      ii. u_{ij} receives $r_{i(j+1)}$ from $u_{i(j+1)}$ sent earlier in this round and initializes $Right[u_{ij}] = r_{i(j+1)}$.
5. For each $i \in [1, t+1]$ and $j \in [0, k_i]$: node u_{ij} calculates its *value*, $Val[u_{ij}] = Left[u_{ij}] - Right[u_{ij}]$.
6. The sender S computes its *grand value*, $Val[S] = \sum_{i=1}^{t+1} Val[u_{i0}]$.

Second round onwards:

1. If S is a leaf node (at level h) in $T_{G^{(r)}}$ and its *grand value*, $Val[S]$, is non zero then S sends $Val[S]$ to its parent which is at $(h-1)^{th}$ level. And, the parent of S receives $Val[S]$.

2. `Else If` S is at k^{th} level ($k \in [1, h-1]$) then S receives *values*, which are non-zero, from its children (sent earlier in that round) which are at $(k+1)^{th}$ level. Subsequently, S *adds* all the received *values* to its *grand value* $Val[S]$ and sends to its parent which is at $(k-1)^{th}$ level. And, the parent of S receives the corresponding sum.

3. For each $i \in [1, t+1]$ and $j \in [1, k_i]$:

   (a) `If` u_{ij} is a leaf node (at level h) in $T_{G^{(r)}}$ and its *value*, $Val[u_{ij}]$, is non zero then u_{ij} sends $Val[u_{ij}]$ to its parent which is at $(h-1)^{th}$ level. And, the parent of u_{ij} receives $Val[u_{ij}]$.

   (b) `If` u_{ij} is at k^{th} level for some $k \in [1, h-1]$ (not a leaf node) then u_{ij} receives *values*, which are non-zero, from its children (sent earlier in that round) which are at $(k+1)^{th}$ level. Subsequently, u_{ij} *adds* all the received values to its *value* $Val[u_{ij}]$ and sends to its parent which is at $(k-1)^{th}$ level. And, the parent of u_{ij} receives the corresponding sum.

4. In the last round, the receiver R adds the sum of all the values it received from its children with the sum of all its *Left Values* (i.e. $\sum_{i=1}^{t+1} Left[u_{i(k_i+1)}]$) to get the message m.

Now we prove the correctness of the protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Opt\_Lin}}$ in the following two theorems.

Theorem 9. *The protocol* $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Opt\_Lin}}$ *is perfectly reliable.*

Proof. It is clear that the receiver R eventually gets the sum of the *grand value* of S and the sum of the *values* of each u_{ij}, for $i \in [1, t+1]$ and $j \in [1, k_i]$. As R computes the $Sum = Val[S] + (\sum_{i=1}^{t+1}\sum_{j=1}^{k_i} Val[u_{ij}]) + \sum_{i=1}^{t+1} Left[u_{i(k_i+1)}]$ to get the message, we should show that the Sum is nothing but the message m. Recall that, for each u_{ij} we have $Right[u_{ij}] = Left[u_{i(j+1)}]$, where $i \in [1, t+1]$ and $j \in [0, k_i]$.

$$
\begin{aligned}
Sum &= Val[S] + \Big(\sum_{i=1}^{t+1}\sum_{j=1}^{k_i} Val[u_{ij}]\Big) + \sum_{i=1}^{t+1} Left[u_{i(k_i+1)}] \\
&= \sum_{i=1}^{t+1} Val[u_{i0}] + \Big(\sum_{i=1}^{t+1}\sum_{j=1}^{k_i} Val[u_{ij}]\Big) + \sum_{i=1}^{t+1} Left[u_{i(k_i+1)}] \\
&= \sum_{i=1}^{t+1} \big(Left[u_{i0}] - Right[u_{i0}]\big) + \Big(\sum_{i=1}^{t+1}\sum_{j=1}^{k_i} \big(Left[u_{ij}] - Right[u_{ij}]\big)\Big) + \sum_{i=1}^{t+1} Left[u_{i(k_i+1)}] \\
&= \sum_{i=1}^{t+1} \big(Left[u_{i0}] - Left[u_{i1}]\big) + \Big(\sum_{i=1}^{t+1}\sum_{j=1}^{k_i} \big(Left[u_{ij}] - Left[u_{i(j+1)}]\big)\Big) + \sum_{i=1}^{t+1} Left[u_{i(k_i+1)}] \\
&= \sum_{i=1}^{t+1} \big(Left[u_{i0}] - Left[u_{i1}]\big) + \Big(\sum_{i=1}^{t+1} \big(Left[u_{i1}] - Left[u_{i(k_i+1)}]\big)\Big) + \sum_{i=1}^{t+1} Left[u_{i(k_i+1)}]
\end{aligned}
$$

$$= \sum_{i=1}^{t+1} Left[u_{i0}] - \sum_{i=1}^{t+1} Left[u_{i1}] + \sum_{i=1}^{t+1} Left[u_{i1}] - \sum_{i=1}^{t+1} Left[u_{i(k_i+1)}] + \sum_{i=1}^{t+1} Left[u_{i(k_i+1)}]$$
$$= \sum_{i=1}^{t+1} Left[u_{i0}] = \sum_{i=1}^{t+1} r_{i0} = m.$$

□

Theorem 10. *The protocol* $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Opt\_Lin}}$ *is perfectly secure.*

