Efficient Perfectly Reliable and Secure Communication Tolerating Mobile Adversary

Arpita Patra **Ashish Choudhary*** **Madhu Gayatri** **C. Pandu Rangan**
Department of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai India 600036
Email:{ arpita,ashishc,madhu }@cse.iitm.ernet.in, rangan@iitm.ernet.in

Abstract

We study the problem of *Perfectly Reliable Message Transmission* (PRMT) and *Perfectly Secure Message Transmission* (PSMT) between two nodes **S** and **R** in an undirected synchronous network, a part of which is under the influence of an *all powerful mobile Byzantine* adversary. In ACISP'2007 Srinathan *et. al.* has proved that the connectivity requirement for PSMT protocols is same for both *static* and *mobile* adversary thus showing that *mobility* of the adversary has *no* effect on the *possibility* of PSMT (also PRMT) protocols. Similarly in CRYPTO 2004, Srinathan *et. al.* has shown that the lower bound on the communication complexity of any multiphase PSMT protocol is same for static and mobile adversary. The authors have also designed a $O(t)$ phase[1] protocol satisfying this bound where t is the maximum number of nodes corrupted by the Byzantine adversary. In this work, we design a *three* phase *bit optimal* PSMT protocol using a novel technique, whose communication complexity matches the lower bound proved in CRYPTO 2004 and thus significantly reducing the number of phases from $O(t)$ to three. Further using our novel technique, we design a three phase *bit optimal* PRMT protocol which achieves reliability with *constant factor* overhead against a mobile adversary. These are the *first* ever constant phase *optimal* PRMT and PSMT protocols against mobile Byzantine adversary. We also characterize PSMT protocols in *directed* networks tolerating mobile adversary.

All the existing PRMT and PSMT protocols abstracts the paths between **S** and **R** as wires, neglecting the intermediate nodes in the paths. However, this causes significant over estimation in the communication complexity as well as round complexity[2] of protocols. Here, we consider the underlying paths as a whole instead of abstracting them as wires and derive a tight bound on the number of rounds required to achieve reliable communication from **S** to **R** tolerating a mobile adversary with arbitrary roaming speed[3]. We show how our constant phase PRMT and PSMT protocols can be easily adapted to design *round optimal* and *bit optimal* PRMT and PSMT protocols provided the network is given as a collection of vertex disjoint paths.

Keywords: Information Theoretic Security, Communication Efficiency, Mobile Adversary.

*Work Supported by Project No. CSE/05-06/076/DITX/CPAN on Protocols for Secure Communication and Computation Sponsored by Department of Information Technology, Government of India.

[1] A phase is a send from **S** to **R** or **R** to **S** or both

[2] Round is different from phase. Round is a send from one node to its immediate neighbor in the network.

[3] By roaming speed we mean the speed with which the adversary changes the set of corrupted node

1 Introduction

Consider the following problem: a sender **S** and a receiver **R**, who want to "talk" to each other via an underlying communication network that they do not trust. Note that if **S** and **R** are connected directly via a private and authenticated link (like in the generic solutions for secure multiparty computation [2, 7, 14, 19]), secure communication is trivially guaranteed. However, in reality, it is not economical to directly connect *every* two players in the network. The sender's distrust in the underlying communication network is usually modeled by a virtual entity called the *adversary* that has *unbounded computing power* and can corrupt some of the players (nodes) in the network. In spite of the presence of such an adversary in the network, **S** wishes to send a message m chosen from a finite field $\mathbb{F}$, reliably to **R**, in a *guaranteed manner*. This problem is called *perfectly reliable message transmission* (PRMT). The problem of *perfectly secure message transmission* (PSMT) has an additional constraint that the adversary should get *no* information about m. Security against such an unbounded powerful adversary is called *information theoretic or perfect security*. Since the adversary has *unbounded computing power*, we cannot use public key cryptography, digital signature, etc to solve PRMT/PSMT problem because they assume that adversary has *polynomial time* computing power.

There have been a variety of adversary models used in the literature, each one catering to a different real-life setting. Dolev *et al* [5], who introduced and studied the problem of PSMT assume that the adversary can corrupt up to any t nodes in the network and that the adversary is *static* Byzantine, i.e., a player once corrupted remains so subsequently. If a node P is under the control of Byzantine adversary, then the adversary fully dictates the action of P. The adversary will have full access to the internal state and computation of P and can force P to deviate from the protocol arbitrarily. More recent efforts using the same (static) adversarial model for the problem of PSMT include [15, 17, 10, 4, 13].

However, as first noticed in [12], the static model implicitly assumes that the number of dishonest players in the network is independent of the protocol's execution time. This is usually not true in practice. Furthermore, since a corrupted player could be corrected given sufficient time, [12] proposed the mobile adversary model wherein the adversary could move around the network whilst still corrupting up to t players at any given instant. Subsequently, extensive research efforts on tolerating mobile adversaries have resulted in what is now well-known as *proactive security* [9, 8, 6, 1].

1.1 Existing Results

It is known that for the existence of any r-phase ($r \geq 2$) PRMT/PSMT protocol, $n \geq 2t+1$ vertex disjoint paths (also called as wires) between **S** and **R** [5] is necessary and sufficient to tolerate a t-active static adversary. Also, as reported in [17], any r phase ($r \geq 2$) PSMT protocol has a communication complexity of $\Omega\left(\frac{n\ell}{n-2t}\right)$ to securely send ℓ field elements against a t-active static adversary. While for PSMT we have a proven lower bound for communication complexity, for PRMT it can be as small as $\Omega(l)$ for communicating message of ℓ field elements. The authors of [13] have designed a three phase PRMT protocol which satisfies the above defined bound and sends a message containing ℓ field elements by communicating $O(\ell)$ field elements. Such a protocol is called *bit-optimal* PRMT protocol. In addition, the authors [13] also reported a three phase

PSMT protocol, whose communication complexity is $O\left(\frac{n\ell}{n-2t}\right)$ (asymptotically touching the lower bound specified for multiphase PSMT) and hence it is *bit optimal* against a static adversary.

Unlike *static adversary*, a t-active *mobile adversary* can corrupt different set of t wires during different phases of the protocol. Thus, a wire once corrupted, may not remain corrupted in subsequent phases. Intuitively, it is more difficult to tolerate a t-active *mobile adversary* in comparison to a t-active *static adversary*. However, in [18], it is shown that $n \geq 2t+1$ wires between $\mathbf{S}$ and $\mathbf{R}$ is necessary and sufficient for the possibility of any r-phase ($r \geq 2$) PRMT/PSMT protocol against a t-active mobile adversary. Thus mobility of adversary *does not* affects its tolerability. In [17], the communication complexity of any r-phase ($r \geq 2$) PSMT protocol is stated to be $\Omega\left(\frac{n\ell}{n-2t}\right)$, where ℓ is the message to be sent securely against a t-active mobile adversary. The authors of the same paper has also designed a $O(t)$ phase PSMT protocol satisfying the bound.

1.2 Our Contribution, Its Motivation and Significance

One of the implicit assumptions made in all the works done on static adversary model is that *the players once faulty, remain so subsequently for the rest of the protocol execution.* While such assumption is justified with respect to short-lived and fast protocols, it is invariably too conservative in adequately capturing the fault patterns in large and time-consuming protocols.This observations naturally motivates the study of PRMT and PSMT with mobile adversary settings which we do in this work. The following are the main contribution of this paper:

(a) A *bit-optimal* three phase PRMT protocol, which sends a message of ℓ field elements by communicating $O(\ell)$ field elements and thus achieves reliability with *constant factor* overhead in three phases even in the presence of mobile adversary.

(b) A *bit-optimal* three phase PSMT protocol satisfying the bound for communication complexity proved in [17].

Both these protocols uses a *novel* technique, very different from the techniques adapted in the three phase *bit-optimal* PRMT and PSMT protocol proposed in [13] tolerating a static adversary.

(c) We give the *first ever* characterization of PSMT protocols in *directed* networks tolerating mobile adversary.

All existing PRMT and PSMT protocols abstract the underlying network as vertex disjoint paths, called *wires*, between $\mathbf{S}$ and $\mathbf{R}$, thus neglecting the intermediate nodes in these paths. However, as shown in [16], such an abstraction gives an incorrect estimation on the communication complexity and round complexity of PRMT and PSMT protocols, in many practical scenarios. Hence, it is essential to consider all the intermediate nodes in each wire for the design and analysis of PRMT and PSMT protocols. Also, considering the intermediate nodes/details of each wire motivates to use more finer notion of *round* in comparison to phase. Accordingly, the behavior of *mobile adversary* is re-defined to allow the adversary to corrupt any set of t nodes after every $\rho \geq 1$ rounds, where ρ is called the roaming speed of the adversary. In this work, our contribution also encompasses:

(d) Computation of a tight bound on the minimum number of rounds r_{min}, required for the existence of any PRMT protocol tolerating mobile adversary, with roaming speed of $\rho = 1$. The same for an adversary with arbitrary roaming speed $\rho \geq 2$.

(e) Finally, adaptation of our constant *phase* PRMT and PSMT protocols into round optimal and communication optimal PRMT and PSMT protocols in a given network, provided the network is given as a collection of disjoint paths.

As mentioned earlier, abstraction of network as wires leads to incorrect estimation on communication and round complexity of protocols. But still wired abstraction eases deriving lower bounds on communication complexity and finding out the connectivity requirement for PRMT/PSMT problem and also simplifies the analysis of protocols. The same reason answers for why we have designed phase-based protocols for PRMT and PSMT and later adapted them to work in terms of rounds.

2 PRMT and PSMT Tolerating Mobile Adversary (in Terms of Phases)

Here we design constant phase *bit-optimal* PRMT and PSMT protocols in undirected network. We first define the network settings and computational model used for designing *phase-based* protocols.

2.1 Network Settings and Computational Model

Recall that a phase is a send from **S** to **R** or vice-versa. While designing protocols in terms of phases, following the approach of [5], we abstract the network as a collection of vertex disjoint paths called wires between **S** and **R**, neglecting the intermediate nodes in these paths. A t-active mobile adversary can corrupt different set of t wires during different phases of the protocol. Hence a wire w, which is corrupted in some phase, may not remain corrupted during subsequent phases and can behave honestly. Also by corrupting a wire w during a particular phase, adversary does not get any information which was transmitted over w in earlier phase(s) (unless w was corrupted in earlier phase(s) also). Thus, if $t = 1$ and adversary has corrupted wire w during first phase and wire $w' \neq w$ during second phase, then adversary will combinedly know the information transmitted through w (w') during first (second) phase. Also w' (w) will behave honestly during first (second) phase. We assume that **S** and **R** are connected by $n \geq 2t + 1$ bi-directional wires $w_1, w_2, \ldots w_n$, which is necessary and sufficient for PRMT/PSMT protocols against a t-active mobile adversary [17, 16]. Any information which is sent over all the n wires is said to be "broadcast". Any information which is "broadcast" over $n > 2t$ wires will always be recovered correctly at the receiving end by taking the majority.

2.2 Black Box Used in Our Protocols

We now briefly describe the black boxes used in our protocols.

2.2.1 Extracting Randomness

Consider the following problem: **S** and **R** agree on a n-tuple $x = [x_1, x_2, \ldots x_n] \in \mathbb{F}^n$ such that the adversary knows $n - f$ components of x, but has no information (in information theoretic sense) about the other f components of x. However, **S** and **R** do not necessarily know which values are known to the adversary. But they want to agree on a sequence of f elements $y_1, y_2, \ldots y_f \in \mathbb{F}$ such that $y_1, y_2, \ldots y_f$ is information theoretically secure. This is achieved by $\mathbf{EXTRAND}_{n,f}(x)$ proposed in [17].

Lemma 1 ([17]) *The adversary has no information about* $[y_1\ y_2\ \ldots\ y_f]$ *computed in EXTRAND.*

Algorithm EXTRAND$_{n,f}(x)$. Let V be a $n \times f$ Vandermonde matrix with members in $\mathbb{F}$, which is a part of protocol specification. **S** and **R** both locally compute the product $[y_1\ y_2\ \ldots\ y_f] = [x_1\ x_2\ \ldots\ x_n]V$.

2.2.2 Communicating Conflict Graph

Consider the following scenario: **S** and **R** are connected by $n = 2t+1$ bi-directional wires. **S** selects at random n polynomials $p_i(x)$, $1 \leq i \leq n$ over $\mathbb{F}$, each of degree t. Next through wire $w_i, 1 \leq i \leq n$, **S** sends to **R** the polynomial $p_i(x)$ and for each j, $1 \leq j \leq n$, the value of $p_j(\alpha_i)$ denoted by r_{ji}, where α_i's are arbitrary distinct publicly specified members of $\mathbb{F}$.

