Perfectly Reliable and Secure Communication Tolerating Static and Mobile Mixed Adversary

Ashish Choudhary[1*] Arpita Patra[1] Ashwinkumar B. V[1]
Kannan Srinathan [2] C. Pandu Rangan[1]

[1] Department of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai India 600036
Email: {ashishc,arpita,ashwin}@cse.iitm.ernet.in, rangan@iitm.ernet.in

[2] Center for Security, Theory and Algorithmic Research
International Institute of Information Technology
Hyderabad India
Email: srinathan@iiit.ac.in

Abstract

In the problem of *perfectly reliable message transmission* (PRMT), a sender **S** and a receiver **R** are connected by n bidirectional synchronous channels. A mixed adversary $\mathcal{A}_{(t_b,t_f,t_p)}$ with *infinite computing power* controls t_b, t_f and t_p channels in Byzantine, fail-stop and passive fashion respectively. Inspite of the presence of $\mathcal{A}_{(t_b,t_f,t_p)}$, **S** wants to reliably send a message m to **R**, using some protocol, without sharing any key with **R** beforehand. After interacting in phases[1] as per the protocol, **R** should output $m' = m$, without any error. In the problem of *perfectly secure message transmission* (PSMT), there is an additional constraint that $\mathcal{A}_{(t_b,t_f,t_p)}$ should not know *any* information about m in *information theoretic* sense. The adversary can be either static[2] or mobile.[3]

The *connectivity requirement*, *phase complexity* and *communication complexity* are three important parameters of any interactive PRMT/PSMT protocol and are well studied in the literature when the channels are controlled by a static/mobile Byzantine adversary. However, when the channels are controlled by mixed adversary $\mathcal{A}_{(t_b,t_f,t_p)}$, we encounter several surprising consequences. In this paper, we study the problem of PRMT and PSMT tolerating "static/mobile mixed adversary". We prove that even though the connectivity requirement for PRMT is same against both static and mobile mixed adversary, the lower bound on communication complexity for PRMT tolerating mobile mixed adversary is more than its static mixed counterpart. This is interesting because against only "Byzantine adversary", the connectivity requirement and the lower bound on the communication complexity of PRMT protocols are same for both static and mobile case. Thus our result shows that for PRMT, mobile mixed adversary is more powerful than its

*Work supported by project No. CSE/05-06/076/DITX/CPAN on Protocols for Secure Communication and Computation sponsored by Department of Information Technology, Govt. of India.

[1] A phase is a send from **S** to **R** or vice-versa

[2] A static adversary corrupts the same set of channels in each phase of the protocol. The choice of the channel to corrupt is decided before the beginning of the protocol.

[3] A mobile adversary can corrupt different set of channels in different phases of the protocol.

static counterpart. As our second contribution, we design a four phase communication optimal PSMT protocol tolerating "static mixed adversary". Comparing this with the existing three phase communication optimal PSMT protocol against "static Byzantine adversary", we find that additional one phase is enough to design communication optimal protocol against static mixed adversary. Finally, we show that the connectivity requirement and lower bound on communication complexity of any PSMT protocol is same against both static and mobile mixed adversary, thus proving that mobility of the adversary has no effect in PSMT. To show that our bound is tight, we also present a worst case nine phase communication optimal PSMT protocol tolerating mobile mixed adversary which is first of it's kind. This also shows that the mobility of the adversary does not hinder to design constant phase communication optimal PSMT protocol. In our protocols, we have used new techniques which can be effectively used against both static and mobile mixed adversary and are of independent interest.

Keywords: Perfect Reliability, Information Theoretic Security, Static and Mobile Mixed Adversary.

1 Introduction

In *perfectly reliable message transmission* (PRMT) problem, a sender **S** is connected to a receiver **R** in an unreliable network by n vertex disjoint paths called wires; **S** wishes to send a message m chosen from a finite field $\mathbb{F}$ reliably to **R**, in a *guaranteed* manner (without any error), in spite of the presence of several kinds of faults in the network. The *perfectly secure message transmission* (PSMT) problem has an additional constraint that the adversary should get *no* information about m. The faults in the network is modeled by an *adversary* who controls the actions of nodes in the network in a variety of ways and have *unbounded computing power*. Security against such an adversary is called *information theoretic security*, which is also known as *perfect security*. Since the adversary has *unbounded computing power*, we cannot use any cryptographic primitive, such as public key cryptography, hash function, etc to solve PSMT problem. The PRMT and PSMT problem was first studied and solved by Dolev et.al [6] against static Byzantine adversary. The PRMT and PSMT problems are very important primitives in various reliable and secure distributed protocols. If **S** and **R** are connected directly via a private and authenticated link (which is generally assumed in generic secure multiparty protocols [2, 8, 16, 24]), reliable and secure communication is trivially guaranteed. However, in reality, it is not economical to directly connect every two players in the network. Therefore such a complete network can only be virtually realized by simulating the missing links using PRMT and PSMT protocols as primitives.

Existing Results: There are various settings in which PRMT and PSMT problem has been studied extensively in the past (see [6, 5, 7, 17, 9, 22]). The most natural and interesting question posed in the context of PRMT/PSMT are: (a) POSSIBILITY: What is the necessary and sufficient condition that a given network should satisfy for the possibility of PRMT/PSMT from **S** to **R**? (b) OPTIMALITY: Once the POSSIBILITY of a protocol is ensured in a given network, what is the communication complexity lower bound for any reliable/secure protocol to send a message of specific length. Moreover, how to design *communication optimal* PRMT/PSMT protocols which satisfies the lower bound? The above questions can be examined in various settings. The questions in (a) and (b) have been completely answered against **static Byzantine** adversarial model in [15, 19, 1, 21, 10]) and against **mobile Byzantine** adversarial model in [22, 14]. In [18], the authors have partially

answered the questions (a) and (b) against **static mixed** adversarial model. However, nothing is known in **mobile mixed** adversarial model. Also in spite of being a very practical adversarial model, **mobile mixed** adversary have got no exposure.

Why to Study Mixed Mobile Adversary: In a typical large network, certain nodes may be strongly protected and few others may be moderately/weakly protected. An adversary may only be able to failstop(/eavesdrop in) a strongly protected node, while he may affect a weakly protected node in Byzantine fashion. Thus, we may capture the abilities of an adversary in a more realistic manner using three parameters t_b, t_f, t_p where t_b, t_f, t_p are the number of nodes under the influence of adversary in Byzantine, failstop and passive fashion respectively. Also it is better to grade different kinds of disruption done by adversary and consider them separately rather than treating every kind of fault as Byzantine fault as this is an "overkill". Also we stress that many times mobile adversary captures practical scenarios better than static adversary. For example when **S** and **R** are engaged in interaction for a long time, then some faults in initial phases can be fixed and in the mean time, a hacker may attack some other nodes.

Recently in [20], the authors have studied the issues related to the POSSIBILITY and OPTIMALITY of *unconditional reliable message transmission*[4] (which is same as PRMT with exponentially small probability of error in reliability) and *unconditional secure message transmission*[5] (which is same as PSMT with exponentially small probability of error in reliability only), in undirected synchronous networks, tolerating static mixed adversary. However, the techniques used in [20] cannot be used to design PRMT/PSMT protocols against static and mobile mixed adversary.

Our Contribution: In this work, we focus our attention on PRMT/PSMT in undirected synchronous networks against static and mobile mixed adversary. Table 1 tabulates both the existing and proposed (in this paper) connectivity requirement and communication complexity lower bound results.

Contribution 1 *We provide a worst case four phase communication optimal PSMT protocol tolerating static mixed adversary which is a first protocol of its kind. Comparing this with the existing three phase communication optimal PSMT protocol against static Byzantine adversary [15], we find that using an additional phase, we get a communication optimal PSMT protocol against static mixed adversary.*

Contribution 2 *We give the characterization for the possibility of any PRMT protocol against mobile mixed adversary and show that it is same as against static mixed adversary. We prove lower bound on the communication complexity of any PRMT protocol against mobile mixed adversary and show it is tight by designing a three phase communication optimal PRMT protocol, whose communication complexity matches this bound. Comparing these results with existing results for PRMT against static mixed adversary, we find that though mobility of mixed adversary has no affect on* POSSIBILITY *of PRMT protocols, it significantly affects its* OPTIMALITY. *This is surprising because* **mobile Byzantine** *and* **static Byzantine** *adversary has same effect in PRMT in terms of* POSSIBILITY *[22] and* OPTIMALITY *[14].*

[4] In [20], the authors have termed it as *probabilistic perfectly reliable message transmission* (PPRMT).
[5] In [20], the authors have termed it as *probabilistic perfectly secure message transmission* (PPSMT).

Table 1: Connectivity and Lower Bound for Communication Complexity for PRMT and PSMT problems; Results with "*" are provided in this paper. Moreover, all the bounds are tight. Here ℓ is the number of field elements in the message. The communication complexity is in terms of field elements.

	Byzantine Adversary		Mixed Adversary	
	Static	Mobile	Static	Mobile
PRMT; Connectivity (n)	$2t_b + 1$ [6]	$2t_b + 1$ [22]	$2t_b + t_f + 1$ [18]	$2t_b + t_f + 1$*
PRMT; Lower Bound	$\Omega(\ell)$ [21, 15]	$\Omega(\ell)$ [14]	$\Omega\left(\frac{(n-t_f)\ell}{n-(t_b+t_f)}\right)$ [18]	$\Omega(\frac{n\ell}{n-(t_b+t_f)})$*
PSMT; Connectivity (n)	$2t_b + 1$ [6]	$2t_b + 1$ [22]	$2t_b + t_f + t_p + 1$ [18]	$2t_b + t_f + t_p + 1$*
PSMT; Lower Bound	$\Omega(\frac{n\ell}{n-2t_b})$ [19]	$\Omega(\frac{n\ell}{n-2t_b})$ [19, 14]	$\Omega(\frac{n\ell}{n-(2t_b+t_f+t_p)})$ [18]*	$\Omega(\frac{n\ell}{n-(2t_b+t_f+t_p)})$*

Contribution 3 *We show that characterization for the possibility and lower bound on communication complexity of PSMT protocols tolerating static mixed adversary remain unchanged even for mobile mixed adversary. We also present a worst case nine phase communication optimal PSMT protocol tolerating mobile mixed adversary, which is first of it's kind. Comparing this with contribution 1, we conclude that mobility of adversary does not hinder the possibility of designing* **constant phase** *communication optimal PSMT protocol against mixed adversary, even though it requires slightly more number of phases.*

To design our protocols, we propose new techniques, which can be effectively used against both static and mobile mixed adversary. These techniques are completely different from the techniques used in [20] to design PPRMT/PPSMT protocols against static mixed adversary. We stress that our results on mixed adversary are not simple and trivial extensions of the existing results for Byzantine adversary.

2 Definitions, Network Settings and Adversarial Model

The underlying network is a connected synchronous network represented by an undirected graph where **S** and **R** are two nodes. A *mixed adversary*, with *unbounded* computing power, controls at most t_b, t_f and t_p nodes (excluding **S, R**) in Byzantine, fail-stop and passive fashion respectively. Following approach of [6], we abstract the network and concentrate on solving PRMT/PSMT problem for a single pair of processors (**S, R**), connected by n vertex disjoint paths $w_1, w_2, \ldots, w_n$, also known as wires.[6] In the worst case, if adversary controls a single node on a wire, then out of n wires, at most t_b, t_f and t_p wires can be under the control of the adversary in Byzantine, failstop and passive fashion respectively.

