Round-Optimal Perfectly Secret Message Transmission with Linear Communication Complexity

Ravi Kishore*, Ashutosh Kumar, Chiranjeevi Vanarasa,
and Srinathan Kannan

International Institute of Information Technology, Hyderabad, India - 500032
{ravikishore.vasala,chiranjeevi.v}@research.iiit.ac.in,
ashutosh.kumar@students.iiit.ac.in, srinathan@iiit.ac.in

Abstract. Consider an arbitrary network of n nodes, up to any t of which are eavesdropped on by an adversary. A sender S wishes to send a message m to a receiver R such that the adversary learns nothing about m (unless it eavesdrops on one among $\{S, R\}$). We prove a necessary and sufficient condition on the (synchronous) network for the existence of r-round protocols for perfect communication, for any given $r > 0$. Our results/protocols are easily adapted to asynchronous networks too and are shown to be optimal in asynchronous "rounds". Further, we show that round-optimality is achieved without trading-off the communication complexity; specifically, our protocols have an overall message complexity of $O(n)$ elements of a finite field to perfectly transmit one field element. Interestingly, optimality (of protocols) also implies: (a) when the shortest path between S and R has $\Omega(n)$ nodes, *perfect secrecy is achieved for "free"*, because any (insecure routing) protocol would also take $O(n)$ rounds and send $O(n)$ messages (one message along each edge in the shortest path) for transmission and (b) it is well-known that $(t+1)$ vertex disjoint paths from S to R are necessary for a protocol to exist; a consequent folklore is that the length of the $(t+1)^{th}$ ranked (disjoint shortest) path would dictate the round complexity of protocols; we show that the folklore is false; round-optimal protocols can be substantially faster than the aforementioned length.

1 Introduction

We address the problem of Perfectly Secret Message Transmission(PSMT),[1] defined as follows: A sender S wishes to send a message m to a receiver R such that an adversary, that eavesdrops on no more than t out of the n nodes, learns nothing about m. Our inquiry includes (a) *characterization*: under what conditions is

* Financial support from TCS is acknowledged.

[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.

A. Lehmann and S. Wolf (Eds.): ICITS 2015, LNCS 9063, pp. 33–50, 2015.
DOI: 10.1007/978-3-319-17470-9_3

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, as yet, known.

Although literature on information theoretically secure message transmission is rich, there are settings where answers to none of the aforementioned four questions are, yet, known. 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; not to mention, design of optimal protocols for the same are still far-fetched. Researchers have therefore attacked the problem in scenarios that are not as general as mentioned above – harder the inquiry, 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 hyper-graphs [1], non-threshold adversaries [2], mobile faults [3,4], mixed/hybrid faults [5,6], asynchronous networks [7], to name a few.

The PSMT problem was conceived and first solved by Dolev *et. al* [8]. They assume that the graph is *undirected*. It is proved that PSMT is possible `if and only if` there are at least $(2t + 1)$ vertex disjoint paths between S and R. Further, the protocols designed in [8] are efficient too. However, designing round optimal protocols for PSMT (even in undirected graphs) still 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 [9] (that is, perfectness is negligibly traded-off). However the design of communication optimal solutions are still open.

A particular setting where communication optimum protocol for PSMT are designed is the following: applying Menger's theorem [10], 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 [11]. While the notion of phase complexity has been studied [12,11,13], we stress that round complexity is markedly different from phase complexity, even in the case of undirected networks (as illustrated in Section 2.1).

Recently, restricting to passive adversaries, Renault *et. al* [14] characterize the digraphs that enable PSMT. In fact they use a more general non-threshold adversary model, characterized via an adversary structure, which is a collection of subsets of nodes in the graph, where in the adversary may choose to corrupt (passively in this case) the nodes in any one subset among the collection. The protocols of [14] are therefore not always efficient (that is, may be super-polynomial in n).

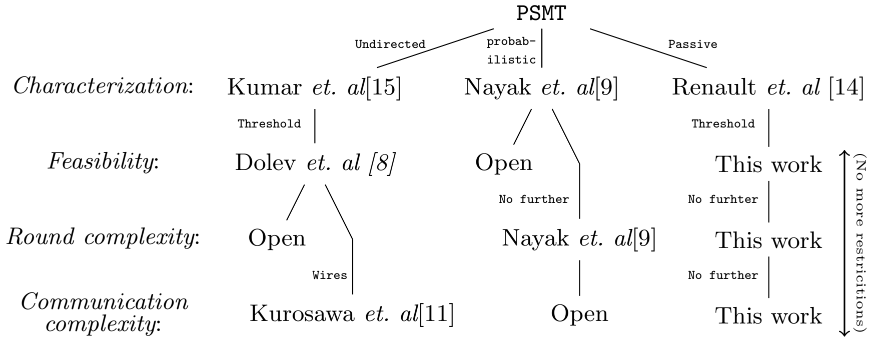


Fig. 1. Restriction based solutions

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 $\mathbb{A}$. However, (im)possibility results are known if one restricts the setting to either undirected graphs [15] or passive adversary or security with error [14,9]. 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* [8] give 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 [9] or computational security [16]. Bit-optimal protocols have been designed in the wires-based abstraction of the undirected graph [11]. While a similar wires-based approach has been used for digraphs too [17], it is known to be inadequate to capture all digraphs on which protocols exists [18].

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 bit-optimal protocols?* (or, are further restrictions like wires-based abstractions still required?)

Interestingly, we design efficient round/bit 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 bit-optimal.[2] Further, the *simplicity* of our protocol ensures the implementability of highly scalable perfectly secret message transmission.

In a nutshell, we address the PSMT problem in such a way that all the four questions, namely, characterization, feasibility, round and bit 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 Secure Communication is Possible

It is well-known that, for passive threshold adversaries, $(t+1)$-vertex disjoint paths are necessary and sufficient for PSMT from S to R in undirected graphs [8]. Consequently, as noted in [8] 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 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 $\langle S(=v_0), v_1, v_2, v_3, \dots v_{n-1}, R(=v_n)\rangle$ have length of 2 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. R gets the two points and hence the message after n rounds. Can 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 3 we know that 3 rounds are necessary and sufficient for PSMT in the graph 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 an interesting problem.