Let **R** receives polynomial $p'_i(x)$ and the values r'_{ji} along w_i. **R** tries to find as many faults as he can find that occurred in the previous phase and communicates his findings reliably to **S**. Towards this, **R** constructs a directed graph called conflict graph $H = (\mathcal{W}, E)$, where $\mathcal{W} = \{w_1, w_2, \ldots, w_n\}$ and arc $(w_i, w_j) \in E$ if $r'_{ij} \neq p'_i(\alpha_j)$; i.e., if the value of the received polynomial $p'_i(x)$ evaluated at $x = \alpha_j$, does not match the corresponding received value r'_{ij}. There can be $\Theta(n^2)$ arcs in the conflict graph. For each $(w_i, w_j) \in E$, **R** adds a four tuple $\{w_i, w_j, p'_i(\alpha_j), r'_{ij}\}$ to a list X. **R** then broadcasts X to **S**. **S** reliably receives X. For each $\{w_i, w_j, p'_i(\alpha_j), r'_{ij}\} \in X$, **S** verifies $r'_{ij} \stackrel{?}{=} r_{ij}$ and $p'_i(\alpha_j) \stackrel{?}{=} p_i(\alpha_j)$. Depending upon the outcome of the test, **S** concludes that either **R** has received incorrect r'_{ij} over wire w_j or **R** has received incorrect $p'_i(x)$ over wire w_i (or both) and hence accordingly adds w_i or w_j (or both) to a list L_{fault}. **S** then broadcasts L_{fault} to **R**. Now we can say the following:

Theorem 1 **S** *will always be able to identify the wires over which* **R** *has received faulty polynomial during first phase. Moreover,* **S** *will be able to reliably send this information to* **R**.

Proof: Suppose wire w_i has been corrupted in first phase; i.e., $p_i(x) \neq p'_i(x)$. Then the two polynomials can intersect at, at most t points, since both are of degree t. Since there are at least $t+1$ honest wires, it may happen that $p_i(x) = p'_i(x)$ for at most t α_k's corresponding to t honest wires, so there is at least one honest wire w_j, such that $r_{ij} = r'_{ij}$ and $p'_i(\alpha_j) \neq r'_{ij}$, which will contradict w_i and so the arc (w_i, w_j) will be present in the conflict graph and hence the four tuple $\{\alpha_i, \alpha_j, p'_i(\alpha_j), r'_{ij}\}$ will be present in the list X. Since X is broadcast over $2t+1$ wires, **S** will correctly receive X and eventually knows all the corrupted polynomials; i.e., **S** knows all the wires w_i over which the **R** has received *corrupted polynomial* $p_i(x)$ and adds them to L_{fault} and then reliably sends L_{fault} to **R** by broadcasting. □

Theorem 2 *The communication complexity of broadcasting the list* X *is* $O(n^3)$.

Proof: The proof follows from the fact that there can be $\Theta(n^2)$ arcs in the conflict graph and corresponding to each arc, there exist a four tuple in X. □

Remark 1 *An efficient way of sending the conflict graph (which contains* $O(n^2)$ *edges) by communicating* $O(n^2)$ *field elements was introduced in [17] and subsequently used in [13]. The method deals with finding maximum matching of conflict graph and a few notions from coding theory. However, the same technique can not be adopted here against mobile adversary, as it can choose to corrupt*

different set of t wires in different phases. So, we introduce a novel technique called **Union technique** *(described in the next section) which enables us to send n conflict graphs by communicating $O(n^3)$ field elements. Later we use this technique to design bit-optimal PRMT/PSMT protocols against a mobile adversary.*

2.3 PRMT with Constant Factor Overhead Tolerating Mobile Adversary

We propose a three phase PRMT protocol **PRMT_Optimal** which sends a message containing $n(t+1)^2 = O(n^3)$ field elements by communicating $O(n^3)$ field elements against a t-active mobile Byzantine adversary, where **S** and **R** are connected by $n = 2t+1$ bi-directional wires. Thus, **PRMT_Optimal** achieves reliability with constant factor overhead in constant phases and thus is *bit-optimal*. In [13], a three phase *bit-optimal* PRMT protocol had been presented against a static adversary which sends $O(n^2)$ messages by communicating $O(n^2)$. Thus, extra adversarial power of *mobility* does not hinder to achieve *bit-optimality* in the same number of phases (three) except that the optimality is achieved for larger message size!! Before describing the protocol **PRMT_Optimal**, we describe a novel technique used in our protocol which we call as **Union Technique** for combining n conflict graphs.

Union Technique: Recall the same scenario described in section 2.2.2. In first phase **R** receives n polynomials $p'_i(x), 1 \leq i \leq n$, each of degree t (out of which at most t can be corrupted) and n values corresponding to each polynomial (out of which at most t can be corrupted) denoted by r'_{ij}. Let B denote the set of n polynomials and their n values as received by **R**. Using B, **R** can construct a conflict graph. Now, in our three phase PRMT protocol **PRMT_Optimal** , during **Phase I**, instead of a single set B, **R** receives n such sets denoted as $B_k, 1 \leq k \leq n$, where B_k contains n polynomials $p'_{ki}(x), 1 \leq i \leq n$ and n values for each $p'_{ki}(x)$ denoted by $r'_{ki,j}, 1 \leq j \leq n$. The flow of information over n wires during **Phase I** is given in Table 1.

Table 1: Data Flow over n wires in **Phase I** of **PRMT_Optimal**

Wire	B_1	...	B_k	...	B_n
w_1	$P_{11}(x)\ r_{11,1}, r_{12,1}, \ldots r_{1n,1}$	...	$P_{k1}(x)\ r_{k1,1}, r_{k2,1}, \ldots r_{kn,1}$	...	$P_{n1}(x)\ r_{n1,1}, r_{n2,1}, \ldots r_{nn,1}$
w_2	$P_{12}(x)\ r_{11,2}, r_{12,2}, \ldots r_{1n,2}$	...	$P_{k2}(x)\ r_{k1,2}, r_{k2,2}, \ldots r_{kn,2}$	...	$P_{n2}(x)\ r_{n1,2}, r_{n2,2}, \ldots r_{nn,2}$
...	...	...	...	...	...
w_i	$P_{1i}(x)\ r_{11,i}, r_{12,i}, \ldots r_{1n,i}$	...	$P_{ki}(x)\ r_{k1,i}, r_{k2,i}, \ldots r_{kn,i}$	...	$P_{ni}(x)\ r_{n1,i}, r_{n2,i}, \ldots r_{nn,i}$
...	...	...	...	...	...
w_n	$P_{1n}(x)\ r_{11,n}, r_{12,n}, \ldots r_{1n,n}$	...	$P_{kn}(x)\ r_{k1,n}, r_{k2,n}, \ldots r_{kn,n}$	...	$P_{nn}(x)\ r_{n1,n}, r_{n2,n}, \ldots r_{nn,n}$

R then constructs conflict graph H_k using the set B_k. For each H_k, we can say the following from Theorem 1: if during **Phase I**, **R** receives a corrupted polynomial $p'_{ki}(x) \neq p_{ki}(x)$ over w_i, then there exist at least one directed arc (w_i, w_j) in H_k, where w_j is an honest wire (out of the $t+1$ honest wires not under the control of the adversary). If **R** broadcasts all conflict graphs, then from Theorem 1, both **S** and **R** would come to know the identity of all faulty wires w_i over which **R** has received at least one faulty $p'_{ki}(x), 1 \leq k \leq n$ during **Phase I**. However, from Theorem 2, broadcasting all of them requires communicating $O(n^4)$ field elements. So we now introduce a very intelligent method of combining n conflict graphs into a single directed conflict graph H. By

broadcasting H to $\mathbf{S}$, $\mathbf{R}$ can ensure that $\mathbf{S}$ will be able to identify all w_i's over which $\mathbf{R}$ has received at least one faulty polynomial $p'_{ki}(x)$. The combined directed conflict graph $H = (V, E)$ will have vertices and edges as follows: $V = \{w_1, w_2, \ldots, w_n\}$ and $E = \{(w_i, w_j)\}$ where arc $(w_i, w_j) \in E$ if (w_i, w_j) occurs in at least one $H_k, 1 \leq k \leq n$. Since an arc (w_i, w_j) can occur in multiple H_k's, $\mathbf{R}$ considers (w_i, w_j) from the minimum indexed H_γ among all such H_k's, keeping a note that (w_i, w_j) is added from H_γ. For each $(w_i, w_j) \in E$, $\mathbf{R}$ adds a five tuple $\{w_i, w_j, \gamma, p'_{\gamma i}(\alpha_j), r'_{\gamma i,j}\}$ to a list X, provided (w_i, w_j) is taken from H_γ. This indicates that in the set B_γ, the value of the polynomial $p'_{\gamma i}(x)$ received over w_i, when evaluated at α_j, does not match with the corresponding value $r'_{\gamma i,j}$ received over w_j. It is easy to see that there can be $\Theta(n^2)$ edges in H and hence $\Theta(n^2)$ tuples in X. In the next theorem, we prove that $\mathbf{S}$ can identify all faulty wires over which $\mathbf{R}$ received at least one faulty polynomial after receiving X.

Theorem 3 *In the* **Union Technique**, *if* $\mathbf{R}$ *broadcasts* X *to* $\mathbf{S}$, *then* $\mathbf{S}$ *identifies all faulty wires* w_i *over which* $\mathbf{R}$ *has received at least one corrupted polynomial* $p'_{ki}(x)$.

Proof: Suppose during **Phase I**, $\mathbf{R}$ receives a faulty polynomial $p'_{ki}(x)$ over w_i. Then from Theorem 1, there exists at least one arc $(w_i, w_j) \in H_k$, where w_j is an honest wire. Since the combined conflict graph H is formed by considering all the arcs in the individual H_k's, $1 \leq k \leq n$, list X must have a five tuple $\{w_i, w_j, \gamma, p'_{\gamma i}(\alpha_j), r'_{\gamma i,j}\}$. Now there are two possibilities: (i) $\gamma = k$ which indicates the five tuple exactly corresponds to the arc $(w_i, w_j) \in H_k$. else (ii) $\gamma < k$ which indicates the five tuple corresponds to the arc $(w_i, w_j) \in H_\gamma$ which in turn implies polynomial $p'_{\gamma i}(x)$ has also been corrupted. Hence, adding five tuple for the arc $(w_i, w_j) \in H_\gamma$ in H will not effect in identifying w_i as *a wire delivering at least one faulty polynomial*. This follows from the fact that no unchanged polynomial over w_i can be contradicted by honest wire w_j. Thus, for each faulty w_i delivering at least one incorrect polynomial during **Phase I**, there exists a five tuple in X. Hence, when $\mathbf{R}$ broadcasts X, $\mathbf{S}$ will identify all faulty wires over which $\mathbf{R}$ received at least one faulty polynomial $p'_{ki}(x)$ after performing local verification. □

Now we are well-equipped to understand **Protocol PRMT_Optimal**. Intuitively, the protocols works as follows: $\mathbf{S}$ selects n bivariate polynomials whose coefficients are the message to be sent. $\mathbf{S}$ then generates n sets B_k, $1 \leq k \leq n$ from n bivariate polynomials and communicates them to $\mathbf{R}$ in **Phase I**. On receiving n B_k's, $\mathbf{R}$ first constructs n conflict graphs H_k's and then combine all of them to a single graph H and broadcast H to $\mathbf{S}$ in **Phase II**. In **Phase III** $\mathbf{S}$ identifies all faulty wires from the knowledge of H and sends them across to $\mathbf{R}$. Finally, $\mathbf{R}$ recovers the message by reconstructing all the n bivariate polynomials using the identity of the faulty wires communicated by $\mathbf{S}$. We now proceed to show the correctness of the protocol.