A wire which is controlled in a failstop fashion may fail to deliver any information, but if it delivers the information then it will be correct. However, the adversary will have no idea about the information that passed through a wire which is controlled in failstop fashion. A wire which is passively controlled will always deliver correct information. However, the adversary will also completely know the information, which passed through a passively controlled wire. A Byzantine corrupted wire may deliver correct information or it may deliver incorrect information. However, in any case, the adversary will completely know the information,

[6]The approach of abstracting the network as a collection of n wires is justifying using Menger's theorem [12] which states that a graph is $c-(\mathbf{S},\mathbf{R})$-connected iff **S** and **R** are connected by at least c vertex disjoint paths.

which passed through a Byzantine corrupted wire. The mixed adversary can be static or mobile. We denote the static and mobile mixed adversary by $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$ and $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ respectively.

Scope of $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$: The static mixed adversary $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$ controls the **same** set of t_b, t_f and t_p wires among n wires, in Byzantine, fail-stop and passive fashion respectively, in different phases of any PRMT/PSMT protocol. The set of wires which it controls is decided before the execution of the protocol. A wire which is under the control of $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$, will remain so throughout the protocol.

Scope of $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$: The mobile mixed adversary $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ controls **different** set of t_b, t_f and t_p wires among n wires, in Byzantine, fail-stop and passive fashion respectively, in different phases of any PRMT/PSMT protocol. A wire which is controlled by $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ in some phase, may become free from $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ in subsequent phase. Though $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ controls different set of wires in different phases of the protocol, it does not allow the adversary to gain any information which has previously passed (in earlier phases of the protocol) through the wires under its control in current phase. This is because the wires (and hence the nodes along these wires) erases all the local information from its memory at the end of each phase. Also any wire which is not under the control of the adversary in current phase will behave correctly, irrespective of the way it behaved in earlier phases of a protocol. The adversary can gain information from the wires in a cumulative fashion. For example, suppose during first phase of a protocol, $\mathcal{A}^{mobile}_{(1,1,1)}$ controls w_1, w_2 and w_3 in Byzantine, failstop and passive fashion respectively in a network, where **S** and **R** are connected by wires $w_1, w_2, \ldots, w_5$. Now suppose during second phase, it controls w_2, w_4 and w_5 in Byzantine, failstop and passive fashion respectively. Then w_1 and w_3 will behave correctly during second phase and adversary has no access to the information passing through them in second phase. At the end of second phase, adversary will know the information which passed through w_1 and w_3 during first phase and the information which passed through w_2 and w_5 during second phase.

Remark 1 *A mobile adversary is different from an adaptive adversary [4], who dynamically corrupt nodes (wires) during the protocol execution and whose choice of corrupting a wire may depend on the data seen so far. This is so because a node (wire) which is once under the control of adaptive adversary, will remain so throughout the protocol whereas in case of mobile adversary, it may become free in subsequent phases of the protocol. Also, adaptive adversary is slightly different from static adversary in the sense that static adversary decides which wires to control before the start of the protocol. Our protocols designed against static mixed adversary will also work against adaptive adversary without any modification.*

Throughout the paper we use m to denote the message that **S** wants to send to **R**. m is a sequence of ℓ field elements from a finite field $\mathbb{F}$. The only restriction on $\mathbb{F}$ is that $|\mathbb{F}| \geq n$. We use $|m|$ to denote the number of field elements in m. Any information which is sent through all the wires is said to be "broadcast". If x is "broadcast" over at least $2t_b + t_f + 1$ wires, then at most t_f wires may fail to deliver x, where as at most t_b wires may deliver incorrect x. But at least $t_b + 1$ wires will deliver correct x. So receiver will be able to correctly receive x by taking majority vote. The communication complexity of any protocol is the total number of field elements communicated by **S** and **R** in the protocol. We say that a wire is **corrupted**, if the information sent over the wire is changed. A wire which is not under the control of the adversary is said to be **honest**.

Definition 1 *(Optimal PRMT/PSMT (OPRMT/OPSMT) Protocol) Let $\mathcal{N}$ be a network, under the influence of $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$ or $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ and Π be a PRMT/PSMT protocol, which sends m from $\mathbf{S}$ to $\mathbf{R}$ in $\mathcal{N}$, by communicating $O(b)$ field elements. Then Π is called an OPRMT/OPSMT protocol if the lower bound on the communication complexity of any PRMT/PSMT protocol in $\mathcal{N}$ to send m is $\Omega(b)$ field elements.*

3 Coding Theory Preliminaries

In our protocols, we have used Reed-Solomon (RS) codes, which are used to reliably send message over a noisy channel. Let $Ch_{(t_b,t_f)}$ denote a noisy channel, where at most t_f and t_b locations of a codeword can be arbitrarily erased and changed respectively during the transmission. We call the later type of errors as Byzantine error.

Definition 2 ([11]) *For message block $M = (m_1\ m_2\ \ldots\ m_k)$ over $\mathbb{F}$, define ReedSolomon polynomial as $P_M(x) = m_1 + m_2x + m_3x^2 + \ldots + m_kx^{k-1}$. Let $\alpha_1, \alpha_2, ..., \alpha_n, n > k$, denote a sequence of distinct and fixed elements from $\mathbb{F}$. Then vector $C = (c_1\ c_2\ \ldots\ c_n)$ where $c_i = P_M(\alpha_i), 1 \leq i \leq n$ is called the Reed-Solomon codeword of size n for the message block M. We denote the size of vector C by $|C|$.*

The next theorem summarizes a known result related to Reed-Solomon codes.

Theorem 1 (Singleton Bound [11]) *Suppose a sender has generated a RS codeword C of size $|C| = N$, for a message block M of size k and sends the codeword C through $Ch_{(t_b,t_f)}$. Let the received codeword be C' of size $|C'| \geq N - t_f$ and different from C in at most t_b locations. Then the receiver can reconstruct the message M from C' iff $N \geq 2t_b + t_f + k$.*

Theorem 2 gives the number of errors which can be corrected and detected by RS codes.

Theorem 2 ([11, 5]) *Let C denote the RS codeword for a message block of size k, where $|C| = n$. Let the codeword be sent over $Ch_{(t_b,t_f)}$. Let n' denotes the size of the received codeword C', where $n' \geq n - t_f$. Then RS decoding can correct upto c Byzantine errors in C' and simultaneously detect additional d Byzantine errors in C' iff $n' - k \geq 2c + d$.*

RS-DECODING ALGORITHM [11, 13]: Berlekamp Welch algorithm is one of the most simple and efficient RS decoding algorithm existing in the literature. In general, we denote the RS decoding algorithm by $RS-DEC(n', c, d, k)$. The algorithm takes an n' length codeword C' received through $Ch_{(t_b,t_f)}$, where C' corresponds to a codeword which was encoded using a polynomial of degree $k-1$ (so the message block size is k). Let $t'_b \leq t_b$ denotes the **actual** number of Byzantine errors that are present in C'. The only information receiver knows about t'_b is that $t'_b \leq t_b$. The variables c and d are passed as parameters to the algorithm, where c represents the number of Byzantine errors that receiver wants to correct in C' and d represents the number of additional Byzantine errors that receiver wants to detect in C'. The variables c and d should satisfy the relation given in Theorem 2. In addition, $c + d \leq t_b$.

The algorithm tries to correct at most c Byzantine errors in C'. In addition to this, it tries to detect at most d additional Byzantine errors (if they are present) in C'. The algorithm either (a) outputs a polynomial of degree $k-1$, along with an error list or (b) fails to output any polynomial of degree $k-1$. The error list (if it is produced) contains at most c entries, where each entry is a pair, indicating an error location in C' along with the value received at that location in C'. We illustrate (a) and (b) in the sequel, in the context of our PRMT and PSMT protocols.

Definition 3 *We call an error list generated by RS−DEC algorithm as "good" if each of the values in the error list, pointed as corrupted/modified value, is indeed corrupted. Otherwise we call the error list as "bad".* **When an error list is "bad", it must point a correct value in C' as corrupted.**

We now design a single phase PRMT protocol called **PRU-SP-Mixed** using RS codes. In the protocol, **S** and **R** are connected by $N \geq 2t_b + t_f + 1$ wires, $w_i, 1 \leq i \leq N$, of which at most t_b and t_f wires can be under the control of a static adversary in Byzantine and fail-stop fashion respectively ($N \geq 2t_b + t_f + 1$ wires are necessary and sufficient for the existence of any PRMT protocol tolerating such a static adversary [18]). The goal is to reliably send a message m containing ℓ field elements from **S** to **R**.

Protocol PRU-SP-Mixed$(m, \ell, N, t_b, t_f, k)$: Single Phase PRMT Tolerating $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$

- **S** breaks up m into blocks $\mathbf{B_1}, \mathbf{B_2}, \ldots, \mathbf{B_{\ell/k}}$, each consisting of k field elements, where $k = N-2t_b-t_f$. If ℓ is not an exact multiple of k, a default padding is used to make ℓ mod $k = 0$.
- For each block $\mathbf{B_j}, 1 \leq j \leq \ell/k$ of size k, **S** computes n length RS codeword of $\mathbf{B_j}$ denoted by $(c_{j1}c_{j2}\ldots c_{jN})$. **S** sends $c_{ji}, 1 \leq j \leq \ell/k$ along the wire $w_i, 1 \leq i \leq N$. Note that the RS codeword of all the blocks of m are computed and sent parallely by **S** to **R** in a single phase.
- **R** parallely receives the (possibly corrupted/erased) c_{ji}'s for all $\mathbf{B_j}$'s and applies the RS decoding algorithm to each of them and reconstructs all $\mathbf{B_j}$'s. **R** then concatenates the $\mathbf{B_j}$'s to recover the message m.

Lemma 1 *Protocol* **PRU-SP-Mixed** *correctly sends m by communicating $O\left(\frac{N\ell}{N-2t_b-t_f}\right)$ field elements.*

Proof: Follows from the working of the protocol and Theorem 1. □

Protocol **PRU-SP-Mixed** has another important property given in the following theorem.

Theorem 3 *If* **R** *in advance knows the identity of $\alpha \leq t_b$ wires which are under the control of Byzantine adversary, then protocol* **PRU-SP-Mixed** *can reliably send m using block size $k \leq (N - 2t_b - t_f) + \alpha$.*

Proof: Since **R** knows α wires which are under the control of Byzantine adversary, it simply ignores these wires and therefore the connectivity (set of active wires) reduces to $N - \alpha$. Also among the values received by **R** along these $N - \alpha$ wires, at most $t_b - \alpha$ could be Byzantine corrupted. Substituting these values in Theorem 1, we get $k \leq N - \alpha - 2(t_b - \alpha) - t_f \leq (N - 2t_b - t_f) + \alpha$. Hence **PRU-SP-Mixed**$(m, \ell, N, t_b, t_f, k)$ will work correctly with $k \leq (N - 2t_b - t_f) + \alpha$.

4 Existing OPSMT Protocols Tolerating $\mathcal{A}^{static}_{t_b}$ and Its Limitations

The existing OPSMT protocol against a t_b active static Byzantine adversary $\mathcal{A}^{static}_{t_b}$ works as follows [15]: **S** and **R** are connected by $n = 2t_b + 1$ wires, of which at most t_b can be under the control of $\mathcal{A}^{static}_{t_b}$. Essentially, **S** sends one random t_b degree polynomial over each of the n wires and their n values distributed over n wires. After a sequence of interaction between **S** and **R** according to the protocol, the constant coefficients of the $t_b + 1$ polynomials which

are not under the control of the adversary, are established as an *information theoretic secure* "one time pad" between **S** and **R**. Moreover the communication complexity of the interaction is $O(n^2)$. Now using this one time pad, **S** securely sends $t_b + 1 = \Theta(n)$ field elements to **R** by communicating $O(n^2)$ field elements [15].

