Unconditionally Reliable and Secure Message Transmission in Undirected Synchronous Networks: Possibility, Feasibility and Optimality*

Arpita Patra[1] Ashish Choudhary[1†] Kannan Srinathan [2] C. Pandu Rangan[1]

[1] Department of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai India 600036
Email: {arpita, ashishc}@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

We study the interplay of network connectivity and the issues related to the possibility, feasibility and optimality for *unconditionally reliable message transmission* (URMT) and *unconditionally secure message transmission* (USMT) in an undirected *synchronous* network, under the influence of an adaptive *mixed* adversary $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$, who has *unbounded computing power* and can corrupt t_b, t_o, t_f and t_p nodes in the network in Byzantine, *omission*, *fail-stop* and passive fashion respectively. In URMT problem, a sender $\mathbf{S}$ and a receiver $\mathbf{R}$ are part of a distributed network, where $\mathbf{S}$ and $\mathbf{R}$ are connected by intermediate nodes, of which at most t_b, t_o, t_f and t_p nodes can be under the control of $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$. $\mathbf{S}$ wants to send a message m which is a sequence of ℓ field elements from a finite field $\mathbb{F}$ to $\mathbf{R}$. The challenge is to design a protocol, such that after interacting in phases[1] as per the protocol, $\mathbf{R}$ should output $m' = m$ with probability at least $1 - \delta$, where $0 < \delta < \frac{1}{2}$. Moreover, this should happen, irrespective of the adversary strategy of $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$. The USMT problem has an additional requirement that $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ should not know anything about m in *information theoretic* sense.

In this paper, we answer the following in context of URMT and USMT: (a) Possibility: when is a protocol possible in a given network? (b) Feasibility: Once the existence of a protocol is ensured then does there exists a polynomial time protocol on the given network? (c) Optimality: Given a message of specific length, what is the minimum communication complexity (lower bound) needed by any protocol to transmit the message and how to design a protocol whose total communication complexity matches the lower bound on the communication complexity? Finally we also show that *allowing a negligible error probability significantly helps in the possibility, feasibility and optimality of both reliable and secure message transmission protocols.* To design our protocols, we propose several new techniques which are of independent interest.

Keywords: Probabilistic Reliability, Information Theoretic Security, Mixed Adversary.

1 Introduction

Achieving reliable and secure communication is a fundamental problem in the theory of communication. In modern applied network security, there is a lot of emphasis on the use of virtual private networks (using

*A preliminary version of this paper appeared in INDOCRYPT 2007.

†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 $\mathbf{S}$ to $\mathbf{R}$ or viceversa.

cryptography), firewalls, virus scanners, etc. However, routers too are vulnerable [40]. Two problems have been identified if routers are hacked. The hacker can shut down these nodes or forward incorrect information [8, 18]. An important problem solved by routers is to find the network graph. Hacked routers can disrupt this by claiming non-existent nodes as part of the network graph. Hence there is a need for considering an adversary who can disrupt the network in variety of ways.

The problem of Byzantine nodes disrupting communications was studied in a broader context by Dolev-Dwork-Waarts-Yung [8] for the first time, by adding the issue of privacy. The research found its origin from secure multiparty computation. Up to that time, one assumed a complete graph for private and robust (reliable) communication [2, 16, 29, 39]. The authors in [8] called the above problems as *perfectly reliable message transmission* (PRMT) and *perfectly secure message transmission* (PSMT). In the problem of *perfectly reliable message transmission* (PRMT), 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 (with zero error probability), in spite of the presence of several kinds of faults in the network. The problem of *perfectly secure message transmission* (PSMT) has an additional constraint that the adversary should get *no* information about m in *information theoretic sense*. 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*.

The PRMT and PSMT are well-motivated problems for it being one of the fundamental primitives used by all fault-tolerant distributed algorithms like Byzantine agreement [23, 22, 9, 10], multiparty computation [39, 3, 2, 29] etc. All these popular fault-tolerant distributed algorithms assume that the underlying network is a complete graph, thereby implicitly assuming the existence of a PRMT protocol that can simulate a complete graph (by filling up the missing links in given incomplete graph) overlayed in the actual network which is seldom a complete graph itself. There is another motivation to study PSMT problem. Currently, all existing public key cryptosystems, digital signature schemes are based on the hardness assumptions of certain number theoretic problems. With the advent of new computing paradigms, such as quantum computing and increase in computing speed, may render these assumptions baseless. In that case, one has to look for information theoretically secure message transmission schemes.

There are various settings in which PRMT and PSMT problem has been studied extensively in the past. For example, the underlying network model may be undirected [8, 27, 1] graph, directed [28, 7] graph or hypergraph [14]. The communication in the network could be synchronous [8, 31] or asynchronous [30]. The faults could be passive, fail-stop, Byzantine or sometimes mixed/hybrid faults [15]. The number of faulty nodes may be bounded by a fixed constant (threshold adversary) [8, 31] or the potential sets of faulty nodes may be described by a collection of subsets of nodes (non-threshold adversary) [19], while the adversary may be mobile [26] or adaptive [8, 31]. The taxonomy of settings in which PRMT and PSMT can be studied are listed in Table 1. Any PRMT/PSMT protocol is analyzed by the following

Underlying Network Model	Adversary Capacity	Adversary Behavior
Undirected Graph	*Threshold Adaptive*	*Byzantine*
Directed Graph	*Threshold Mobile*	*Fail-Stop*
Undirected Hypergraph	*Non-Threshold Adaptive*	*Passive*
Directed Hypergraph	*Non-Threshold Mobile*	*Mixed*

Table 1: The taxonomy of the settings in which PRMT/PSMT can be studied.

parameters: (a) connectivity of the underlying network (b) number of phases taken by the protocol, where a phase is a communication from **S** to **R** or vice-versa (c) communication complexity, which is the total number of field elements communicated by **S** and **R** in the protocol (d) amount of computation done by **S** and **R** in the protocol. Irrespective of the settings in which PRMT and PSMT is studied, the following issues are common:

(i) POSSIBILITY: When is a protocol possible in the given network?

(ii) FEASIBILITY: Once the existence of a protocol is ensured then does there exists a polynomial time protocol on the given network?

(iii) OPTIMALITY: Given a message of specific length, what is the minimum communication complexity (lower bound) needed by any protocol to transmit the message and how to design a protocol whose total communication complexity matches the lower bound on the communication complexity?

The issues (a), (b) and (c) has been completely resolved for certain settings. For certain network settings, these issues has been partly resolved where as for certain settings, nothing is known. For example, the issues (a), (b) and (c) for PRMT in undirected synchronous networks tolerating threshold adaptive Byzantine adversary in solved in [8, 27]. Similarly, issues (a), (b), (c) for PSMT in undirected synchronous networks tolerating threshold adaptive Byzantine adversary in solved in [8, 27, 34, 36, 11, 20]. Desmedt et.al [7] and Arpita et.al [28] has studied the issues related to the possibility and feasibility of PSMT protocols in directed networks tolerating threshold adaptive Byzantine adversary. On the other hand, nobody has addressed the issues (a), (b) and (c) for PSMT in arbitrary directed hypergraphs mobile mixed adversary. Note that the techniques used to address (a), (b) and (c) in one setting cannot be directly adapted to address the same issues in other settings. For example, the techniques used to design feasible PSMT protocols in directed networks [28] are very different from the one which are used to design PSMT protocols in undirected networks [27].

It's a well-renowned fact that in numerous situations *randomization* helps unbelievably in making the life simple. The testimonials ranges from famous number theoretic randomized primality testing algorithms to various distributed computation tasks like verifiable secret sharing (VSS) [29], multiparty computation [6] to name a few. In this work, we focus on to expose the effect of randomization on PRMT and PSMT problems. We name the probabilistic PRMT and PSMT as *unconditionally reliable message transmission* (URMT) and *unconditionally secure message transmission* (USMT) respectively. The problem of URMT is identical to the problem of PRMT except that $\mathbf{R}$ should correctly receive $\mathbf{S}$'s message with probability at least $1-\delta$ (for any $0 < \delta < 1/2$), instead of probability 1, as in case of PRMT. In USMT, in addition to the conditions of URMT, $\mathbf{S}$'s message must be hidden *information theoretically* from the adversary. The difference between PRMT, URMT, PSMT and USMT is summarized in Table 2.

	Probability of Error in Reliability (δ)	Probability of Error in Security (ϵ)
PRMT	0	No issue of security
URMT	$0 < \delta < 1/2$	No issue of security
PSMT	0	0
USMT	$0 < \delta < 1/2$	0

Table 2: Difference between the different terminologies used in the paper

Intuitively, the allowance of a small probability of error in the transmission (only in the reliability) should result in improvements in both the fault tolerance as well as the efficiency aspects of reliable and secure protocols. What exactly is the improvement? — this is the central question addressed in this paper. More specifically, in this paper, we address issues related to POSSIBILITY, FEASIBILITY and OPTIMALITY in the context of URMT and USMT. Now as in the case of PRMT and PSMT, URMT and USMT can also be studied in various network settings and adversary model. In this paper, we completely resolve issues (a), (b) and (c) in the context of URMT and USMT, in undirected synchronous network, tolerating an adaptive threshold mixed adversary $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$.

why to study mixed 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 fail-stop(/eavesdrop in) a strongly protected node, while he may affect in a Byzantine fashion a weakly

protected node. Thus, we may capture the abilities of an adversary in a more realistic manner using four parameters t_b, t_o, t_f, t_p where t_b, t_o, t_f, t_p are the number of nodes under the influence of the adversary in Byzantine, omission, failstop and passive adversary, respectively (for more formal definition see Section 2). 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".

Comparing our results with the existing results for PRMT and PSMT in undirected networks show that randomness and probabilistic approaches lead to improved fault tolerance, communication, phase and computational complexities.

1.1 Existing Literature

We first recall the existing results for PRMT and PSMT in undirected synchronous networks tolerating threshold Byzantine and mixed adversary in Table 3 and Table 4.