Theorem 4 *In the protocol* **PRMT_Optimal**, $\mathbf{R}$ *will always be able to correctly recover the message.*

Proof: In **PRMT_Optimal**, to recover m, $\mathbf{R}$ should be able to interpolate each bivariate polynomial $q_k(x, y), 1 \leq k \leq n$. Since each $q_k(x, y)$ is of degree t in both x and y, $\mathbf{R}$ requires $t + 1$ correct $q_k(x, \alpha_i) = p_{ki}(x)$'s to recover $q_k(x, y)$. Since among n wires at most t can be corrupted, $\mathbf{R}$ will receive at least $t + 1$ correct $p_{ki}(x)$'s. Now $\mathbf{R}$ wants to know the identity of $t + 1$ correct $p_{ki}(x)$'s. During **Phase II**, $\mathbf{R}$ constructs n conflict graph $H_k, 1 \leq k \leq n$ and combine them into a single conflict graph H using **Union Technique**, forms X and broadcasts it to $\mathbf{S}$. From Theorem 3, on receiving X, $\mathbf{S}$ identifies all faulty wires over which $\mathbf{R}$ has received at least one faulty polynomial

Protocol PRMT_Optimal : A Three Phase Optimal PRMT Protocol Against Mobile Adversary

Let the sequence of $n(t+1)^2$ field elements that **S** wishes to transmit be denoted by $m_{k,ij}$, $0 \leq i,j \leq t$ and $1 \leq k \leq n$.

Phase I: (S to R)

- Using the $m_{k,ij}$ values, **S** defines n bivariate polynomials $q_k(x,y), 1 \leq k \leq n$ as follows: $q_k(x,y) = \sum_{i=0,j=0}^{i=t,j=t} m_{k,ij} x^i y^j$
- **S** then evaluates each $q_k(x,y), 1 \leq k \leq n$ at n publicly known distinct values $\alpha_1, \alpha_2, \ldots, \alpha_n$ to obtain total n^2 polynomials denoted as $p_{ki}(x), 1 \leq k \leq n, 1 \leq i \leq n$ over $\mathbb{F}$, each of degree t where $p_{ki}(x) = q_k(x, \alpha_i)$. Over wire $w_i, 1 \leq i \leq n$, **S** sends the polynomials $p_{ki}(x), 1 \leq k \leq n$ and the values $p_{kj}(\alpha_i)$, denoted by $r_{kj,i}$, for $1 \leq k,j \leq n$ (see Table 1).

Phase II (**R** to **S**)

- Let **R** receives over wire $w_i, 1 \leq i \leq n$ the polynomials $p'_{ki}(x)$ and the values $r'_{kj,i}, 1 \leq k,j \leq n$. **R** then considers the polynomials $p'_{11}(x), p'_{12}(x), \ldots, p'_{1n}(x)$ and the values $r'_{1j,i}, 1 \leq j,i \leq n$ and constructs the conflict graph H_1 as explained in section 2.2.2; i.e., $(w_i, w_j) \in H_1$ if $p'_{1i}(\alpha_j) \neq r'_{1i,j}$. Similarly, **R** considers the polynomials $p'_{21}(x), p'_{22}(x), \ldots, p'_{2n}(x)$ and the values $r'_{2j,i}, 1 \leq j,i \leq n$ and constructs the conflict graph H_2. In general, **R** considers the polynomials $p'_{k1}(x), p'_{k2}(x), \ldots, p'_{kn}(x)$ and the values $r'_{kj,i}, 1 \leq j,i \leq n$ and constructs the conflict graph $H_k, 1 \leq k \leq n$.
- **R** combines $H_k, 1 \leq k \leq n$ into a single directed conflict graph H using **Union Technique** and forms the corresponding list of five tuples X and broadcasts X to **S**.

Phase III (**S** to **R**)

• **S** reliably receives the list X. **S** then creates a list L_{fault} which is initialized to $\emptyset$. For each tuple $\{w_i, w_j, k, p'_{ki}(\alpha_j), r'_{ki,j}\} \in X$, **S** locally verifies $r'_{ki,j} \stackrel{?}{=} r_{ki,j}$ and $p'_{ki}(\alpha_j) \stackrel{?}{=} p_{ki}(\alpha_j)$. Depending upon the output of the verification, **S** concludes that w_i or w_j or both are faulty and add to L_{fault}. **S** finally broadcasts the list L_{fault} to **R** and terminates the protocol.

Message Recovery by R.

• **R** reliably receives L_{fault} and identifies all w_i over which it had received at least one faulty polynomial $p'_{ki}(x), 1 \leq k \leq n$ during **Phase I** (see Theorem 4). **R** neglects all the polynomials $p'_{ki}(x), 1 \leq k \leq n$ for each $w_i \in L_{fault}$,. Using the remaining (at least) $t+1$ p'_{ki}'s, $1 \leq k \leq n$, **R** correctly recovers the bivariate polynomials $q_k(x,y)$'s, $1 \leq k \leq n$ and hence the message.

during **Phase I** and adds them to L_{fault} and broadcasts to **R**. **R** neglects all the n polynomials received over each $w_i \in L_{fault}$. Since $|L_{fault}| \leq t$, **R** will have at least $t+1$ correct $p_{ki}(x)$ for each $1 \leq k \leq n$, using which **R** recovers each $q_k(x,y)$ and hence m. □

Theorem 5 *The communication complexity of* **PRMT_Optimal** *is* $O(n^3)$.

Proof: During **Phase I**, **S** sends over each wire n polynomials of degree t and n^2 values. So communication complexity of **Phase I** is $O(n^3)$. During **Phase II**, **R** broadcasts the list X. As explained earlier, X contains $\Theta(n^2)$ tuples. Hence broadcasting X requires $O(n^3)$ communication complexity. During **Phase III**, **S** broadcasts the list L_{fault}. Since $|L_{fault}| \leq t$, this involves communicating $O(nt) = O(n^2)$. Hence the overall communication complexity of **PRMT_Optimal** is $O(n^3)$. □

Remark 2 **PRMT_Optimal** *sends* $n(t+1)^2 = O(n^3)$ *field elements by communicating* $O(n^3)$ *field elements. Since any PRMT protocol for communicating* ℓ *field elements has to communicate* $\Omega(\ell)$ *field elements [16], our protocol is asymptotically optimal. Since any field element is represented*

by $log_2(|\mathbb{F}|)$ bits, **PRMT_Optimal** *sends $n^3 log_2(|\mathbb{F}|)$ bits by communicating $O(n^3 log_2(|\mathbb{F}|))$ bits. Hence* **PRMT_Optimal** *is bit optimal.*

2.4 Constant Phase Bit Optimal Proactive PSMT Protocol

We now present a three phase PSMT protocol **PSMT_Optimal** which securely sends $n(t+1) = O(n^2)$ field elements by communicating $O(n^3)$ field elements against a t-active mobile Byzantine adversary where **S** and **R** are connected by $n = 2t+1$ wires, thus matching the lower bound on the communication complexity of multiphase PSMT protocol against a t-active mobile adversary, as proved in [17]. This significantly reduces the $O(t)$ phase communication optimal PSMT protocol given in [17].

Protocol PSMT_Optimal: A Three Phase Optimal PSMT Protocol Against Mobile Adversary

Let the sequence of $n(t+1)$ field elements that **S** wishes to transmit securely be denoted by $m_i, 1 \leq i \leq n(t+1)$.

Phase I: S to R

• **S** selects n^2 polynomials $p_{ki}(x), 1 \leq k, i \leq n$ over $\mathbb{F}$ each of degree t where the coefficients of $p_{ki}(x)$'s are randomly chosen from $\mathbb{F}$. Over $w_i, 1 \leq i \leq n$, **S** sends the polynomial $p_{ki}(x), 1 \leq k \leq n$ and the values $p_{kj}(\alpha_i)$, denoted by $r_{kj,i}$, for $1 \leq k, j \leq n$ (the flow of data is similar to Table 1 except that here $p_{ki}(x)$'s are independent of m).

Phase II (R to S)

• Let **R** receives over $w_i, 1 \leq i \leq n$ the polynomials $p'_{ki}(x)$ and the values $r'_{kj,i}, 1 \leq k, j \leq n$. Then similar to **PRMT_Optimal** protocol, **R** constructs the conflict graphs $H_1, H_2, \ldots, H_n$ and combine them to a single conflict graph H using **Union Technique** , forms the list of five tuples X and broadcasts X to **S**.

Phase III (S to R)

- Similar to the **PRMT_Optimal** protocol, **S** correctly receives X and identifies all faulty wires w_i over which **R** must have received at least one faulty polynomial during **Phase I**. **S** adds all such wires L_{fault}. **S** neglects all $p_{ki}(x),\ 1 \leq k \leq n$ sent during **Phase I** if $w_i \in L_{fault}$.
- **S** is left with $(n - |L_{fault}|)$ wires after neglecting all the faulty wires in the previous step. **S** then forms a vector x of length $(n - |L_{fault}|) * n$ which is the concatenation of the constant terms of all the polynomials $p_{ki}(x), 1 \leq k \leq n$ such that $w_i \notin L_{fault}$.
- **S** computes a pad y of length $n(t+1)$ by executing $\mathbf{EXTRAND}_{n(n-|L_{fault}|),n(t+1)}(x)$ algorithm of section 2.2.1. **S** computes $c = [c_1 c_2 \ldots c_{n(t+1)}] = y \oplus m$, where $c_i = y_i \oplus m_i$ and broadcasts L_{fault} and c to **R**.

Message Recovery by R.

• **R** reliably receives the list L_{fault} and identifies all the wires w_i over which it has received at least one faulty polynomial $p'_{ki}(x), 1 \leq k \leq n$ during **Phase I** (see Theorem 6). **R** neglects all $p'_{ki}(x), 1 \leq k \leq n$ if $w_i \in L_{fault}$. **R** generates the pad y of length $n(t+1)$ following the same procedure as done by **S** and finally recovers the message m by computing $m = c \oplus y$.

Theorem 6 *In the protocol* **PSMT_Optimal**, **R** *will always be able to correctly receive the message.*

Proof: **PSMT_Optimal** is similar to **PRMT_Optimal**, except that in **PSMT_Optimal**, the coefficients of the polynomials $p_{ki}(x), 1 \leq k, i \leq n$ are arbitrary field elements, used to establish a one time pad of length $n(t+1)$ between **S** and **R**. From Theorem 4, on receiving X, **S** identifies all w_i's over which **R** has received at least one faulty $p'_{ki}(x), 1 \leq k \leq n$ during **Phase I**. **S** adds all

such wires to L_{fault} and then neglects all the n polynomials which are sent over such faulty wires. **S** then forms the vector x which is the concatenation of constant terms of the polynomials $p_{ki}(x)$, $1 \leq k \leq n$ if $w_i \notin L_{fault}$ and generates a one time pad y of length $n(t+1)$. **S** then blinds m with y to generate c and broadcasts c and L_{fault} to **R**. On receiving L_{fault}, **R** identifies the faulty wires w_i and neglects all the n polynomials received over w_i where $w_i \in L_{fault}$, construct the vector x, regenerates the one time pad y and thus recovers the message by XOR-ing c with y. □.

Theorem 7 *In the protocol* **PSMT_Optimal**, *any adversary $\mathcal{A}$ who can control different set of t wires during different phases of the protocol will get no information about the message m.*

Proof: Without loss of generality, assume during **Phase I**, $\mathcal{A}$ controls $w_1, w_2, \ldots, w_t$. Thus $\mathcal{A}$ knows the constant terms of the polynomials $p_{ki}(x), 1 \leq k \leq n, 1 \leq i \leq t$. Moreover, for the remaining polynomials $p_{kj}, t+1 \leq j \leq n$, $\mathcal{A}$ receives t points over $w_1, w_2, \ldots w_t$. Since the degree of each $p_{kj}, t+1 \leq j \leq n$ is t, $\mathcal{A}$ lacks one point for each of these polynomials implying information theoretic security for the constant terms of these polynomials. From Theorem 6, during **Phase III**, **S** will be able to identify all the faulty wires over which **R** had received at least one faulty polynomial during **Phase I**. **S** adds all such wires to L_{fault} and neglects them. **S** is left with $n - |L_{fault}|$ wires, out of which at most $t - |L_{fault}|$ wires were passively listened by the adversary. So **S** forms the vector x which is the list of constant terms of all the polynomials which were delivered correctly to **R** during **Phase I**. Since, there are $t+1$ honest (not controlled by adversary) wires, **S** generates a one time pad of length $n(t+1)$ from x by executing **EXTRAND**. The proof now follows from the correctness of the **EXTRAND** algorithm. Note that during **Phase II**, the list X broadcast by **R**, reveals no new information to $\mathcal{A}$. Suppose $\{w_i, w_j, k, p'_{ki}(\alpha_j), r'_{ki,j}\} \in X$. It implies that either w_i or w_j or both had been corrupted by $\mathcal{A}$ during **Phase I**. If w_i was corrupted by $\mathcal{A}$ then $\mathcal{A}$ knows $p_{ki}(x)$ and hence the value $r'_{ki,j} = r_{ki,j}$. On the other hand, if $\mathcal{A}$ had corrupted w_j, then $\mathcal{A}$ already knows $r_{ki,j} = p_{ki}(\alpha_j)$ which it had changed to $r'_{ki,j}$. Hence, X reveals no new information to the adversary whatsoever. □

Theorem 8 *The communication complexity of the protocol* **PSMT_Optimal** *is $O(n^3)$.*

Proof: The communication complexity of **PSMT_Optimal** is exactly same as **PRMT_Optimal**. The third phase of **PSMT_Optimal** has an additional cost of broadcasting the blinded message c of size $n(t+1)$ which requires sending $O(n^3)$ field elements. So overall communication complexity is $O(n^3)$.