A Remark on Extending to Asynchronous "Rounds". Due to the absence of fail-stop and/or Byzantine corrupt nodes in our setting, it is fairly straight-forward to adapt all our protocols (and hence our characterizations) to the asynchronous setting too. Indeed, several of our protocols are directly designed assuming that the network is asynchronous; this is the reason that commands like `wait` appear in our algorithms (these can be safely ignored in case of full-fledged synchrony). On the other hand in asynchronous networks, there is no formal notion of global round, and therefore our claims of round-optimality have to be understood accordingly. Specifically, we define an asynchronous 'hop'

[2] Linear communication complexity is equivalent to bit-optimality only when we consider optimally fault-tolerant protocols, that is, using the maximum t-adversary that is tolerable. Otherwise, *sub-linear* communication complexity is achieved by trading-off fault-tolerance using multi-secret sharing (analogous to [19]).

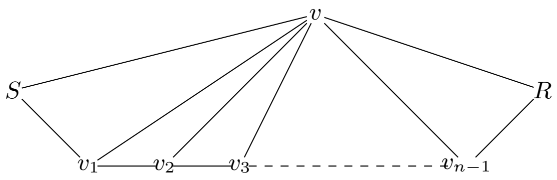


Fig. 2. An undirected graph tolerating one passive fault

as a round with an in-built `wait-for-the-message`. Though, these hops are not globally in lock-step, we may easily use it as a measure of asynchronous round-complexity of a protocol – the length of longest nested hop sequence. We see that our protocol are hop-optimal, and we can derive the same using the same algorithms used for deriving the round optimal protocols.

2.2 Linear Communication Complexity

Folklore suggests that optimizing the number of rounds for a distributed protocol, typically increases the communication complexity (total numbers of bits transferred across all edges in the network during the execution of protocol). In rare cases, round optimality can co-exist with bit-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 (Section 4.2). 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). At any rate, each of these edges is *critical*, in the sense that, if deleted, PSMT is rendered impossible – hence they need to be used at least once. Thus, we arrive at a surprising protocol for secure 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 $O(n)$ rounds and send $O(n)$ messages (one message along each edge in the shortest path) for transmission.

2.3 Efficient Discriminant Algorithms

Specifying the necessary and sufficient condition does not imply that there exists an efficient algorithm for checking the same. Indeed, the literature (on possibility of 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 [18,9]. However, no efficient algorithms to test these conditions are known; in fact they may be NP-hard too, 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 5.3 is a polynomial-time algorithm for testing the existence of an r-round secure communication protocol in a given network (and if yes, obtaining a round optimal one). All the reductions mentioned in the paper can be easily done in polynomial time, as all of them involve creation of a sub-graph of the given graph.

3 Network Model and Definitions

Definition 1. ***Passive Corruption:*** *Following [12], a node v is said to be passively corrupted if the adversary has full access to the information and internal state of v. But v will honestly follow the protocol execution.*

Definition 2. *Following [12], define the* `VIEW` *of a node $v \in V$ at any point of the execution of a protocol Π, to be the information the node can get from its local input (if any) to the protocol, all the messages that it had earlier sent or received, the protocol code executed by the node and its random coins.* `VIEW` *of a set of nodes $W(\subseteq V)$ is the information that the nodes in W can get together from their individual* `VIEWS` *and is denoted by $\mathtt{VIEW}^{\Pi}_{G}(W)$. The* `VIEW` *of an adversary $\mathbb{A}$ is the* `VIEW` *of the set of nodes controlled by adversary, denoted by $\mathtt{VIEW}^{\Pi}_{G}(\mathbb{A})$.*

Definition 3. ***Perfect Security:*** *Following [12], a message transmission protocol Π for sending message m from sender S to receiver R is said to be perfectly secure if it satisfies the following two conditions:*

1. *Perfect Reliability: At the end of the protocol Π receiver should learn the correct message m.*
2. *Perfect Secrecy: Adversary should not learn any information about the message m(i.e. adversary should not be able to distinguish whether S sent message m or m' for any two messages m and m').*

Definition 4. *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 5. *A sequence of nodes $p : \langle v_0(= u), v_1, v_2, \ldots, v_k, v_{k+1}(= v)\rangle$ is said to be a weak path from u to v in a directed graph $G(V, E)$, if $\forall j \in \{0, 1, \ldots, k\}$, either $(v_j, v_{j+1}) \in E$ or $(v_{j+1}, v_j) \in E$.*
We say that 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 $p : \langle v_0(= u), v_1, v_2, \ldots, v_k, v_{k+1}(= v)\rangle$.

Notations:

1. In a directed graph $G(V, E)$
   (a) The set of all corrupted nodes is denoted by $V_{\mathcal{C}} \subseteq V$ and $V \setminus V_{\mathcal{C}}$ denotes the set of honest nodes. We also have $|V_{\mathcal{C}}| \leq t$.
   (b) $G[V^{'}]$ denotes the induced sub graph of G induced by the vertex set $V^{'}$.
   (c) V_v denotes the set of vertices from which vertex v is reachable.
2. d_v denotes the length of a shortest path from $v(\in V_R)$ to receiver R.
3. $[l, u] = \{m \in \mathbb{Z} \mid l \leq m \leq u\}$.

We model our communication network as a directed graph $G(V, E)$, $|V| = n$, where each edge is a private, authentic and reliable channel. Our network is synchronous and every node knows the topology of the network. Communication happens in a sequence of rounds. In any round, a player can receive the messages sent to it by its in-neighbours in the previous round, perform some computation and finally send a message to its out-neighbours. The set of faults in the network is characterized by a central (fictitious) adversary who can eavesdrop or passively corrupt no more than t nodes in the network. Throughout this paper, by a "faulty node" we mean that the node is "passively corrupted by the adversary" and by "secure" we mean "perfectly secure". For brevity, by "`PSMT is possible`", we mean "`PSMT tolerating` t`-threshold passive adversary is possible`". By "a number r is chosen randomly" we mean "r is chosen uniformly at random from field $\mathbb{F}$". Our message space is a large enough field $\langle \mathbb{F}, +, \star \rangle$ and all the calculations are done in this field $\mathbb{F}$ only.