Table 3: Connectivity Requirement and Lower Bounds for PRMT and PSMT in Undirected Networks. r denotes number of phases and ℓ denotes the message size in terms of field elements

Model	Connectivity Requirement between **S** and **R** (n)	Lower Bound on Communication Complexity
PRMT(Byzantine Adversary)	$n \geq 2t_b + 1, \forall r \geq 1$ [8]	$\Omega(\frac{n\ell}{n-2t_b})$ for $r = 1, 2$ [34] $\Omega(\frac{n\ell}{n-t_b})$ for $r \geq 3$[36]
PSMT(Byzantine Adversary)	$n \geq 3t_b + 1$ for $r = 1$ [8] $n \geq 2t_b + 1$ for $r \geq 2$ [8]	$\Omega(\frac{n\ell}{n-3t_b})$ for $r = 1$ [11] $\Omega(\frac{n\ell}{n-2t_b})$ for $r \geq 2$ [36]
PRMT(Mixed Adversary)	$n \geq 2t_b + t_o + t_f + 1, \forall r \geq 1$ [33]	$\Omega\left(\frac{n\ell}{n-(2t_b+t_o+t_f)}\right)$ for $r = 1, 2$[33] $\Omega\left(\frac{(n-t_f-t_o)\ell}{n-(t_b+t_o+t_f)}\right)$ for $r \geq 3$[33]
PSMT(Mixed Adversary)	$n \geq 3t_b + 2t_o + t_f + t_p + 1$ for $r = 1$ [33] $n \geq 2t_b + t_o + t_f + t_p + 1$ for $r \geq 2$ [27]	$\Omega(\frac{n\ell}{n-(3t_b+2t_0+t_f+t_p)})$ for $r = 1$ [33] $\Omega(\frac{n\ell}{n-(2t_b+t_0+t_f+t_p)})$ for $r \geq 2$ [33]

The problem of URMT and USMT in undirected synchronous networks in the presence of threshold adaptive[2] Byzantine adversary was first defined and solved by Franklin *et al* [12]. As one of the key results, they have proved, that over undirected graphs, URMT (USMT) is possible if and only if PRMT (PSMT) is possible! Subsequent works on URMT and USMT include [13, 38, 7]. However, all these results try to address the issue of POSSIBILITY and FEASIBILITY of URMT and USMT protocols and that too only in the presence of threshold Byzantine adversary. In [21], Kurosawa et.al have addressed the issue of OPTIMALITY of USMT protocols in undirected networks tolerating threshold Byzantine adversary. Most recently, Srinathan et.al [35] and Shankar et.al [32] have given the characterization for the POSSIBILITY of URMT in arbitrary directed graphs tolerating non-threshold and threshold Byzantine adversary respectively. However, as far as our knowledge is concerned, no body has ever simultaneously addressed the issue of POSSIBILITY, FEASIBILITY and OPTIMALITY of URMT and USMT protocols in any network model tolerating threshold mixed adversary.

1.2 Our Contribution

As mentioned earlier, any reliable/secure protocol is analyzed by the the connectivity requirement of the network, the number of phases required by the protocol, the total number of field elements communicated by **S** and **R** throughout the protocol and the computation done by **S** and **R**. The *trade-offs* among these parameter are well studied in the literature in the context of PRMT and PSMT in undirected synchronous

[2]By adaptive adversary, we mean an adversary that decides on the set of players to corrupt on the fly during the protocol execution.

Table 4: Protocols with Optimum Communication Complexity. ℓ is the message size in terms of field elements and n is the corresponding connectivity requirement from Table 3

Model	Communication Complexity in Terms of Field Elements	Number of Phases	Remarks
PRMT (Byzantine Adversary)	$O(\frac{n\ell}{n-2t_b})$	≤ 2	$\ell \geq n$ [34].
	$O(\frac{n\ell}{n-t_b})$	3	$\ell \geq n^2$ [27].
PSMT (Byzantine Adversary)	$O\left(\frac{n\ell}{n-3t_b}\right)$	1	$\ell \geq n$ [11].
	$O\left(\frac{n\ell}{n-2t_b}\right)$	2	• Exponential computation and communication complexity [1]. Polynomial computation and communication complexity [20]
	$O\left(\frac{n\ell}{n-2t_b}\right)$	3	• Polynomial computation and communication complexity [27].
PRMT (Mixed Adversary)	$O\left(\frac{n\ell}{n-(2t_b+t_o+t_f)}\right)$	1	$\ell \geq n$ [33].
	$O\left(\frac{(n-t_f-t_o)\ell}{n-(t_b+t_o+t_f)}\right)$	$\log\left(\frac{t_f+t_o}{n-(t_f+t_o)}\right)$	$\ell \geq n^2$ [37].
PSMT (Mixed Adversary)	$O\left(\frac{n\ell}{n-(2t_b+t_o+t_f+t_p)}\right)$	4	$\ell \geq n$ [4]

network tolerating threshold Byzantine adversary [27, 36, 1, 20]. In this paper, we investigate the trade-off for URMT and USMT in the presence of threshold adaptive *mixed* adversary, which is to our knowledge, the *first* attempt in the literature of URMT and USMT.

So we provide characterization, lower bound on communication complexity and protocols that matches the lower bound for URMT and USMT. In some of the cases the protocols presented here, achieves their task in exactly optimal number of phases (the minimum number of phases that is required to achieve the communication complexity lower bound for a specific message size). Such protocols are simultaneously communication and phase optimal. In summary, for URMT we show the following:

- URMT between **S** and **R** tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ is possible iff the network is $(2t_b + t_o + t_f + 1)$-(**S**,**R**)-connected.
- Any single phase URMT protocol tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$, from **S** to **R** over $n \geq 2t_b + t_o + t_f + 1$ wires communicates $\Omega(\frac{n\ell}{n-(t_b+t_o+t_f)})$ field elements to reliably transmit (with high probability) ℓ field elements.
- Any multiphase URMT protocol, from **S** to **R** over $n \geq 2t_b + t_o + t_f + 1$ wires communicates $\Omega(\ell)$ field elements to reliably transmit (with high probability) ℓ field elements.

We also design *polynomial* time *communication optimal and phase optimal* single phase URMT protocol whose communication complexity satisfy our proven lower bound. As a corollary, we show that our *single* phase URMT protocol has a *special* property that it achieves reliability with *constant factor* overhead (i.e. sending ℓ field elements by communicating $O(\ell)$ field elements) when executed *only* under the presence of Byzantine adversary (i.e., $t_o = t_f = t_p = 0$). An $O(\log \frac{t_f+t_o}{n-t_f-t_o})$ phase PRMT protocol which sends ℓ field elements by communicating $O(\ell)$ field elements is presented in [37]. The protocol of [37] is also a valid multiphase URMT protocol satisfying the communication complexity lower bound for multiphase URMT. Design of a bit optimal multiphase URMT protocol with lesser number of phases is left as an open problem.

For USMT problem, we show the following:

- Any single phase USMT protocol that achieves perfect secrecy (with negligible error probability of

$\delta > 0$ in **reliability**), tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ is possible iff there exists $n \geq 2t_b + 2t_o + t_f + t_p + 1$ vertex disjoint paths between **S** and **R**.

- Any single phase USMT protocol over $n \geq 2t_b + 2t_o + t_f + t_p + 1$ vertex disjoint paths between **S** and **R**, tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$, must communicate $\Omega(\frac{n\ell}{n-(2t_b+2t_o+t_f+t_p)})$ field elements in order to securely send an ℓ-field element message with very high probability.
- Multiphase USMT between **S** and **R** in an undirected network tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ is possible if and only if the network is $(t_b + \max(t_b, t_p) + t_o + t_f + 1)$-(**S,R**)-connected.
- Any r-phase ($r \geq 2$) USMT protocol which securely sends ℓ field elements in the presence of $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ needs to communicate $\Omega\left(\frac{n\ell}{n-(t_b+t_o+t_f+t_p)}\right)$ field elements, where **S** and **R** are connected by $n \geq (t_b + \max(t_b, t_p) + t_o + t_f + 1)$ vertex disjoint paths.

We also design *polynomial* time *communication optimal* single phase and four phase USMT protocols whose communication complexity satisfies our proven lower bounds for single phase and multiphase USMT respectively, thus showing that our bounds are tight. So the single phase USMT is both communication and phase optimal. Similarly our *four* phase USMT protocol against $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ has a *special* property that it achieves *secrecy* with *constant factor* overhead (sending ℓ field elements by communicating $O(\ell)$ field elements) when executed *only* under the presence of Byzantine adversary (i.e. $t_o = t_f = t_p = 0$). However, against only Byzantine adversary, USMT with constant factor overhead in communication complexity can be achieved in two phases itself. One such protocol is also presented in this paper. So, the results on URMT and USMT is tabulated in Table 5 and Table 6.

Remark 1 *In any URMT and USMT protocol, the communication complexity should be a function of δ which is the error probability of the protocol. However, in the results summarized in Table 5 and Table 6, δ is not appearing explicitly in the communication complexity expressions. The reason is that the communication complexity expressions are given in terms of field elements. This is done for the ease of comparing the communication complexities of URMT and USMT protocols with the communication complexities of PRMT and PSMT protocols (in terms of field elements). In any URMT and USMT protocol, the field size is always a function of δ (as illustrated in our protocols). So though the communication complexity expressions in Table 5 and Table 6 does not contain δ explicitly, they are actually function of δ.*

Table 5: Connectivity Requirement and Lower Bound on Communication Complexity for URMT and USMT. r denotes number of phases and ℓ is the message size in terms of field elements. All the $*$ marked results are presented in this paper.

Model	Connectivity (n)	Lower Bounds
URMT(Byzantine Adversary)	$n \geq 2t_b + 1, \forall r \geq 1$ [12]	$\Omega(\frac{n\ell}{n-t_b})$ for $r = 1$ $*$
USMT(Byzantine Adversary)	$n \geq 2t_b + 1, \forall r \geq 1$ [12]	$\Omega(\frac{n\ell}{n-2t_b})$ for $r = 1$ $*$ $\Omega(\frac{n\ell}{n-t_b})$ for $r \geq 2$ $*$
URMT(Mixed Adversary)	$n \geq 2t_b + t_o + t_f + 1, \forall r \geq 1$ $*$	$\Omega\left(\frac{n\ell}{n-(t_b+t_o+t_f)}\right)$ for $r = 1$ $*$ $\Omega(\ell)$ for $r \geq 2$ $*$
USMT(Mixed Adversary)	$n \geq 2t_b + 2t_o + t_f + t_p + 1$ for $r = 1$ $*$ $n \geq t_b + \max(t_b, t_p) + t_o + t_f + 1$ for $r \geq 2$ $*$	$\Omega(\frac{n\ell}{n-(2t_b+2t_0+t_f+t_p)})$ for $r = 1$ $*$ $\Omega(\frac{n\ell}{n-(t_b+t_0+t_f+t_p)})$ for $r \geq 2$ $*$

Now, comparing Table 3 with Table 5 and Table 4 with Table 6, we find that allowing a negligible error probability has tremendous effect on reliable and secure message transmission in terms of POSSIBILITY, FEASIBILITY and OPTIMALITY. We show many practical scenarios where no optimal PRMT or PSMT protocol exist but optimal URMT and USMT protocol do exists, thus showing the power of allowing negligible error probability in the reliability of the protocols (without sacrificing secrecy).