Remark 3 *In [17], it is reported that any three phase PSMT protocol which securely sends $n(t+1) = O(n^2)$ field elements, in the presence of mobile adversary, need to communicate $\Omega(n^3)$ field elements. Since, the communication complexity of* **PSMT_Optimal** *is $O(n^3)$, it is asymptotically optimal. As* **PSMT_Optimal** *sends $O(n^2 log_2|\mathbb{F}|)$ bits by communicating $O(n^3 log_2|\mathbb{F}|)$ bits, it is bit optimal also.*

3 PSMT Tolerating Mobile Adversary in Directed Networks

In [3], the authors have studied PSMT in directed networks in the presence of a static adversary, where the network is abstracted in the form of directed wires, directed either from **S** to **R** or vice-versa. Modeling the underlying network in the form of a directed graph is important in many

practical scenarios. For instance, a base-station may communicate to even a far-off hand-held device but the other way round is not possible. Hence the digraph model is practically well-motivated. We now characterize multiphase PSMT between $\mathbf{S}$ and $\mathbf{R}$ in a directed network against a t-active mobile adversary.

Theorem 9 *Let $G = (V, E)$ be a directed network, where $\mathbf{S}$ and $\mathbf{R}$ are two nodes. Then a r-phase ($r \geq 2$) PSMT protocol between $\mathbf{S}$ and $\mathbf{R}$ against a t-active mobile adversary is possible iff there exist $2t + 1$ directed wires from $\mathbf{S}$ to $\mathbf{R}$ and $2t + 1$ directed wires from $\mathbf{R}$ to $\mathbf{S}$.*

Proof: **Sufficiency:** Suppose there exist $2t+1$ directed wires $f_1, f_2, \ldots, f_{2t+1}$ from $\mathbf{S}$ to $\mathbf{R}$ and $2t+1$ directed wires $b_1, b_2, \ldots, b_{2t+1}$ from $\mathbf{R}$ to $\mathbf{S}$. Then irrespective of whether the wires $f_i, 1 \leq i \leq 2t+1$ and $b_j, 1 \leq j \leq 2t+1$ share any vertex or not, the protocol **PSMT_Optimal** can be executed in G. It is easy to see that Theorem 6 and Theorem 7 will hold here.

Necessity: Since any PSMT protocol should communicate the message reliably, $\mathbf{S}$ and $\mathbf{R}$ should be $2t + 1$ connected in forward direction which is necessary for PRMT [5]. We prove that $2t + 1$ wires are necessary from $\mathbf{R}$ to $\mathbf{S}$ also by contradiction. Assume there are $2t$ wires from $\mathbf{R}$ to $\mathbf{S}$. Since the wires corrupted by the adversary from $\mathbf{S}$ to $\mathbf{R}$ can be totally independent of the wires corrupted from $\mathbf{R}$ to $\mathbf{S}$ (the adversary is mobile), the adversary can fail reliable communication from $\mathbf{R}$ to $\mathbf{S}$ by corrupting t wires from $\mathbf{R}$ to $\mathbf{S}$ [5]. Hence, any communication from $\mathbf{R}$ to $\mathbf{S}$ is useless for $\mathbf{S}$. This reduces any multiphase protocol to a single phase protocol where $\mathbf{S}$ has to securely send a message over $(2t + 1)$ wires tolerating a t-active Byzantine adversary. This is again impossible from the results of [5], as $3t + 1$ wires is necessary for single phase secure communication.Therefore, there should be at least $(2t + 1)$ wires from $\mathbf{R}$ to $\mathbf{S}$. Hence the theorem holds. □

4 PRMT and PSMT Tolerating Mobile Adversary (in Terms of Rounds)

Till the previous section, we concentrated to design *bit-optimal phase-based* PRMT/PSMT protocols on an network abstracted in terms of wires. The merits in working in such a model are as follows: (i) It eases deriving the connectivity requirement for the possibility of PRMT/PSMT protocols and also lower bounds for the communication complexity for protocols. (ii) It simplifies the analysis of any protocol designed on such model. But this model has its own demerits which are brought to the fore by providing a motivating example in the next section.

4.1 Motivating Example

In many practical scenarios, modeling the network as wires, does not give correct estimation on the communication complexity of PRMT (PSMT) protocols [16]. To understand the statement, we provide a motivating example. Consider the network on $(2t + 8)$ vertices given in Figure 1. Suppose the network in Figure 1 is abstracted as a collection of $(2t + 2)$ wires, under the control of a t-active mobile adversary. From [17], there exist an *optimal* single phase PRMT protocol with communication complexity of $O(n\ell)$ to send ℓ field elements, where n is the number of wires from $\mathbf{S}$ to $\mathbf{R}$ (which in this case is $2t + 2$). Now suppose that the protocol execution take place in a sequence of rounds, where at the beginning of each round, each node send messages to their

neighbors. Thus, the messages sent by a player in round k reaches its neighbor at the beginning of round $k+1$. Then the so called single phase "optimal" protocol of [17] runs in *six* rounds (which is the length of the longest path), with a communication complexity of $O(n)$ times the message size. Now the question is whether there exists a 6-round PRMT protocol in the network of Figure 1 with a better communication complexity. The answer is yes! Consider the following protocol: **S** and **R** run the 3-phase **PRMT_Optimal** protocol using the wires $P_1, P_2, \ldots, P_{2t+1}$, neglecting the path of length six (The longest path takes 6 rounds! While all other paths delivers message in two rounds). Thus while the single phase protocol has a complexity of $O(n\ell)$, the 3-phase protocol has a communication complexity of $O(\ell)$. Thus in Figure 1, an $O(\ell)$ 6-round protocol is possible. However, the information regarding the *length* of each of the paths (wires) in the actual network is

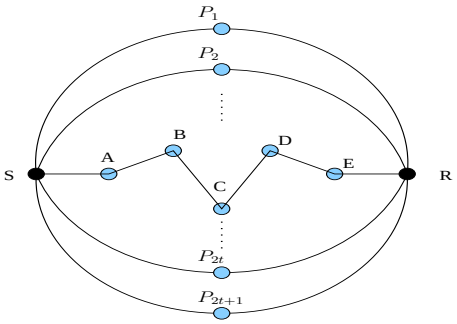

Figure 1: A $(2t+2)$-(**S,R**)-connected Network.

completely lost in the wired abstraction. Similarly, in the above network, if we consider all $2t+2$ wires from **S** to **R**, then we may declare that there does not exist any single phase and hence six round PSMT protocol from **S** to **R** because it is well known from [5], that any single phase PSMT protocol against a t-active adversary requires $3t+1$ wires from **S** to **R**. However, **PSMT_Optimal** when executed over the paths $P_1, P_2, \ldots, P_{2t+1}$, excluding the path of length six, terminates in six rounds (each phase of the **PSMT_Optimal** takes two rounds), which is also bit optimal. Thus in many practical scenarios, wired abstraction causes an over estimation in the round complexity and communication complexity of PRMT (PSMT) protocols in the original network. We thus redefine our network model and adversary settings.

4.2 Round Based Network and Adversary Settings

As shown in the motivating example, it is necessary to use more fine-grained and hence stronger model, namely the graph based one (in comparison to the collection of wires) for designing and analyzing *optimal* PRMT and PSMT protocols. So we consider a graph with internal details in the following way. Let H be an undirected graph under the control of a t-active mobile adversary. From [18], H should be $(2t+1)$-(**S, R**) connected which is necessary and sufficient for PRMT and PSMT. Let G be the subgraph of H induced by the $2t+1$ vertex disjoint paths. If there are more than $2t+1$ vertex disjoint paths in H, then G will also contain these paths. In the following sections, we work on G to derive tight lower bound on round complexity for reliable communication and design protocols on G.

The system is assumed to be synchronous, that is, the protocol is executed in a sequence of

rounds wherein in each round, a player can perform some local computation, send new messages to his out-neighbors, receive the messages sent in previous round by his in-neighbors (and if necessary perform some more local computation), in that order. The distrust in the network is modeled by a mobile Byzantine adversary. The behavior of *mobile adversary* is re-defined to allow it to corrupt any set of t nodes after every $\rho \geq 1$ rounds, where ρ is called the roaming speed of the adversary. We first consider the worst case that of $\rho = 1$, later on, we will consider any arbitrary value of ρ. More formally before the beginning of round k, the adversary can corrupt any subset $\mathcal{P}_{corrupt}$ consisting of t players. Then the adversary has access to the messages sent to the players in $\mathcal{P}_{corrupt}$ in round $k-1$ and can alter the behavior of the players in $\mathcal{P}_{corrupt}$ arbitrarily in the round k. However by corrupting a player P in a round k the adversary does not obtain information about the messages to and from the node P in all the previous rounds, i.e., the protocol can choose to delete some information from the (honest) node at the end of a round, to make sure that the information is not available to the adversary even if he corrupts the node at a later round. Before computing the minimum number of rounds for reliable communication, we explain the concept of transmission graph.

4.3 Transmission Graph

Graphs always have been used as a very powerful abstraction of the network by modeling the physical link between two nodes as an edge between the corresponding vertices of the graph. However it does not contain any temporal information. Especially in the case of mobile adversary, where the adversary can corrupt different set of nodes at different times, a graph representation of the network is inadequate. However since the protocol itself discretizes time in terms of rounds, it is sufficient to model the system at each round rather than each time instant. Hence, in [16], the author have introduced the concept of transmission graph $\mathcal{G}^d$ to study the execution of a protocol that has run d rounds. In the transmission graph $\mathcal{G}^d$, each node P is represented by a set of nodes $\{P_0, P_1, P_2 \dots P_d\}$. The node P_r corresponds to the node P at round r. For any two neighboring nodes P and Q and any $1 \leq r \leq d$, a message sent by P to Q in round $r-1$ is available to Q only at round r. Hence there is an edge in $\mathcal{G}^d$ connecting the node P_{r-1} to the node Q_r for all $1 \leq r \leq d$. Note that the transmission graph is a directed graph, because of the directed nature of time. So the edges between the nodes at consecutive time steps are always oriented towards increasing time. We now recall the definition of transmission graph from [16].