For tolerating $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$, **S** and **R** must be connected by at least $n = 2t_b + t_f + t_p + 1$ wires (see Theorem 4). Now if we use the same technique of sending polynomials as well as their values (as used in OPSMT protocol againt $\mathcal{A}^{static}_{t_b}$), **S** and **R** end up in establishing a secure "one time pad" of length $t_b + 1$ after communicating $O(n^2)$ field elements. The reason is that adversary can crash t_f wires and passively listen the polynomials over $(t_b + t_p)$ wires. Therefore only $n - t_f - t_b - t_p = t_b + 1$ polynomials will be unknown to the adversary. Since $n = 2t_b + t_f + t_p + 1$, t_b may not be $\Theta(n)$ and can even be a constant. Thus the resulting PSMT protocol may send a message of very small size with very high communication complexity of $O(n^2)$, which will not be an OPSMT protocol against $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$. In the next section, we propose certain new protocols based on some new techniques, using which we can design OPSMT protocols tolerating both $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$ and $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$.

5 OPSMT Tolerating Static Mixed Adversary $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$

Here we first recall the characterization for the possibility and the lower bound on communication complexity of any multiphase PSMT protocol tolerating $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$ [18].

Theorem 4 ([18]) *Any r-phase ($r \geq 2$) PSMT protocol between* **S** *and* **R** *in an undirected network $\mathcal{N}$ tolerating $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$ is possible iff $\mathcal{N}$ is $(2t_b + t_f + t_p + 1)$-(**S,R**)-connected.*

Proof: If part: We now show that if the network is not $(2t_b + t_f + t_p + 1)$-(**S,R**)-connected, then no PSMT protocol exists. For this, we make use of the result by Dolev *et al.* [6], which states that PSMT against a static adversary who can corrupt up to any t_b and t_p nodes in the network in Byzantine and passive fashion respectively, is possible if and only if the network $\mathcal{N}$ is $(2t_b + t_p + 1)$-(**S,R**)-connected.[7] Assume that a PSMT protocol Π exists in a network $\mathcal{N}$ which is not $(2t_b + t_f + t_p + 1)$-(**S,R**)-connected. Consider the network $\mathcal{N}'$ that is induced by $\mathcal{N}$ on deleting t_f vertices from a minimal vertex cutset of $\mathcal{N}$ (this can be interpreted as an adversary blocking the communication over t_f vertex disjoint paths). It follows that $\mathcal{N}'$ is not a $(2t_b + t_p + 1)$-(**S,R**)-connected network. Evidently, if Π is a PSMT protocol on $\mathcal{N}$, then Π' is a PSMT protocol on $\mathcal{N}'$, where Π' is the protocol Π restricted to the players in $\mathcal{N}'$. But from [6], we know that Π' exists only if the network $\mathcal{N}'$ is $(2t_b + t_p + 1)$-(**S,R**)-connected. Thus no such Π' is possible and hence no protocol Π over $\mathcal{N}$ tolerating the original adversary is possible.

Only If Part: Let the underlying network be $(2t_b + t_f + t_p + 1)$-(**S,R**) - connected. We design a four phase OPSMT protocol $OPSMT\_\Pi^{static}_{(t_b,t_f,t_p)}$ tolerating $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$ in section 5. □

[7] The actual expression is $(t_a + \max(t_a, t_e) + 1)$-(**S,R**)-connected, where the adversary can corrupt up to t_a nodes in active (Byzantine) fashion and t_e nodes in passive fashion in the "containment" model, where the set of actively corrupted nodes is a subset of the set of passively corrupted nodes or vice-versa. However, in this paper, we assume that the set of actively corrupted nodes are disjoint from the set of passively corrupted nodes

Theorem 5 ([18]) *Any r-phase ($r \geq 2$) PSMT protocol which securely sends ℓ field elements in the presence of $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$ needs to communicate $\Omega\left(\frac{n\ell}{n-(2t_b+t_f+t_p)}\right)$ field elements, where $n \geq 2t_b + t_f + t_p + 1$.*

PROOF: The proof follows by extending the entropy based argument used in [21] to prove the lower bound on the communication complexity of any r-phase ($r \geq 2$) PSMT protocol against $\mathcal{A}^{static}_{t_b}$. □

Let **S** and **R** be connected by $n = 2t_b + t_f + t_p + 1$ wires $w_i, 1 \leq i \leq n$. We design a four phase OPSMT protocol $OPSMT\_\Pi^{static}_{(t_b,t_f,t_p)}$ which securely sends n field elements by communicating $O(n^2)$ field elements, tolerating $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$. Comparison of this with three phase OPSMT protocol tolerating $\mathcal{A}^{static}_{t_b}$ presented in [15] shows that additional one phase is enough to design OPSMT protocol against $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$. We first design few sub-protocols and finally combine them to get $OPSMT\_\Pi^{static}_{(t_b,t_f,t_p)}$.

Assumption 1 *In our protocols, we assume that whenever sender sends some information to receiver through n wires, then the receiver receives information over first $n - t_f \leq N' \leq n$ wires and the last $n - N'$ fails to deliver any information to the receiver. This is without loss of generality because receiver can always broadcast back the index of the wires over which it has not received any information. This does not affect the communication complexity of our protocols.*

5.1 $Pad\_Establishment\_\Pi^{static}_{(t_b,t_f,t_p)}$- A Conditional Single Phase PSMT Protocol

Suppose **A** and **B** are connected by $n = 2t_b + t_f + t_p + 1$ wires under the influence of $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$. Also **A** in advance knows the identity of at least $\frac{t_b}{2}$ wires which are Byzantine corrupted. We now design a single phase sub-protocol $Pad\_Establishment\_\Pi^{static}_{(t_b,t_f,t_p)}$ which securely establishes a random one time pad of length n between **A** and **B**, which is *information theoretically secure* from $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$.

Protocol $Pad\_Establishment\_\Pi^{static}_{(t_b,t_f,t_p)}$: Single Phase Protocol to Establish a One Time Pad

Computation and Communication by A

• **A** saves the identity of the known faulty wires in a list L_{fault}. According to the problem specification, $\frac{t_b}{2} \leq |L_{fault}| \leq t_b$. **A** selects n random polynomials $q_j(x), 1 \leq j \leq n$, over $\mathbb{F}$, each of degree $t_b - |L_{fault}| + t_p$. For each $q_j(x), 1 \leq j \leq n$, **A** computes a RS codeword $[q_{j1}\ q_{j2}\ \ldots\ q_{jn}]$ of size n, such that $q_{ji} = q_j(\alpha_i), 1 \leq i \leq n$. For $1 \leq i \leq n$, if wire $w_i \notin L_{fault}$, then **A** sends to **B** the values q_{ji} over w_i. Finally **A** broadcasts L_{fault} to **B**.

Computation by B

• **B** receives L_{fault} and neglects any information received over $w_i \in L_{fault}$. Among the remaining wires, at most t_f wires can fail to deliver any information. Suppose **B** receives values over the first $n' \geq n - |L_{fault}| - t_f$ wires. Let **B** receives q'_{ji} over $w_i, 1 \leq i \leq n'$. Let $Q'_j = [q'_{j1}\ q'_{j2}\ \ldots\ q'_{jn'}], 1 \leq j \leq n$ denote the j^{th} received codeword. **B** applies $RS - DEC(n', t_b - |L_{fault}|, 0, t_b - |L_{fault}| + t_p + 1)$ algorithm to Q'_j, recovers $q_j(x)$ and hence $q_j(0)$. The n tuple $q = [q_1(0)\ q_2(0)\ \ldots\ q_n(0)]$ is established correctly and securely between **A** and **B**.

Theorem 6 *$Pad\_Establishment\_\Pi^{static}_{(t_b,t_f,t_p)}$ establishes the information theoretic secure n tuple $q = [q_1(0)\ \ldots\ q_n(0)]$ between **A** and **B** against $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$ in single phase by communicating $O(n^2)$ field elements.*

Proof: From L_{fault}, **B** identifies $|L_{fault}| \geq \frac{t_b}{2}$ Byzantine corrupted wires and neglects them. Among the remaining wires, at most t_f can fail to deliver any information. So in the worst case $n' = 2t_b + t_p + 1 - |L_{fault}|$. The codeword $Q'_j, 1 \leq j \leq n$ received by **B**, represents a RS codeword, which is RS encoded using a polynomial of degree $k - 1 = t_b - |L_{fault}| + t_p$. Also **B** knows that in Q'_j, at most $t_b - |L_{fault}|$ values could be corrupted and tries to correct these errors by applying $RS - DEC$ with $c = t_b - |L_{fault}|$ and $d = 0$. Substituting the values of n', c, d and k in the inequality of Theorem 2, we find that $RS - DEC(n', t_b - |L_{fault}|, 0, t_b - |L_{fault}| + t_p + 1)$ will be able to correct all the $t_b - |L_{fault}| \leq \frac{t_b}{2}$ errors in Q'_j and outputs $q_j(x)$ correctly.

The adversary gets at most $t_b - |L_{fault}| + t_p$ distinct points on each $t_b - |L_{fault}| + t_p$ degree polynomial $q_j(x)$, implying information theoretic security for each $q_j(0)$. For each $q_j(x), 1 \leq j \leq n$, **A** sends $n - |L_{fault}| = O(n)$ values which incurs a total communication complexity of $O(n^2)$. Also communication complexity of broadcasting L_{fault} is $O(n^2)$. □

5.2 $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$ - A Three Phase Protocol to Identify at least $\frac{t_b}{2}$ Byzantine Corrupted Wires

As before **A** and **B** are connected by $n = 2t_b + t_f + t_p + 1$ wires. We now design a *novel* three phase protocol $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$ tolerating $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$, which has the following properties: (a) If at most $\frac{t_b}{2}$ wires get Byzantine corrupted during first phase then **A** securely establishes a one time pad of length n with **B** at the end of second phase. (b) If more than $\frac{t_b}{2}$ wires get corrupted during first phase, then the pad will not be established. However, either **A** comes to know the identity of at least $\frac{t_b}{2}$ Byzantine corrupted wires at the end of second phase or **B** comes to know the identity of at least $\frac{t_b}{2}$ Byzantine corrupted wires at the end of third phase. We now formally prove the properties of protocol $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$.

Theorem 7 *1. If at most $\frac{t_b}{2}$ Byzantine wires are corrupted during* **Phase I**, *then an n length information theoretically secure pad $p = [p_1(0)\ p_2(0)\ \ldots\ p_n(0)]$ is established between* **A** *and* **B** *at the end of* **Phase II**.