Proof. The proof directly follows from the security proof of the protocol $\mathbf{\Pi}_{\mathbf{Rnd\_Eff\_Sim}}$ given in *Theorem* 5. In *Theorem* 5, we showed that, once sharing of random numbers is done in the first round, in subsequent rounds even if the adversary gets $Val[u_i]$, for each uncorrupted node u_i of the weak path p, the adversary learns nothing (additional) about the message m being transmitted to R. The protocol code of the first round of the current protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Opt\_Lin}}$ is same as that of the protocol $\mathbf{\Pi}_{\mathbf{Rnd\_Eff\_Sim}}$. Also, as we have $t+1$ vertex disjoint weak paths from S to R, there exists at least one weak path p_i such that no node of it is corrupted, for some $i \in [1, t+1]$. Combining all together, we get, the adversary learns nothing about r_{i0}. Thus, the adversary learns nothing (additional) about the message $m = \sum_{j=1}^{t+1} r_{j0}$. □

The communication complexity of the protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Opt\_Lin}}$ is linear. The reason is as follows. In the first round, each edge of the weak paths carries exactly one field element r_{ij}, for some $i \in [1, t+1]$ and $j \in [0, k_i+1]$. As the weak paths are disjoint, the number of edges is bounded by $|V|$. Also, each edge of the Reverse Directed Rooted Tree $T_{G^{(r)}}$ carries at most one field element. And, the number of edges in $T_{G^{(r)}}$ is also bounded by $|V|-1$. Therefore, to transmit a single field element secretly, all the edges together carry at most $\mathcal{O}(|V|)$ field elements.

An interesting implication of this protocol is the following. If the shortest distance from $S$to R is $\Omega(|V|)$, then we achieve perfect secrecy for *free*. Because any reliable but insecure routing protocol would also takes $\mathcal{O}(|V|)$ rounds and send $\mathcal{O}(|V|)$ messages (one message along each edge in the shortest path) for transmission.

6.2 An example of the round optimal protocol with linear communication complexity

In this section we illustrate the protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Opt\_Lin}}$ with an example. Consider the graph G given in Fig. 4. In earlier section, we have seen that the minimum number of rounds required for `PSMT` possibility in G is four. Accordingly, we execute the protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Opt\_Lin}}$ on communication graph $\mathcal{G}^{(3)}$ of the graph $G^{(4)}$ given in Fig. 8. We represent $\mathcal{G}^{(3)}$ with three vertex disjoint weak paths and a Reverse Directed Rooted Tree $T_{G^{(4)}}$ rooted at R in Fig. 9. Furthermore, a value sent by a node u to a node v as per the protocol code, is depicted over an edge $(u, v) \in E$. The first round computation of the protocol, that is, sharing of random numbers and calculating corresponding *values*, is depicted at the top of the Fig. 9 using three disjoint weak paths. Whereas, the computations of the second and subsequent rounds are depicted at the bottom using the tree $T_{G^{(4)}}$.