Definition 1 *Given a graph $G = (V, E)$ and a positive integer d, the transmission Graph $\mathcal{G}^d$ is a directed graph defined as follows*

- *Nodes of $\mathcal{G}^d$ belong to $V \times \{0 \dots d\}$ where the node $(P, r) \in V \times \{0 \dots d\}$ is denoted by P_r.*
- *The edge set of $\mathcal{G}^d$ is $E^d = E_1 \cup E_2$ where, $E_1 = \{(P_{a_{r-1}}, P_{b_r}) \ | (P_a, P_b) \in E$ and $1 \leq r \leq d\}$ and $E_2 = \{(P_{a_{r-1}}, P_{a_r}) | P_a \in V$ and $1 \leq r \leq d\}$.*

Let $\mathcal{P}^r$ denote the set of nodes corresponding to nodes at round r, $\mathcal{P}^r = \{P_{a_r} \ | P_a \in V\}$. Let $\mathcal{ADV}_{mobile}$ be a threshold mobile adversary acting on a network G that can corrupt any t nodes in a single round. Consider an execution Γ of a d-round protocol on G. Suppose $\mathcal{ADV}_{mobile}$ corrupts a set of nodes $Adv_r = \{P_1, P_2, \dots P_t\}$ in round r in G, then the same effect is obtained by corrupting the nodes $Adv^r = \{P_{1_r}, P_{2_r}, \dots P_{t_r}\}$ in $\mathcal{G}^d$. Hence the effect of $\mathcal{ADV}_{mobile}$ on execution Γ can be

simulated by a static general adversary who corrupts $\bigcup_{r=1}^{d} Adv^r$ on $\mathcal{G}^d$. More formally, we have the following lemma:

Lemma 2 *Mobile adversary $\mathcal{ADV}_{mobile}$ acting on the original network graph G for d rounds can be simulated by a static adversary given by the adversary structure $\mathcal{ADV}^d_{static} = \{Adv^1 \cup Adv^2 \cup Adv^3 \ldots \cup Adv^d | Adv^r \in \Pi_t(\mathcal{P}^r), 1 \leq r \leq d\}$ on $\mathcal{G}^d$, where $\Pi_t(\mathcal{P}^r)$ denotes the set of all subsets of cardinality t of the set $\mathcal{P}^r$ excluding $\mathbf{S}$ and $\mathbf{R}$.*

Example 1 *Consider the network shown in Figure 2: The network is 3-(**S**,**R**)-connected and hence from [18], at most one mobile adversary ($t = 1$) can be tolerated by any PRMT (PSMT) protocol. Consider $\mathcal{G}^4$, where the adversary structure $\mathcal{ADV}^4_{static} = \{Adv^1 \cup Adv^2 \cup Adv^3 \cup Adv^4\}$ where each $Adv^r \in \Pi_1(\mathcal{P}^r), 1 \leq r \leq 4$ where $\Pi_1(\mathcal{P}^r)$ denotes the set of all subsets of cardinality 1 of the set $\mathcal{P}^r$. For example, $\{A_1, A_2, A_3, A_4\}, \{A_1, D_2, G_3, H_4\}, \{H_1, E_2, B_3, A_4\}$ are some of the elements of $\mathcal{ADV}^4_{static}$ in $\mathcal{G}^4$. Here $\{A_1, A_2, A_3, A_4\}$ denotes an adversarial strategy where in the original network, the adversary corrupts the same node A in all the four rounds. Similarly $\{H_1, E_2, B_3, A_4\}$ denotes an adversarial strategy where in the original network, the adversary corrupts the nodes H, E, B and A during first, second, third and fourth round respectively. In fact there are 11^4 possible elements of $\mathcal{ADV}^4_{static}$ in $\mathcal{G}^4$ since there are 11 nodes in G (excluding $\mathbf{S}$ and $\mathbf{R}$) and in each of the four rounds, adversary can choose any one of the 11 nodes to corrupt.*

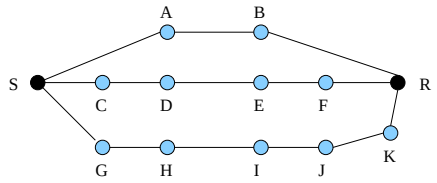

Figure 2: A 3-(**S**,**R**)-connected Network G.

In general let G be a graph with $2t + 1$ (or more) vertex disjoint paths between $\mathbf{S}$ and $\mathbf{R}$ and N be the total number of nodes in these paths. Then in $\mathcal{G}^d$, there will be $\binom{N}{t}^d$ possible elements in the adversary structure $\mathcal{ADV}^d_{static}$. In order to find the minimum number of rounds for reliable communication, we slightly modify the definition of transmission graph as follows:

Definition 2 *Given a graph G and an integer $d > 0$ the modified Transmission Graph G^d is the graph $\mathcal{G}^d$ along with two additional nodes $\mathbf{S}$, $\mathbf{R}$. $\mathbf{S}$ is connected to all $\mathbf{S}_r$,$0 \leq r \leq d$ and each $\mathbf{R}_r$, $0 \leq r \leq d$ is connected to $\mathbf{R}$. Further the edges between $(\mathbf{S}_{r-1}, \mathbf{S}_r)$ and $(\mathbf{R}_{r-1}, \mathbf{R}_r)$ for $1 \leq r \leq d$ are removed.*

Definition 3 *Two paths Γ_1 and Γ_2 between the nodes $\mathbf{S}$ and $\mathbf{R}$ in the modified transmission graph G^d are said to be securely disjoint if the only common nodes between the two paths are $\mathbf{S}_a$ and $\mathbf{R}_b$ for some value of a and b. That is, $\Gamma_1 \cap \Gamma_2 \subset \{\mathbf{S}_0, \mathbf{S}_1, \mathbf{S}_2 \ldots \mathbf{S}_d\} \cup \{\mathbf{R}_0, \mathbf{R}_1, \mathbf{R}_2 \ldots \mathbf{R}_d\}$*

Definition 4 *Given a path $\Gamma = \{\mathbf{S}, P_1, P_2 \ldots P_z, \mathbf{R}\}$ from $\mathbf{S}$ to $\mathbf{R}$ in the underlying graph G, the space-time path Γ^i in graph G^d is defined as $\Gamma^i = \{\mathbf{S}, \mathbf{S}_i, P_{1_{i+1}}, P_{2_{i+2}}, \ldots P_{z_{i+z}}, \mathbf{R}_{i+z+1}, \mathbf{R}\}$, $0 \leq i \leq d - z - 1$.*

Example 2 *Consider the path* $\Gamma = \{\mathbf{S}, A, B, \mathbf{R}\}$ *in Figure 2. Now in* G^5*, there are three space time paths corresponding to the path* Γ*, namely* $\Gamma^0 = \{\mathbf{S}, \mathbf{S}_0, A_1, B_2, \mathbf{R}_3, \mathbf{R}\}$*,* $\Gamma^1 = \{\mathbf{S}, \mathbf{S}_1, A_2, B_3, \mathbf{R}_4, \mathbf{R}\}$ *and* $\Gamma^2 = \{\mathbf{S}, \mathbf{S}_2, A_3, B_4, \mathbf{R}_5, \mathbf{R}\}$*. The space time path* Γ^0 *can be interpreted as* $\mathbf{S}$ *communicating to* A *in the* 0^{th} *round,* A *communicating to* B *in the first round,* B *communicating to* $\mathbf{R}$ *in the second round which is received by* $\mathbf{R}$ *in the third round. Similarly, the paths* Γ^1 *and* Γ^2 *can be interpreted. Note that in* G^5*, there are only three space time paths corresponding to the path* Γ *in* G*. This is so because if any protocol is executed for five rounds, then* $\mathbf{R}$ *will stop receiving anything from* B *after fifth round. In general, let* G *be a graph and* Γ *be a path between* $\mathbf{S}$ *and* $\mathbf{R}$ *containing* z *nodes (i.e., the path length is* $z+1$*). Then in the transmission graph* $G^d, d > z$*, there will be* $d - z$ *space time paths corresponding to the path* Γ*, namely* $\Gamma^i, 0 \leq i \leq d - z - 1$*.*

Lemma 3 ([18]) *For any path* Γ *of length* z *(containing* $z+1$ *nodes) from* $\mathbf{S}$ *to* $\mathbf{R}$ *in* G*, the paths* $\Gamma^i, 0 \leq i \leq d - z$ *are pairwise securely disjoint. Further, for any two vertex disjoint paths* Γ_1*,* Γ_2 *and for any* i, j *the paths* Γ_1^i *and* Γ_2^j *are securely disjoint.*

Example 3 *Consider the paths* $\Gamma_1 = \{\mathbf{S}, A, B, \mathbf{R}\}$ *and* $\Gamma_2 = \{\mathbf{S}, C, D, E, F, \mathbf{R}\}$ *in the network shown in Figure 2. Suppose we consider the transmission graph* G^6*, then there are following space time paths corresponding to* Γ_1 *in* G^6*:* $\Gamma_1^0 = \{\mathbf{S}, \mathbf{S}_0, A_1, B_2, \mathbf{R}_3, \mathbf{R}\}$*,* $\Gamma_1^1 = \{\mathbf{S}, \mathbf{S}_1, A_2, B_3, \mathbf{R}_4, \mathbf{R}\}$*,* $\Gamma_1^2 = \{\mathbf{S}, \mathbf{S}_2, A_3, B_4, \mathbf{R}_5, \mathbf{R}\}$ *and* $\Gamma_1^3 = \{\mathbf{S}, \mathbf{S}_3, A_4, B_5, \mathbf{R}_6, \mathbf{R}\}$*. Similarly, there are following space time paths corresponding to* Γ_2 *in* G^6*:* $\Gamma_2^0 = \{\mathbf{S}, \mathbf{S}_0, C_1, D_2, E_3, F_4, \mathbf{R}_5, \mathbf{R}\}$ *and* $\Gamma_2^1 = \{\mathbf{S}, \mathbf{S}_1, C_2, D_3, E_4, F_5, \mathbf{R}_6, \mathbf{R}\}$*. It is clear that all* $\Gamma_1^i, 0 \leq i \leq 3$ *are securely disjoint. Similarly, all* $\Gamma_2^i, 0 \leq i \leq 1$ *are securely disjoint. Also all the space time paths* $\Gamma_1^i, \Gamma_2^j, 0 \leq i \leq 3, 0 \leq j \leq 1$ *are securely disjoint.*

4.4 Computing Minimum Number of Rounds for PRMT with $\rho = 1$

In [18], the authors have computed the minimum number of rounds d for reliable communication from $\mathbf{S}$ to $\mathbf{R}$ which is $d > (2t+1)N$ (see Lemma 4.1 of [18]), where $\mathbf{S}$ and $\mathbf{R}$ are connected by $2t+1$ paths and N is the total number of nodes in the given network. However, we show that the bound in [18] is not tight. So, we derive tight bound on the minimum number of rounds, denoted by r_{min} required for reliable communication from $\mathbf{S}$ to $\mathbf{R}$. Consider a graph G where $\mathbf{S}$ and $\mathbf{R}$ are connected by $2t+1$ vertex disjoint paths $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_{2t+1}\}$. Without loss of generality, assume that the paths are arranged in ascending order of path length. Let N_i denotes the number of nodes in $\Gamma_i, 1 \leq i \leq 2t+1$. Then in G^d, as explained earlier, there will be $d - N_i$ space time paths corresponding to $\Gamma_i, 1 \leq i \leq 2t+1$ in G provided $d - N_i > 0$. If $d - N_i \leq 0$ then there will be no space time path corresponding to Γ_i in G^d. Assuming that each of the term $d - N_i$ is positive, the total number of the space time paths in G^d is $\sum_{i=1}^{i=2t+1}(d - N_i)$. From Lemma 3, all these paths are securely disjoint. Now if any reliable protocol is executed on the original graph G for d rounds, then the adversary can make corruption only up to $(d-1)$ rounds because in any reliable protocol, which is executed for d rounds, $\mathbf{R}$ will receive information from its neighboring nodes in round d, which they sent to $\mathbf{R}$ in round $d-1$ and terminates the protocol. So even if adversary corrupts some node in round d, it will not effect the protocol, because the protocol will terminate in the d^{th} round itself. Note that if at least one node in a space time path in G^d is corrupted, it implies that the entire space time path is corrupted because the corrupted data introduced by the corrupted node will be forwarded by other nodes of the path in subsequent rounds. In general, since the adversary can corrupt at most t nodes in each round of any reliable protocol, it can corrupt at most

$t(d-1)$ nodes in G^d which can be in worst case distributed on $t(d-1)$ secure disjoint paths and hence each element in $\mathcal{ADV}^d_{static}$ is of maximum cardinality $t(d-1)$. We now state the following theorem.

Theorem 10 *Let G be an undirected network where $\mathbf{S}$ and $\mathbf{R}$ are connected by $2t+1$ vertex disjoint paths $\Gamma_1, \Gamma_2, \ldots, \Gamma_{2t+1}$ with N_i nodes in $\Gamma_i, 1 \leq i \leq 2t+1$. Let $\mathcal{ADV}_{mobile}$ be a mobile adversary corrupting any set (probably different) of t nodes in each round. Then the minimum number of rounds required for reliable communication is r_{min} iff $r_{min} \geq N - 2t + 1$ where $N = \sum_{i=1}^{i=2t+1} N_i$.*

Proof: **Necessity**: Let r_{min} be the minimum number of rounds required for reliable communication in G. Then as explained above, any mobile adversary $\mathcal{ADV}_{mobile}$ can be simulated by a static adversary structure $\mathcal{ADV}^{r_{min}}_{static}$ where each element of it is of cardinality $t(r_{min}-1)$. Also in $G^{r_{min}}$, there will be $\sum_{i=1}^{i=2t+1}(r_{min} - N_i)$ securely disjoint paths between $\mathbf{S}$ and $\mathbf{R}$ out of which at most $t(r_{min}-1)$ can be under the control of the adversary. Now it is known from [10], that reliable communication between $\mathbf{S}$ and $\mathbf{R}$ in a network in the presence of a static adversary given by an adversary structure is possible iff removal of of any two adversarial sets from the adversary structure does not disconnect $\mathbf{S}$ and $\mathbf{R}$. It implies that reliable communication in G under the presence of $\mathcal{ADV}_{mobile}$ is possible in r_{min} rounds if $\sum_{i=1}^{i=2t+1}(r_{min} - N_i) \geq 2t(r_{min}-1) + 1$. Solving this we get $r_{min} \geq N - 2t + 1$ where $N = \sum_{i=1}^{i=2t+1} N_i$.