2. If more than $\frac{t_b}{2}$ errors occurred during **Phase I**, *then either* **A** *or* **B** *comes to know the identity of more than $\frac{t_b}{2}$ corrupted wires at the end of* **Phase II** *or* **Phase III** *respectively.*

Proof: We prove the theorem for the worst case where during **Phase I**, t_f wires failed to deliver any information to **B**. Hence **B** receives information over $n' = n - t_f = 2t_b + t_p + 1$ wires during first phase. Hence each of the received codewords $P'_j, 1 \leq j \leq n$ will contain $n' = 2t_b + t_p + 1$ values, of which at most t_b could be corrupted. Also, each P'_j is originally RS encoded using a polynomial $p_j(x)$ of degree $k - 1 = t_b + t_p$. During **Phase II**, **B** tries to correct at most $c = \frac{t_b}{2}$ and detect additional $d = 0$ errors in each P'_j by applying $RS - DEC$. By substituting the values of n', c, d and k in the inequality of Theorem 2, we find that $RS - DEC(n', \frac{t_b}{2}, 0, t_b + t_p + 1)$ will be able to correct at most $\frac{t_b}{2}$ errors in P'_j and detects additional 0 errors in P'_j. Now we consider the following two cases:

Case I: At most $\frac{t_b}{2}$ errors occurred during Phase I: In this case at most $\frac{t_b}{2}$ values in each P'_j could be corrupted. Hence $RS - DEC$ will successfully correct all errors in each P'_j. Hence **B** will recover each $p_j(x)$ correctly and all the error lists will be "good". When **A** gets the error lists from **B**, it finds that they are "good" and concludes that **B** has recovered each

Protocol $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$

Phase I: (A to B)

• **A** randomly selects n polynomial $p_1(x), p_2(x), \ldots, p_n(x)$ over $\mathbb{F}$, each of degree $t_b + t_p$. For each $p_j(x), 1 \leq j \leq n$, **A** computes a RS codeword $[p_{j1}\ p_{j2}\ \ldots\ p_{jn}]$ of size n. Over wire $w_i, 1 \leq i \leq n$, **A** sends the values p_{ji}.

Phase II: (**B** to **A**)

• Let **B** receives information over first n' wires where $n - t_f \leq n' \leq n$. Let **B** receives $p'_{ji}, 1 \leq j \leq n$ over wire $w_i, 1 \leq i \leq n'$. **B** then forms the received codewords $P'_j = [p'_{j1}\ p'_{j2}\ \ldots\ p'_{jn'}], 1 \leq j \leq n$.

• In each $P'_j, 1 \leq j \leq n$, **B** assumes at most $\frac{t_b}{2}$ values to be corrupted, applies $RS-DEC(n', \frac{t_b}{2}, 0, t_b+t_p+1)$ algorithm to each P'_j and tries to reconstruct some polynomial $\bar{p}_j(x)$ of degree $t_b + t_p$.

• If there exists some $j \in \{1, 2, \ldots, n\}$, such that **B** fails to recover a $t_b + t_p$ degree polynomial after applying $RS - DEC(n', \frac{t_b}{2}, 0, t_b + t_p + 1)$ to codeword P'_j, then **B** broadcasts to **A**, "ERROR-R" signal and received codeword P'_j, along with its index j. /* At least $t_b/2 + 1$ Byzantine errors are present in P'_j. */

• If some polynomial of degree $t_b + t_p$ is reconstructed after applying RS decoding algorithm to each of n received codewords, then **B** proceeds as follows:

Let $Error\_List_j$ denotes the error list obtained by applying RS decoding algorithm to P'_j. Also let L_j be the number of pairs in $Error\_List_j$. Since RS decoding is applied to P'_j, assuming the number of errors in P'_j to be at most $\frac{t_b}{2}$, $L_j \leq \frac{t_b}{2}$. **B** broadcasts $Error\_List_j, 1 \leq j \leq n$ to **A**.

Computation by A

• If **A** receives "ERROR-R" signal and index j along with P'_j, then **A** locally compares P'_j with P_j (the original j^{th} codeword restricted to first n' locations), finds the identity of at least $\frac{t_b}{2} + 1$ faulty wires which delivered incorrect components of P_j during first phase and TERMINATES the protocol.

• If **A** receives n error-lists and all the n error lists are "good", then **A** concludes that **B** has recovered each $p_j(x), 1 \leq j \leq n$ correctly and the protocol terminates. Otherwise, **A** finds at least one $j \in \{1, 2, \ldots, n\}$, such that $Error\_List_j$ is "bad". If there are multiple such j's, **A** randomly selects one. In this case, **A** concludes that **B** reconstructed $\bar{p}_j(x) \neq p_j(x)$ and initiates **Phase III**.

Conditional Phase III: A to B

• If **A** has identified a j such that **B** has reconstructed $\bar{p}_j(x) \neq p_j(x)$, then **A** broadcasts to **B** the tuple $[p_{j1}\ p_{j2}\ \ldots\ p_{jn}]$, which is the original codeword corresponding to $p_j(x)$ (which **A** had sent during **Phase I**). In this case, **B** correctly receives the actual codeword corresponding to $p_j(x)$, compares it with the codeword P'_j (corresponding to $p_j(x)$) which it has received during **Phase I**, identifies more than $\frac{t_b}{2}$ faulty wires and terminates the protocol.

$p_j(x)$ correctly. Hence the vector $p = [p_1(0)\ p_2(0)\ \ldots\ p_n(0)]$ is established correctly between **A** and **B**. The security of p follows from the fact that during **Phase I**, the adversary gets at most $t_b + t_p$ points (by passively listening $t_b + t_p$ wires) on each $p_j(x)$ which are of degree $t_b + t_p$, thus each $p_j(0)$ is information theoretic secure. Also notice that each of the n error lists are "good", thus they leak no extra information about $p_j(x)$'s to $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$.

Case II: More than $\frac{t_b}{2}$ errors occurred during Phase I: Without loss of generality, let $p_j(x)$ be one of the polynomials for which at least $\frac{t_b}{2}+1$ values has been corrupted by adversary during **Phase I**. Thus j^{th} received codeword P'_j will have more than $\frac{t_b}{2}$ corrupted values. So **B** fails to reconstruct $p_j(x)$ correctly by applying $RS - DEC(n', \frac{t_b}{2}, 0, t_b + t_p + 1)$ to the P'_j. Now there are two possible cases:

1. Suppose the values in P_j' are corrupted in such a way that $RS-DEC$, when applied to P_j', fails to output any t_b+t_p degree polynomial. In this case, **B** knows that more than $\frac{t_b}{2}$ values in P_j' are corrupted. So it broadcasts P_j' to **A**, along with "ERROR-R" signal and index j. Once **A** correctly receives these values, after local verification, **A** will come to know the identity of at least $\frac{t_b}{2}+1$ faulty wires, which delivered incorrect values of polynomial $p_j(x)$ to **B** during first phase.

2. Suppose the values in P_j' are corrupted in such a way that $RS-DEC$, when applied to P_j', outputs a t_b+t_p degree polynomial $\bar{p}_j(x)$, along with $Error\_List_j$. In this case $\bar{p}_j(x) \neq p_j(x)$ and $Error\_List_j$ is "bad" and contains a correct value in it. This is so because $p_j(x)$ and $\bar{p}_j(x)$ can have same value in at most t_b+t_p points, as they are polynomials of degree t_b+t_p. So when **A** receives $Error\_List_j$, it concludes that more than $\frac{t_b}{2}$ wires delivered incorrect values of $p_j(x)$ to **B** during first phase. Hence during third phase, **A** broadcasts the original codeword for $p_j(x)$ to **B**. **B** then identifies more than $\frac{t_b}{2}$ faults after local verification. □

Theorem 8 *The communication complexity of protocol* $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$ *is* $O(n^2t_b)$.

Proof: During first phase, **A** sends a RS codeword of length n for n polynomials, thus communicating $O(n^2)$ field elements. During the second phase, **B** broadcasts n error-lists, each containing at most $\frac{t_b}{2}$ pairs, thus communicating $O(n^2t_b)$ field elements. Communication complexity of third phase again is $O(n^2)$ field elements. Hence overall complexity is $O(n^2t_b)$ field elements. □.

5.3 Reducing the Communication Complexity of Protocol $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$

We now present a nice trick to reduce the communication complexity of sending n error-lists from $O(n^2t_b)$ to $O(n^2)$ in **Phase II** of protocol $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$ (previously, it has been broadcast). Let $ERROR\_List_J$ be the the error-list with maximum number of pairs L_J, where $J \in \{1,2,\ldots,n\}$. If there are several error-lists with L_J pairs, then **B** arbitrarily selects one. **B** then broadcasts *only* $Error\_List_J$ and sends the remaining error-lists after concatenating them into a list Y and executing the protocol **PRU-SP-Mixed**$(Y,|Y|,n,t_b,t_f,L_J)$. **A** correctly receives $Error\_List_J$ and verifies whether it is "good" . If it is, then **A** concludes that **B** has correctly recovered $p_J(x)$. In this case, **A** also identifies L_J faulty wires from $Error\_List_J$. Thus from Theorem 3, protocol **PRU-SP-Mixed** will correctly deliver the list Y containing the remaining error-lists. On the other hand, if **A** finds that $Error\_List_J$ is "bad", then **A** concludes that **B** has not recovered $p_J(x)$ correctly. In this case, **A** fails to know L_J faults from $Error\_List_J$ and hence can not recover list Y delivered using **PRU-SP-Mixed**. But still **A** identifies one polynomial $(p_J(x))$ which is not received correctly by **B** (due to more than $\frac{t_b}{2}$ errors during **Phase I**). *Note that while the properties of protocol* $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$ *(Theorem 7) remain intact by incorporating these changes, the communication complexity reduces to* $O(n^2)$.

Lemma 2 *The above steps when incorporated in* **Phase II** *of protocol* $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$, *reduces its communication complexity to* $O(n^2)$.

Proof: Broadcasting a *single* error-list ($Error\_List_J$) requires communicating $O(n^2)$ field elements. From Lemma 1 and Theorem 3, sending the remaining error-lists by **PRU-SP-Mixed**$(Y, |Y|, n, t_b, t_f, L_J)$ will require communicating $O\left(\frac{|Y|}{L_J} * n\right) = O(n^2)$ field elements because $|Y| \leq (n-1) * (2L_J)$. □

Changed Steps in Communication Efficient $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$

If during **Phase II**, some polynomial of degree $t_b + t_p$ is obtained after applying RS decoding algorithm to each of the n received codewords, then **B** proceeds as follows:

• Let $J \in \{1, 2, \ldots, n\}$ be the index of the error-list with maximum number of pairs L_J. **B** concatenates all the error-lists, except $Error\_List_J$ to form a list Y. **B** broadcasts $Error\_List_J$ along with its index J to **A** and sends the list Y by executing the protocol **PRU-SP-Mixed**$(Y, |Y|, n, t_b, t_f, L_J)$.

Computation by A

• **A** correctly receives the index J and $Error\_List_J$. He checks whether $Error\_List_J$ is "good". **IF** YES, then **A** concludes that **B** has correctly recovered $p_J(x)$. **A** also identifies L_J faulty wires from $Error\_List_J$ and hence from Theorem 3, correctly receives Y delivered by **PRU-SP-Mixed**. **A** then separates the individual error-lists from Y. The rest of the computation by **A** are now same as in protocol $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$.
ELSE if $Error\_List_J$ is "bad" then **A** concludes that **B** has reconstructed $\bar{p}_J(x) \neq p_J(x)$ and executes conditional **Phase III**.

5.4 Designing Four Phase OPSMT Protocol $OPSMT\_\Pi^{static}_{(t_b,t_f,t_p)}$

We now combine $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$ and $Pad\_Establishment\_\Pi^{static}_{(t_b,t_f,t_p)}$ to design a four phase OPSMT protocol called $OPSMT\_\Pi^{static}_{(t_b,t_f,t_p)}$ which is secure against $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$. In the protocol, we show, how a one time pad is established between **S** and **R**. Once **S** knows that the pad is going to be established, **S** can blind the message by XORing it with the pad and broadcasts the blinded message to **R** in the last phase of the protocol. On receiving the blinded message, **R** extracts the message by XORing the blinded message with the pad.