Table 6: Protocols with Optimum Communication Complexity. ℓ is the message size in terms of field elements. n denotes respective connectivity requirement specified in Table 5. All the $*$ marked results are presented in this paper.

Model	Communication Complexity	Number of Phases	Remarks
URMT (Byzantine Adversary)	$O(\frac{n\ell}{n-t_b})$	1	$\ell \geq n^2$ *.
USMT (Byzantine Adversary)	$O\left(\frac{n\ell}{n-2t_b}\right)$	1	$\ell \geq n$ *
	$O(\ell)$	2	$\ell \geq n^2$ *
PRMT (Mixed Adversary)	$O\left(\frac{n\ell}{n-(t_b+t_o+t_f)}\right)$	1	$\ell \geq n(t_b+1)$ *
	$O(\ell)$	$O\left(\log\left(\frac{t_f+t_o}{n-(t_f+t_o)}\right)\right)$	$\ell \geq n^2$ [37].
USMT (Mixed Adversary)	$O\left(\frac{n\ell}{n-(2t_b+2t_o+t_f+t_p)}\right)$	1	$\ell \geq n$ *
	$O\left(\frac{n\ell}{n-(t_b+t_o+t_f+t_p)}\right)$	4	$\ell = n^2$ if $t_p \geq t_b$ or $\ell = (t_b - t_p)n^2$ if $t_b > t_p$ *

1.3 Techniques Used

The techniques used for designing PRMT and PSMT protocols are completely different from the techniques used for designing URMT and USMT protocols. The existing URMT and USMT protocols [12, 7] use the idea of *information theoretic* authentication schemes and check vectors along with error correcting codes. The check vectors are introduced in [29] for information checking (IC) protocols, which are used to generate IC signatures. The IC signatures can be used as a semi digital signature[6, 29]. Using these ideas, one can design FEASIBLE URMT and USMT protocols in undirected networks tolerating mixed adversary. However, the resultant protocols will be cumbersome and will not be COMMUNICATION OPTIMAL against mixed adversary. To design optimal protocols against mixed adversary, we introduce a new technique, called **Extrapolation Technique**. Using **Extrapolation Technique**, we can design communication optimal URMT protocol against mixed adversary. By using a slight variant of **Extrapolation Technique**, we can also design communication optimal USMT protocol tolerating mixed adversary. The **Extrapolation Technique** is first of its kind and is of independent interest.

1.4 Organization of the Paper

This paper is mainly divided into four main sections, namely on single phase URMT, multiphase URMT, single phase USMT and multiphase USMT. A vivid description has been penned down listing the tasks that can be realized in *probabilistic* scenarios (URMT/USMT) but can't be achieved by their *perfect* counterpart (PRMT/PSMT) and the tasks that are significantly improved by the application of probabilistic approaches (URMT/USMT) (as compared to PRMT/PSMT) at the end of each section.

2 Network Model, Adversary Model and Definitions

The underlying network is a connected synchronous network represented by an undirected graph where **S** and **R** are two nodes of the graph. A *mixed adversary* $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$, with *unbounded* computing power controls t_b, t_o, t_f and t_p nodes in the graph (excluding **S, R**) in Byzantine, omission, fail-stop and passive fashion respectively.

Definition 1 ([17]) FAILSTOP CORRUPTION*: A node P is said to be fail-stop corrupted if the adversary can crash P at will at any time during the execution of the protocol. But as long as P is alive, it will honestly follow the protocol and the adversary will have no access to any information or internal state of P. Once P is crashed, then it will remain inactive for the rest of the protocol.*

Definition 2 ([17]) OMISSION CORRUPTION*: We say that a node P is omission corrupted, if the adversary can block the working of P at will at any time during the execution of the protocol. But as long as P is alive, it will follow the instructions of the protocol honestly. The adversary can eavesdrop the internal data/computation of P but cannot make P to deviate from the proper execution of the protocol. Once P is blocked, it can again become alive at some later stage of the protocol and start following the protocol honestly.*

Definition 3 PASSIVE CORRUPTION*: A node P is said to be passively corrupted if it honestly follows the protocol. But the adversary will have full access to any information or internal state of P.*

Definition 4 BYZANTINE CORRUPTION*: A node P is said to be Byzantine corrupted if the adversary fully control the actions of P. The adversary will have full access to the computation and communication of P and can force P to deviate from the protocol and behave arbitrarily.*

The fail-stop error models a hardware failure caused by any natural calamity or manual shutdown. Also the nodes which are fail-stop corrupted cannot be passively listened by the adversary. On the other hand, nodes corrupted in omission fashion can be eavesdropped by the adversary. Thus omission error can be considered as a combination of fail-stop and passive corruption with the exception that unlike fail-stop error, a node which is crashed once due omission error may become alive during later stages of the protocol. Note that though omission adversary has eavesdropping capability, it also has blocking capability. Thus it is stronger than passive and failstop corruption. But it weaker than Byzantine corruption. Thus, the adversary is a mixed adversary, represented as $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$. Note that a Byzantine adversary is a special type of mixed adversary where $t_o = t_f = t_p = 0$. Since Byzantine and omission corrupted nodes can also be eavesdropped, the maximum number of nodes which can be eavesdropped by the adversary is bounded by $t_b + t_o + t_p$. We assume that the adversary is a *centralized* adversary and can collectively pool the data from the nodes under its control and use it according to his own choice in any manner. The adversary is adaptive [6]. Thus he is allowed to *dynamically* corrupt nodes during the protocol execution. Moreover, his choice of nodes (to corrupt) may depend on the data seen so far. However, the total number of nodes that can be under the control of the adversary throughout the protocol is bounded by the threshold. Also one a node is under the control of the adversary in some fashion, then it will remain so throughout the protocol.

Following the approach of Dolev et. al. [8], we abstract away the network and concentrate on solving URMT and USMT problem for a single pair of processors, the *sender* $\mathbf{S}$ and the *receiver* $\mathbf{R}$, connected by n parallel and synchronous bi-directional channels $w_1, w_2, \ldots, w_n$, also known as *wires.* [3] In the worst case, the adversary can compromise an entire wire by controlling a single node (say the first node) on the wire. Hence, $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ having unbounded computing power can corrupt upto t_b, t_o, t_f and t_p wires in Byzantine, omission, failstop and passive fashion respectively. Moreover, we assume that the wires that are under the control of the adversary in Byzantine, omission, failstop and passive fashion are mutually disjoint.

A wire which is controlled in a failstop fashion may fail to deliver any information, but if it delivers any 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. Also, once a failstop controlled wire crashes, then it will remain inactive for the rest of the protocol. 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 wire which is controlled in omission fashion behaves in a similar fashion as a failstop controlled wire. However, the adversary will also know the information that passed through a omission controlled wire. Moreover, a crashed wire which is controlled in omission fashion, may again become alive later. 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, which passed through a Byzantine corrupted wire.

[3] The approach of abstracting the network as a collection of n wires is justifying using Menger's theorem [24] which states that a graph is $c-(\mathbf{S},\mathbf{R})$-connected iff $\mathbf{S}$ and $\mathbf{R}$ are connected by at least c vertex disjoint paths.

Throughout this paper, we use m to denote the message that $\mathbf{S}$ wishes to send to $\mathbf{R}$. The message is assumed to be a sequence of ℓ elements from the finite field $\mathbb{F}$. The only constraint on $\mathbb{F}$ is that its size must be no less than the number of wires n. Moreover, the size of $\mathbb{F}$ is a function of δ which is the error probability of the URMT and USMT protocol. In our protocols, we show how to set $\mathbb{F}$ as a function of δ, to achieve an error probability of at most δ. Since we measure the size of the message in terms of the number of field elements, we also measure the communication complexity in units of field elements.

Broadcast: If some information is sent over all the wires then it is said to be "broadcast". If x is "broadcast" over at least $2t_b + t_o + t_f + 1$ wires, then at most $t_f + t_o$ wires may crash and fail to deliver x (if these wires does not crash, then they will deliver correct 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.

Definition 5 PRMT: *In perfectly reliable message transmission (PRMT) over a sufficiently connected network $\mathcal{N} = (V, E)$, tolerating mixed adversary $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$, $\mathbf{S} \in V$ intends to transmit a message m which is a sequence of ℓ field elements from a finite field $\mathbb{F}$ to $\mathbf{R} \in V$ using some protocol, such that after interacting in phases as per the protocol, the following condition must hold:*
PERFECT RELIABILITY: $\mathbf{R}$ *should correctly output $m' = m$ with probability 1.*

Definition 6 PSMT: *The problem of perfectly secure message transmission (PSMT) over a sufficiently connected network $\mathcal{N}$ requires perfect reliability of PRMT and the following condition:*
PERFECT SECRECY: *The message should be hidden from the adversary in information theoretic sense.*

Definition 7 URMT: *The problem of URMT is same as PRMT, except that it should satisfy a weaker version of perfect reliability called probabilistic reliability:*
PROBABILISTIC RELIABILITY: $\mathbf{R}$ *should correctly output $m' = m$ with probability at least $1 - \delta$, for any $0 < \delta < 1/2$.*

Notice that "Probabilistic Reliability" says that $\mathbf{R}$ can obtain a wrong message with small probability δ. We now define a strictly stronger notion of "Probabilistic Reliability" which we call as "Strong Probabilistic Reliability". A URMT protocol that achieves "Strong Probabilistic Reliability" always outputs the correct message ; otherwise it fails with output NULL, but it never outputs an incorrect message.