Sufficiency: Suppose $r_{min} \geq N - 2t + 1$ where $N = \sum_{i=1}^{i=2t+1} N_i$. Then in $G^{r_{min}}$ there are $2t(r_{min}-1)+1$ securely disjoint paths from $\mathbf{S}$ to $\mathbf{R}$, out of which at most $t(r_{min}-1)$ can be under the control of the adversary $\mathcal{ADV}^{r_{min}}_{static}$. Let us denote these paths by $w_1, w_2, \ldots, w_{2q+1}$, where $q = t(r_{min}-1)$. We now describe a reliable protocol **REL** on the graph $G^{r_{min}}$ and show how it can be executed on the real network G to reliably send m. **REL** can be emulated on G in the following way: if a node P_{1_b} and $P_{2_{b+1}}$ are consecutive nodes in $G^{r_{min}}$ along some path w_i, where w_i is the space time path corresponding to some physical path $\Gamma_j, 1 \leq j \leq 2t+1$, then P_1 on receiving m' (possibly changed m) along the path Γ_j at the beginning of round b forward it to the node P_2 at the end of round b which is received by P_2 in round $b+1$. The protocol has a communication complexity of $O((2t(r_{min}-1)|m|)$ and this is polynomial in N. The correctness of the protocol is obvious. □

Protocol REL: Round-Optimal Reliable Message transmission of message m.

- The sender $\mathbf{S}$ sends the message m along all the paths w_i, $1 \leq i \leq 2q+1$.
- All nodes P_{a_b} along a path w_i just forward the message to the next node along w_i.
- The receiver on receiving the values along all the paths takes the majority value as the message m.

Example 4 *For the network in Figure 2, $r_{min} = 10$. This is because in G^9, there are sixteen space time paths of which the adversary can corrupt at most eight paths. So exactly half of the paths can be under the control of the adversary. However, in G^{10}, there are nineteen space time paths out of which the adversary can corrupt at most nine paths. Hence, majority of the paths will be error free.*

4.4.1 Finding r_{min} in the Presence of more than $2t+1$ Paths for $\rho = 1$

In many practical scenarios there may be more than $2t+1$ vertex disjoint paths between $\mathbf{S}$ and $\mathbf{R}$. Even then we can find r_{min} by using the same argument as above. Suppose G is a network where there are $n > 2t+1$ paths $\Gamma_1, \Gamma_2, \ldots, \Gamma_{2t+1}, \ldots, \Gamma_n$ between $\mathbf{S}$ and $\mathbf{R}$, arranged in ascending order

of path length, such that there are N_i nodes in path Γ_i. We call the algorithm for computing r_{min} in this case as **Algorithm_Round_Complexity**.

Algorithm_Round_Complexity: Computing r_{min} where **S** and **R** are connected by $n > 2t+1$ vertex disjoint paths.

1. Set $r_{min} = N - 2t + 1$ where $N = \sum_{i=1}^{i=2t+1} N_i$.
2. For $i = 2t+2$ to n do:
   (a) If $N_i + 1 > r_{min}$ then output r_{min} and EXIT.
   (b) If $N_i + 1 \leq r_{min}$ then do the following:
      i. Compute $r = \lceil \frac{(N_1+N_2+...+N_i)-2t+1}{i-2t} \rceil$
      ii. If $r \leq r_{min}$ then set $r_{min} = r$ else GOTO step 3.
3. Output r_{min}.

Theorem 11 **Algorithm_Round_Complexity** *correctly computes* r_{min} *when* **S** *are* **R** *connected by more than* $2t+1$ *vertex disjoint paths.*

Proof: In **Algorithm_Round_Complexity**, r_{min} is first set to $N - 2t + 1$, which according to Theorem 10 is the minimum number of rounds for reliable communication in the presence of $2t+1$ paths $\Gamma_j, 1 \leq j \leq 2t+1$ between **S** and **R**. Note that the paths $\Gamma_j, 1 \leq j \leq n$ are arranged in ascending order of path length. Now there are following cases to be considered for Γ_{2t+2}:

• $r_{min} < N_{2t+2} + 1$: Note that $N_{2t+2} + 1$ is the path length of Γ_{2t+2}. So if in any reliable protocol, the path Γ_{2t+2} is involved then it will take at least $N_{2t+2} + 1$ rounds to send any information from **S** to **R** through the path Γ_{2t+2}. However, since $r_{min} < N_{2t+2} + 1$, including the path Γ_{2t+2} will increase r_{min}. Since the path lengths of remaining $\Gamma_j, 2t+3 \leq j \leq n$ is at least $N_{2t+2} + 1$, using the above argument, any round optimal protocol should only consider the first $2t+1$ paths and hence $r_{min} = N - 2t + 1$.

• $r_{min} \geq N_{2t+2} + 1$: In this case, including Γ_{2t+2} may reduce the value of r_{min}. Using the argument of Theorem 10, we first compute minimum number of rounds r required for reliable communication considering the first $2t+2$ paths. Now r is computed as $\sum_{i=1}^{i=2t+2}(r - N_i) \geq 2t(r-1) + 1$ which implies $r \geq \lceil \frac{(N_1+N_2+...+N_{2t+2})-2t+1}{2} \rceil$. If the minimum value of r is less than or equal to r_{min}, then considering Γ_{2t+2} reduces or does not change r_{min} and hence r_{min} is updated to r. Otherwise Γ_{2t+2} is neglected and r_{min} is not updated. However, if $r > r_{min}$, then including Γ_{2t+2} in any reliable protocol protocol will increase r_{min}. Hence Γ_{2t+2} is not considered. Since the path lengths of remaining $\Gamma_j, 2t+3 \leq j \leq n$ is at least $N_{2t+2} + 1$, including any of them will increase r_{min}. Hence all of them are neglected.

In the algorithm, the above two checking is done for all $\Gamma_i, 2t+2 \leq i \leq n$. Once r_{min} is computed, **S** will know which paths to consider for reliably sending any message to **R**. In the corresponding transmission graph $G^{r_{min}}$ there will be $2t(r_{min}-1)+1$ securely disjoint paths. So the protocol **REL** can be executed on $G^{r_{min}}$ which can be simulated on original network G as specified in Theorem 10. □

Example 5 *Intuitively,* r_{min} *can be computed considering the first* $2t+1$ *shortest paths between* **S** *and* **R**. *However, this is not always true!! For example in Figure 2, if we add one more vertex*

disjoint path of six nodes between **S** *and* **R**, *then from* **Algorithm_Round_Complexity**, $r_{min} = 8$ *(assuming* $t = 1$*) as opposed to* $r_{min} = 10$ *considering the first three shortest paths (according to Theorem 10).*

4.4.2 Bit Optimal PRMT and PSMT Protocols in Terms of Rounds

From Theorem 10, in $G^{r_{min}}$ there will be $2t(r_{min} - 1) + 1$ securely disjoint paths out of which at most $t(r_{min} - 1)$ can be corrupted. However each of these paths are temporal and hence can be used at most once. We now present the modified version of three phase protocol **PRMT_Optimal**, called **PRMT_Round**, tolerating a mobile adversary who can corrupt any t nodes in every round. **PRMT_Round** is executed for $3r_{min}$ rounds on G where G is the original network consisting $2t + 1$ vertex disjoint paths between **S** and **R**. The first phase of **PRMT_Optimal** is executed in the first r_{min} rounds from **S** to **R**, the second phase of **PRMT_Optimal** is executed in the next r_{min} rounds from **R** to **S** and finally the third phase in the last r_{min} rounds from **S** to **R**. This can be visualized as executing a $3r_{min}$ round protocol on $G^{3r_{min}}$, where first r_{min} rounds are executed from **S** to **R**, next r_{min} rounds from **R** to **S** and finally last r_{min} rounds from **S** to **R**. Let $q = t(r_{min} - 1)$ and $n = 2q + 1$. We refer to the nodes corresponding to the first r_{min} rounds from **S** to **R** as the first half denoted by $\Gamma_i^{(1)}, 1 \leq i \leq 2q + 1$, the nodes in the next r_{min} rounds from **R** to **S** as second half denoted by $\Gamma_i^{(2)}, 1 \leq i \leq 2q + 1$ and the nodes in the last r_{min} rounds from **S** to **R** as third half denoted by $\Gamma_i^{(3)}, 1 \leq i \leq 2q + 1$. From Theorem 10, $r_{min} = N - 2t + 1$. The protocol is same as **PRMT_Optimal** except that degree of each bi-variate polynomial is q. Moreover, **Phase** $i, 1 \leq i \leq 3$ is executed in r_{min} rounds on $\Gamma_j^{(i)}, 1 \leq j \leq 2q + 1$. **PRMT_Round** can be simulated on G following the explanation provided earlier for **REL** protocol. Note that Theorem 4 and Theorem 5 will hold for **PRMT_Round** with q in the place of t. The protocol reliably sends $n(q + 1)^2 = O(n^3)$ field elements by communicating $O(n^3)$ field elements in $3r_{min}$ rounds.

Similarly the three phase **PSMT_Optimal** can also be adapted to a $3r_{min}$ round PSMT protocol **PSMT_Round**, which securely sends $n(q+1) = O(n^2)$ field elements by communicating $O(n^3)$ field elements. All theorems w.r.t **PSMT_Optimal** will hold for **PSMT_Round** with $t = q$.

Remark 4 *In* **PRMT_Round**, *each phase of* **PRMT_Optimal** *is simulated in* r_{min} *rounds over* $n \geq 2q + 1$ *securely disjoint space time paths, of which the adversary can corrupt at most* q *paths. Treating space time paths as wires, from the results of [17], it is impossible to reliably send* ℓ *field elements by communicating* $O(\ell)$ *field elements in two phases. Thus the minimum number of phases required to do so is three. Since to correctly simulate a phase we require* r_{min} *rounds, our* $3r_{min}$ *round PRMT protocol* **PRMT_Optimal** *is both round optimal and bit optimal. Similarly from the results of [17], our* $3r_{min}$ *round PSMT protocol* **PSMT_Round** *is bit-optimal. However it is not round optimal because minimum number of phases required to tolerate a* t*-active mobile adversary with* $2t+1$ *wires is two. The two phase PSMT protocol of [15] against static adversary will also work for mobile adversary by adapting it into a* $2r_{min}$ *round PSMT protocol as done in [18]. However two phase protocol of [15] (and hence the* $2r_{min}$ *round protocol of [18]) is not bit-optimal.*

Protocol PRMT_Round : A $3r_{min}$ Round PRMT Protocol

Let the sequence of $n(q+1)^2$ field elements that **S** wishes to transmit be denoted by $m_{k,ij}$, $0 \leq i,j \leq q$ and $1 \leq k \leq n$.

First r_{min} rounds: (S to R) executed over space time paths $\Gamma_i^{(1)}, 1 \leq i \leq 2q+1$

- Using the $m_{k,ij}$ values, **S** defines n bivariate polynomials $q_k(x,y), 1 \leq k \leq n$ as follows: $q_k(x,y) = \sum_{i=0,j=0}^{i=q,j=q} m_{k,ij} x^i y^j$
- **S** then evaluates each $q_k(x,y), 1 \leq k \leq n$ at n publicly known distinct values $\alpha_1, \alpha_2, \ldots, \alpha_n$ to obtain total n^2 polynomials denoted as $p_{ki}(x), 1 \leq k \leq n, 1 \leq i \leq n$ over $\mathbb{F}$, each of degree q where $p_{ki}(x) = q_k(x, \alpha_i)$. Over space time paths $\Gamma_i^{(1)}, 1 \leq i \leq 2q+1$, **S** sends $p_{ki}(x), 1 \leq k \leq n$ and the values $p_{kj}(\alpha_i)$, denoted by $r_{kj,i}$, for $1 \leq k,j \leq n$.