Protocol $OPSMT\_\Pi^{static}_{(t_b,t_f,t_p)}$ **- A Four Phase OPSMT Protocol Tolerating** $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$

• **R** and **S** starts executing protocol $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$, where **Phase I** is initiated by **R**. IF at the end of **Phase II** of protocol $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$, pad $p = [p_1(0)\ p_2(0)\ \ldots\ p_n(0)]$ is established securely between **R** and **S**, then **R** terminates the protocol by broadcasting "SUCCESS-R" signal to **S**.

• IF at the end of **Phase II** of protocol $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$, **R** identifies at least $\frac{t_b}{2} + 1$ faulty wires, then **R** securely establishes a one time pad $q = [q_1(0)\ q_2(0)\ \ldots\ q_n(0)]$ with **S** by executing protocol $Pad\_Establishment\_\Pi^{static}_{(t_b,t_f,t_p)}$.

• IF at the end of **Phase III** of protocol $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$, **S** identifies at least $\frac{t_b}{2} + 1$ faulty wires, then **S** securely establishes a one time pad $q = [q_1(0)\ q_2(0)\ \ldots\ q_n(0)]$ with **R** by executing protocol $Pad\_Establishment\_\Pi^{static}_{(t_b,t_f,t_p)}$ and terminates the protocol.

We now prove the correctness and security of protocol $OPSMT\_\Pi^{static}_{(t_b,t_f,t_p)}$.

Theorem 9 *In* $OPSMT\_\Pi^{static}_{(t_b,t_f,t_p)}$, **S** *correctly establishes a random, information theoretically secure one time pad of length* n *with* **R** *in at most four phases.*

Proof: In $OPSMT\_\Pi^{static}_{(t_b,t_f,t_p)}$, the sub-protocol $Error\_Identification\_\Pi^{static}_{(t_b,t_f,t_p)}$ terminates in either two phases or three phases. If it terminates in two phases, then there are two possibilities: If at the end of second phase, **R** concludes that the pad p is *securely* established with **S**, then **R** terminates the protocol in third phase by broadcasting "SUCCESS-R" signal. Otherwise at the end of **Phase II**, **R** will know the identity of at least $\frac{t_b}{2}+1$ faulty wires (see Theorem 7). With this knowledge, **R** *securely* establishes the pad q with **S** during **phase III** using sub-protocol $Pad\_Establishment\_\Pi^{static}_{(t_b,t_f,t_p)}$ (Theorem 6).

If $Error\_Identification$ terminates in three phases, then at the end of **Phase III**, **S** identifies at least $\frac{t_b}{2}+1$ faulty wires (Theorem 7). Now **S** establishes the pad q with **R** during fourth phase, using $Pad\_Establishment$ (see Theorem 6). The security of pad p (q) follows from Theorem 7 (Theorem 6).

Theorem 10 *Protocol* $OPSMT\_\Pi^{static}_{(t_b,t_f,t_p)}$ *is an OPSMT protocol communicating* $O(n^2)$ *field elements.*

Proof: The communication complexity follows from Lemma 2, Theorem 6 and working of the protocol. From Theorem 5, in an $n = 2t_b + t_f + t_p + 1$ connected network, any four phase PSMT protocol has to communicate $\Omega(n^2)$ field elements to securely send n field elements against $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$. Since the communication complexity of $OPSMT\_\Pi^{static}_{(t_b,t_f,t_p)}$ is $O(n^2)$, it is an OPSMT protocol. □

Remark 2 *Noticeably* $OPSMT\_\Pi^{static}_{(t_b,t_f,t_p)}$ *sends only codeword of polynomials, in contrast to the existing protocol summarized in section 4, which sends both polynomial and its codeword. The advantage that we get by sending only codeword is that we obtain one information theoretic secure value per codeword (after some intermediate information exchanges and then applying RS decoding), thus establishing a secure one time pad of size* $\Theta(n)$ *between* **S** *and* **R**. *Soon, we will show that this technique can be used to design OPRMT and OPSMT protocols even against mobile mixed adversary.*

6 OPRMT Tolerating Mobile Mixed Adversary $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$

We first recall that for the existence of any PRMT protocol tolerating $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$, the network should be $(2t_b + t_f + 1)$-(**S, R**)-connected [18]. For full details see **APPENDIX B**. In the presence of $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$, the necessary and sufficient condition for the existence of any PRMT protocol between **S** and **R** is given by the following theorem.

Theorem 11 *Any PRMT protocol between* **S** *and* **R** *in an undirected network* $\mathcal{N} = (\mathbb{P}, E)$, *tolerating a mobile mixed adversary* $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ *is possible if and only if* $\mathcal{N}$ *is* $(2t_b + t_f + 1)$-*(***S,R***)-connected.*

Proof: If part: $(2t_b+t_f+1)$-(**S,R**)-connected network is required for the existence of PRMT protocols against a weaker adversary $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$ [18]. Hence it is required against more stronger $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ also.

Only If Part: Suppose that the network is $(2t_b + t_f + 1)$-(**S, R**)-connected. **S** broadcasts the message over all the wires. **R** then recovers the message by taking the majority voting. □

As a sufficiency proof, we specified broadcasting which is a naive protocol. It communicates $n\ell$ field elements for transmitting ℓ elements reliably. Therefore broadcasting is not

an efficient PRMT protocol against $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$. So, the important question here is: can we reliably send a message containing ℓ field elements by communicating less than $O(n\ell)$ field elements against $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$? We answer this question by proving the lower bound on communication complexity of PRMT protocols tolerating $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ and show that it is different from the existing lower bound against $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$. This shows that as far as lower bound on communication complexity of PRMT is concerned, $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ is more powerful than $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$. We also design a three phase OPRMT protocol tolerating $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$.

Remark 3 *In [18], it is shown that any PRMT protocol in a n-(***S, R***)-connected network ($n \geq 2t_b + t_f + 1$), communicates $\Omega\left(\frac{(n-t_f)\ell}{n-(t_b+t_f)}\right)$ field elements in order to reliably send ℓ field elements against $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$. If $n = 2t_b + t_f + 1$, then this it implies that any PRMT protocol has to communicate $\Omega(\ell)$ field elements to reliably send a message containing ℓ field elements against $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$. Moreover, in [23], the authors have shown that this bound is tight by designing an $O\left(\log(\frac{t_f}{n-t_f})\right)$ phase OPRMT protocol, which sends a message of size ℓ field elements by communicating $O(\ell)$ field elements, where $n = 2t_b + t_f + 1$. However, we next show that lower bound for communication complexity is different for $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$.*

Theorem 12 *Any PRMT protocol between* **S** *and* **R** *connected by $n \geq 2t_b + t_f + 1$ wires under the influence of $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ must communicate $\Omega(\frac{n\ell}{n-(t_b+t_f)})$ field elements in order to transmit a message containing ℓ field elements.*

Proof: We now prove the lower bound on the communication complexity of any r-phase ($r \geq 2$) PRMT protocol which sends ℓ field elements tolerating $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$. The proof of the theorem is inspired by entropy based argument, used to prove the lower bound on the communication complexity of PRMT/PSMT protocols against $\mathcal{A}^{static}_{t_b}$ [21]. Before providing the proof, we first try to quantify the reason behind different lower bound for static and mobile mixed adversary. In static case, the lower bound is derived by assuming that both **S** and **R** knows the set of wires which are fail-stop corrupted in *advance*. Hence the term $(n-t_f)$ appears in the numerator of the lower bound expression against $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$ (see Remark 3). This is a reasonable assumption because against static adversary, we can always strategies protocols to remember faults caught in earlier phases and use that knowledge to amortize the overall communication complexity and message size in later phases (the OPRMT protocol of [23] is based on this important principle). However, protocols tolerating mobile mixed adversary is memoryless because adversary corrupts different set of wires in different phases of the protocol. Hence, the protocols against mobile mixed adversary cannot use the knowledge of the faults, specially fail-stop faults, which occurred in previous phases, to amortize the communication complexity and message size in later phases. We now present our formal proof for this theorem.

Lemma 3 *The communication complexity of any multi-phase PRMT protocol to send a message against a mobile adversary corrupting up to $b(\leq t_b), F(\leq t_f)$ and $P(\leq t_b + t_p)$ of the wires in Byzantine, Fail-stop and passive manner respectively (in each phase of the protocol) is not less than the share complexity (sum of the length of the shares) of distributing n shares for the message such that any set of $n - b - F$ shares has full information about the message.*

To prove the lemma, we begin with defining a weaker version of single-phase PRMT called PRMT with Error Detection (PRMTED). We then prove the equivalence of communication

complexity of PRMTED protocol to send message $\mathbf{M}$ and the share complexity (sum of the length of all shares) of distributing n shares for $\mathbf{M}$ such that any set of $n-b-F$ correct shares has full information about $\mathbf{M}$. To prove the aforementioned statement, we show their equivalence (Claim 4). We then show the equivalence of single-phase protocol PRMTED and multiphase PRMT protocol in terms of communication complexity and also answer the question: why it is weaker than multiphase PRMT protocol (Claim 6). These two equivalence will prove the desired equivalence as stated in this lemma.

Definition 4 *A single phase PRMT protocol is called PRMTED if it satisfies the following:*

1. *If the adversary is passive throughout the protocol then* $\mathbf{R}$ *correctly receives the message sent by* $\mathbf{S}$.

2. *If the adversary corrupts information over some* b *wires* ($b \leq t_b$)*, then* $\mathbf{R}$ *detects it, and aborts.*

3. *If adversary blocks some* $F \leq t_f$ *wires, without doing any other corruption, then* $\mathbf{R}$ *recovers message correctly. Else if adversary blocks more than* t_f *wires or do some corruption (or both), then* $\mathbf{R}$ *aborts.*

Observe that PRMTED is a strictly weaker version of PRMT because a PRMT protocol not only detects errors but also corrects them. We next show that the properties of PRMTED protocol for sending message $\mathbf{M}$ is equivalent to the problem of distributing n shares for $\mathbf{M}$ such that any set of $n-b-F$ correct shares has full information about $\mathbf{M}$.

Claim 1 *Let* Π *be a PRMTED protocol tolerating an adversary that can corrupt up to any* b, F *and* P *of the* n *wires connecting* $\mathbf{S}$ *and* $\mathbf{R}$ *in Byzantine, fail-stop and passive manner respectively. In an execution of* Π *for sending a message* $\mathbf{M}$*, the data* $s_i, 1 \leq i \leq n$ *sent by the* $\mathbf{S}$ *along wires* $w_i, 1 \leq i \leq n$ *form* n *shares for* $\mathbf{M}$ *such that any set of* $n-b-F$ *correct shares has full information about* $\mathbf{M}$.