Definition 8 STRONG PROBABILISTIC RELIABILITY: $\mathbf{R}$ *should either correctly receive $\mathbf{S}$'s message or otherwise output NULL, where the probability of receiving correct message is at least $1 - \delta$, for any $0 < \delta < 1/2$.*

Definition 9 STRONG URMT: *Strong URMT satisfies "Strong Probabilistic Reliability" property instead of "Probabilistic Reliability".*

Definition 10 USMT: *USMT requires probabilistic reliability property of URMT and perfect secrecy property of PSMT.*

Definition 11 STRONG USMT: *Strong USMT requires* PERFECT SECRECY *of PSMT and should satisfy "Strong Probabilistic Reliability".*

All the protocols presented in this paper are strong URMT and strong USMT protocols.

Definition 12 COMMUNICATION OPTIMAL URMT/USMT PROTOCOL: *Let Π be an r ($r \geq 1$) phase URMT (USMT) protocol which reliably (securely) sends a message m containing ℓ field elements by communicating $O(b)$ field elements. If the lower bound on the communication complexity of any r phase URMT (USMT) protocol to send m is $\Omega(b)$ field elements, then Π is said to be a communication optimal URMT (USMT) protocol to reliably (securely)send m.*

3 URMT in Undirected Network Tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$

In this section, we characterize the POSSIBILITY of single phase URMT tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$. We then prove the lower bound on the communication complexity of any single phase URMT protocol tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ and show that our bound is *tight* by designing a communication optimal single phase URMT whose total communication complexity matches this bound. We then briefly discuss multiphase URMT tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$. Finally, we end the section by comparing our results with the existing results for PRMT.

3.1 Characterization for single phase URMT

The existing characterization for URMT tolerating threshold adaptive Byzantine adversary in undirected network is as follows:

Theorem 1 ([12]) *Any $r \geq 1$ phase URMT between* **S** *and* **R** *against a t_b active adaptive Byzantine adversary $\mathcal{A}_{t_b}$ is possible iff the network is $(2t_b+1)$-(***S,R***)-connected.*

The characterization for URMT tolerating mixed adversary is as follows:

Theorem 2 *Any $r \geq 1$ phase URMT between* **S** *and* **R** *against a threshold adaptive mixed adversary $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ is possible iff the network is $(2t_b+t_o+t_f+1)$-(***S,R***)-connected.*

PROOF: If part: Consider a network which is $(2t_b + t_o + t_f + 1)$-(**S,R**)-connected. So there exists $n \geq 2t_b + t_o + t_f + 1$ wires between **S** and **R**. To send a message m, **S** simply *broadcasts* m to **R** over the n wires. It is easy to see that **R** will receive m with probability one by taking majority [4].
Only if part: We now show that if the network is not $(2t_b+t_o+t_f+1)$-(**S,R**)-connected, then no URMT protocol exists. Assume that a URMT protocol Π exists in a network $\mathcal{N}$ that is not $(2t_b + t_o + t_f + 1)$-(**S,R**)-connected. Consider the network $\mathcal{N}'$, induced by $\mathcal{N}$, on deleting $(t_o + t_f)$ vertices from a minimal vertex cutset of $\mathcal{N}$. This can be viewed as an adversary crashing the communication over $t_o + t_f$ wires, which are under its control in omission and failstop fashion respectively. It follows that $\mathcal{N}'$ is not $(2t_b+1)$-(**S,R**)-connected. Evidently, if Π is a URMT protocol on $\mathcal{N}$, then Π' is a URMT protocol on $\mathcal{N}'$, where Π' is the protocol Π restricted to the players in $\mathcal{N}'$. However, from Theorem 1, Π' is non-existent. Thus Π is impossible too. □

Significance of Theorem 2: Theorem 2 *strictly generalizes* Theorem 1 because we obtain the latter by substituting $t_o = t_f = 0$ in the former. Now consider a network, which is 4-(**S,R**)-connected. From Theorem 1, on this network, any URMT protocol can tolerate at most *one* Byzantine fault. However, according to Theorem 2, it is possible to tolerate *one additional* faulty node, which can be either omission or fail-stop faulty. Thus our characterization shows *more fault tolerance* in comparison to the existing results and also shows the motivation for studying URMT and USMT in the context of mixed adversary.

Comparison 1 (POSSIBILITY of PRMT vs POSSIBILITY of URMT) *From Table 3 (third row), for the existence of any $r \geq 1$ phase PRMT against $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$, there should exist $n \geq 2t_b + t_o + t_f + 1$ wires between* **S** *and* **R**. *From Theorem 2, the same number of error is required even for the existence of URMT protocol against $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$. This shows that allowing a negligible error probability in the reliability does not help in the* POSSIBILITY *of reliable message transmission protocols.*

Though allowing a negligible error helps in the POSSIBILITY of reliable message transmission protocols, in the sequel, we show that allowance of a negligible error probability in transmission *significantly* reduces the communication complexity in comparison to perfect (zero error) transmission.

[4] The protocol described here is a naive protocol which does not take the advantage of allowing small error probability in the reliability.

3.2 Lower Bound on Communication Complexity of Single phase URMT Protocol

We now prove the lower bound on the communication complexity of any single phase URMT protocol tolerating mixed adversary $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$.

Theorem 3 *Any single phase URMT protocol, from* $\mathbf{S}$ *to* $\mathbf{R}$ *over* $n \geq 2t_b+t_o+t_f+1$ *wires, communicates* $\Omega(\frac{n\ell}{n-(t_b+t_o+t_f)})$ *field elements to transmit a message containing* ℓ *field elements tolerating* $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$.

Proof: In any single phase URMT protocol, the concatenation of the information sent over n wires can be viewed as an (probabilistic) error correcting code which can correct t_b Byzantine errors and $t_o + t_f$ erasures with an arbitrarily high probability. Without loss of generality, the domain of the set of possible values of the data sent along a wire can be assumed to be the same for all the wires [5]. Let $\mathbb{S}$ be the set of possible values of the data sent along the wires. Thus, each codeword can be viewed as concatenation of n elements from $\mathbb{S}$ which can be represented by $n \log |\mathbb{S}|$ bits. Now, the removal of any $(t_b+t_o+t_f)$ elements from each of the codewords, which corresponds to an adversary blocking $t_b + t_o + t_f$ wires (a Byzantine adversary can also block communication) should result in shortened codewords that are all distinct. For if any two are identical, then the original codewords could have differed only in at most $(t_b + t_o + t_f)$ elements, implying that there exist two codewords c_1 and c_2 and an adversarial strategy such that the receiver's view is the *same* on the receipt of c_1 and c_2. Specifically, without loss of generality assume that c_1 and c_2 differ only in their last $(t_b + t_o + t_f)$ elements. That is, $c_1 = \alpha \circ \beta$ and $c_2 = \alpha \circ \gamma$, where $\circ$ denotes concatenation and $|\beta| = |\gamma| = (t_b + t_o + t_f)$ elements. Now, consider the two cases: (a) c_1 is sent and the adversary corrupts it to $\alpha \circ \perp$ by completely blocking the last $(t_b + t_o + t_f)$ elements (wires) and (b) c_2 is sent and the adversary again corrupts it to $\alpha \circ \perp$. Thus, $\mathbf{R}$ can not distinguish between the receipt of c_1 and c_2 with probability greater than $\frac{1}{2}$, which violates the URMT communication property (in any URMT protocol, receiver should be able to receive the message with probability more than $\frac{1}{2}$). Therefore, all shortened codewords containing $n - (t_b + t_o + t_f)$ elements from $\mathbb{S}$ are distinct. This implies that there are same number of shortened codewords as original codewords. But the number of shortened codewords can be at most $C = |\mathbb{S}|^{(n-(t_b+t_o+t_f))}$. Now each shortened codeword can be represented by $\log C = (n-(t_b+t_o+t_f)) \log |\mathbb{S}|$ bits. Since, for error-correction, we need to communicate the longer codeword containing $n \log |\mathbb{S}|$ bits, reliable communication of shortened codeword of $k = \log C$ bits incurs a communication cost of at least $n \log |\mathbb{S}|$ bits. Hence communication of a single bit incurs communication of $\frac{n}{(n-(t_b+t_o+t_f)}$ bits. So to communicate ℓ elements from a field $\mathbb{F}$, represented by $\ell \log |\mathbb{F}|$ bits, $\Omega(\frac{n\ell}{(n-(t_b+t_o+t_f))} \log |\mathbb{F}|)$ bits need to be sent. Since $\log |\mathbb{F}|$ bits represents one field element from $\mathbb{F}$, communicating ℓ elements from $\mathbb{F}$ requires a communicating $\Omega(\frac{n\ell}{(n-(t_b+t_o+t_f)})$ field elements. □

Remark 2 *In any URMT protocol designed over a field* $\mathbb{F}$*, the size of the field depends upon the error probability* δ *of the protocol (this is demonstrated in next section). From Theorem 3, any URMT protocol to send* ℓ *field elements from* $\mathbb{F}$ *need to communicate* $\Omega(\frac{n\ell}{(n-(t_b+t_o+t_f))} \log |\mathbb{F}|)$ *bits. Thus the communication complexity of any single phase URMT protocol is a function of* δ *(since* $|\mathbb{F}|$ *is a function of* δ*), though it is not explicitly mentioned in the expression derived in Theorem 3. It should also be noted that communication complexity explicitly depends upon the message size* ℓ*.*

Comparison 2 (Communication Complexity of Single Phase PRMT and URMT:) *While the lower bound on the communication complexity of any single phase PRMT tolerating mixed adversary is* $\Omega(\frac{n\ell}{(n-(2t_b+t_o+t_f)})$ *field elements (see Table 3, third row), the same for URMT is* $\Omega(\frac{n\ell}{(n-(t_b+t_o+t_f)})$ *field elements (Theorem 3). This clearly shows that allowing a negligible error probability helps in significant reduction in the communication complexity of reliable protocols.*

[5] Suppose, however, that there is exists a protocol Π that does not have this symmetry property among the data sent along the wires. Then consider the protocol Π' which consists of n parallel executions of protocol Π with the identities or numbers of the wires being "rotated" by a distance of i in the i^{th} execution. Clearly, this protocol achieves the symmetry property by "spreading the load"; further its message expansion factor is equal to that of Π. Thus, one may without loss of generality, assume that the domains of all the wires are the same.