4 Communication Efficient PSMT Protocol in G

In any protocol Π, if there is no path from a node v to receiver R, then v can't convey any information to R. Therefore with out loss of generality we can assume that from every node to R there is at least one path. Once this assumption is made, in this section we present a communication efficient protocol Π_{Eff} in G with communication complexity of $O(n^2)$ whenever PSMT is possible in G_u. First we present a protocol Π_{Sim}, which simulates the corresponding path p' of a weak path p. Then we run protocol Π_{Sim} for simulating the corresponding path of each such weak path, to get protocol Π_{Eff}. We show that if every node in a weak path p is an honest node then protocol Π_{Sim}, securely transmits message m from S to R in G. Thus, executing Π_{Sim} for $t + 1$ (or more) times results in a PSMT protocol.

4.1 Protocol Π_{Sim}

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 S wants to send to R using the corresponding path p'.

1. `if` weak path p is a path in G then S simply sends message m using path p.
2. `otherwise` let $\{u_{i_1}, u_{i_2}, \ldots, u_{i_k}\}$ be the set of all nodes in the weak path p such that $(u_{i_j}, u_{i_j+1}) \notin E$ for $j \in [1, k]$ and without loss of generality assume that $i_m < i_n$ for $m < n$.
   (a) As $(u_{i_j}, u_{i_j+1}) \notin E$, we have (i) $(u_{i_j+1}, u_{i_j}) \in E$ and (ii) a subpath p_{i_j+1} from u_{i_j+1} to $u_{i_{j+1}}$ in G having only nodes of weak path p.
   (b) u_{i_j+1} chooses a random number r_{i_j+1} and sends to $u_{i_{j+1}}$ using path p_{i_j+1} and to u_{i_j} using edge (u_{i_j+1}, u_{i_j}).
   (c) $u_{i_j} (j \neq 1)$ calculates $r_{i_{j-1}+1} + r_{i_j+1}$ and sends to R using some path as there exists at least one path.
3. S will send m to u_{i_1} and u_{i_1} sends $m + r_{i_1+1}$ to R.

4. `for` $j = k-1, k-2, \ldots, 1$; R computes $r_{i_j+1} = (r_{i_j+1} + r_{i_{j+1}+1}) - r_{i_{j+1}+1}$.
5. Once R gets r_{i_1+1} for $j = 1$, it finally computes $m = (m + r_{i_1+1}) - r_{i_1+1}$.

Lemma 1. *In graph G, for three honest nodes u, v, w; if* PSMT *is possible from w to u and w to v then* PSMT *is possible from u to v if there is a path p from u to v(i.e. $u \in V_v$).*

Proof. Let m be the message that u wants to communicate to v secretly. First w chooses a random number r and sends the same to both u and v secretly. Now u masks the message m with r as $m + r$ and sends to v using path p. Finally v unmasks the message m by subtracting r from $m + r$. This protocol is perfectly secure even if path p contains malicious nodes, since in a field $\langle \mathbb{F}, +, * \rangle$; for given $x, z \in \mathbb{F}$, $\exists$ unique $y \in \mathbb{F}$ such that $x + y = z$.

Corollary 1. *Protocol Π_{Sim} for simulating the corresponding path p' of a weak path $p : \langle S(= u_0), u_1, \ldots, u_l, u_{l+1}(= R) \rangle$, securely transmits a message m from S to R if every node u_i in p is an honest node.*

Proof. As i_k is maximum, u_{i_k} is the last node in p such that $(u_{i_k}, u_{i_k+1}) \notin E$, we have (i) Secure edge $(u_{i_k+1}, u_{i_k}) \in E$ and (ii) secure path from u_{i_k+1} to R containing only nodes of weak path p, which implies PSMT from u_{i_k+1} to R is possible in G. Therefore, from Lemma 1, PSMT is possible from u_{i_k} to R.

1. `for` $j = k-1, k-2, \ldots, 1$ we have:
   (a) a secure path p_{i_j+1} from u_{i_j+1} to $u_{i_{j+1}}$ in G containing only nodes of weak path p.
   (b) PSMT is possible from $u_{i_{j+1}}$ to R.
   (c) Above two steps together gives, PSMT is possible from u_{i_j+1} to R.
   (d) Secure edge $(u_{i_j+1}, u_{i_j}) \in E$.
   (e) From Lemma 1, we get, PSMT is possible from u_{i_j} to R.
2. In particular when $j = 1$ we get, PSMT is possible from u_{i_1} to R in G.
3. We have a secure path from S to u_{i_1} containing only nodes of weak path p.
4. From the above two steps we get, PSMT is possible from S to R.

4.2 Efficient Protocol

We now present a PSMT protocol Π_{Eff} in G whenever PSMT is possible in G_u. Dolev *et. al* [8] show that PSMT is possible in G_u `if and only if` there exists $(t+1)$ vertex disjoint paths from S to R in G_u. Let p'_i be a vertex disjoint path in G_u corresponding to weak path p_i, for each $i \in [1, t+1]$.

Protocol Π_{Eff}:

1. S chooses a random t-degree polynomial $p(x)$ and replaces constant term $p(0)$ with the message m.
2. S sends $p(i)$ to R by simulating the corresponding path p'_i of a weak path p_i using protocol Π_{Sim}.
3. R reconstructs $p(x)$ once it receives all $(t+1)$ points to get message m.

Lemma 2. *The protocol Π_{Eff} is reliable and secure.*

Proof. Protocol Π_{Eff} is reliable since the protocol Π_{Sim} is reliable. Protocol Π_{Eff} is secure since there exists at least one secure weak path p_i for some $i \in [1, t+1]$, so $p(i)$ is secure by Corollary 1. On a t-degree polynomial, t or fewer points reveals nothing about constant term [20], which is the message m.

The communication complexity of the above protocol Π_{Eff} is $O(n^2)$ since these $(t+1)$ paths may contain all the n nodes and each node may need to send the masked value to the receiver R using some path which in turn can contain $O(n)$ nodes.