Proof: We show that any set of $n-b-F$ shares has full information about $\mathbf{M}$. The proof is by contradiction. For a set of wires A, let $Message(\mathbf{M}, A)$, denotes the set of messages sent along the wires in A during the execution of PRMTED to send $\mathbf{M}$. Now for any set of C wires with $|C| \geq n-b-F$, $Message(\mathbf{M}, C)$ should uniquely determine the message $\mathbf{M}$. If not, then there exists another message $\mathbf{M}'$ such that $Message(\mathbf{M}, C) = Message(\mathbf{M}', C)$. By definition the adversary can block all the messages sent along the F wires not in C and change the messages along b wires not in C, such that the set of set of all messages received by $\mathbf{R}$ is identical to $Message(\mathbf{M}', C)$. In this case, $\mathbf{R}$ receives the message M', while $\mathbf{S}$ sent M. This is a contradiction since $\mathbf{R}$ must detect that there has been a corruption. □

The above claim also says that the communication complexity of PRMTED protocol to send $\mathbf{M}$ is same as the share complexity (sum of the length of all shares) of distributing n shares for a message $\mathbf{M}$ such that any set of $n-b-F$ shares has full information about $\mathbf{M}$. Now we step forward to show the communication complexity of PRMTED protocol is the lower bound on the communication complexity of any multiphase PRMT protocol against $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$.

Before that we take a closer look at the execution of any multi-phase PRMT protocol against $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$. $\mathbf{S}$ and $\mathbf{R}$ are modeled as polynomial time Turing machines with access to a random tape. The number of random bits used by the $\mathbf{S}$ and $\mathbf{R}$ are bounded by a polynomial

$q(n)$. Let $r_1, r_2 \in \{0,1\}^{q(n)}$ denote the contents of the random tapes of $\mathbf{S}$ and $\mathbf{R}$ respectively. The message $\mathbf{M}$ is an element from the set $\{0,1\}^{p(n)}$, where $p(n)$ is a polynomial. A transcript for an execution of a multiphase PRMT protocol Π is the concatenation of all the messages sent by $\mathbf{S}$ and $\mathbf{R}$ along all the wires.

Definition 5 *A passive transcript $\mathcal{T}(\Pi, \mathbf{M}, r_1, r_2)$ is a transcript for the execution of the multiphase protocol Π with $\mathbf{M}$ as the message to be sent, r_1, r_2 as the contents of the random tapes of sender $\mathbf{S}$ and the receiver $\mathbf{R}$ and the adversary $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ remaining passive throughout the execution of Π. Let $\mathcal{T}(\Pi, \mathbf{M}, r_1, r_2, w_i)$ denote the passive transcript restricted to messages exchanged along the wire w_i. When $\Pi, \mathbf{M}, r_1, r_2$ are obvious from the context, we drop them and denote the passive transcript restricted to a wire w_i by $\mathcal{T}_{w_i}$. Similarly, $\mathcal{T}_B$ denotes the passive transcript $\mathcal{T}$ restricted to a set of wires in B.*

Given $(\mathbf{M}, r_1, r_2)$ it is possible for $\mathbf{S}$ to compute $\mathcal{T}(\Pi, \mathbf{M}, r_1, r_2)$ by simulating $\mathbf{R}$ with random tape r_2. Similarly given $(\mathbf{M}, r_1, r_2)$, $\mathbf{R}$ can compute $\mathcal{T}(\Pi, \mathbf{M}, r_1, r_2)$ by simulating $\mathbf{S}$ with r_1. Note that although $\mathbf{S}$ and $\mathbf{R}$ require both r_1, r_2 to generate the transcript $\mathcal{T}$, $\mathbf{R}$ requires only r_2 in order to obtain the message $\mathbf{M}$ from the transcript $\mathcal{T}$. This is clear since $\mathbf{R}$ does not have access to r_1 during the execution of Π but still can retrieve the message $\mathbf{M}$ from the messages exchanged.

Definition 6 *A passive transcript $\mathcal{T}_B$, with $n - F \leq |B| \leq n$ is said to be a valid fault-free transcript with respect to $\mathbf{R}$ if there exists random string r_2 and message $\mathbf{M}$ such that protocol Π at $\mathbf{R}$ with r_2 as the contents of the random tape and $\mathcal{T}_B$ as the messages exchanged, terminates by outputting the message $\mathbf{M}$.*

Definition 7 *Two passive transcripts $\mathcal{T}_B$ and $\mathcal{T}'_B$, where $n - F \leq B \leq n$ are said to be* adversely close *if the two transcripts differ only on a set of wires A such that $|A| \leq b + (|B| - (n - F))$. Formally $|\{w_i | \mathcal{T}_{w_i} \neq \mathcal{T}'_{w_i}\}| \leq b + (|B| - (n - F))$.*

Claim 2 *Two valid fault-free transcripts $\mathcal{T}_B(\Pi, \mathbf{M}, r_1, r_2)$ and $\mathcal{T}_B(\Pi, \mathbf{M}', r'_1, r'_2)$ with two different message inputs $\mathbf{M}, \mathbf{M}'$, cannot be adversely close to each other, where $n - F \leq B \leq n$.*

Proof: Suppose two valid fault-free transcripts $\mathcal{T}_B(\Pi, \mathbf{M}, r_1, r_2)$ and $\mathcal{T}_B(\Pi, \mathbf{M}', r'_1, r'_2)$ are adversely close, then there is a set of wires A, $|A| \leq b + (|B| - (n - F))$ such that the two transcripts differ only on messages sent along the wires in A. Without loss of generality, assume last $b + (|B| - (n - F))$ wires belong to A with $A = X \circ Y$ where $|X| = b$ and $|Y| = (|B| - (n - F))$. Consider the following two executions of Π where the contents of $\mathbf{S}$'s and $\mathbf{R}$'s random tapes are r_1, r_2 respectively

• $\mathbf{S}$ wants to send $\mathbf{M}$. $\mathbf{S}$ and $\mathbf{R}$ executes Π while the adversary stop the wires in Y to deliver any message. As $\mathcal{T}_{B-Y}(\Pi, \mathbf{M}, r_1, r_2)$ is a valid transcript with respect to $\mathbf{M}$, $\mathbf{R}$ terminates with output $\mathbf{M}$.

• $\mathbf{S}$ wants to send $\mathbf{M}$. $\mathbf{S}$ and $\mathbf{R}$ executes Π. The adversary blocks messages over Y and changes the messages along wires in X such that the view of $\mathbf{S}$ is $\mathcal{T}_{B-Y}(\Pi, \mathbf{M}, r_1, r_2)$ but the view of $\mathbf{R}$ is $\mathcal{T}_{B-Y}(\Pi, \mathbf{M}', r'_1, r'_2)$. Since $\mathcal{T}_{B-Y}(\Pi, \mathbf{M}', r'_1, r'_2)$ is a valid transcript with respect to $\mathbf{M}'$, $\mathbf{R}$ will terminate with output $\mathbf{M}'$.

The two scenarios differ only in the adversarial behavior and in the contents of **R**'s random tape. In both the scenarios **S** wanted to send message **M**. But the message received by receiver **R** in the second case is an incorrect message $\mathbf{M}'$. This is a contradiction because Π is a PRMT protocol. □

Till now, we have shown that a passive transcript over at least $n - b - F$ wires allows **R** to output **M** correctly. We now show how to reduce a multiphase PRMT protocol into a single phase PRMTED protocol.

Protocol PRMTED

• **S** computes the passive transcript $\mathcal{T}(\Pi, \mathbf{M}, r_1, r_2)$ for some random r_1 and r_2 and sends $\mathcal{T}(\Pi, \mathbf{M}, r_1, r_2, w_i)$ to **R** along wire w_i.

• If **R** does not receives information through at least $n - F$ wires then **R** outputs ERROR and stop. Otherwise, let **R** receives information over the set of wires $B = \{w_{i_1}, w_{i_2}, \ldots, w_{i_\alpha}\}$ where $n - F \leq |B| \leq n$. **R** concatenates the values received along these wires to obtain a transcript $\mathcal{T}_B$ (which may be corrupted along b wires) and does the following:

- for each $\mathbf{M} \in \{0,1\}^{p(n)}$ and $r_2 \in \{0,1\}^{q(n)}$ do:
  If $\mathcal{T}_B$ is a valid transcript with random tape contents r_2 for message **M** then output **M** and stop.
  Output ERROR.

Claim 3 *The Communication complexity of any multiphase PRMT protocol* Π *against* $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ *is at least the communication complexity of* **PRMTED** *protocol.*

Proof: The communication complexity of any multiphase PRMT protocol Π assuming the adversary to be passive during the complete execution, is trivially a lower bound for any multiphase PRMT protocol with corruption in any phase. In **PRMTED**, **S** communicates the transcript generated by him assuming adversary to be passive throughout the execution of Π to **R**. It is easy to see that the cost of communicating such a transcript by **PRMTED** is same as of Π with the assumption that adversary remain passive throughout the execution of Π.

From Claim 5, we know that valid transcripts of two different messages cannot be adversely close to each other. So irrespective of the actions of the adversary, the transcript received by **R** cannot be a valid transcript for any message other than **M** for any value of r_2. Hence if **R** outputs a message **M** then it is the same message sent by **S**. □

This completes the proof of Lemma 5. We now prove the share complexity of distributing n shares for a message such that any set of $n-b-F$ correct shares has full information about the message.

Lemma 4 *The share-complexity (that is sum of the length of all shares) of distributing* n *shares for a message of size* ℓ *field elements from* $\mathbb{F}$ *such that any set of* $n - b - F$ *correct shares has full information about the message is* $\Omega(\frac{n\ell}{(n-b-F)})$.

Proof: Let X_i denotes the i^{th} share. For any subset $A \subseteq \{1, 2 \ldots n\}$, let X_A denotes the set of variables $\{X_i | i \in A\}$. Let **M** be a value drawn uniformly at random from $\mathbb{F}^l$. Then the message **M** and the shares X_i are random variables. Let $H(X)$ for a random variable denote its entropy. Let $H(X|Y)$ denotes the entropy of X conditional on Y. The conditional entropy measures how much entropy a random variable X has remaining if we have already

learned completely the value of a second random variable Y [3]. Since any set B consisting of $n-b-F$ shares has full information about $\mathbf{M}$, we have $H(\mathbf{M}|X_B) = 0$. Since $\mathbf{M}$ is chosen uniformly from $\mathbb{F}^\ell$, we have

$$H(\mathbf{M}|X_\emptyset) = H(\mathbf{M}) = \ell \tag{1}$$

From the chain rule of the entropy [3], for any two random variable X_1, X_2, we have $H(X_1, X_2) = H(X_2) + H(X_1|X_2)$. Substituting $X_1 = \mathbf{M}$ and $X_2 = X_B$, we get

$$H(\mathbf{M}, X_B) \quad = \quad H(X_B) + H(\mathbf{M}|X_B)$$

From the properties of joint entropy [3], for any two variables X_1, X_2, we have $H(X_1, X_2) \geq H(X_1)$ and $H(X_1, X_2) \geq H(X_2)$. Thus, $H(\mathbf{M}, X_B) \geq H(\mathbf{M})$. Substituting in the above equation, we get

$$\begin{aligned} H(\mathbf{M}) &\leq H(X_B) + H(\mathbf{M}|X_B) \\ &\leq H(X_B) + 0 \text{ because } \mathbf{M} \text{ can be known completely from} X_B \end{aligned}$$

Consequently, $H(\mathbf{M}) \leq H(X_B)$. Therefore for any set B of cardinality $n-b-F$, we have

$$H(X_B) \geq H(\mathbf{M}) \Rightarrow \sum_{i \in B} H(X_i) \geq H(\mathbf{M})$$

Since there are $\binom{n}{n-b-F}$ possible subsets of cardinality $n-b-F$, summing the above equation over all possible subsets of cardinality $n-b-F$ we get

$$\sum_{B} \sum_{i \in B} H(X_i) \quad \geq \quad \binom{n}{n-b-F} H(\mathbf{M})$$