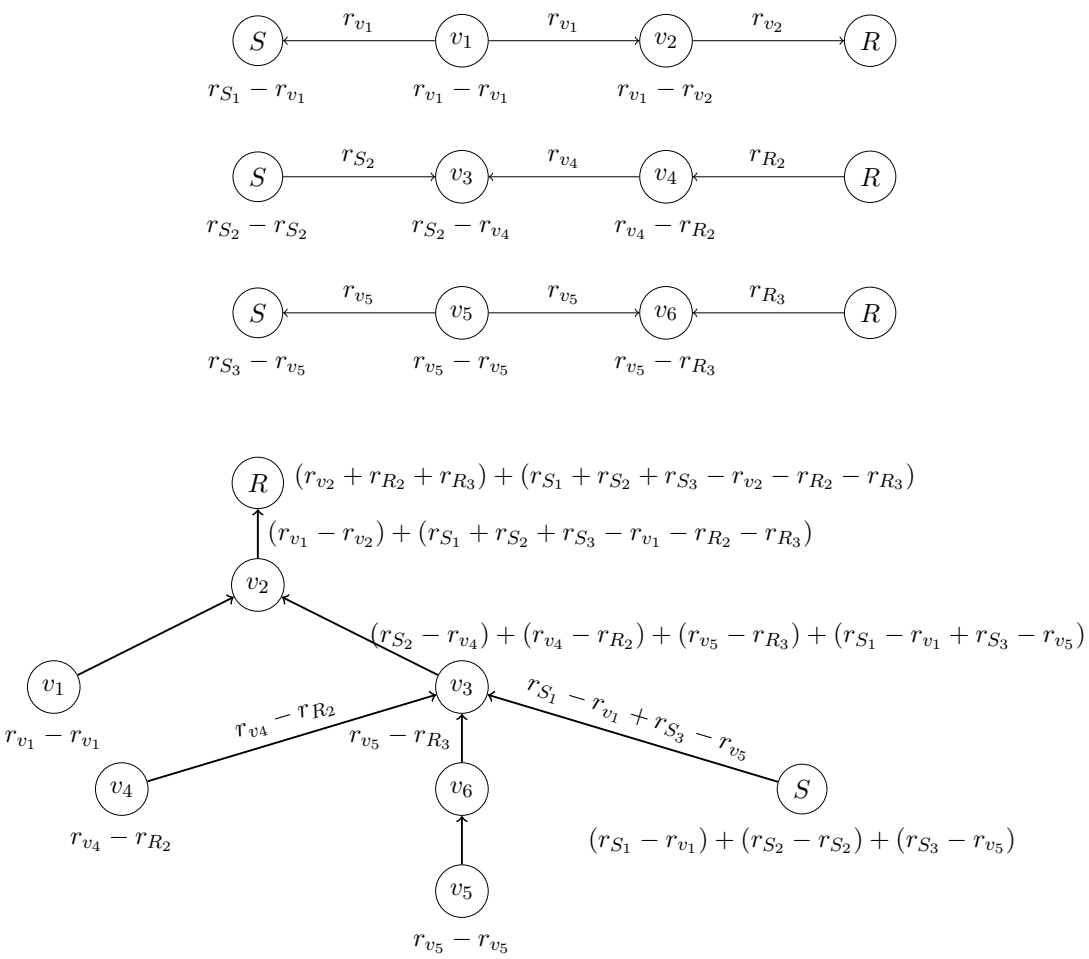

Figure 9: An execution of the protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Opt\_Lin}}$ on $\mathcal{G}^{(3)}$ of the graph $G^{(4)}$ given in Fig. 8

An execution of the protocol $\Pi^{\mathbf{Static}}_{\mathbf{Rnd\_Opt\_Lin}}$ on $\mathcal{G}^{(3)}$

First round:

1. The sender S chooses two random numbers r_{S_1}, r_{S_2} and initializes $r_{S_3} = m - (r_{S_1} + r_{S_2})$.
2. Every node v, except S and R, chooses a random number r_v.
3. The receiver R chooses three random numbers $r_{R_1}, r_{R_2}, r_{R_3}$.
4. The node v_1 sends r_{v_1} to the node v_2 as well as the sender S and both the nodes v_2 and S receive r_{v_1}.
5. The node v_2 sends r_{v_2} to the receiver R and R receives r_{v_2}.
6. The node S sends r_{S_2} to the node v_3 and v_3 receives r_{S_2}.
7. The node v_4 sends r_{v_4} to the node v_3 and v_3 receives r_{v_4}.
8. The receiver R sends r_{R_2} to the node v_4 and v_4 receives r_{R_2}.
9. The node v_5 sends r_{v_5} to the node v_6 as well as the sender S and both the nodes v_6 and S receive r_{v_5}.
10. The receiver R sends r_{R_3} to the node v_6 and v_6 receives r_{R_3}.
11. The sender S computes its *grand value*, $Val[S] = (r_{S_1} - r_{v_1}) + (r_{S_2} - r_{S_2}) + (r_{S_3} - r_{v_5}) = r_{S_1} - r_{v_1} + r_{S_3} - r_{v_5}$.
12. Every node v, except S and R, calculates its *value* $Val[v]$:
    (a) $Val[v_1] = r_{v_1} - r_{v_1}$, $Val[v_2] = r_{v_1} - r_{v_2}$ and $Val[v_3] = r_{S_2} - r_{v_4}$.
    (b) $Val[v_4] = r_{v_4} - r_{R_2}$, $Val[v_5] = r_{v_5} - r_{v_5}$ and $Val[v_6] = r_{v_5} - r_{R_3}$.

Second round:

1. The node v_4 sends $Val[v_4] = r_{v_4} - r_{R_2}$ to the node v_3 and v_3 receives $Val[v_4]$.
2. The node v_6 sends $Val[v_6] = r_{v_5} - r_{R_3}$ to the node v_3 and v_3 receives $Val[v_6]$.
3. The sender S sends $Val[S] = r_{S_1} - r_{v_1} + r_{S_3} - r_{v_5}$ to the node v_3 and v_3 receives $Val[S]$.
4. The node v_3 calculates: $Sum(v_3) = Val[v_3] + Val[v_4] + Val[v_6] + Val[S] = (r_{S_2} - r_{v_4}) + (r_{v_4} - r_{R_2}) + (r_{v_5} - r_{R_3}) + (r_{S_1} - r_{v_1} + r_{S_3} - r_{v_5}) = r_{S_2} - r_{R_2} - r_{R_3} + r_{S_1} - r_{v_1} + r_{S_3}$.

Third round:

1. The node v_3 sends $Sum(v_3) = r_{S_1} + r_{S_2} + r_{S_3} - r_{v_1} - r_{R_2} - r_{R_3}$ to the node v_2 and v_2 receives $Sum(v_3)$.

2. The node v_2 calculates $Sum(v_2) = Val[v_2] + Sum(v_3) = (r_{v_1} - r_{v_2}) + (r_{S_1} + r_{S_2} + r_{S_3} - r_{v_1} - r_{R_2} - r_{R_3})$.

Fourth round:

1. The node v_2 sends $Sum(v_2) = r_{S_1} + r_{S_2} + r_{S_3} - r_{v_2} - r_{R_2} - r_{R_3}$ to the receiver R and R receives $Sum(v_2)$.

2. The receiver R calculates the message $m = (r_{v_2} + r_{R_2} + r_{R_3}) + Sum(v_2) = (r_{v_2} + r_{R_2} + r_{R_3}) + (r_{S_1} + r_{S_2} + r_{S_3} - r_{v_2} - r_{R_2} - r_{R_3}) = r_{S_1} + r_{S_2} + r_{S_3}$.

7 PSMT tolerating mobile adversary

We have been considering the static adversary so far. In static adversary, a node once corrupted remains corrupted subsequently. Thus, the static adversary can corrupt only one fixed set of t-nodes throughout the protocol execution. Here, we relax this requirement by allowing the adversary to corrupt a different set of t nodes of its choice in different rounds. The adversary of this kind is called the mobile adversary. Intuitively we know that it is difficult to tolerate the mobile adversary due to its dynamic nature. And, the design of appropriate PSMT protocols for tolerating the mobile adversary offers better security guarantees compared to the static adversary counterpart.