Second r_{min} rounds: (R to S) executed over space time paths $\Gamma_i^{(2)}, 1 \leq i \leq 2q+1$

- Let **R** receives over space time path $\Gamma_i^{(1)}, 1 \leq i \leq n$ the polynomials $p'_{ki}(x)$ and the values $r'_{kj,i}, 1 \leq k,j \leq n$. **R** then considers the polynomials $p'_{11}(x), p'_{12}(x), \ldots, p'_{1n}(x)$ and the values $r'_{1j,i}, 1 \leq j,i \leq n$ and constructs the conflict graph H_1 as explained in section 2.2.2. Similarly, **R** considers the polynomials $p'_{21}(x), p'_{22}(x), \ldots, p'_{2n}(x)$ and the values $r'_{2j,i}, 1 \leq j,i \leq n$ and constructs the conflict graph H_2. In general, **R** considers the polynomials $p'_{k1}(x), p'_{k2}(x), \ldots, p'_{kn}(x)$ and the values $r'_{kj,i}, 1 \leq j,i \leq n$ and constructs the conflict graph $H_k, 1 \leq k \leq n$.
- **R** combines $H_k, 1 \leq k \leq n$ into a single directed conflict graph H using **Union Technique** and forms the corresponding list of five tuples X and reliably sends X to **S** by executing **REL** protocol over the space time paths $\Gamma_i^{(2)}, 1 \leq i \leq 2q+1$.

Last r_{min} rounds: S to R executed over space time paths $\Gamma_i^{(3)}, 1 \leq i \leq 2q+1$

• **S** reliably receives the list X and identifies all faulty space time paths $\Gamma_i^{(1)}$ over which **R** has received at least one faulty polynomial $p'_{ki}(x), 1 \leq k \leq n$ during first r_{min} rounds. **S** adds all such paths to a list L_{fault}. Note that $|L_{fault}| \leq q$. **S** then reliably sends L_{fault} to **R** by executing **REL** protocol over the space time paths $\Gamma_i^{(3)}, 1 \leq i \leq 2q+1$.

Message Recovery by R.

• **R** reliably receives L_{fault} and identifies all space time path $\Gamma_i^{(1)}$ over which it has received at least one faulty polynomial $p'_{ki}(x), 1 \leq k \leq n$ during first r_{min} rounds (proof is similar to Theorem 4). **R** neglects all the polynomials $p'_{ki}(x), 1 \leq k \leq n$ for each $\Gamma_i^{(1)} \in L_{fault}$,. Using the remaining (at least) $q+1$ p'_{ki}'s, $1 \leq k \leq n$, **R** correctly recovers the bivariate polynomials $q_k(x,y)$'s, $1 \leq k \leq n$ and hence the message.

4.5 Computing r_{min} for Arbitrary Roaming Speed

We now consider a mobile adversary with roaming speed $\rho > 1$ and compute r^{ρ}_{min} which is the minimum number of rounds required for reliable communication from **S** to **R**, against a t-active mobile adversary, corrupting t nodes after every ρ rounds. Note that a mobile adversary with roaming speed one is the strongest adversary. Intuitively, reducing roaming speed of adversary will reduce the minimum number of rounds required for PRMT between **S** and **R**. We support our intuition by computing r^{ρ}_{min} for an arbitrary $\rho(>1)$.

Assume **S** and **R** are connected by $n = 2t+1$ vertex disjoint paths $\Gamma_i, 1 \leq i \leq 2t+1$ which are in ascending order of path length and Γ_i has N_i nodes. Without loss of generality, we assume that the adversary starts corruption from the first round. Thus, if $\rho = 2$ and if a protocol is executed for six rounds, then adversary will corrupt t nodes in round one, three and five. Note that the t nodes which are corrupted in round one, three and five will also remain corrupted in the second, fourth and sixth round respectively. In general, any mobile adversary $\mathcal{ADV}^{\rho}_{mobile}$ who corrupts any t nodes in the network in every ρ rounds in any r round protocol can be simulated by a static

Protocol PSMT_Round: A $3r_{min}$ Round PSMT Protocol

Let the sequence of $n(q+1)$ field elements that **S** wishes to transmit securely be denoted by $m_i, 1 \leq i \leq n(q+1)$.

First r_{min} rounds: S to R executed over space time paths $\Gamma_i^{(1)}, 1 \leq i \leq 2q+1$

• **S** selects n^2 polynomials $p_{ij}(x), 1 \leq i,j \leq n$ over $\mathbb{F}$ each of degree q where the coefficients of each $p_{ij}(x)$ are independent of each other and the message. **S** then sends through each of the securely disjoint paths $\Gamma_i^{(1)}, 1 \leq i \leq 2q+1$ the polynomial $p_{ki}(x), 1 \leq k \leq n$ and the values $p_{kj}(\alpha_i)$, denoted by $r_{kj,i}$, for $1 \leq k,j \leq n$.

Second r_{min} rounds: R to S executed over space time paths $\Gamma_i^{(2)}, 1 \leq i \leq 2q+1$

Let **R** receives over each path $\Gamma_i^{(1)}, 1 \leq i \leq n$ the polynomials $p'_{ki}(x)$ and the values $r'_{kj,i}, 1 \leq k,j \leq n$. **R** then constructs the conflict graphs $H_1, H_2, \ldots, H_n$. From the n conflict graphs, **R** constructs a list of five tuples X in the same way as done in **PSMT_Optimal**. **R** then reliably sends the list X by executing the protocol **REL** using the paths $\Gamma_i^{(2)}, 1 \leq i \leq 2q+1$.

Last r_{min} rounds: S to R executed over space time paths $\Gamma_i^{(3)}, 1 \leq i \leq 2q+1$

• **S** correctly receives the list X and identifies all the faulty paths $\Gamma_i^{(1)}$ over which **R** must had received at least one faulty polynomial $p'_{ki}(x), 1 \leq k \leq n$ during first r_{min} rounds. **S** adds all such paths to a list L_{fault}. Note than $|L_{fault}| \leq q$.

• **S** then forms a vector x of length $(n - |L_{fault}|) * n$ which is the concatenation of the constant terms of all the polynomials $p_{ki}(x), 1 \leq k \leq n$ such that $\Gamma_i^{(1)} \notin L_{fault}$. **S** then computes a pad y of length $n(q+1)$ by executing $\mathbf{EXTRAND}_{n(n-|L_{fault}|),n(q+1)}(x)$ algorithm of section 2.2.1. **S** then computes $c = [c_1 c_2 \ldots c_{n(q+1)}] = y \oplus m$, where $c_i = y_i \oplus m_i$. **S** finally reliably sends the list L_{fault} and c to **R** over the paths $\Gamma_i^{(3)}, 1 \leq i \leq 2q+1$ by executing the **REL** protocol.

Message Recovery by R

• **R** reliably receives L_{fault} and identifies all the paths $\Gamma_i^{(1)}$ over which it had received at least one faulty polynomial $p'_{ki}(x), 1 \leq k \leq n$ during first r_{min} rounds. Corresponding to each path $\Gamma_i^{(1)} \in L_{fault}$, **R** neglects all the polynomials $p'_{ki}(x), 1 \leq k \leq n$. **R** generates the pad y of length $n(q+1)$ following the same procedure as done by **S** and finally recovers the message m by computing $m = c \oplus y$.

adversary structure $\mathcal{ADV}^{\rho}_{static}$ with size $\binom{N}{t}^{\lceil \frac{r}{\rho} \rceil}$ where N is total number of nodes in $2t+1$ paths since in r rounds, adversary will change the set of corrupted nodes after every $\lceil \frac{r}{\rho} \rceil$ rounds.
We now show how the roaming speed of the adversary changes its control over space time paths. In G^r, each $\Gamma_i, 1 \leq i \leq 2t+1$ will have $r - N_i$ securely disjoint space time paths.

Example 6 *Consider two space time paths* $\Gamma_1^0 = \{\mathbf{S}, \mathbf{S}_0, A_1, B_2, C_3, \mathbf{R}_4, \mathbf{R}\}$ *and* $\Gamma_1^1 = \{\mathbf{S}, \mathbf{S}_1, A_2, B_3, C_4, \mathbf{R}_5, \mathbf{R}\}$ *in* G^5 *corresponding to some path* $\Gamma_1 = \{\mathbf{S}, A, B, C, \mathbf{R}\}$ *in a network* G. *If* $\rho = 1$ *and if the adversary corrupts node* A *during first round, then* Γ_1^0 *is corrupted. However, it does not imply that* Γ_1^1 *is also corrupted until and unless the adversary corrupts node* A *in the second round also. However, if* $\rho = 2$ *and if the adversary corrupts node* A *during the first round, then both* Γ_1^0 *and* Γ_1^1 *will be corrupted because node* A *will remain corrupted during the second round also. Thus for* $\rho = 1$, *the two paths are independent of each other but for* $\rho = 2$, *the two paths can be treated as one set, which will be corrupted if adversary corrupts the first node of the path during the first round.*

In general, if any reliable protocol is executed for r rounds, then in G^r, each $\Gamma_i, 1 \leq i \leq 2t+1$ will have $\lceil \frac{r-N_i}{\rho} \rceil$ independent securely disjoint set of space time paths. Notice that if $\rho = 1$, then

each space time path is itself an independent set and hence we get $r - N_i$ independent sets for each Γ_i. Since the adversary can corrupt up to $r-1$ rounds, in G^r, at most $\lceil \frac{r-1}{\rho} \rceil * t$ independent sets can be corrupted because out of $r-1$ rounds, the adversary will change the corrupted set of nodes $\lceil \frac{r-1}{\rho} \rceil$ times.

Theorem 12 *Let G be a $(2t+1)$-(**S, R**) connected undirected network under the influence of a t-active mobile adversary with roaming speed of $\rho > 1$. Then the minimum number of rounds r^{ρ}_{min} required for reliable communication is given by $r^{\rho}_{min} = \ min\ \{r, r^{\rho-1}_{min}\}$ where r is the minimum value satisfying $\sum_{i=1}^{i=2t+1} \lceil \frac{r-N_i}{\rho} \rceil \geq 2\lceil \frac{r-1}{\rho} \rceil * t + 1$.*

Proof: Necessity: In G^r, there will be $\sum_{i=1}^{i=2t+1} \lceil \frac{r-N_i}{\rho} \rceil$ independent set of securely disjoint paths out of which at most $\lceil \frac{r-1}{\rho} \rceil t$ independent sets could be corrupted. Considering each independent set as wires, from [5], r will be r^{ρ}_{min} if $\sum_{i=1}^{i=2t+1} \lceil \frac{r-N_i}{\rho} \rceil \geq 2\lceil \frac{r-1}{\rho} \rceil * t + 1$. If the minimum value of r satisfying this inequality is greater than $r^{\rho-1}_{min}$, then $r^{\rho}_{min} = r^{\rho-1}_{min}$ because a mobile adversary with roaming speed ρ is always weaker in capability than one with roaming speed $\rho - 1$. Hence any round optimal PRMT protocol tolerating a mobile adversary with roaming speed $\rho - 1$ can always withstand the same with lesser roaming speed.

Sufficiency: We design protocol $\mathbf{REL}^{\rho}$ which reliably sends a message from **S** to **R** in r^{ρ}_{min} rounds. If $r^{\rho}_{min} = r^{\rho-1}_{min}$, then $\mathbf{REL}^{\rho}$ is replication of $\mathbf{REL}^{\rho-1}$. Otherwise, $\mathbf{REL}^{\rho}$ is defined as follows:

Protocol $\mathbf{REL}^{\rho}$: Round-Optimal Reliable Message transmission of m.

- **S** sends m along the first space time path of each $\sum_{i=1}^{i=2t+1} \lceil \frac{r^{\rho}_{min}-N_i}{\rho} \rceil$ securely disjoint independent set of space time paths.
- **R** only considers the values received along the first space time path of each of the $\sum_{i=1}^{i=2t+1} \lceil \frac{r^{\rho}_{min}-N_i}{\rho} \rceil$ securely disjoint independent set of space time paths and outputs the majority as m.