3.3 Upper Bound on Communication Complexity of Single Phase URMT Tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$

Let **S** and **R** be connected by $n = 2t_b + t_o + t_f + 1$ wires, denoted as $\mathcal{W} = \{w_1, w_2, \ldots, w_n\}$, of which at most t_b, t_o, t_f and t_p are under the control of $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ in Byzantine, omission, failstop and passive fashion respectively. We now present a *communication optimal* single phase URMT protocol **URMT_Single_Phase**, which delivers a message containing $(t_b + 1)n$ field elements by communicating $O(n^2)$ field elements in single phase with (arbitrarily) high probability. This shows that the lower bound on the communication complexity of single phase URMT proved in the previous section is tight. **URMT_Single_Phase** has a special feature that it achieves reliability with *constant factor* overhead, when executed only under the presence of Byzantine adversary (i.e., $t_o = t_f = t_p = 0$). Let δ be a bound on the probability that the protocol fails to deliver the correct message. We require the size of the field $\mathbb{F}$ to be at least $\frac{n^3}{\delta}$. The message block is represented by $\mathbf{M} = [m_1\ m_2\ \ldots\ m_n\ m_{n+1}\ m_{n+2} \ldots\ m_{2n} \ldots\ m_{t_bn+1}\ m_{t_bn+2} \ldots m_{t_bn+n}]$.

Remark 3 *Our single phase protocol* **URMT_Single_Phase** *is a strong URMT protocol (see Definition 9).*

Before presenting the protocol, we describe a novel technique, called as **Extrapolation Technique** which we use in designing the protocol **URMT_Single_Phase**.

Extrapolation Technique: We visually represent **M** as a rectangular array A of size $(t_b+1) \times n$ where the j^{th} row, $1 \leq j \leq t_b + 1$ contains the elements $m_{(j-1)n+1}\, m_{(j-1)n+2} \ldots m_{(j-1)n+n}$. For each column i of A, $1 \leq i \leq n$ we do the following: we construct the unique t_b degree polynomial $q_i(x)$ passing through the points $(1, m_i), (2, m_{n+i}), \ldots, (t_b + 1, m_{t_bn+i})$ where $m_i, m_{n+i}, \ldots, m_{t_bn+i}$ belong to the i^{th} column A. Then $q_i(x)$ is evaluated at $t_b + t_o + t_f$ values of x namely, $x = t_b + 2, t_b + 3, \ldots, n$ to obtain $c_{1i}, c_{2i}, \ldots, c_{(t_b+t_o+t_f)i}$. Finally, we obtain a square array D of size $n \times n$ containing n^2 elements, where

$$D = \begin{bmatrix} m_1 & m_2 & \ldots & m_i & \ldots & m_n \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ m_{(j-1)n+1} & m_{(j-1)n+2} & \ldots & m_{(j-1)n+i} & \ldots & m_{(j-1)n+n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ m_{t_bn+1} & m_{t_bn+2} & \ldots & m_{t_bn+i} & \ldots & m_{t_bn+n} \\ c_{11} & c_{12} & \ldots & c_{1i} & \ldots & c_{1n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ c_{j1} & c_{j2} & \ldots & c_{ji} & \ldots & c_{jn} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ c_{(t_b+t_o+t_f)1} & c_{(t_b+t_o+t_f)2} & \ldots & c_{(t_b+t_o+t_f)i} & \ldots & c_{(t_b+t_o+t_f)n} \end{bmatrix} = \begin{bmatrix} A \\ C \end{bmatrix} \text{ where}$$

C is the sub-matrix of D containing last $t_b + t_o + t_f$ rows. Thus D is the row concatenation of matrix A of size $(t_b + 1) \times n$ (containing elements of **M**) and matrix C, whose elements are obtained from A by **Extrapolation Technique**. We now prove certain properties of the array D.

Lemma 1 *In D, all the n elements of any column can be uniquely generated from any $t_b + 1$ elements of the same column.*

Proof: The proof follows from the simple observation that the n elements along any column of D lie on a t_b degree polynomial and any $t_b + 1$ points on a t_b degree polynomial are enough to reconstruct the t_b degree polynomial. □

Lemma 2 *The elements of message* **M** *can be uniquely determined from any $t_b + 1$ rows of D.*

Proof: From the construction of D, the elements of **M** are arranged in the first $t_b + 1$ rows. If the first $t_b + 1$ rows are known then the lemma holds trivially. On the other hand, if some other $t_b + 1$ rows are known, then from Lemma 1, i^{th} column, $1 \leq i \leq n$, of D can be completely generated from $t_b + 1$ elements of the same column. Hence, knowledge of any $t_b + 1$ rows can reconstruct the whole matrix D and hence the message **M** (first $t_b + 1$ rows of D). □

Lemma 3 *Modification of at most t_b elements along any column of D is detectable.*

PROOF: Recall that in D, the values along i^{th} column lie on a unique t_b degree polynomial $q_i(x)$. Now suppose t_b values along i^{th} column are changed in such a manner that they lie on some other t_b degree polynomial $q_i'(x)$, where $q_i(x) \neq q_i'(x)$. Since both $q_i(x)$ and $q_i'(x)$ are of degree t_b, they can match on additional t_b common points. But still there are at least $n - 2t_b = t_o + t_f + 1$ points which lie on the original polynomial $q_i(x)$ (but not on $q_i'(x)$). Hence any attempt to interpolate a t_b degree polynomial passing through the elements of i^{th} column (in which at most t_b values has been changed) will be futile, clearly indicating that at most t_b values are changed along the column. Hence the lemma holds. □

We are now ready to describe our single phase URMT protocol.

Protocol URMT_Single_Phase - The Single Phase URMT Protocol

Computation and Communication by S:

1. **S** generates a rectangular array D containing n^2 field elements, from the $(t_b + 1) \times n$ elements of message **M** using **Extrapolation Technique**. **S** then forms n polynomials $p_j(x), 1 \leq j \leq n$, each of degree $n-1$, where $p_j(x)$ is formed using the j^{th} row of D as follows: the coefficient of $x^i, 0 \leq i \leq n-1$ in $p_j(x)$ is the $(i+1)^{th}$ element of j^{th} row of D.

2. **S** chooses another n^2 field elements at random, say r_{ji}, $1 \leq i, j \leq n$. Over w_j, **S** sends the following to **R**: the polynomial $p_j(x)$ and the n ordered pairs $(r_{ji}, p_i(r_{ji}))$, for $1 \leq i \leq n$. Let $v_{ji} = p_i(r_{ji})$.

Message Recovery by R:

1. Let $\mathcal{F}$ denotes the set of wires that delivered nothing and let $\mathcal{B}$ denotes the set of wires that delivered invalid information (like higher degree polynomials etc.). Note that the wires in $\mathcal{B}$ are Byzantine corrupted because omission or fail-stop controlled wires are not allowed to modify the information passing over them. **R** removes all the wires in $(\mathcal{F} \cup \mathcal{B})$ from $\mathcal{W}$, to work on the remaining wires in $\mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$, out of which at most $t_b - |\mathcal{B}|$ could be Byzantine corrupted. Let **R** receives $p_j'(x)$ and (r_{ji}', v_{ji}'), $1 \leq i \leq n$ over $w_j \in \mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$. We say that w_j *contradicts* w_i if: $v_{ji}' \neq p_i'(r_{ji}')$ where $w_i, w_j \in \mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$. Among all the wires in $\mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$, **R** checks if there is a wire contradicted by at least $(t_b - |\mathcal{B}|) + 1$ wires. All such wires are Byzantine corrupted and removed (see Lemma 4).

2. To retrieve **M**, **R** tries to reconstruct the array D as generated originally by **S**. Let D' represents the corresponding array which **R** tries to recover at his end. Corresponding to each $w_j \in \mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$, which is not removed in previous step, **R** fills the j^{th} row of D' in the following manner: coefficient of $x^i, 0 \leq i \leq n-1$ in $p_j'(x)$ occupies $(i+1)^{th}$ column in the j^{th} row of D'; i.e., the coefficients of $p_j'(x)$ are inserted in j^{th} row of D' such that the coefficient of x^i in $p_j'(x)$ occupies $(i+1)^{th}$ column in the j^{th} row of D'.

3. After doing the above step for each $w_j \in \mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$, which is not removed in step 1 of message recovery, **R** has at least $t_b + 1$ rows inserted in D' (see Lemma 6). **R** then checks the validity of these rows as follows: let $i_1, i_2, \ldots, i_k, k \geq t_b + 1$ denote the index of the rows which are inserted by **R** in D'. Let $y_{i_1}^j, y_{i_2}^j, \ldots, y_{i_k}^j, 1 \leq j \leq n$ denote the values along $j^{th}, 1 \leq j \leq n$ column of D'. **R** checks whether the points $(i_1, y_{i_1}^j), (i_2, y_{i_2}^j), \ldots, (i_k, y_{i_k}^j)$ lie on a t_b degree polynomial. Note that at this point, each column will have at least $t_b + 1$ elements, which are enough to do the checking.

4. If the above test fails for at least one column of D', then **R** outputs "NULL" and halts. Otherwise, **R** regenerates the complete D' correctly and recovers **M** from the first $t_b + 1$ rows (see Lemma 6).

Lemma 4 *In* **URMT_Single_Phase**, *if any $w_j \in \mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$ is contradicted by at least $(t_b - |\mathcal{B}|) + 1$ wires from the set $\mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$, then the polynomial $p_j(x)$ over w_j has been changed by adversary or in effect w_j is Byzantine corrupted.*

PROOF: The wires in $\mathcal{B}$ are already identified to be Byzantine corrupted and hence neglected by **R**. Also the wires in $\mathcal{F}$ delivers nothing and hence neglected by **R**. So among the remaining $\mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$ wires, at most $(t_b - |\mathcal{B}|)$ could be Byzantine corrupted. Also there cannot be any contradiction between two honest wires (which has correctly delivered the values to **R**) and hence any honest wire can be contradicted by at most $(t_b - |\mathcal{B}|)$ wires. Thus if a wire is contradicted by at least $(t_b - |\mathcal{B}|) + 1$ wires then it is Byzantine corrupted. □

Lemma 5 *In the protocol, if the adversary corrupts a polynomial over wire w_j in such a way that w_j is not removed during step 1 of message recovery, then* **R** *will always be able to detect it at the end of step 3 of message recovery and outputs "NULL".*

PROOF: We consider the worst case, where t_o+t_f wires which are omission and failstop controlled crashed and failed to deliver any information (if they do not crash then they deliver correct information). So **R** will receive information over $2t_b+1$ wires, of which at most t_b could be Byzantine corrupted. At the beginning of step 3 of message recovery, there are at least t_b+1 rows present in D'. This follows from the fact there always exist t_b+1 honest wires which will deliver correct polynomials to **R**. As mentioned in Lemma 4, any honest wire can be contradicted by at most $(t_b-|\mathcal{B}|)$ wires and hence is not be removed by **R** during step 1 of message recovery. So the coefficients of the polynomials corresponding to these honest wires will be present in D'.