Now we will give one example for the simulation of a corresponding of a weak path using the protocol Π_{Sim}. Consider the graph G in the Fig. 3 which has three disjoint weak paths namely $p_1 : \langle S, v_1, v_2, R\rangle$, $p_2 : \langle S, v_3, v_4, R\rangle$ and $p_3 : \langle S, v_5, v_6, R\rangle$. Therefore it can tolerate up to two faulty nodes. Due to space constraints, we only give the simulation of the corresponding path of weak path p_3, which is as follows:

1. R chooses a random number r_8 and sends to v_6.
2. v_5 chooses a random number r_5 and sends to both S and v_6.
3. v_6 masks r_5 with r_8 as $r_5 + r_8$ and sends to R using the path $\langle v_6, v_4, v_1, v_2, R\rangle$.
4. S sends the masked value $p(3) + r_5$ to R using the path $\langle S, v_3, v_1, v_2, R\rangle$.
5. R first unmasks r_5 by just subtracting r_8 from $r_5 + r_8$ and gets r_5 and then R similarly unmasks $p(3)$.

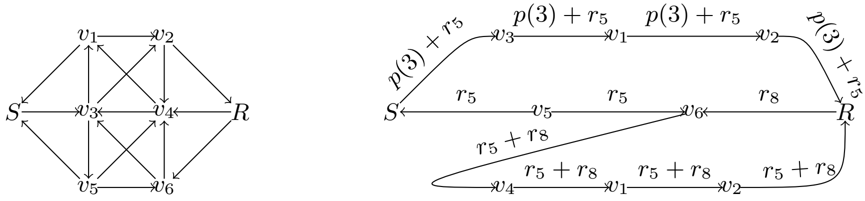


Fig. 3. An example graph G with simulation of the corresponding path of p_3

Theorem 1. PSMT *from S to R is possible in G if and only if in G_u.*

Proof. Run the protocol Π_{Eff} for PSMT in G if PSMT is possible in G_u.

4.3 Polynomial Time Algorithm for Verifying PSMT Possibility in G

1. Compute V_R using Breadth First Search from R, using opposite direction of edges.
2. `if` edge $(S, R) \in E$ or $(R, S) \in E$ then `return true`.

3. `else` create auxiliary graph $G^{aux}(V^{aux}, E^{aux})$ of G as follows :
    (a) split each vertex $v_i \in V$ except S and R, into two vertices v_{i1} and v_{i2} and add an edge from v_{i1} to v_{i2}. $V^{aux} = \{S, R\} \cup_{i=1}^{n} \{v_{i1}, v_{i2}\}$.
    (b) point all incoming edges of v_i to v_{i1} as incoming edges of v_{i1}.
    (c) point all out going edges of v_i as out going edges of v_{i2}.
    (d) for every edge add uniform edge capacity of 1.
4. Run `Max flow algorithm` to find `Max flow` in G_u^{aux}.
5. If `Max flow` $\geq t+1$ then return `true` else `false`.
6. Note that $(t+1)$ Vertex disjoint paths also can be found easily.

This is a polynomial time algorithm as breadth first search takes in worst case $O(n^2)$ time [21], construction of graph G^{aux} takes $O(n^2)$ time and max flow takes $O(n^3)$ time [22,23].

5 Round Optimality

In first subsection, we present a `generic round efficient PSMT protocol` in G. In later subsection we bring the notion of `round evolution graph`, a subgraph of G which evolves as number of rounds increases. We show that if `PSMT` is possible in `round evolution graph` then we can simulate the `generic round efficient PSMT protocol` in that `round evolution graph`. Finally we give a polynomial time algorithm for identifying the optimal rounds number. Combing all together, we obtain round optimal protocol as well as the optimal rounds number.

5.1 Round Efficient Protocol

In this section we present a `round efficient` protocol Π_{Rnd_Eff} whenever `PSMT` is possible in G_u. The main idea is every node v in G, will start its computation and/or communication from first round itself and if anything needs to be sent to R directly it will send using a shortest path, so that it conveys the required information to R possibly in least number of rounds. In the first round, for an edge $(u, v) \in E_u$, both nodes u and v chooses random numbers r_u, r_v respectively such that:

1. `if` $(u, v) \in E$ (forward edge with respect to u) then u sends r_u to v and initializes its `Right Value`, $R[u] = r_u$ and v initializes its `Left Value`, $L[v] = r_u$ once it receives r_u from u.
2. `else` if(i.e. there is no forward edge) v sends r_v to u and initializes its `Left Value`, $L[v] = r_v$, this is possible since $(u, v) \in E_u$ and $(u, v) \notin E$; edge (v, u) must be in E(backward edge with respect to u). Node u initializes its `Right Value`, $R[u] = r_v$ once it receives r_v from v.
3. It is clear that in both the cases $R[u] = L[v]$.

Protocol Π_{Rnd_Eff}:

1. Since PSMT is possible in G_u, there exists $(t{+}1)$ vertex disjoint paths from S to R in G_u namely $p_i : \langle u_{i0}(= S), u_{i1}, \ldots, u_{ik_i}, u_{i(k_i+1)}(= R)\rangle$, for $i \in [1, t+1]$. As $u_{ij} \in V_R$, there exists at least one path from u_{ij} to R, for $j \in [0, k_i]$ and let $p_{u_{ij}}$ be a shortest path from u_{ij} to R. Note that for $i \in [1, t+1]$, $u_{i0} = S$ and $u_{i(k_i+1)} = R$.
2. Let m be the message S wants to send to R securely.
3. S chooses a random t degree polynomial $p(x)$ and replaces constant term $p(0)$ with m.
4. For each path p_i, $i \in [1, t+1]$:
   (a) Every node $u_{ij}(\neq u_{i0})$, chooses a random number r_{ij}, for $j \in [1, k_i + 1]$.
   (b) $S(= u_{i0})$ initializes $r_{i0} = p(i)$ and also initializes $L[u_{i0}] = p(i)$.
   (c) For $j \in [1, k_i + 1]$, if $(u_{i(j-1)}, u_{ij}) \in E$ then $u_{i(j-1)}$ sends $r_{i(j-1)}$ to u_{ij} and initializes $R[u_{i(j-1)}] = r_{i(j-1)}$. u_{ij} waits to receive $r_{i(j-1)}$ and initializes $L[u_{ij}] = r_{i(j-1)}$ once it is received.
   (d) For $j \in [1, k_i + 1]$, if $(u_{i(j-1)}, u_{ij}) \notin E$ then u_{ij} sends r_{ij} to $u_{i(j-1)}$ and initializes $L[u_{ij}] = r_{ij}$. $u_{i(j-1)}$ waits to receive r_{ij} and initializes $R[u_{i(j-1)}] = r_{ij}$ once it is received.
   (e) Observe that in both cases $R[u_{i(j-1)}] = L[u_{ij}]$, for $j \in [1, k_i + 1]$.
   (f) For $j \in [0, k_i]$ every node u_{ij} calculates its Value, $Val[u_{ij}] = L[u_{ij}] - R[u_{ij}]$ and if it is non-zero(i.e. $L[u_{ij}] \neq R[u_{ij}]$) then it sends $Val[u_{ij}]$ to receiver R using shortest a path $p_{u_{ij}}$. R waits till it receives $Val[u_{ij}]$.
5. R computes $p(i) = \sum_{j=0}^{k_i} Val[u_{ij}] + L[u_{i(k_i+1)}]$. This is possible for R to compute since R knows $L[u_{i(k_i+1)}]$ as $u_{i(k_i+1)} = R$.