We notice that, if nodes cannot *wipe/delete* the data from their memory, then given sufficient time (rounds) the adversary can eavesdrop the required number of nodes to get the secret (in the worst case the adversary can eavesdrop each node of the network after certain number of rounds though the protocol might have terminated). Therefore, it is necessary for the nodes to have the *data deletion* capability to tolerate the mobile adversary. Also, we assume that once data is deleted it cannot be recovered by any means. We use the notation DEL[U] to denote the *deletion* of every element from the subset U of the field $\mathbb{F}$. We now present the intuition behind the protocol $\mathbf{\Pi}^{\mathbf{Mobile}}_{\mathbf{Rnd\_Opt\_Lin}}$, which tolerates the mobile t-adversary if PSMT from S to R is possible tolerating the static t-adversary.

Assume that there exist $t+1$ vertex-disjoint weak paths from S to R in G then there exists at least one weak path (say p) such that no node of it is corrupted in the first round. Then, before the adversary corrupts any node from the weak path p in subsequent rounds, each pair of adjacent nodes of p exchange information, which eventually guarantees PSMT from S to R tolerating mobile adversary. Once adjacent nodes are done with exchanging information, each node u: (1) locally computes a function f on its local information (2) stores the output of f and (3) completely *deletes* the local information. The function f has the property that looking at the output of the function adversary learns nothing about the corresponding inputs. More precisely, let u be a node from the uncorrupted weak path p. Consider the following two cases.

Case 1: Suppose node u is an in-neighbour of R (i.e., $(u, R) \in E$) and wants to send the message m to R. Then, u simply forwards the message m to R and *deletes* m from its memory. The receiver R gets the message by the end of the round but the adversary learns nothing (additional) about the message even if it eavesdrops the node u in subsequent rounds as the message is already deleted by node u.

Case 2: Suppose u is not directly connected to R but u is an out-neighbour of R (i.e., $(R, u) \in E$) and the node u wishes to send a message m to R. The protocol works as follows. In first round, R sends a random key K_R to u and u receives K_R. Then, the node u calculates $m - K_R$ and *deletes*

both m and K_R from its memory. Observe that by corrupting node u in any subsequent rounds, the adversary gets $m - K_R$ which reveals nothing about either m or K_R. Therefore, in subsequent rounds node u can send $m - K_R$ to R along any path (from u to R), which may be eavesdropped by the adversary. Once R receives $m - K_R$, R adds K_R to $m - K_R$ and gets the message m.

We use this simple idea to design the protocol $\mathbf{\Pi}^{\mathbf{Mobile}}_{\mathbf{Rnd\_Opt\_Lin}}$ to tolerate the mobile t-adversary. This protocol is the same as the protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Opt\_Lin}}$ except that at the end of the first round each node u deletes its left-value $Left[u]$ and right-value $Right[u]$ after calculating its value $Val[u]$.

Consider a communication graph $\mathcal{G}^{(t+1)}$ of the digraph $G^{(r)}(V, E)$, where r be the optimal number of rounds required for PSMT possibility in G tolerating t-threshold static adversary. Let $(t+1)$ vertex disjoint weak paths of $\mathcal{G}^{(t+1)}$ are, namely $p_i : \langle u_{i0}(= S), u_{i1}, \ldots, u_{ik_i}, u_{i(k_i+1)}(= R)\rangle$, for each $i \in [1, t+1]$. And, the height of a Reverse Directed Rooted Tree $T_{G^{(r)}}$ of $G^{(r)}$ be h with root R is at the 0^{th} level. Then, the protocol code is as follows.

7.1 The Protocol $\mathbf{\Pi}^{\mathbf{Mobile}}_{\mathbf{Rnd\_Opt\_Lin}}$

The Protocol $\mathbf{\Pi}^{\mathbf{Static}}_{\mathbf{Rnd\_Opt\_Lin}}$

First round:

1. For each $i \in [1, t+1]$ and $j \in [0, k_i + 1]$: node u_{ij}, except $u_{(t+1)0}$, picks a random number $r_{ij} \in \mathbb{F}$.
2. The sender $S(= u_{(t+1)0})$ computes $r_{(t+1)0} = m - \sum_{i=1}^{t} r_{i0}$.
3. For each $i \in [1, t+1]$: $S(= u_{i0})$ initializes $Left[u_{i0}] = r_{i0}$.
4. For each $i \in [1, t+1]$ and $j \in [0, k_i]$:
   (a) if $(u_{ij}, u_{i(j+1)}) \in E^{(r)}$, then:
      i. u_{ij} sends r_{ij} to $u_{i(j+1)}$ and initializes $Right[u_{ij}] = r_{ij}$.
      ii. $u_{i(j+1)}$ receives r_{ij} from u_{ij} sent earlier in this round and initializes $Left[u_{i(j+1)}] = r_{ij}$.
   (b) if $(u_{ij}, u_{i(j+1)}) \notin E^{(r)}$, then:
      i. $u_{i(j+1)}$ sends $r_{i(j+1)}$ to u_{ij} and initializes $Left[u_{i(j+1)}] = r_{i(j+1)}$.
      ii. u_{ij} receives $r_{i(j+1)}$ from $u_{i(j+1)}$ sent earlier in this round and initializes $Right[u_{ij}] = r_{i(j+1)}$.
5. For each $i \in [1, t+1]$ and $j \in [0, k_i]$, node u_{ij} :
   (a) calculates its *value*, $Val[u_{ij}] = Left[u_{ij}] - Right[u_{ij}]$.
   (b) performs the deletion operation, DEL$[\{Left[u_{ij}], Right[u_{ij}]\}]$.
6. The sender S computes its *grand value*, $Val[S] = \sum_{i=1}^{t+1} Val[u_{i0}]$.