Now in all the possible $\binom{n}{n-b-F}$ subsets of size $n-b-F$, each of the term $H(X_i)$ appears $\binom{n-1}{n-b-F-1}$ times. So

$$\binom{n-1}{n-b-F-1} \sum_{i=1}^{n} H(X_i) \geq \binom{n}{n-b-F} H(\mathbf{M}) \quad \Rightarrow \quad \sum_{i=1}^{n} H(X_i) \geq \frac{n}{n-b-F} H(\mathbf{M})$$

Since $H(\mathbf{M}) = \ell$, we get

$$\sum_{i=1}^{n} H(X_i) \geq \frac{n\ell}{n-b-F}$$

Thus the share-complexity for any $\mathbf{M} \in \mathbb{F}^\ell$ is $\Omega\left(\frac{n\ell}{n-b-F}\right)$. □

Since $b \leq t_b$ and $F \leq t_f$, $\Omega\left(\frac{n\ell}{n-b-F}\right) = \Omega\left(\frac{n\ell}{n-(t_b+t_f)}\right)$. Theorem 12 now follows from Lemma 5 and Lemma 6. □

□

We now design a three phase OPRMT protocol $OPRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$, which reliably sends a message m containing nt_b ($t_b \geq 1$) field elements by communicating $O(n^2)$ field elements, where $n = 2t_b + t_f + 1$. If $t_b = \Theta(n)$, then the protocol sends $\Theta(n^2)$ field elements by communicating $O(n^2)$ field elements. Note that if $t_b = 0$, then we can directly send a message of size ℓ by broadcasting it over n wires, incurring a communication cost of $O(n\ell)$, which will be an OPRMT protocol (for $t_b = 0$).

Protocol $OPRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$ - Three Phase OPRMT Protocol Tolerating $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$

Phase I: S to R

- **S** divides m into blocks $B_1, B_2, \ldots, B_z$, each containing $1 + \frac{t_b}{2}$ field elements. A default pad can be used to make the size of the last block to be $1 + \frac{t_b}{2}$. For each $B_j, 1 \leq j \leq z$, **S** computes a RS codeword of size n denoted by $[c_{j1}\ c_{j2}\ \ldots\ c_{jn}]$ and sends c_{ji} through w_i.

Phase II: R to S: **R** receives information over the first $n - t_f \leq n' \leq n$ wires. Through these n' wires, **R** receives the values $c'_{ji}, 1 \leq j \leq z, 1 \leq i \leq n'$. Let $C'_j, 1 \leq j \leq z$ denotes j^{th} received codeword where $C'_j = [c'_{j1}\ c'_{j2}\ \ldots\ c'_{jn'}]$. **R** applies $RS - DEC(n', \frac{t_b}{2}, \frac{t_b}{2}, \frac{t_b}{2} + 1)$ algorithm to each C'_j and tries to correct $\frac{t_b}{2}$ errors and simultaneously detect additional $\frac{t_b}{2}$ errors in C'_j.

- If $RS - DEC$ does not detects additional errors ($\leq \frac{t_b}{2}$) in any C'_j, after correcting at most $\frac{t_b}{2}$ errors, then $RS - DEC$ recovers each block B_j of m correctly. **R** recovers m by concatenating all B_j's and broadcasts "TERMINATE" signal to **S**.
- If $\exists J \in \{1, 2, \ldots, z\}$, such that $RS - DEC$ detects additional errors in C'_J, after correcting at most $\frac{t_b}{2}$ errors, then **R** broadcasts C'_J and index J.

Phase III: S to R: If **S** receives "TERMINATE" signal, then he terminates the protocol. Else **S** does the following:

- **S** receives C'_J and index J. After locally comparing C'_J with its corresponding original codeword C_J, **S** identifies at least $\frac{t_b}{2} + 1$ wires which were Byzantine corrupted during **Phase I** and broadcasts their identity to **R**.

Local Computation by R If during second phase, **R** has broadcasted C'_J, then it does the following:

- **R** correctly receives the identity of at least $\frac{t_b}{2} + 1$ wires, which delivered incorrect values during **Phase I**. From each codeword C'_j received during first phase, **R** removes the c'_{ji}'s received over these corrupted wires. **R** applies $RS - DEC$ to the new C'_j's, assuming the number of errors c (to be corrected) to be at most $\frac{t_b}{2}$ and the number of additional errors d (to be detected) to 0 and correctly recovers all B_j's and hence m.

Theorem 13 *Protocol $OPRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$ reliably sends nt_b field elements by communicating $O(n^2)$ field elements in at most three phases tolerating $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$.*

Proof: In the worst case, **R** receives information over $n' = n - t_f = 2t_b + 1$ wires during **Phase I**. Now each received codeword C'_j is RS encoded by a polynomial of degree $k - 1 = \frac{t_b}{2}$. **R** sets $c = d = \frac{t_b}{2}$ (which along with the value of n' and k, satisfies the inequality of Theorem 2) and applies RS-DEC to C'_j. If at most $\frac{t_b}{2}$ errors occur during **Phase I**, then $RS - DEC$ will correct them ($c = \frac{t_b}{2}$) and successfully output each B_j. Moreover **R** will know that it has recovered B_j's correctly because $RS - DEC$ has not detected the presence of any additional error in C'_j. Otherwise, there exists an $J \in \{1, 2, \ldots, z\}$, such that $RS - DEC$ detects additional faults ($d = \frac{t_b}{2}$) in C'_J after correcting $c = \frac{t_b}{2}$ errors in C'_J and does not outputs B_J. In this case, **R** broadcasts C'_J to **S**, who after local verification, identifies at least $\frac{t_b}{2} + 1$ faulty wires which were Byzantine corrupted during **Phase I**. After knowing the identity of these wires from **S**, **R** neglects the values in each C'_j's which were received along those faulty wires during **Phase I**. Now each C'_j will be of length at least $n' = n' - \frac{t_b}{2} = 2t_b + 1 - (\frac{t_b}{2}) = \frac{3t_b}{2} + 1$ and at most $\frac{t_b}{2}$ values in them could be corrupted. Substituting $d = 0, c = \frac{t_b}{2}$ and values of n' and k in the inequality in Theorem 2, we find that $RS - DEC$ can correct the remaining $c \leq \frac{t_b}{2}$ errors in each C'_j and outputs B_j's correctly. This proves the correctness.

During **Phase I**, **S** sends an n length codeword for each B_j of size $1 + \frac{t_b}{2}$. So the total communication cost of **Phase I** is $O\left(\frac{|m|}{\frac{t_b}{2}} * n\right) = O(n^2)$ because $|m| = nt_b$. It is easy to verify that in the remaining two phases, $O(n^2)$ field elements are communicated. Hence the

theorem. □

Remark 4 *In [14], it is shown that against only Byzantine adversary (i.e. $t_f = t_p = 0$), mobility of the adversary has no effect in comparison to its static counterpart in terms of communication complexity and phase complexity of PRMT protocols. However, Remark 3 and Theorem 12 shows that in mixed adversarial model, mobile adversary is more powerful than its static counterpart in terms of the communication complexity of PRMT protocols.*

In the next section we use $PRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$ as a black box to design OPSMT tolerating $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$.

7 OPSMT Tolerating Mobile Mixed Adversary $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$

The characterization for the possibility of any multiphase PSMT protocol tolerating $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ is same as the characterization for PSMT against $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$ (see Theorem 4). The fact that $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ is more powerful than $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$ proves the necessity of the characterization. To prove the sufficiency, we present an OPSMT protocol in the sequel tolerating $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$. Before that we note that the lower bound on communication complexity of PSMT protocols against $\mathcal{A}^{static}_{(t_b,t_f,t_p)}$ (specified in Theorem 5) holds good in case of PSMT protocols tolerating $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$. Since $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ can corrupt different set of wires in each phase, the protocol cannot adapt as it finds corrupted wires; thus it can be considered to be memoryless. In general, a mobile adversary is more powerful than static adversary. So the lower bound given in Theorem 5 is trivially a lower bound against mobile adversary. We now show that this bound is *tight*. We present a *constant* phase OPSMT protocol $OPSMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$ which securely sends $\Theta(n)$ field elements by communicating $O(n^2)$ field elements against $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$, where **S** and **R** are connected by $n = 2t_b + t_f + t_p + 1$ wires. The protocol terminates in at most nine phases and establishes an information theoretically secure one time pad of length either $n-1$ or $\frac{n}{2}$ between **S** and **R**.

Protocol $OPSMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$ - A Constant Phase OPSMT Protocol Tolerating $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$

Phase I: S to R **S** selects n random polynomials $p_j(x), 1 \leq j \leq n$ over $\mathbb{F}$, each of degree $t_b + t_p$, such that $p_j(0) = s_j$. For each $p_j(x)$, **S** forms a RS codeword $[c_{j1}\ c_{j2}\ \ldots\ c_{jn}]$ of size n and sends c_{ji} over $w_i, 1 \leq i \leq n$.

Phase II: R to S **R** receives c'_{ji}'s over the first $n - t_f \leq n' \leq n$ wires. **R** applies $RS-DEC(n', \frac{t_b}{2}, 0, t_b + t_p + 1)$ to the $j^{th}, 1 \leq j \leq n$ received codeword $C'_j = [c'_{j1}\ c_{j2}\ \ldots\ c'_{jn'}]$. There are two possible cases:

1. Corresponding to each $C'_j, 1 \leq j \leq n$, $RS-DEC$ outputs some polynomial $\bar{p}_j(x)$ of degree $t_b + t_p$, along with error list $Error\_List_j$ containing at most $\frac{t_b}{2}$ pairs. **R** then combines **only the first** $\frac{n}{2}$ error lists and reliably sends them to **S** using three phase PRMT protocol $OPRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$.
2. There exists at least one $J \in \{1, 2, \ldots, n\}$, such that $RS-DEC$, when applied to C'_J, fails to output any $t_b + t_p$ degree polynomial. In this case, **R** broadcasts C'_J and its index J.

Notice that the technique proposed in section 5.3 for sending n error lists in a single phase incurring $O(n^2)$ communication complexity, can not be adopted against mobile adversary. This is so because the technique used the knowledge of the Byzantine corruption done in earlier phases. However, mobile adversary can corrupt different set of wires in different phases. So, here we use the three phase reliable protocol $OPRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$ to send the error lists in three phases with same communication complexity of $O(n^2)$. Also note that

Execution I: Remaining Execution of $OPSMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$, when step 2 of Phase II has been executed

Phase III: S to R

- **S** correctly receives index J and codeword C'_J. After locally comparing C'_J with its corresponding actual codeword C_J, **S** identifies at least $\frac{t_b}{2}+1$ wires which delivered incorrect values to **R** during **Phase I**. **S** saves the identity of these wires in a list L_{fault} and broadcasts L_{fault} to **R**.
- **S** also lists all c_{ji}'s, $j \in \{1,2,\ldots,n\}-\{J\}$, sent during **Phase I**, over $w_i \in L_{fault}$. **S** then re-sends these $(n-1)\times|L_{fault}| = O(nt_b)$ values by executing the three phase PRMT protocol $OPRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$. This will occupy the next three phases. /* The re-send values are already known to $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ because the wires in L_{fault} were under the control of $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ during **Phase I**. */

Local Computation by R (At the end of Phase V)

• After receiving list L_{fault}, **R** identifies $|L_{fault}| > \frac{t_b}{2}$ wires which has delivered incorrect information during **Phase I**. **R** removes from the $n-1$ codewords C'_j's, $j \in \{1,2,\ldots,n\}-\{J\}$ (received during **Phase I**), the values c'_{ji}'s, which **R** has received along $w_i \in L_{fault}$ during **Phase I**. **R** replaces them with the corresponding actual c_{ji}'s, which **S** has re-send through PRMT protocol $PRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$.