Now if w_j (which has delivered a faulty polynomial $p'_j(x) \neq p_j(x)$) is not removed during step 1 of message recovery, then during step 2 of message recovery, the coefficients of $p'_j(x)$ are inserted in the j^{th} row of D'. Since $p_j(x) \neq p'_j(x)$, there exists at least one coefficient in $p'_j(x)$ which is different from the corresponding coefficient in $p_j(x)$. Let $p_j(x)$ differs from $p'_j(x)$ in the coefficient of x^i. Then $(i+1)^{th}$ column of D' differs from the $(i+1)^{th}$ column of original D at j^{th} position. Also the $(i+1)^{th}$ column of D' may differ from the $(i+1)^{th}$ column of original D in at most t_b locations (including j^{th} location). This is because in the worst case, out of the $2t_b+1$ wires, the adversary may change the polynomials along at most t_b wires (which are Byzantine corrupted), such that the coefficient of x^i in all these changed polynomials differ from their corresponding coefficient of x^i in the original polynomials. So, in the worst case, at most t_b elements of the $(i+1)^{th}$ column of D' can be different from $(i+1)^{th}$ column of D. The proof now follows from Lemma 3. Hence **R** will detect that at atmost t_b of the received polynomials are incorrect and outputs "NULL". □

Lemma 6 *In* **URMT_Single_Phase**, *if the test in step 4 of message recovery succeeds for all the n columns of D', then* **R** *will never output "NULL" and always recovers* **M** *correctly.*

PROOF: As explained in previous Lemma, at the beginning of step 4 of message recovery, there will be at least t_b+1 rows present in D'. Now if the test in step 4 succeeds for all the n columns of D', it implies that all the rows present in D' are same as the corresponding rows in the original D. From Lemma 1, **R** will be able to completely regenerate all the n columns of original D. The proof now follows from Lemma 2. It is easy to see that **R** does not outputs "NULL" in this case.

Theorem 4 *If* **URMT_Single_Phase** *is executed over a field $\mathbb{F}$ of size $|\mathbb{F}| \geq \frac{n^3}{\delta}$, then it is a strong URMT protocol and terminates with a non-"NULL" output with probability at least $1-\delta$.*

PROOF. Since no honest wire contradicts another honest wire, from Lemma 4, all the wires removed by **R** during step 1 of message recovery are indeed faulty. We need to show that if a wire is corrupted and delivered incorrect polynomial, then it will be contradicted by all the honest wires with high probability. Let π_{ij} be the probability that a corrupted wire w_j will not be contradicted by a honest wire w_i. This means that the adversary can ensure that $p_j(r_{ij}) = p'_j(r_{ij})$ with a probability of π_{ij}. Since there are only $n-1$ points at which these two polynomials intersect, this allows the adversary to guess the value of r_{ij} with a probability of at least $\frac{\pi_{ij}}{n-1}$. But since r_{ij} was selected uniformly in $\mathbb{F}$, the probability of guessing it is at most $\frac{1}{|\mathbb{F}|}$. Therefore we have $\pi_{ij} \leq \frac{n-1}{|\mathbb{F}|}$ for each i,j. Thus the total probability that the adversary can find w_i, w_j such that corrupted wire w_j will not be contradicted by an honest wire w_i is at most $\sum_{i,j} \pi_{ij} \leq \frac{n^2(n-1)}{|\mathbb{F}|}$ which is bounded by $\frac{n^3}{|\mathbb{F}|}$. Since $\mathbb{F}$ is chosen such that $|\mathbb{F}| \geq \frac{n^3}{\delta}$, it follows that a Byzantine corrupted wire w_j and hence a corrupted $p'_j(x) \neq p_j(x)$, received over w_j can be included in D' with probability at most δ. However, if such a $p'_j(x)$ is included in D', then from Lemma 5, **R** will detect this and will output "NULL". Thus protocol **URMT_Single_Phase** is a strong URMT protocol and outputs a non-"NULL" output with probability at least $1-\delta$. □

Theorem 5 **URMT_Single_Phase** *reliably sends* **M** *containing $n(t_b+1)$ field elements by communicating $O(n^2)$ field elements. In terms of bits, the protocol sends $n(t_b+1)\log|\mathbb{F}|$ bits by communicating $O(n^2 \log|\mathbb{F}|)$ bits.*

PROOF: Over each wire, **S** sends a polynomial of degree $n-1$ and n ordered pair. Thus the total communication complexity is $O(n^2)$. Since each element from field $\mathbb{F}$ can be represented by $\log|\mathbb{F}|$ bits, the communication complexity of the protocol is $O(n^2 log|\mathbb{F}|)$ bits. □

Theorem 6 **URMT_Single_Phase** *is a communication and phase optimal URMT protocol.*

PROOF: It is obvious that **URMT_Single_Phase** is a phase optimal protocol. Now we show that **URMT_Single_Phase** is *communication optimal* also. In Theorem 3, substituting $n = 2t_b + t_o + t_f + 1$ and $\ell = n(t_b+1)$, we find that any single phase URMT protocol must communicate $\Omega(n^2)$ elements to send $n(t_b + 1)$ elements. Now, from Theorem 5, the communication complexity of **URMT_Single_Phase** is $O(n^2)$. Hence our protocol has **optimal communication complexity**. In terms of bits, **URMT_Single_Phase** sends $n(t_b + 1) \log |\mathbb{F}|$ bits by communicating $O(n^2 \log |\mathbb{F}|)$ bits where $|\mathbb{F}| = \frac{n^3}{\delta}$ and $1 - \delta$ is the least probability with which the protocol terminates without "NULL". So, our protocol is both phase and communication optimal. □

Corollary 1 *Protocol* **URMT_Single_Phase** *when executed under the presence of only Byzantine adversary, achieves reliability with "constant factor overhead" by sending* $\Theta(n^2)$ *field elements with a communication overhead of* $O(n^2)$ *field elements.*

PROOF: From Theorem 5, **URMT_Single_Phase** reliably sends $n(t_b + 1)$ field elements by communicating $O(n^2)$ field elements when $n = 2t_b + t_o + t_f + 1$. If $t_o = t_f = 0$, then **URMT_Single_Phase** sends $(t_b + 1)n = \Theta(n^2)$ field elements (when $t_o = 0, t_f = 0$, $n = 2t_b + 1$ and so $t_b = \Theta(n)$) by communicating $O(n^2)$ field elements. Thus it achieves reliability with "constant factor overhead". □

3.4 Multiphase URMT tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$

We now briefly discuss about the communication complexity of multiphase URMT protocols tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$.

Theorem 7 *Any multiphase URMT protocol between* **S** *and* **R** *over* $n \geq 2t_b + t_o + t_f + 1$ *wires communicates* $\Omega(\ell)$ *field elements to send a message containing* ℓ *field elements against* $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$.

PROOF: The lower bound of $\Omega(\ell)$ for sending ℓ field elements is obvious, since any URMT protocol must send at least the message. An $O(\log \frac{t_f+t_o}{n-t_f-t_o})$ phase PRMT protocol which sends ℓ field elements by communicating $O(\ell)$ field elements is presented in [37]. The protocol of [37] is also a valid multiphase URMT, thus satisfying the communication complexity lower bound for multiphase URMT. □

We do not know whether there exists a URMT protocol with less number of phases, which sends ℓ field elements by communicating $O(\ell)$ field elements. Design of such a protocol is left as an open problem.

3.5 Comparison of URMT with PRMT

We now compare the results of URMT presented in this section, with the existing results for PRMT. The comparison can be listed as follows:

1. Allowing a negligible error probability in the reliability does not help in the POSSIBILITY of reliable message transmission protocols (see Comparison 1).

2. Allowing a negligible error probability in the reliability *significantly* reduces the communication complexity of reliable message transmission protocols (see Comparison 2).

3. In the presence of *only* Byzantine adversary (i.e., $t_o = t_f = t_p = 0$), it is impossible to design any single phase PRMT protocol which achieves reliability with "constant factor overhead"; i.e., sending ℓ field elements by communicating $O(\ell)$ field elements. The minimum number of phases required by any PRMT protocol to achieve reliability with "constant factor overhead" is 3 [27]. However, it is possible to design a single phase URMT, which under the presence of only Byzantine adversary, achieves reliability with "constant factor overhead" (see Corollary 1). This again shows the power of allowing a negligible error probability in the context of reliable message transmission.

4 Single Phase USMT Tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$

In this section, we prove the necessary and sufficient condition for the existence of any single phase USMT protocol in the presence of mixed adversary. We then prove the lower bound on the communication complexity of any single phase USMT protocol and show that our bound is *tight* by designing a *communication optimal* single phase USMT protocol whose total communication complexity satisfy this bound. As a special case, we show that in the presence of only Byzantine adversary (i.e., $t_o = t_f = t_p = 0$), if $3t_b + 1$ wires are available, then our single phase USMT protocol achieves security with constat factor overhead. Finally we compare our results on single phase USMT with the existing results for single phase PSMT. Our comparison shows that allowing a negligible error probability *only* in the reliability, *significantly* helps in the POSSIBILITY and reducing the communication complexity of single phase secure message transmission protocols.