Second round onwards:

1. `If` S is a leaf node (at level h) in $T_{G^{(r)}}$ and its *grand value*, $Val[S]$, is non zero then S sends $Val[S]$ to its parent which is at $(h-1)^{th}$ level. And, the parent of S receives $Val[S]$.

2. `Else If` S is at k^{th} level ($k \in [1, h-1]$) then S receives *values*, which are non-zero, from its children (sent earlier in that round) which are at $(k+1)^{th}$ level. Subsequently, S *adds* all the received *values* to its *grand value* $Val[S]$ and sends to its parent which is at $(k-1)^{th}$ level. And, the parent of S receives the corresponding sum.

3. For each $i \in [1, t+1]$ and $j \in [1, k_i]$:

   (a) `If` u_{ij} is a leaf node (at level h) in $T_{G^{(r)}}$ and its *value*, $Val[u_{ij}]$, is non zero then u_{ij} sends $Val[u_{ij}]$ to its parent which is at $(h-1)^{th}$ level. And, the parent of u_{ij} receives $Val[u_{ij}]$.

   (b) `If` u_{ij} is at k^{th} level for some $k \in [1, h-1]$ (not a leaf node) then u_{ij} receives *values*, which are non-zero, from its children (sent earlier in that round) which are at $(k+1)^{th}$ level. Subsequently, u_{ij} *adds* all the received values to its *value* $Val[u_{ij}]$ and sends to its parent which is at $(k-1)^{th}$ level. And, the parent of u_{ij} receives the corresponding sum.

4. In the last round, the receiver R adds the sum of all the values it received from its children with the sum of all its *Left Values* (i.e. $\sum_{i=1}^{t+1} Left[u_{i(k_i+1)}]$) to get the message m.

Theorem 11. *The protocol* $\mathbf{\Pi^{Mobile}_{Rnd\_Opt\_Lin}}$ *is perfectly reliable and perfectly secure.*

Proof. **Perfect Reliability**: The protocol $\mathbf{\Pi^{Mobile}_{Rnd\_Opt\_Lin}}$ is same as the protocol $\mathbf{\Pi^{Static}_{Rnd\_Opt\_Lin}}$, except that, each node u_{ij}, after computing $Val[u_{ij}]$ performs an extra *delete* operation, DEL[$\{Left[u_{ij}]$, $Right[u_{ij}]\}$]. However, in the protocol $\mathbf{\Pi^{Static}_{Rnd\_Opt\_Lin}}$, for each node u_{ij}, $Val[u_{ij}]$ is enough to reconstruct the message m, whereas individual values $Left[u_{ij}]$ and $Right[u_{ij}]$ are not necessary. Therefore, perfect reliability is guaranteed.
Perfect Security: We know that there exists at least one weak path p_i (for some $i \in [1, t+1)$ such that no node of it is corrupted in the first round. Then, notice that, by the end of the first round, each node u_{ij} of p_i calculates its *value*, $Val[u_{ij}] = Left[u_{ij}] - Right[u_{ij}]$ and performs the delete operation, DEL[$\{Left[u_{ij}], Right[u_{ij}]\}$]. Therefore by corrupting p_i in subsequent rounds, the adversary gets $Val[u_{ij}] = Left[u_{ij}] - Right[u_{ij}]$ but nothing about either $Left[u_{ij}]$ or $Right[u_{ij}]$. In *Theorem* 10, we already showed that even if the adversary gets $Val[u_{ij}]$ for each u_{ij}, it learns nothing about r_{i0} and thus nothing (additional) about the message m. □

8 Multicast

Although point to point transmission is common, there are numerous applications in which the same message needs to be delivered to many receivers. To encompass this natural generalization we define Secret Multicast, from the sender S to the set of receivers $\hat{\mathbf{R}} = \{R_1, R_2, \ldots, R_k\}$, $k \geq 1$, as follows: The sender S wishes to secretly communicate a message m to each receiver

$R_i \in \hat{\mathbf{R}}$ such that the adversary, who can passively corrupt up to t nodes other than sender S and any receiver in $\hat{\mathbf{R}}$, learns nothing (additional) about the message m.

The simple idea of achieving SECRET MULTICAST by doing separate PSMT from the sender to each of the receivers does not work. In fact, there are graphs in which PSMT from the sender S to a single receiver R_1 is not possible, however, after making node R_2 (of the same graph) the second receiver, SECRET MULTICAST from S to $\{R_1, R_2\}$ becomes possible. For example, we can consider a graph in which, individually, PSMT from S to R_2 is possible and PSMT from R_2 to R_1 is possible, whereas PSMT from S to R_1 is not possible. The main idea in characterizing multicast is realizing the fact that the receivers can never be corrupted by the adversary and hence can be used as intermediate (uncorrupted) senders. Now, we define the following notion to help us model this transitive communication.