• After replacement, **R** knows that out of the n' values in each $C'_j, j \in \{1,2,\ldots,n\}-\{J\}$, at most $t_b-|L_{fault}|$ could be corrupted. **R** applies $RS-DEC(n', t_b-|L_{fault}|, 0, t_b+t_p+1)$ algorithm to these $n-1$ C'_j's and correctly recovers $p_j(x)$'s. The constant term of these $n-1$ polynomials constitute an $n-1$ length information theoretically secure pad established between **S** and **R** and the protocol terminates here.

while executing $OPRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$, **S** and **R** can neglect a pre-determined set of t_p wires and run the protocol on the remaining $2t_b+t_f+1$ wires (the PRMT protocol requires only $2t_b+t_f+1$ wires between **S** and **R**). This does not affects the correctness and working of the protocol.

Theorem 14 *Protocol $OPSMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$ correctly and securely establishes a one time pad of length $\Theta(n)$ between* **S** *and* **R** *in at most nine phases by communicating $O(n^2)$ field elements tolerating $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$.*

Proof: We prove the theorem for the worst case where exactly t_f wires (probably different set) failed to deliver any information in each phase due to fail-stop corruption. Thus each codeword C'_j received during first phase will be of length $n'=n-t_f=2t_b+t_p+1$, which are RS encoded using a polynomial of degree $k-1=t_b+t_p$. Consider the following two cases:

Case I: At most $\frac{t_b}{2}$ wires are Byzantine Corrupted During Phase I: In this case, the proof is same as the proof of **Case I** in Theorem 7. The only difference is instead of an n length pad, an $\frac{n}{2}$ length pad, consisting of the constant coefficients of the first $\frac{n}{2}$ polynomials will be established between **S** and **R**. Also since $\frac{n}{2}$ error lists are sent using $OPRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$, protocol takes extra two phases and terminates in **Phase V**.

Case II: More than $\frac{t_b}{2}$ wires are Byzantine Corrupted During Phase I: Now effect of more than $\frac{t_b}{2}$ Byzantine corruption during **Phase I** can be categorized into two cases. (1) $RS-DEC$ outputs some polynomial of degree t_b+t_p for each C'_j (2) There exists a $J \in \{1,2,\ldots,n\}$ for which $RS-DEC$ fails to output any polynomial. While in (1), occurrence of more than $\frac{t_b}{2}$ faults cannot be immediately detected (as $RS-DEC$ is applied with $d=0$), in (2) it is immediately detected. Now again in (1), if more than $\frac{t_b}{2}$ Byzantine errors occurs in the codewords of *only* last $\frac{n}{2}$ polynomials i.e for $p_j(x)$ such that $\frac{n}{2}+1 \le j \le n$ (this implies that at most $\frac{t_b}{2}$ Byzantine errors took place in the first $\frac{n}{2}$ codewords), then the

Execution II: Remaining Execution of $OPSMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$, when step 1 of Phase II has been executed

/* **R** has initiated three phase $OPRMT - \Pi^{mobile}_{(t_b,t_f,t_p)}$ protocol to reliably send first $\frac{n}{2}$ error lists during **Phase II**. The $OPRMT$ protocol will be over at the end of **Phase IV**.*/

Local Computation by S (At the end of Phase IV)

S reliably receives first $\frac{n}{2}$ error lists through $OPRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$ and locally checks the status of these error lists.

- If all error lists are "good", then **S** concludes that **R** has correctly recovered $p_j(x), 1 \leq j \leq \frac{n}{2}$ correctly and an information theoretically secure pad $[p_1(0)\ p_2(0)\ p_{n/2}(0)]$ is established with **R**. **S** terminates the protocol by broadcasting terminating signal to **R**. Accordingly **R** terminates the protocol.
- If $\exists J \in \{1, 2, \ldots, \frac{n}{2}\}$, such that $Error\_List_J$ is "bad", then **S** concludes that more than $\frac{t_b}{2}$ values has been changed in J^{th} codeword during **Phase I**.

Phase V: S to R (If second case happens in the above computation) **S** asks **R** to broadcast the J^{th} codeword as received by **R** during **Phase I**. **S** does this by broadcasting index J along with "ERROR" signal.

Phase VI: R to S

On receiving "ERROR" signal and index J during **Phase V**, **R** broadcasts C'_J, received during **Phase I**.

Phase VII: S to R

On receiving C'_J, **S** identifies more than $\frac{t_b}{2}$ wires which were Byzantine corrupted during **Phase I** and saves them in a list L_{fault}. From here onwards the execution is similar as in **Execution I**. We specify only the small differences:

- If $w_i \in L_{fault}$, then **S** lists the i^{th} component of the codewords corresponding to the last $\frac{n}{2}$ polynomials $p_j(x), n/2+1 \leq j \leq n$. **S** reliably re-sends these components by executing $OPRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$. Recall that in this execution sequence, **R** had not sent the last $\frac{n}{2}$ error lists during **Phase II** (step 1). The re-send values are already known to the adversary and does not give any extra information about $p_j(x), n/2 + 1 \leq j \leq n$.
- $OPRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$ terminates in **Phase IX** (since it takes 3 phases) and therefore at the end of **phase IX**, **R** performs the same local computation as done in **Execution I** to correctly recover the polynomials $p_j(x), n/2 + 1 \leq j \leq n$ to establish a pad of size $\frac{n}{2}$. The $\frac{n}{2}$ size pad constitutes the constant term of the recovered polynomials $p_j(x), n/2 + 1 \leq j \leq n$.

proof is same as in **Case I**. On the other hand, if more than $\frac{t_b}{2}$ faults occurs for J^{th} codeword, where $J \in \{1, 2, \ldots, \frac{n}{2}\}$, the proof is given below.

1. **During Phase II, R reconstructs** $\bar{p}_J(x) \neq p_J(x),\ J \in \{1, 2, \ldots, \frac{n}{2}\}$: In this case, $Error - List_J$ is a "bad" error list which contain at least one correct value of original $p_J(x)$. Since **R** reliably sends back first $\frac{n}{2}$ error lists using $OPRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$, **S** correctly receives $Error - List_J$ and finds that it is "bad", implying that **R** has reconstructed some $\bar{p}_J(x) \neq p_J(x)$. So **S** asks **R** to broadcast the J^{th} codeword C'_J, as received during **Phase I**. On receiving C'_J, **S** compares it with its corresponding original codeword C_J and identifies $|L_{fault}| \geq \frac{t_b}{2} + 1$ wires which delivered incorrect values to **R** during **Phase I**. Now by executing $OPRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$, **S** re-sends the components of the last $\frac{n}{2}$ codewords, which were sent through these corrupted wires in **Phase I**. **S** also broadcasts the identity of these corrupted wires. *Note that re-sending these values, does not leak any additional information about the last $\frac{n}{2}$ $p_j(x)$'s to $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ because adversary already came to know these values during* **Phase I**. But now with the new values received, **R** have $n' = 2t_b + t_p + 1$ components for each of the last $\frac{n}{2}$ codewords and at

most $t_b - |L_{fault}| \leq \frac{t_b}{2} - 1$ of these n' components could be corrupted. By substituting $d = 0$ and values of n' and $k = t_b + t_p + 1$ in the inequality of Theorem 2, we find that $RS - DEC(n', t_b - |L_{fault}|, 0, t_b + t_p + 1)$, when applied to last $\frac{n}{2}$ codewords, can correct $c \leq \frac{t_b}{2}$ errors present in them and correctly outputs the corresponding polynomial $p_j(x)$. **R** then considers the constant term of these last $\frac{n}{2}$ $p_j(x)$'s as the secret pad established with **S**. The secrecy of the pad follows from the fact that at any stage of the execution, $\mathcal{A}^{mobile}_{(t_b,t_f,t_p)}$ will not get more than $t_b + t_p$ points on the last $\frac{n}{2}$ $p_j(x)$'s, each of which are of degree $t_b + t_p$.

2. **During Phase II, R is Unable to Recover** $\bar{p}_J(x)$: In this case **R** broadcasts only the J^{th} received codeword C'_J, from which **S** (after local verification) identifies at least $\frac{t_b}{2} + 1$ wires, which delivered incorrect values to **R** during **Phase I**. Now the rest of the proof is same as in the above case. The only difference is, here a pad of length $n - 1$ will be established between **S** and **R**.

From Theorem 13, re-sending $O(nt_b)$ values by executing $OPRMT\_\Pi^{mobile}_{(t_b,t_f,t_p)}$ protocol requires communicating $O(n^2)$ field elements. Also no more than $O(n^2)$ field elements are communicated in any other phase. Hence the overall communication complexity is $O(n^2)$. □

8 Conclusion

In this paper we have contributed significantly to the progress of the state of the art in the problem of PRMT and PSMT. We presented a number of constant phase protocols which are first of their kind and enjoys the property of being communication optimal against static and mobile mixed adversary . To design the protocols, we proposed several new techniques, which can be effectively used against both static and mobile mixed adversary. One can try to design optimal PRMT and PSMT protocols tolerating static and mobile mixed adversary with lesser number of phases than the ones we presented here or may try to prove exact phase complexity of the optimal protocols.

References

[1] S. Agarwal, R. Cramer, and R. de Haan. Asymptotically optimal two-round perfectly secure message transmission. In C. Dwork, editor, *Proc. of Advances in Cryptology: CRYPTO 2006*, LNCS 4117, pages 394–408. Springer-Verlag, 2006.

[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] T. H. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 2004.

[4] R. Cramer, I. Damgård, S. Dziembowski, M. Hirt, and T. Rabin. Efficient multiparty computations secure against an adaptive adversary. In *Proc. of EUROCRYPT 1999*, volume 1592 of *LNCS*, pages 311–326. Springer Verlag, 1999.

[5] 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.

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

[7] M. Franklin and R. Wright. Secure communication in minimal connectivity models. *Journal of Cryptology*, 13(1):9–30, 2000.

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

[9] 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.

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

[11] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. North-Holland Publishing Company, 1978.

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

[13] C. Papadimitriou and U. Vazirani. Discrete mathematics for computer science. Lecture Notes.

[14] A. Patra, A. Choudhary, M. Gayatri, and C. Pandu Rangan. Efficient perfectly reliable and secure communication tolerating mobile adversary. Cryptology ePrint Archive, Report 2008/086, 2008. http://eprint.iacr.org/.

[15] 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.

[16] 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.

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

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

[19] 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.

[20] K. Srinathan, A. Patra, A. Choudhary, and C. Pandu Rangan. Probabilistic perfectly reliable and secure message transmission - possibility, feasibility and optimality. In *INDOCRYPT*, pages 101–122, 2007.

[21] K. Srinathan, N. R. Prasad, and C. Pandu Rangan. On the optimal communication complexity of multiphase protocols for perfect communication. In *IEEE Symposium on Security and Privacy*, pages 311–320, 2007.

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

[23] AshwinKumar B. V, Arpita Patra, Ashish Choudhary, K. Srinathan, and C. Pandu Rangan. On phase complexity of perfectly reliable communication tolerating adaptive mixed adversary. Manuscript.

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