Protocol Π_{Rnd_Eff} *runs in maximum of* $|V|$ *rounds.* This is because, in first round nodes share their random numbers with neighbours as explained in protocol and then each node u_{ij} sends $Val[u_{ij}]$ to R using shortest path $p_{u_{ij}}$, if required. Sending $Val[u_{ij}]$ can take in worst case(when every node in V appears in path $p_{u_{ij}}$ for some values of i, j) $|V|$-1 rounds. Therefore protocol Π_{Rnd_Eff}, achieves PSMT in a total of $|V|$ or fewer rounds.

Lemma 3. *Protocol* Π_{Rnd_Eff} *for sending message* m *from* S *to* R *is reliable.*

Proof. For each u_{ij}, $(j \neq k_i + 1)$ in path p_i, we have $R[u_{ij}] = L[u_{i(j+1)}]$

Let $Sum = \sum_{j=0}^{k_i} Val[u_{ij}] + L[u_{i(k_i+1)}]$. Now we will show that $Sum = p(i)$.

$$
\begin{aligned}
Sum &= \sum_{j=0}^{k_i} (L[u_{ij}] - R[u_{ij}]) + L[u_{i(k_i+1)}] \\
&= \sum_{j=0}^{k_i} (L[u_{ij}] - L[u_{i(j+1)}]) + L[u_{i(k_i+1)}] \\
&= L[u_{i0}] - L[u_{i(k_i+1)}] + L[u_{i(k_i+1)}] = L[u_{i0}] = p(i)
\end{aligned}
$$

Lemma 4. *Protocol Π_{Rnd_Eff} for sending message m from S to R is secure.*

Proof. As the adversary can corrupt at most t nodes, there exists a path p_i from S to R in $G_u[V \setminus V_{\mathcal{C}}]$ for some $i \in [1, t+1]$(i.e. path p_i is not under the control of adversary). Each u_{ij} in path p_i except $u_{i(k_i+1)}$ sends $Val[u_{ij}]$ to R using path $p_{u_{ij}}$ if required, which may be under the control of adversary. For $j \in [0, k_i]$, even if adversary gets $Val[u_{ij}]$, in a field $\mathbb{F}$, $\exists$ unique x such that $\sum_{j=0}^{k_i} (\lambda_j * Val[u_{ij}]) + x = p(i)$, for any $\lambda_j \in \mathbb{F}$. Alternatively we can think as adversary gets $k_i + 1$ system of linear independent equations in $k_i + 2$ variables namely for each $j \in [0, k_i]$, $Val[u_{ij}] = L[u_{ij}] - R[u_{ij}] = L[u_{ij}] - L[u_{i(j+1)}]$. Therefore the adversary learns nothing about $p(i)$ and so nothing about m as well [20].

5.2 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 6. *Given the round number r, and a network represented by a directed graph $G(V, E)$, with receiver R, the round evolution graph of order r, $G^{(r)}(V, E^{(r)})$ is defined as subgraph of G, where $E^{(r)} = E \setminus \{(u, v) \in E \mid d_v \geq r\}$. i.e. Remove edges from which R can't receive information in r rounds.*

Theorem 2. PSMT *is possible in $G^{(r)}$ $\iff$ r-round* PSMT *protocol exists in $G^{(r)}$.*

Proof. Sufficiency: It is clear that if r-round protocol exists then PSMT is trivially possible in $G^{(r)}$.
Necessity: Suppose PSMT is possible in $G^{(r)}$, then we show that in r-rounds we can simulate the protocol Π_{Rnd_Eff} given in Section 5.1. Observe that in protocol Π_{Rnd_Eff}, every node u_{ij} in path p_i, in first round sends the chosen random number r_{ij} to its neighbour(s) if required. We have three cases for each u_{ij}:

1. $(u_{i(j-1)}, u_{ij}) \in E^{(r)}$. By our construction of $G^{(r)}$, $d_{u_{ij}} \leq r - 1$, therefore even if in first round u_{ij} waits to receive random numbers from neighbours, it can send $Val[u_{ij}]$ in total of r-rounds.
2. $(u_{ij}, u_{i(j+1)}) \notin E^{(r)}$ which implies $(u_{i(j+1)}, u_{ij}) \in E^{(r)}$. By our construction of $G^{(r)}$, $d_{u_{ij}} \leq r - 1$. Rest follows as in case(1).
3. $(u_{i(j-1)}, u_{ij}) \notin E^{(r)}$ and $(u_{ij}, u_{i(j+1)}) \in E^{(r)}$, In this case u_{ij} is not required to send its `value` to receiver R, since $Val[u_{ij}] = L[u_{ij}] - R[u_{ij}] = r_{ij} - r_{ij} = 0$.