Definition 18. *Let $V_1, V_2, \ldots, V_k$ and W be any subsets of V. We say that $V_1, V_2, \ldots, V_k$ are pair-wise disjoint* modulo *W if $V_i \cap V_j \subseteq W$ for every $i, j(\neq i) \in [1, k]$. By extending this definition to weak paths in G, we say that any k weak paths $p_1, p_2, \ldots, p_k$ are pair-wise vertex disjoint* modulo *a set W, if $V(p_1), V(p_2), \ldots, V(p_k)$ are pair-wise disjoint* modulo *W, where $V(p_i)$ is the set of all vertices of the weak path p_i. In other words, no two distinct weak paths can share a common node except the nodes from W.*

Now we present the theorem which characterizes the SECRET MULTICAST from S to $\hat{\mathbf{R}} = \{R_1, R_2, \ldots, R_k\}$, $k \geq 1$.

Theorem 12. *Let $G(V, E)$ be a directed graph in which $S, R_1, R_2, \ldots, R_k$ are $k+1$ special nodes. Then,* SECRET MULTICAST *from S to $\hat{\mathbf{R}} = \{R_1, R_2, \ldots, R_k\}$ is possible in G tolerating up to t passive faults if and only if at least one of the following two conditions hold for each $R_i \in \hat{\mathbf{R}}$:*

1. *There exists a path from S to R_i containing nodes only from $\hat{\mathbf{R}} \cup \{S\}$.*
2. *There exist at least $t+1$ vertex disjoint weak paths modulo $\hat{\mathbf{R}} \cup \{S\}$ from S to R_i such that each node on these weak paths must have a path to R_i in G.*

Proof. **Necessity**: Consider a weak path p from S to R_i such that some node u of p has no path to R_i in G. Then clearly, the sender S can never convey any information to R_i along p. At best, node u may receive message from the sender S but would not be able to forward it to the receiver R_i, making the weak path p useless for S to R_i communication. Hence, we consider only the weak paths in which every node has a path to R_i. Let us assume on contrary that, for some $R_i \in \hat{\mathbf{R}}$, there exist only t vertex disjoint weak paths modulo $\hat{\mathbf{R}} \cup \{S\}$ from S to R_i such that each node on these t weak paths has a path to R_i and there is no path containing only nodes from $\hat{\mathbf{R}} \cup \{S\}$. Then, there exist a vertex cut of size t between S and R_i. Thus, by corrupting each node from vertex cut, the adversary learns each piece of information exchanged between S and R_i. Therefore, the view of the adversary is the same as the view of the receiver.

Sufficiency: We give a *modified* PSMT protocol for secretly transmitting the message m to each R_i.

Suppose, there exists a path p from S to R_i containing nodes only from $\hat{\mathbf{R}} \cup \{S\}$. Then, S simply forwards the message to R_i along the nodes on the path p. As no node of p is corrupted, perfect reliability and security are guaranteed.

Otherwise, assume that there exist at least $t+1$ vertex disjoint weak paths modulo $\hat{\mathbf{R}} \cup \{S\}$ from S to R_i, namely p_i for each $i \in [1, t+1]$ such that each node on these $t+1$ weak paths has

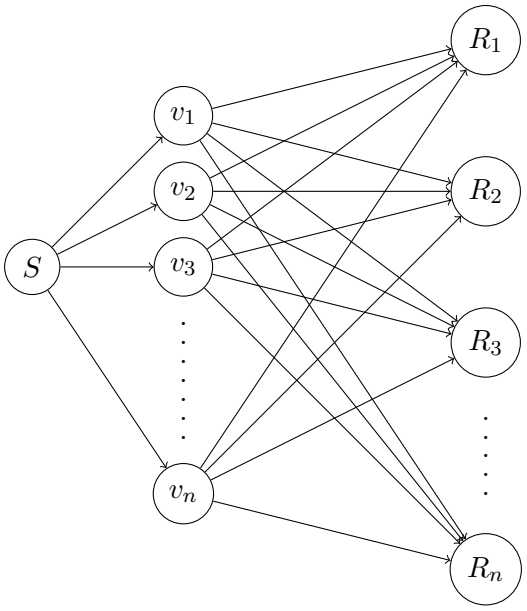

Figure 10: An example graph in which all the edges are critical for SECRET MULTICAST

a path to R_i. The sender S simply runs the linear communication protocol $\mathbf{\Pi}^{\mathbf{Mobile}}_{\mathbf{Rnd\_Opt\_Lin}}$ given in section 7.1.

Notice that these $t+1$ weak paths are pairwise vertex disjoint modulo $\hat{R} \cup \{S\}$. Therefore each node which is common to any of these weak paths must be from $\hat{R} \cup \{S\}$. Moreover, the adversary cannot corrupt the nodes from $\hat{R} \cup \{S\}$, this implies that, the adversary must corrupt at least two nodes to corrupt any two (vertex) disjoint weak paths, one from each of the two disjoint paths. Therefore by corrupting t nodes, the adversary can corrupt maximum of t weak paths. Also, the protocol $\mathbf{\Pi}^{\mathbf{Mobile}}_{\mathbf{Rnd\_Opt\_Lin}}$ reveals nothing about the message m to the adversary which can corrupt up to t different weak paths in each round. Therefore, this *modified* PSMT protocol is perfectly reliable and perfectly secure, completing the proof. □