4.1 Single Phase USMT Protocol Tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$: Characterization and Lower Bound on Communication Complexity

Theorem 8 *Any single phase USMT protocol tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ from $\mathbf{S}$ to $\mathbf{R}$ over n wires is possible if and only if $n > 2t_b + 2t_o + t_f + t_p$. Moreover any such single phase USMT protocol is required to communicate $\Omega(\frac{n\ell}{n-(2t_b+2t_o+t_f+t_p)})$ field elements in order to send a message containing ℓ-field elements.*

Remark 4 *In any USMT protocol designed over a field $\mathbb{F}$, the size of the field depends upon the error probability (in reliability) δ of the protocol. Since each field element from a field $\mathbb{F}$ can be represented by $log|\mathbb{F}|$ bits, from Theorem 8, any single phase USMT protocol to send $\ell log|\mathbb{F}|$ bits, need to communicate $\Omega\left(\frac{n\ell}{n-(2t_b+2t_o+t_f+t_p)} log|\mathbb{F}|\right)$ bits. Thus the communication complexity of any single phase USMT protocol is a function of δ (since $|\mathbb{F}|$ is a function of δ), though it is not explicitly mentioned in the expression derived in Theorem 8.*

PROOF: We first prove the lower bound on the communication complexity. Since perfect secrecy is required, the data (or shares) sent along the n wires in any single phase USMT protocol must be such that information on any set of $(t_b + t_o + t_p)$ wires has no information about the secret message, otherwise the adversary will also know the secret message by passively listening the contents of these wires (recall that the eavesdropping capability of the adversary is at most $t_b + t_o + t_p$). Similarly, the data (shares) sent over any $(n - (t_b + t_o + t_f))$ honest wires during the protocol has full information about the secret message. The latter requirement ensures that even if the adversary simply blocks/corrupts all the data that he can, the secret message is not lost and therefore the receiver's ability to recover the message is not completely ruled out.

Let X_i denotes the i^{th} share of some valid distribution scheme and let m denote the secret message containing ℓ field elements. Then m can be viewed as a value drawn uniformly at random from $\mathbb{F}^\ell$. For any subset $A \subseteq \{1, 2 \ldots n\}$ let X_A denote the set of variables $\{X_i | i \in A\}$. Then the secret m and the shares X_i are random variables. For a random variable X, let $H(X)$ denote its entropy [5]. Roughly speaking, entropy quantifies the information contained in a message, usually in bits or bits/symbol. Since m is drawn uniformly at random from $\mathbb{F}^\ell$, we have $H(m) = \ell$.

Since in any single phase USMT protocol, the data sent along any set B consisting of $(n-(t_b+t_o+t_f))$ honest wires have full information about m, we have

$$H(m|X_B) = 0.$$

Consider any subset $A \subset B$ such that $|A| = (t_b + t_o + t_p)$. Since the data sent along the wires in A is insufficient to retrieve any information about the message m we get

$$H(m|X_A) = H(m).$$

From the chain rule of the entropy [5], for any two random variable X_1, X_2, we have $H(X_1, X_2) = H(X_2) + H(X_1|X_2)$. Here $H(X_1, X_2)$ denotes the joint entropy of X_1, X_2. Informally, the joint entropy

measures how much entropy is contained in a joint system of two random variables. Similarly, $H(X_1|X_2)$ denotes conditional entropy of X_1 on X_2. Informally, it quantifies the remaining entropy (i.e. uncertainty) of X_1 given that the value of a second random variable X_2 is known. Substituting $X_1 = m|X_A$ and $X_2 = X_{B-A}$, we get

$$H(m|X_A, X_{B-A}) \quad = \quad H(X_{B-A}) + H(m|X_A|X_{B-A})$$

From the properties of joint entropy [5], 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, $H(m|X_A, X_{B-A}) \geq H(m|X_A)$. Substituting in the above equation, we get

$$\begin{aligned} H(m|X_A) &\leq H(m|X_A|X_{B-A}) + H(X_{B-A}) \\ &\leq 0 + H(X_{B-A}) \text{ because } m \text{ can be known completely from } X_A \text{ and } X_{B-A} \end{aligned}$$

Consequently, $H(m) \leq H(X_{B-A})$ because $H(m|X_A) = H(m)$. Therefore for all the sets C of cardinality $|B| - |A| = ((n - (t_b + t_o + t_f)) - (t_b + t_o + t_p)) = n - (2t_b + 2t_o + t_f + t_p)$, we have

$$\begin{aligned} H(X_C) &\geq H(m) \\ \sum_{i \in C} H(X_i) &\geq H(m) \end{aligned}$$

Summing the above equation over all possible sets of size $n - (2t_b + 2t_o + t_f + t_p)$ we get

$$\sum_{C} \sum_{i \in C} H(X_i) \quad \geq \quad \binom{n}{n - (2t_b + 2t_o + t_f + t_p)} H(m)$$

Now in all the possible $\binom{n}{n-(2t_b+2t_o+t_f+t_p)}$ subsets of size $n - (2t_b + 2t_o + t_f + t_p)$, each of the term $H(X_i), 1 \leq i \leq n$ will appear $\binom{n-1}{n-(2t_b+2t_o+t_f+t_p)-1}$ times. So we get

$$\begin{aligned} \binom{n-1}{n - (2t_b + 2t_o + t_f + t_p) - 1} \sum_{i=1}^{n} H(X_i) &\geq \binom{n}{n - (2t_b + 2t_o + t_f + t_p)} H(m.) \\ \text{Thus } \sum_{i=1}^{n} H(X_i) &\geq \frac{n}{n - (2t_b + 2t_o + t_f + t_p)} H(m). \end{aligned}$$

Now, right hand side of the equation is nothing but $\left(\frac{n\ell}{n-(2t_b+2t_o+t_f+t_p)}\right)$ because $H(m) = \ell$. Since $\sum_{i=1}^{n} H(X_i)$ defines the information content over n wires, which is sent during any single phase USMT protocol, the lower bound on the communication complexity of any single phase USMT protocol is $\Omega\left(\frac{n\ell}{n-(2t_b+2t_o+t_f+t_p)}\right)$. The proof of the lower bound completes at this point. We now derive the necessary condition for the possibility of single phase USMT protocol directly from the lower bound expression.

Since the communication complexity of any single phase USMT protocol should be positive, we have $n - (2t_b + 2t_o + t_f + t_p) > 0$, which gives $n > 2t_b + 2t_o + t_f + t_p$. This proves the necessity condition. To prove the sufficiency condition, we design a *communication optimal* single phase USMT protocol **USMT_Single_Phase** with $n = 2t_b + 2t_o + t_f + t_p + 1$ wires in next section. This completes the theorem. □

Comparison 3 (POSSIBILITY **of Single Phase USMT and PSMT)** *From [33], a single phase PSMT protocol tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ is possible iff there exists $n \geq 3t_b + 2t_o + t_f + t_p + 1$ wires between* **S** *and* **R**. *Comparing this with Theorem 8, we find that allowing a negligible error probability (only in the reliability), significantly helps in the* POSSIBILITY *of single phase secure message transmission protocols.*

Comparison 4 (Communication Complexity of Single Phase USMT and PSMT) *In [33], it is shown that any single phase PSMT tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ over $n \geq 3t_b + 2t_o + t_f + t_p + 1$ wires has to communicate $\Omega\left(\frac{n\ell}{n-(3t_b+2t_o+t_f+t_p)}\right)$ field elements to send a message containing ℓ field elements. Comparing this with Theorem 8, we find that allowing a negligible error probability (only in the reliability), significantly helps in reducing the communication complexity of single phase secure message transmission protocols.*

In the sequel, we design a single phase *communication optimal* USMT protocol, whose total communication complexity matches the bound proved in Theorem 8, thus showing that the bound is tight.

4.2 Upper Bound on the Communication Complexity of Single Phase USMT Tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$

We now present a single phase *communication optimal* USMT protocol **USMT_Single_Phase** which securely sends a message containing $t_b + t_o + t_f + t_p + 1 = \Theta(n)$ field elements by communicating $O(n^2)$ field elements, where **S** and **R** are connected by $n = 2t_b + 2t_o + t_f + t_p + 1$ wires. This shows that the lower bound on the communication complexity, established in Theorem 8 is *tight*. We require the field size $|\mathbb{F}| \geq \frac{2n^3}{\delta}$, to realize an error probability of at most δ in **USMT_Single_Phase**.

Remark 5 *In [21], the authors have designed an optimal single phase USMT protocol tolerating a t_b active Byzantine adversary $\mathcal{A}_{t_b}$, where **S** and **R** are connected by $n = 2t_b + 1$ wires. Their protocol is based on secret sharing against cheaters [25]. However, our single phase optimal USMT tolerates mixed adversary and is designed using a different technique. It is not an extension of the protocol given in [21].*

We first briefly recall an algorithm from [34], which we have used as a black-box in our USMT protocol. Consider the following problem: Suppose **S** and **R** by some means agree on a sequence of n values $x = [x_1 x_2 \ldots x_n] \in \mathbb{F}^n$ such that the adversary *only* knows $n - f$ values in x. But neither **S** nor **R** knows the identity of the values which are known to the adversary. The goal is for **S** and **R** to agree on a sequence of f values $[y_1\ y_2\ \ldots\ y_f] \in \mathbb{F}^f$, such that the adversary has *no* information about $[y_1\ y_2\ \ldots\ y_f]$ in information theoretic sense. This is achieved by the following algorithm [34]:

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

Lemma 7 ([34]) *The adversary has no information about $[y_1\ y_2\ \ldots\ y_f]$ computed in algorithm EXTRAND in information theoretic sense.*

Proof: The proof follows from the fact that any $f \times f$ sub-determinant in a $n \times f$ Vandermonde matrix is non-zero. □

Now we explain a method which is used to establish a one time pad between **S** and **R** and used in our single phase USMT protocol. We call our method as **Pad Establishment Technique** which is very similar to **Extrapolation Technique** discussed in section 3.