Theorem 3. *r-round* PSMT *protocol exists in G $\iff$* PSMT *is possible in $G^{(r)}$.*

Proof. Sufficiency: Suppose PSMT is possible in $G^{(r)}$, from Theorem 2 we know that r-round protocol exists in $G^{(r)}$ and so r-round protocol exists in G since $G^{(r)}$ is a sub graph of G.
Necessity: Suppose r-round protocols exists for G. Note that any node v with $d_v > r$, never conveys any information to R in an r-round protocol(Since it needs at least $r + 1$ send commands to send to R). Therefore in r-round protocol, an edge (u, v) in E but not in $E^{(r)}$, $\forall v \in V$ can be safely ignored when $d_v \geq r$. In other words, at the end of the r-round protocol Π, $\mathbf{VIEW}_G^{\Pi}(\{R\})$ does not change whether these edges are present or not.
Therefore round optimal protocol in G is a protocol in $G^{(r)}$, where r is the minimum number of rounds required for PSMT in G.

5.3 Polynomial Time Algorithm for Identifying the Optimal Number of Rounds

We will find minimum r for which PSMT is possible in $G^{(r)}$ by doing binary search on r for $r \in [1, |V|]$. This can be done in polynomial time since in each iteration:

1. We are constructing sub graph $G^{(r)}$ of G, this can be done in polynomial time.
2. We are checking whether PSMT is possible or not in $G^{(r)}$, this also can be done in polynomial time as explained in Section 4.3.

5.4 An Example of Round Optimal Protocol

In this section we give a round optimal protocol for the graph G given in Fig. 3. In Fig. 4 we can see a shortest path from each node to R and its distance. Fig. 5 represents the round evolution graphs $G^{(3)}$ and $G^{(4)}$ corresponding to the same graph in Fig. 3. Now in graph G we show that 4 is the optimal number of rounds by showing that PSMT is not possible in $G^{(3)}$ but possible in $G^{(4)}$.

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	$p_{v_5} : \langle v_5, v_4, v_3, v_2, R\rangle$	4
v_6	$p_{v_6} : \langle v_6, v_3, v_2, R\rangle$	3

Fig. 4. A shortest path from each node to receiver R in graph G

As shortest distance from S to R is 3, any protocol will take at least three rounds. Clearly PSMT is not possible in $G^{(3)}$ tolerating 2-adversary as there is

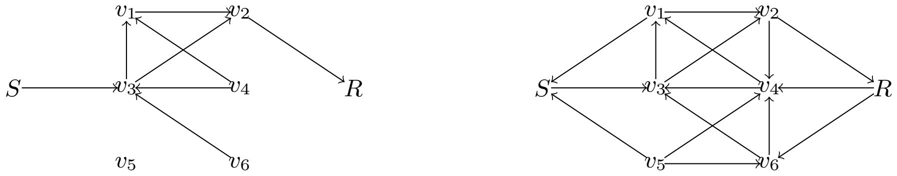


Fig. 5. An example of round evolution graphs $G^{(3)}$ and $G^{(4)}$

only one vertex disjoint path from S to R in $G_u^{(3)}$. In $G_u^{(4)}$ there are 3 vertex disjoint paths from S to R, also R is reachable from each of the nodes in these paths, so by Theorem 1, `PSMT` is possible in G. Now we present a 4 round protocol in the graph G as an example.

1. `Round 1`:
   (a) Node S chooses a random two degree polynomial $p(x)$ and replaces constant term $p(0)$ with m.
   (b) Every node v except S and R chooses a random number r_v. R chooses three random numbers $r_{R_1}, r_{R_2}, r_{R_3}$.
   (c) Node v_1 sends r_{v_1} to v_2 and $S(= S_1)$; node v_2 sends r_{v_2} to R.
   (d) Node $S(= S_2)$ sends $p(2)$ to v_3; node v_4 sends r_{v_4} to v_3; node R sends r_{R_2} to v_4.
   (e) Node v_5 sends r_{v_5} to v_6 and $S(= S_3)$; node R sends r_{R_3} to v_6.
2. `Round 2, Round 3 and Round 4:`
   (a) Every node v calculates its `value` $Val[v]$: $Val[S_1] = p(1) - r_{v_1}$, $Val[v_1] = r_{v_1} - r_{v_1}$, $Val[v_2] = r_{v_1} - r_{v_2}$, $Val[S_2] = p(2) - p(2)$, $Val[v_3] = p(2) - r_{v_4}$, $Val[v_4] = r_{v_4} - r_{R_2}$, $Val[S_3] = p(3) - r_{v_5}$, $Val[v_5] = r_{v_5} - r_{v_5}$, $Val[v_6] = r_{v_5} - r_{R_3}$
   (b) Every node $v \notin \{v_1, S_2, v_5, R\}$, sends its value $Val[v]$ to R using shortest path p_v from v to R, this is possible since the distance of shortest path p_v is less than or equal to 3.
3. R calculates $p(1) = Val[S_1] + Val[v_1] + Val[v_2] + r_{v_2} = p(1) - r_{v_1} + 0 + r_{v_1} - r_{v_2} + r_{v_2} = p(1)$, $p(2) = Val[S_2] + Val[v_3] + Val[v_4] + r_{R_2} = 0 + p(2) - r_{v_4} + r_{v_4} - r_{R_2} + r_{R_2} = p(2)$ and $p(3) = Val[S_3] + Val[v_5] + Val[v_6] + r_{R_3} = p(3) - r_{v_5} + 0 + r_{v_5} - r_{R_3} + r_{R_3} = p(3)$

All the nodes whose shortest distance to receiver R, is less than or equal to 3 can send their values to R in less than or equal to 4 rounds by simply forwarding via shortest path(they may wait in first round for receiving the random numbers from neighbour). Every node except v_5 is at a distance of less than or equal to 3 so they can send their `values` to R in 4 rounds if required. Since, for node v_5, $Val[v_5] = 0$, it is not required to send any value to R. Therefore, this protocol runs in a total of 4 rounds.

6 Linear Communication Complexity

In Section. 4.2, it has already been noted that, communication complexity of protocol Π_{Eff} is $O(n^2)$. Now we modify our protocol to get a generic $O(n)$ communication protocol. We achieve this by ensuring that PSMT is possible in a sub-graph of G, which has only $O(n)$ edges and no edge is used more than once, for transmitting a single field element. As this is a generic protocol, when we apply in $G^{r_{min}}(r_{min}$ is the minimum number of rounds required for PSMT) we get a round optimal linear communication protocol.