8.1 Communication Complexity

Multicast can be achieved by executing *modified* PSMT protocol from S to each of the receivers. For this, we use the protocol $\mathbf{\Pi}^{\mathbf{Mobile}}_{\mathbf{Rnd\_Opt\_Lin}}$ from Section 7.1, which has $\mathcal{O}(|V|)$ communication complexity. As the number of receivers can be $\mathcal{O}(|V|)$, the communication complexity of our protocol becomes $\mathcal{O}(|V|^2)$. To show that this is asymptotically the best we can achieve, we present a graph which has $\Omega(|V|^2)$ critical edges. Consider the digraph $G(V, E)$ represented in Fig.10, which has vertex set $V = \{S, v_1, v_2, \ldots, v_n, R_1, R_2, \ldots, R_n\}$ and edge set $E = \Big\{\{S\} \times \{v_1, \ldots, v_n\}\Big\} \cup \Big\{\{v_1, \ldots, v_n\} \times \{R_1, \ldots, R_n\}\Big\}$. Here the sender S wishes to SECRET MULTICAST to $\hat{\mathbf{R}} = \{R_1, R_2, \ldots, R_n\}$ tolerating up to $n-1$ passive faults. As this graph satisfies the necessary conditions for *Theorem* 12, S to $\hat{\mathbf{R}}$ SECRET MULTICAST is possible, however, removing even a single edge makes it impossible.

8.2 Round Complexity

As the *modified* PSMT protocol for each of the receivers can be executed concurrently, the optimal protocol for multicast can be as fast as the slowest among these protocols. More formally, if r_i is the optimal number of rounds required for PSMT from S to R_i, then the optimal number of rounds for Secret Multicast from S to $\hat{\mathbf{R}}$ is $r = Max\{r_1, r_2, \ldots, r_{|\hat{R}|}\}$.

9 Concluding Remarks

We completely characterized the feasibility and optimality of protocols for perfectly secret message transmission in arbitrary networks under the influence of passive static adversary. We subsequently extended the same to tolerate mobile faults. We proved that empowering the adversary to move around and corrupt different set of t nodes (of its choice) in each round, alters neither the connectivity requirements nor the efficiency parameters. We also extended our ideas to incorporate multiple receivers (multicast), and arrive at optimal protocols in the more general setting. Notwithstanding, the problem of characterizing digraphs over which PSMT tolerating (static) t-Byzantine faults is possible, remains a hard open problem.

Acknowledgements

We wish to thank Prof. Alfred Menezes for his useful comments, specifically for correcting some inaccuracy in the proof of Theorem 2. We also wish to thank other anonymous reviewers for their suggestions that have substantially improved the presentation of this article.

References

[1] Shashank Agrawal, Abhinav Mehta, and Kannan Srinathan. Secure message transmission in asynchronous directed graphs. In *Progress in Cryptology - INDOCRYPT 2011 - 12th International Conference on Cryptology in India, Chennai, India, December 11-14, 2011, Proceedings*, pages 359–378.

[2] Ashwinkumar Badanidiyuru, Arpita Patra, Ashish Choudhury, Kannan Srinathan, and C. Pandu Rangan. On the trade-off between network connectivity, round complexity, and communication complexity of reliable message transmission. *J. ACM*, 59(5):22, 2012.

[3] Ashish Choudhary, Arpita Patra, B. V. Ashwinkumar, K. Srinathan, and C. Pandu Rangan. Perfectly reliable and secure communication tolerating static and mobile mixed adversary. In *Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10-13, 2008, Proceedings*, pages 137–155.

[4] Ashish Choudhary, Arpita Patra, B. V. Ashwinkumar, Kannan Srinathan, and C. Pandu Rangan. On minimal connectivity requirement for secure message transmission in asynchronous networks. In *Distributed Computing and Networking, 10th International Conference, ICDCN 2009, Hyderabad, India, January 3-6, 2009, Proceedings*, pages 148–162.

[5] Ashish Choudhury, Arpita Patra, B. V. Ashwinkumar, Kannan Srinathan, and C. Pandu Rangan. Secure message transmission in asynchronous networks. *J. Parallel Distrib. Comput.*, 71(8):1067–1074, 2011.

[6] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[7] Danny Dolev, Cynthia Dwork, Orli Waarts, and Moti Yung. Perfectly secure message transmission. *J. ACM*, 40(1):17–47, 1993.

[8] Matthias Fitzi, Matthew K. Franklin, Juan A. Garay, and Harsha Vardhan Simhadri. Towards optimal and efficient perfectly secure message transmission. In *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, pages 311–322.

[9] Matthias Fitzi, Martin Hirt, and Ueli M. Maurer. Trading correctness for privacy in unconditional multi-party computation (extended abstract). In *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, pages 121–136.

[10] Matthew K. Franklin and Rebecca N. Wright. Secure communication in minimal connectivity models. *J. Cryptology*, 13(1):9–30, 2000.

[11] Matthew K. Franklin and Moti Yung. Secure hypergraphs: privacy from partial broadcast (extended abstract). In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*, pages 36–44.

[12] Andrew V. Goldberg and Robert Endre Tarjan. Efficient maximum flow algorithms. *Commun. ACM*, 57(8):82–89, 2014.

[13] Martin Hirt and Ueli M. Maurer. Complete characterization of adversaries tolerable in secure multi-party computation (extended abstract). In *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing, Santa Barbara, California, USA, August 21-24, 1997*, pages 25–34.