The correctness of the protocol follows from the fact that **R** will receive $\sum_{i=1}^{i=2t+1} \lceil \frac{r^{\rho}_{min}-N_i}{\rho} \rceil$ different copies of the message m out of which at most $\lceil \frac{r^{\rho}_{min}-1}{\rho} \rceil * t$ can be corrupted. However since $\sum_{i=1}^{i=2t+1} \lceil \frac{r^{\rho}_{min}-N_i}{\rho} \rceil \geq 2\lceil \frac{r^{\rho}_{min}-1}{\rho} \rceil * t + 1$, **R** will always receive the correct message m along the majority of the paths. □

Example 7 *Consider the network G in Figure 2. If $\rho = 1$, then from Theorem 10, $r^1_{min} = 10$. However, if $\rho = 2$, then from Theorem 12, $r^2_{min} = 9$. For the network G in Figure 2, in the transmission graph G^8, there will be the following space time paths between **S** and **R**:*

$\Gamma_1^0 = \{\mathbf{S}, \mathbf{S}_0, A_1, B_2, \mathbf{R}_3, \mathbf{R}\}$	$\Gamma_2^0 = \{\mathbf{S}, \mathbf{S}_0, C_1, D_2, E_3, F_4, \mathbf{R}_5, \mathbf{R}\}$	$\Gamma_3^0 = \{\mathbf{S}, \mathbf{S}_0, G_1, H_2, I_3, J_4, K_5, \mathbf{R}_6, \mathbf{R}\}$
$\Gamma_1^1 = \{\mathbf{S}, \mathbf{S}_1, A_2, B_3, \mathbf{R}_4, \mathbf{R}\}$	$\Gamma_2^1 = \{\mathbf{S}, \mathbf{S}_1, C_2, D_3, E_4, F_5, \mathbf{R}_6, \mathbf{R}\}$	$\Gamma_3^1 = \{\mathbf{S}, \mathbf{S}_1, G_2, H_3, I_4, J_5, K_6, \mathbf{R}_7, \mathbf{R}\}$
$\Gamma_1^2 = \{\mathbf{S}, \mathbf{S}_2, A_3, B_4, \mathbf{R}_5, \mathbf{R}\}$	$\Gamma_2^2 = \{\mathbf{S}, \mathbf{S}_2, C_3, D_4, E_5, F_6, \mathbf{R}_7, \mathbf{R}\}$	$\Gamma_3^2 = \{\mathbf{S}, \mathbf{S}_2, G_3, H_4, I_5, J_6, K_7, \mathbf{R}_8, \mathbf{R}\}$
$\Gamma_1^3 = \{\mathbf{S}, \mathbf{S}_3, A_4, B_5, \mathbf{R}_6, \mathbf{R}\}$	$\Gamma_2^3 = \{\mathbf{S}, \mathbf{S}_3, C_4, D_5, E_6, F_7, \mathbf{R}_8, \mathbf{R}\}$	
$\Gamma_1^4 = \{\mathbf{S}, \mathbf{S}_4, A_5, B_6, \mathbf{R}_7, \mathbf{R}\}$		
$\Gamma_1^5 = \{\mathbf{S}, \mathbf{S}_5, A_6, B_7, \mathbf{R}_8, \mathbf{R}\}$		

If $\rho = 2$, then there will be total seven independent set of securely disjoint paths (three corresponding to Γ_1, two corresponding to Γ_2 and two corresponding to Γ_3). Note that the last set of securely disjoint path corresponding to Γ_3 will have only one path unlike the other sets, each of which will have two paths. Now out of the eight rounds, adversary can do corruption in round one, three, five and seven. Hence, there can be at most four sets of securely disjoint paths out of the seven sets which can be under the control of the adversary. Since majority of the sets will be under the control of the adversary, no reliable protocol is possible in eight rounds. More formally, there exists two elements in the the static adversary structure $\mathcal{ADV}^8_{static}$ corresponding to the transmission graph G^8, such that removal of all the securely disjoint paths passing through these nodes in G^8 disconnects $\mathbf{S}$ *and* $\mathbf{R}$*. For example, consider the sets $\{A_1, A_2, A_3, A_4, A_5, A_6, K_7, K_8\}$ and $\{C_1, C_2, C_3, C_4, K_5, K_6, K_7, K_8\}$ belonging to the adversary structure $\mathcal{ADV}^8_{static}$. The set $\{A_1, A_2, A_3, A_4, A_5, A_6, K_7, K_8\}$ denotes an adversary who corrupts nodes A in the first round (and hence in the second round also because $\rho = 2$), node A in the third round (and hence in the fourth round also), node A in the fifth round (and hence in the sixth round also) and finally node K in the seventh round. Similarly, the other adversary element can be interpreted. Now it is clear to see that all the space time paths in G^8 passes through one of the nodes in $\{A_1, A_2, A_3, A_4, A_5, A_6, K_7, K_8\} \cup \{C_1, C_2, C_3, C_4, K_5, K_6, K_7, K_8\}$. Hence removal of these nodes will disconnect* $\mathbf{S}$ *and* $\mathbf{R}$ *and hence no reliable protocol will exist in G^8 and hence $r^2_{min} \neq 8$. However, if we consider the transmission graph G^9, then there will be nine independent set of securely disjoint paths between* $\mathbf{S}$ *and* $\mathbf{R}$ *(four corresponding to Γ_1, three corresponding to Γ_2 and two corresponding to Γ_3), out of which at most four sets can be under the control of the adversary. Hence majority of the sets will not be under the control of the adversary and hence reliable protocol is possible between* $\mathbf{S}$ *and* $\mathbf{R}$ *in G^9. Since the protocol can be simulated in the original network G in nine rounds, $r^2_{min} = 9$. Note that for G, $r^1_{min} = 10$. Hence $r^2_{min} < r^1_{min}$.*

Once we know how to compute r^{ρ}_{min}, **Algorithm_Round_Complexity** and protocols **PRMT_Round** and **PSMT_Round** can be adapted to tolerate a mobile adversary with arbitrary roaming speed.

4.6 Computing Minimum Number of Rounds for Static Adversary

Here we compute r_{min} for reliable communication against a t-active static adversary. If a node is corrupted by the static adversary in some round, then it remains corrupted for the remaining rounds of the protocol. Hence, the total number of nodes that will be corrupted throughout the protocol during is t.

Theorem 13 *Let G be a $(2t+1)$-(*$\mathbf{S}$, $\mathbf{R}$*) connected undirected network under the influence of a t-active static adversary. Let $\Gamma_1, \Gamma_2, \ldots, \Gamma_{2t+1}$ be the $2t+1$ vertex disjoint paths with N_i nodes in $\Gamma_i, 1 \leq i \leq 2t+1$, arranged in ascending order of path length. Then $r_{min} = N_{2t+1} + 1$, the length of the longest path Γ_{2t+1}.*

Proof: *Necessity*: A node once corrupted by static adversary remains so for the remaining rounds of the protocol. Hence all the space time paths passing through the node remain corrupted. Thus, if the adversary corrupts the first node of Γ_i during the first round of a r round PRMT protocol, then all the $r - N_i + 1$ space time paths $\Gamma^j_i, 0 \leq j \leq r - N_i$ will be corrupted (this is the worst adversary strategy). So all these paths can be considered as a single set controlled by the adversary. Likewise, all the individual space time paths corresponding to each Γ_i can be considered as a single set. Hence r_{min} is the minimum value of r such that after r rounds, there exists $2t+1$ such independent sets (corresponding to each of the $2t+1$ physical paths in G). It is easy to verify

that r_{min} is $N_{2t+1}+1$ which is the length of the longest path Γ_{2t+1} in G. The reason is that the independent set corresponding to Γ_{2t+1} will be generated only in $G^{N_{2t+1}+1}$; i.e., after $N_{2t+1}+1$ rounds. Before that in $G^r, r = N_{2t+1}$, only the independent sets corresponding to $\Gamma_i, 1 \leq i \leq 2t$ will be generated. Hence there will be only $2t$ such independent sets in $G^{N_{2t+1}}$, out of which at most t can be corrupted by the adversary. Hence no reliable protocol will be possible in $G^{N_{2t+1}}$ and hence r_{min} will be at least $N_{2t+1}+1$.

Sufficiency: Consider the following protocol in $G^{N_{2t+1}+1}$: **S** sends the message along the first space time path corresponding to each of the $2t+1$ independent sets. On receiving, **R** will output majority as the message. The correctness of the protocol follows from the fact that in $G^{N_{2t+1}+1}$, there will be $2t+1$ independent sets of paths, of which at most t could be corrupted. □

5 Conclusion and Open Problems

On the first look, a mobile adversary appears to be much more powerful and demanding than a static adversary with the same threshold. However the equivalence in terms of tolerability for these two kind of adversaries has been shown in [18]. In this paper we have shown the equivalence in terms of designing optimal PRMT and PSMT protocols. Our contributions are summarized as follows: We have designed constant phase PRMT and PSMT bit optimal protocols tolerating mobile adversary. Our second major contribution comes in terms of providing a generic method to compute the minimum number of rounds for PRMT tolerating a mobile adversary with different roaming speeds. Though we have presented efficient protocols for reliable and secure communication for every tolerable adversary, we are able to show that the round optimal protocols are efficient only if the network is given as a collection of disjoint paths. It is an interesting open problem to design (or prove the non-existence of) an efficient round optimal protocol for secure communication for all possible networks. Another challenging problem is to design a two phase (and hence $2r_{min}$ round) PSMT protocol which securely sends ℓ field elements by communicating $O\left(\frac{n\ell}{n-2t}\right)$ field elements against t-active mobile adversary. Recently, in [11], Kurosawa et.al have designed a two phase PSMT protocol achieving this bound against a t-active static adversary. However, their protocol cannot be adapted against mobile adversary.

References

[1] Michael Backes, Christian Cachin, and Reto Strobl. Proactive secure message transmission in asynchronous networks. In *PODC '03: Proceedings of the twenty-second annual symposium on Principles of distributed computing*, pages 223–232. ACM Press, 2003.

[2] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. of 20th ACM STOC*, pages 1–10, 1988.

[3] Y. Desmedt and Y. Wang. Perfectly secure message transmission revisited. Cryptology ePrint Archive, Report 2002/128, 2002. url - http://eprint.iacr.org.

[4] Y. Desmedt and Y. Wang. Perfectly secure message transmission revisited. In *Proc. of Advances in Cryptology: Eurocrypt 2002*, LNCS 2332, pages 502–517. Springer-Verlag, 2003.

[5] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *JACM*, 40(1):17–47, 1993.

[6] Yair Frankel, Peter Gemmell, Philip D. MacKenzie, and Moti Yung. Proactive RSA. In *Proceedings of Advances in Cryptology - CRYPTO 97*, volume 1294 of *Lecture Notes in Computer Science (LNCS)*, pages 440–452. Springer-Verlag, 1997.

[7] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proc. of 19th ACM STOC*, pages 218–229, 1987.

[8] A. Herzberg, M. Jakobson, S. Jarecki, H. Krawczyk, and M. Yung. Proactive Public Key and Signature Systems. In *Proceedings of 4th Conference on Computer and Communications Security*, pages 100–110, Zurich, Switzerland, April 1997. ACM Press.

[9] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive Secret Sharing, or: How to Cope with Perpetual Leakage. In D. Coppersmith, editor, *Proceedings of Advances in Cryptology - CRYPTO 95*, volume 963 of *Lecture Notes in Computer Science (LNCS)*, pages 339–352. Springer-Verlag, 1995.

[10] M. V. N. A. Kumar, P. R. Goundan, K. Srinathan, and C. Pandu Rangan. On perfectly secure communication over arbitrary networks. In *Proc. of 21st PODC*, pages 193–202. ACM Press, 2002.

[11] K. Kurosawa and K. Suzuki. Truly efficient 2-round perfectly secure message transmission scheme. To appear in Proc. of EUROCRYPT 2008.

[12] R. Ostrovsky and M. Yung. How to withstand mobile virus attacks. In *Proc. of 10th PODC*, pages 51–61. ACM Press, 1991.

[13] A. Patra, A. Choudhary, K. Srinathan, and C. Pandu Rangan. Constant phase bit optimal protocols for perfectly reliable and secure message transmission. In *Proc. of INDOCRYPT 2006*, volume 4329 of *LNCS*, pages 221–235. Springer Verlag, 2006.

[14] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proc. of 21st ACM STOC*, pages 73–85, 1989.

[15] H. Sayeed and H. Abu-Amara. Efficient perfectly secure message transmission in synchronous networks. *Information and Computation*, 126(1):53–61, 1996.

[16] K. Srinathan. Secure distributed communication. PhD Thesis, IIT Madras, 2006.

[17] K. Srinathan, A. Narayanan, and C. Pandu Rangan. Optimal perfectly secure message transmission. In *Proc. of Advances in Cryptology: CRYPTO 2004*, LNCS 3152, pages 545–561. Springer-Verlag, 2004.

[18] K. Srinathan, P. Raghavendra, and C. Pandu Rangan. On proactive perfectly secure message transmission. In *ACISP*, pages 461–473, 2007.

[19] A. C. Yao. Protocols for secure computations. In *Proc. of 23rd IEEE FOCS*, pages 160–164, 1982.