Definition 7. *Spanning tree of a graph G_u is denoted by $T = (V, E_T)$ is defined as a tree with R as its root(which is at 0^{th} level) and node $v \in V_R$ is at i^{th} level(in any order) if $d_v = i$(distance from v to R in digraph). Note that each node in i^{th} level points to only one node in the $(i-1)^{th}$ level, else we get cycles.*

Definition 8. *Suppose if we have k-vertex disjoint weak paths from S to R in G, namely p_i for $i \in [1, k]$. Communication graph of the graph $G(V, E)$ of order k is denoted by $\mathcal{G}^k(V, \mathcal{E})$ and is defined as $\mathcal{E} = E \cap (E_p \cup E_T)$. Where $E_p = \bigcup_{i=1}^{k} E(p_i)$ and E_{p_i} is the set of edges in weak path p_i.*

Theorem 4. PSMT *from S to R is possible in G* if and only if *it is possible in $\mathcal{G}^{(t+1)}$.*

Proof. Sufficiency: Suppose PSMT is possible from S to R in G then from Theorem 1, we have $(t+1)$-vertex disjoint weak paths from S to R in G. By construction of graph $\mathcal{G}^{(t+1)}$, every edge in these paths present in $\mathcal{G}^{(t+1)}$. We now show that we can simulate the protocol Π_{Rnd_Eff} given in Section 5.1. By using the edges in these $t+1$ paths, if required nodes can share their random numbers with neighbours. Also every node $v \in V_R$ is at i^{th} level in spanning tree T for some $i \in [1, n-1]$ and so has a shortest path to R for sending $Val[u_{ij}]$ to R.
Necessity: Suppose PSMT is not possible in G then clearly PSMT is not possible in sub graph $\mathcal{G}^{(t+1)}$ of G.

6.1 Round Optimal Protocol with Linear Communication Complexity $\Pi_{Rnd\_Opt\_Lin}$:

1. Every node u_{ij} except $u_{(t+1)0}$, in these $(t+1)$ paths pick a number $r_{ij} \in_R \mathbb{F}$, for $i \in [1, t+1]$ and $j \in [0, k_i+1]$. $S(= u_{(t+1)0})$ computes $r_{(t+1)0} = m - \sum_{i=1}^{t} r_{i0}$, this is possible for S to compute since r_{i0} is chosen by u_{i0} which is S itself.
2. For $i \in [1, t+1]$, $S(= u_{i0})$ initializes $L[u_{i0}] = r_{i0}$.
3. S computes $Val[S] = \sum_{i=1}^{t+1} Val[u_{i0}]$.
4. Let the height of the spanning tree T of G_u is h with root R is at 0^{th} level.
5. For each u_{ij} in path p_i, follow the protocol Π_{Rnd_Eff} exactly as in Section 5.1, except that instead of sending $Val[u_{ij}]$ to R using path $p_{u_{ij}}$ separately:

   (a) `If` u_{ij} is at h^{th} level(leaf node), u_{ij} sends $Val[u_{ij}]$ to its parent at $(h-1)^{th}$ level.
   (b) `Else` each u_{ij} which is in $k^{th}(k \in [1, h-1])$ level waits till it receives *Values* $Val[u_{ij}]$ from its children in $(k+1)^{th}$ level if required. Once u_{ij} receives *Values* from its children it *adds* all the received values to its `Value` $Val[u_{ij}]$ and sends to its parent which is at $(k-1)^{th}$ level.

6. In last round, R adds all the `values` received from its children which are at level one with its sum of `Left Values`(i.e. $\sum_{i=1}^{t+1} L[u_{i(k_i+1)}]$) to get message m.

Lemma 5. *The protocol $\Pi_{Rnd\_Opt\_Lin}$ is reliable.*

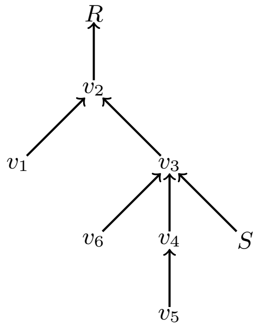
Proof. For each u_{ij}, $(j \neq k_i + 1)$ in path p_i, we have $R[u_{ij}] = L[u_{i(j+1)}]$
Let $Sum = \sum_{i=1}^{t+1} \sum_{j=0}^{k_i} Val[u_{ij}] + \sum_{i=1}^{t+1} L[u_{i(k_i+1)}]$. Now we show that $Sum = m$

$$\begin{aligned} Sum &= \sum_{i=1}^{t+1} \sum_{j=0}^{k_i} (L[u_{ij}] - R[u_{ij}]) + \sum_{i=1}^{t+1} L[u_{i(k_i+1)}] \\ &= \sum_{i=1}^{t+1} \sum_{j=0}^{k_i} (L[u_{ij}] - L[u_{i(j+1)}]) + \sum_{i=1}^{t+1} L[u_{i(k_i+1)}] \\ &= \sum_{i=1}^{t+1} (L[u_{i0}] - L[u_{i(k_i+1)}]) + \sum_{i=1}^{t+1} L[u_{i(k_i+1)}] \\ &= \sum_{i=1}^{t+1} L[u_{i0}] = \sum_{i=1}^{t+1} r_{i0} = m. \end{aligned}$$

Lemma 6. *The protocol $\Pi_{Rnd\_Opt\_Lin}$ is secure.*

Proof. Proof is analogous to the proof in Lemma 4. We shown that, in Lemma 4, protocol Π_{Eff} simulates the corresponding path p_i' of a secure weak path p_i, to securely transmit $p(i)$. In protocol $\Pi_{Rnd\_Opt\_Lin}$, $p(i)$ is replaced with r_{i0} and so r_{i0} is secure. This implies m is secure, since $m = r_{i0} + \sum_{j(\neq i)=1}^{t+1} r_{j0}$ and such r_{i0} is unique in any field $\mathbb{F}$.

The communication complexity of the above protocol $\Pi_{Rnd\_Opt\_Lin}$ is linear. Since, in first round as every node u_{ij} sends r_{ij} to its neighbours if require, which is $O(n)$ and each edge in spanning tree T is used only once to send a value from child to parent. We know that in a spanning tree with n nodes can not have more than n-1 edges, therefore total communication complexity is $O(n)$. Suppose shortest distance d_S from S to R is $\Omega(n)$ then we achieve `security for free` since the reliable communication itself requires O(n) communication. We give an example of the protocol $\Pi_{Rnd\_Opt\_Lin}$ in graph G given in Fig. 3. Tree constructed based on the shortest distances from each node to R is given in Fig. 6. S chooses two random numbers r_1, r_2 and initializes $r_3 = m - (r_1 + r_2)$. S replaces $p(i)$ with r_i, for $i \in [1, 3]$.