[14] Ravi Kishore, Ashutosh Kumar, Chiranjeevi Vanarasa, and Kannan Srinathan. Round-optimal perfectly secret message transmission with linear communication complexity. In *Information Theoretic Security - 8th International Conference, ICITS Lugano, Switzerland, May 2-5, 2015, Proceedings*, pages 33–50.

[15] Ravi Kishore, Ashutosh Kumar, Chiranjeevi Vanarasa, and Kannan Srinathan. On the price of proactivizing round-optimal perfectly secret message transmission. *IEEE Transactions on Information Theory*, 64(2):1404–1422, 2018.

[16] M. V. N. Ashwin Kumar, Pranava R. Goundan, K. Srinathan, and C. Pandu Rangan. On perfectly secure cmmunication over arbitrary networks. In *Proceedings of the Twenty-First Annual ACM Symposium on Principles of Distributed Computing, PODC 2002, Monterey, California, USA, July 21-24, 2002*, pages 193–202.

[17] M. V. N. Ashwin Kumar, K. Srinathan, and C. Pandu Rangan. Asynchronous perfectly secure computation tolerating generalized adversaries. In *Information Security and Privacy, 7th Australian Conference, ACISP 2002, Melbourne, Australia, July 3-5, 2002, Proceedings*, pages 497–512.

[18] Kaoru Kurosawa and Kazuhiro Suzuki. Truly efficient 2-round perfectly secure message transmission scheme. *IEEE Transactions on Information Theory*, 55(11):5223–5232, 2009.

[19] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[20] Abhinav Mehta, Shashank Agrawal, and Kannan Srinathan. Brief announcement: Synchronous las vegas URMT iff asynchronous monte carlo URMT. In *Distributed Computing, 24th International Symposium, DISC 2010, Cambridge, MA, USA, September 13-15, 2010, Proceedings*, pages 201–203.

[21] Abhinav Mehta, Shashank Agrawal, and Kannan Srinathan. Interplay between (im)perfectness, synchrony and connectivity: The case of reliable message transmission. *Theor. Comput. Sci.*, 496:2–16, 2013.

[22] K. Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10:96–115, 1927.

[23] Manan Nayak, Shashank Agrawal, and Kannan Srinathan. Minimal connectivity for unconditionally secure message transmission in synchronous directed networks. In *Information Theoretic Security - 5th International Conference, ICITS 2011, Amsterdam, The Netherlands, May 21-24, 2011, Proceedings*, pages 32–51.

[24] Rafail Ostrovsky and Moti Yung. How to withstand mobile virus attacks (extended abstract). In *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing, Montreal, Quebec, Canada, August 19-21, 1991*, pages 51–59.

[25] Arpita Patra, Ashish Choudhary, Madhu Vaidyanathan, and C. Pandu Rangan. Efficient perfectly reliable and secure message transmission tolerating mobile adversary. In *Information Security and Privacy, 13th Australasian Conference, ACISP 2008, Wollongong, Australia, July 7-9, 2008, Proceedings*, pages 170–186.

[26] Arpita Patra, Bhavani Shankar, Ashish Choudhary, K. Srinathan, and C. Pandu Rangan. Perfectly secure message transmission in directed networks tolerating threshold and non threshold adversary. In *Cryptology and Network Security, 6th International Conference, CANS 2007, Singapore, December 8-10, 2007, Proceedings*, pages 80–101.

[27] Jérôme Renault, Ludovic Renou, and Tristan Tomala. Secure message transmission on directed networks. *Games and Economic Behavior*, 85:1–18, 2014.

[28] Hasan Md. Sayeed and Hosame Abu-Amara. Perfectly secure message transmission in asynchronous networks. In *Proceedings of the Seventh IEEE Symposium on Parallel and Distributed Processing, SPDP 1995, San Antonio, Texas , USA, October 25-28, 1995*, pages 100–105.

[29] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[30] Hongsong Shi, Shaoquan Jiang, Reihaneh Safavi-Naini, and Mohammed Ashraful Tuhin. On optimal secure message transmission by public discussion. *IEEE Transactions on Information Theory*, 57(1):572–585, 2011.

[31] K. Srinathan, M. V. N. Ashwin Kumar, and C. Pandu Rangan. Asynchronous secure communication tolerating mixed adversaries. In *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, pages 224–242.

[32] Kannan Srinathan, Arpita Patra, Ashish Choudhary, and C. Pandu Rangan. Unconditionally reliable message transmission in directed hypergraphs. In *Cryptology and Network Security, 7th International Conference, CANS 2008, Hong-Kong, China, December 2-4, 2008, Proceedings*, pages 285–303.

[33] Kannan Srinathan, Arpita Patra, Ashish Choudhary, and C. Pandu Rangan. Unconditionally secure message transmission in arbitrary directed synchronous networks tolerating generalized mixed adversary. In *Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2009, Sydney, Australia, March 10-12, 2009*, pages 171–182.

[34] Kannan Srinathan, Prasad Raghavendra, and C. Pandu Rangan. On proactive perfectly secure message transmission. In *Information Security and Privacy, 12th Australasian Conference, ACISP 2007, Townsville, Australia, July 2-4, 2007, Proceedings*, pages 461–473.

[35] Kannan Srinathan and C. Pandu Rangan. Possibility and complexity of probabilistic reliable communication in directed networks. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, Denver, CO, USA, July 23-26, 2006*, pages 265–274.

[36] Yongge Wang and Yvo Desmedt. Perfectly secure message transmission revisited. *IEEE Transactions on Information Theory*, 54(6):2582–2595, 2008.