1. Round 1: Every node shares their random numbers, exactly as in examle given in section 5.4, except that $p(i)$ is replaced with r_i.
2. Round 2: Every node v calculates its value $Val[v]$. Nodes v_4, v_6, S will send $Val[v_4], Val[v_6], Val[S]$ to v_3 respectively.
3. Round 3: Node v_3 calculates the sum, $Sum(v_3) = Val[v_3] + Val[v_4] + Val[v_6] + Val[S]$ and sends to v_2.
4. Round 4: Node v_2 calculates $sum(v_2) = Sum(v_3) + Val[v_2]$ and sends to R. Finally R computes $m = sum(v_2) + r_{v_2} + r_{R_2} + r_{R_3}$

Fig. 6. An example of protocol $\Pi_{Rnd\_Opt\_Lin}$ in graph G of Fig. 3

7 Conclusions and Open Problems

We have completely characterized the feasibility and optimality of PSMT in arbitrary networks under the influence of passive adversary. Similar characterization for the case of Byzantine and/or Mobile adversary has been left as an interesting open problem.

References

1. Franklin, M.K., Yung, M.: Secure hypergraphs: privacy from partial broadcast (extended abstract). In: Leighton, F.T., Borodin, A. (eds.) STOC, pp. 36–44. ACM (1995)
2. Hirt, M., Maurer, U.: Complete Characterization of Adversaries Tolerable in Secure Multi-party Computation. In: Proceedings of the 16th Symposium on Principles of Distributed Computing (PODC), pp. 25–34. ACM Press (August 1997)
3. Ostrovsky, R., Yung, M.: How to Withstand Mobile Virus Attacks. In: Proceedings of the 10th Symposium on Principles of Distributed Computing (PODC), pp. 51–61. ACM Press (1991)
4. Srinathan, K., Raghavendra, P., Chandrasekaran, P.R.: On proactive perfectly secure message transmission. In: Pieprzyk, J., Ghodosi, H., Dawson, E. (eds.) ACISP 2007. LNCS, vol. 4586, pp. 461–473. Springer, Heidelberg (2007)
5. Fitzi, M., Hirt, M., Maurer, U.M.: Trading Correctness for Privacy in Unconditional multi-party Computation. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 121–136. Springer, Heidelberg (1998)
6. Choudhary, A., Patra, A., Ashwinkumar, B.V., Srinathan, K., Rangan, C.P.: Perfectly reliable and secure communication tolerating static and mobile mixed adversary. In: Safavi-Naini, R. (ed.) ICITS 2008. LNCS, vol. 5155, pp. 137–155. Springer, Heidelberg (2008)
7. Sayeed, H., Abu-Amara, H.: Perfectly Secure Message Transmission in Asynchronous Networks. In: Seventh IEEE Symposium on Parallel and Distributed Processing (1995)

8. Dolev, D., Dwork, C., Waarts, O., Yung, M.: Perfectly Secure Message Transmission. Journal of the Association for Computing Machinery (JACM) 40(1), 17–47 (1993)
9. Nayak, M., Agrawal, S., Srinathan, K.: Minimal connectivity for unconditionally secure message transmission in synchronous directed networks. In: Fehr, S. (ed.) ICITS 2011. LNCS, vol. 6673, pp. 32–51. Springer, Heidelberg (2011)
10. Menger, K.: Zur allgemeinen kurventheorie. Fundamenta Mathematicae 10, 96–115 (1927)
11. Kurosawa, K., Suzuki, K.: Truly efficient 2-round perfectly secure message transmission scheme. IEEE Trans. Inf. Theor. 55(11), 5223–5232 (2009)
12. Badanidiyuru, A., Patra, A., Choudhury, A., Srinathan, K., Rangan, C.P.: On the trade-off between network connectivity, round complexity, and communication complexity of reliable message transmission. J. ACM 59(5), 22 (2012)
13. Fitzi, M., Franklin, M.K., Garay, J.A., Vardhan, S.H.: Towards optimal and efficient perfectly secure message transmission. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 311–322. Springer, Heidelberg (2007)
14. Renault, J., Renou, L., Tomala, T.: Secure message transmission on directed networks. Games and Economic Behavior 85, 1–18 (2014)
15. Kumar, M.V.N.A., Goundan, P.R., Srinathan, K., Pandu Rangan, C.: On perfectly secure communication over arbitrary networks. In: Proceedings of the 21st Symposium on Principles of Distributed Computing (PODC), Monterey, California, USA, pp. 193–202. ACM Press (July 2002)
16. Diffie, W., Hellman, M.E.: New Directions in Cryptography. IEEE Transactions on Information Theory IT-22, 644–654 (1976)
17. Desmedt, Y.G., Wang, Y.: Perfectly Secure Message Transmission Revisited. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 502–517. Springer, Heidelberg (2002)
18. Srinathan, K., Rangan, C.P.: Possibility and complexity of probabilistic reliable communications in directed networks. In: Proceedings of 25th ACM Symposium on Principles of Distributed Computing, PODC 2006 (2006)
19. Franklin, M.K., Yung, M.: Communication complexity of secure computation (extended abstract). In: Kosaraju, S.R., Fellows, M., Wigderson, A., Ellis, J.A. (eds.) Proceedings of the 24th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 4-6, pp. 699–710. ACM (1992)
20. Shamir, A.: How to Share a Secret. Communications of the ACM 22, 612–613 (1979)
21. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 3rd edn. MIT Press (2009)
22. Endre Tarjan, R.: Testing graph connectivity. In: Proceedings of the Sixth Annual ACM Symposium on Theory of Computing, STOC 1974, pp. 185–193. ACM, New York (1974)
23. Goldberg, A.V., Tarjan, R.E.: A new approach to the maximum flow problem. Journal of the ACM 35, 921–940 (1988)