Pad Establishment Technique: Suppose $n = 2t_b + 2t_o + t_f + t_p + 1$. We first randomly choose $(t_b + t_o + t_p + 1) \times (n + t_p)$ field elements from the field $\mathbb{F}$ denoted by $M_{j1}, M_{j2}, \ldots, M_{j(n+t_p)}, 1 \leq j \leq t_b + t_o + t_p + 1$. We then construct a rectangular array A of size $(t_b + t_o + t_p + 1) \times (n + t_p)$ where the $j^{th}, 1 \leq j \leq t_b+t_o+t_p+1$ row contains the elements $M_{j1}, M_{j2}, \ldots, M_{j(n+t_p)}$. Now consider the first column of A, containing $M_{11}, M_{21}, \ldots, M_{(t_b+t_o+t_p+1)1}$. We construct the unique $t_b+t_o+t_p$ degree polynomial $q_1(x)$ passing through the points $(1, M_{11}), (2, M_{21}), \ldots, (t_b+t_o+t_p+1, M_{(t_b+t_o+t_p+1)1})$. We then evaluate $q_1(x)$ at $t_b+t_o+t_f$ values of x, namely at $x = t_b+t_o+t_p+2, t_b+t_o+t_p+3, \ldots, n$ to obtain $c_{11}, c_{21}, \ldots, c_{(t_b+t_o+t_f)1}$. We repeat the procedure for all the $n + t_p$ columns of A. In general, considering the $i^{th}, 1 \leq i \leq n + t_p$ column of A consisting of the elements $M_{1i}, M_{2i}, \ldots, M_{(t_b+t_o+t_p+1)i}$, we construct the unique $t_b + t_o + t_p$

degree polynomial $q_i(x)$ passing through the points $(1, M_{1i}), (2, M_{2i}), \ldots, ((t_b+t_o+t_p+1), M_{(t_b+t_o+t_p+1)i})$. Then $q_i(x)$ is evaluated at $t_b + t_o + t_f$ values of x, namely at $x = t_b + t_o + t_p + 2, t_b + t_o + t_p + 3, \ldots, n$ to obtain $c_{1i}, c_{2i}, \ldots, c_{(t_b+t_o+t_f)i}$. Finally, we obtain a rectangular array D of size $n \times (n + t_p)$ containing $n \times (n + t_p)$ elements, where

$$D = \begin{bmatrix} M_{11} & M_{12} & \ldots & M_{1i} & \ldots & M_{1(n+t_p)} \\ M_{21} & M_{22} & \ldots & M_{2i} & \ldots & M_{2(n+t_p)} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ M_{j1} & M_{j2} & \ldots & M_{ji} & \ldots & M_{j(n+t_p)} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ M_{(t_b+t_o+t_p+1)1} & M_{(t_b+t_o+t_p+1)2} & \ldots & M_{(t_b+t_o+t_p+1)i} & \ldots & M_{(t_b+t_o+t_p+1)(n+t_p)} \\ c_{11} & c_{12} & \ldots & c_{1i} & \ldots & c_{1(n+t_p)} \\ c_{21} & c_{22} & \ldots & c_{2i} & \ldots & c_{2(n+t_p)} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ c_{j1} & c_{j2} & \ldots & c_{ji} & \ldots & c_{j(n+t_p)} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ c_{(t_b+t_o+t_f)1} & c_{(t_b+t_o+t_f)2} & \ldots & c_{(t_b+t_o+t_f)i} & \ldots & c_{(t_b+t_o+t_f)(n+t_p)} \end{bmatrix} = \begin{bmatrix} A \\ C \end{bmatrix} \text{ where}$$

C is the sub-matrix of D containing last $t_b + t_o + t_f$ rows. Thus D is the row concatenation of matrix A of size $(t_b + t_o + t_p + 1) \times (n + t_p)$ and matrix C, whose elements are obtained from A.

Remark 6 (Difference between Extrapolation Technique and Pad Establishment Technique) *: In* **Extrapolation Technique***, the size of the matrix A is $(t_b + 1) \times n$ and its elements constitutes the message, that* **S** *wants to reliably send to* **R***. On the other hand, in* **Pad Establishment Technique***, the size of the matrix A is $(t_b + t_o + t_p + 1) \times (n + t_p)$. Moreover, the elements of A are random elements, independent of the message that* **S** *wants to securely send to* **R***. In* **Extrapolation Technique***, the rest of the rows of matrix D are obtained by fitting t_b degree polynomials to the elements along each column of A, where as in* **Pad Establishment Technique***, the rest of the rows of D are obtained by fitting polynomials of degree $t_b + t_o + t_p$ to the elements along each column of A. The reason for these differences if that* **Extrapolation Technique** *is used in reliable protocol, where there is no issue of secrecy. But* **Pad Establishment Technique** *is used for secure protocol, where at most $t_b + t_o + t_p$ wires can be passively listened by the adversary.*

Since **Pad Establishment Technique** is similar to the **Extrapolation Technique**, all the properties of later holds for the former. For the sake of completeness, we mention them.

Lemma 8 *In D, all the $n = 2t_b + 2t_o + t_f + t_p + 1$ elements of any column can be uniquely generated from any $t_b + t_o + t_p + 1$ elements of the same column.*

Proof: The proof follows using similar argument as in the proof of Lemma 1. □

Lemma 9 *In D, if at most t_b elements along any column are changed, then it can be always detected.*

Proof: The proof follows using similar argument as in Lemma 3. □

We now present our single phase USMT protocol called **USMT_Single_Phase**. Let the message be denoted by $m = (m_1\ m_2\ \ldots\ m_{t_b+t_o+t_f+t_p+1})$ and the set of n wires be denoted as $\mathcal{W} = \{w_1, w_1, \ldots, w_n\}$.

Lemma 10 *In* **USMT_Single_Phase***, if any $w_j \in \mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$ is contradicted by at least $(t_b - |\mathcal{B}|) + 1$ wires in the set $\mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$, then the polynomial $p_j(x)$ over w_j has been changed by adversary or in effect w_j is Byzantine corrupted.*

Proof: The proof is similar to the proof of Lemma 4 and is omitted. □

Lemma 11 *In the protocol, if the adversary corrupts a polynomial over wire w_j in such a way that w_j is not removed during step 2 of message recovery, then* **R** *will always be able to detect it at the end of step 4 of message recovery and outputs "NULL".*

Protocol USMT_Single_Phase - The Single Phase USMT Protocol

Computation and Communication by S

1. **S** selects at random $(t_b + t_o + t_p + 1) \times (n + t_p)$ field elements from $\mathbb{F}$ denoted by $M_{11}, M_{12}, \ldots, M_{1(n+t_p)}, M_{21}, M_{22}, \ldots, M_{2(n+t_p)}, \ldots, M_{(t_b+t_o+t_p+1)1}, M_{(t_b+t_o+t_p+1)2}, \ldots, M_{(t_b+t_o+t_p+1)(n+t_p)}$, which are independent of each other and the secret message m. From these elements **S** generates the rectangular array D containing $n \times (n + t_p)$ field elements using **Pad Establishment Technique**.
2. **S** then forms n polynomials $p_j(x), 1 \leq j \leq n$, each of degree $n - 1 + t_p$ where $p_j(x)$ is formed using the j^{th} row of D as follows: the coefficient of $x^i, 0 \leq i \leq n - 1 + t_p$ in $p_j(x)$ is the $(i + 1)^{th}$ element of j^{th} row of D.
3. **S** chooses another n *secret* and random field elements, $\alpha_1, \alpha_2, \ldots, \alpha_n$, which are independent of the message m and the elements of rectangular array D. Over w_j, **S** sends the following to **R**: the polynomial $p_j(x)$, the secret value α_j and the n tuple $\{p_i(\alpha_j)\}$, for $1 \leq i \leq n$. Let $v_{ji} = p_i(\alpha_j)$.
4. **S** then prepares a list E which consist of coefficients of all n polynomials; i.e., concatenation of the rows of D. **S** finally computes $y = [y_1\ y_2\ \ldots\ y_{t_b+t_o+t_f+t_p+1}] = \text{EXTRAND}_{n(n+t_p),t_b+t_o+t_f+t_p+1}(E)$ and broadcasts $d = m \oplus y$ to **R**.

Message Recovery by R

1. Let $\mathcal{F}$ denotes the set of wires that delivered nothing and let $\mathcal{B}$ denotes the set of wires that delivered invalid information (like higher degree polynomials etc.) to **R**. Note that the wires in $\mathcal{B}$ are Byzantine corrupted because omission or fail-stop controlled wires are not allowed to modify the information passing over them. **R** removes all the wires in $(\mathcal{F} \cup \mathcal{B})$ from $\mathcal{W}$ to work on the remaining wires in $\mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$ out of which at most $t_b - |\mathcal{B}|$ could be Byzantine corrupted.
2. Let **R** receives $p'_j(x)$, α'_j and the n tuple $\{v'_{ji}\}$, $1 \leq i \leq n$ over $w_j \in \mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$. **R** also correctly receives $d = m \oplus y$, which is broadcast by **S**. We say that w_j *contradicts* w_i if: $v'_{ji} \neq p'_i(\alpha'_j)$, where $w_i, w_j \in \mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$. Among all the wires in $\mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$, **R** checks if there is a wire contradicted by at least $(t_b - |\mathcal{B}|) + 1$ wires. All such wires are Byzantine corrupted and removed (see Lemma 10).
3. To retrieve m, **R** needs the vector y, which in turn is constructed from the list E. So to get the list E, **R** tries to reconstruct the array D as generated originally by **S**. Let D' be the array, corresponding to D which **R** tries to recover at his end. D' is constructed as follows: Corresponding to each $w_j \in \mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$, which is not removed in previous step, **R** fills the j^{th} row of D' in the following manner: coefficient of $x^i, 0 \leq i \leq n - 1 + t_p$ in $p'_j(x)$ occupies $(i+1)^{th}$ column in the j^{th} row of D'; i.e., the coefficients of $p'_j(x)$ are inserted in j^{th} row of D' such that the coefficient of x^i in $p'_j(x)$ occupies $(i + 1)^{th}$ column in the j^{th} row of D'.
4. After doing the above step for each $w_j \in \mathcal{W} \setminus (\mathcal{F} \cup \mathcal{B})$, which is not removed in step 2 of message recovery, **R** will have at least $t_b + t_o + t_p + 1$ rows inserted in D' (see Lemma 12). **R** then checks the validity of these rows as follows: let $i_1, i_2, \ldots, i_k, k \geq t_b + t_o + t_p + 1$ denote the index of the rows which are inserted by **R** in D'. Let $y^j_{i_1}, y^j_{i_2}, \ldots, y^j_{i_k}, 1 \leq j \leq n + t_p$ denote the values along $j^{th}, 1 \leq j \leq n$ column of D'. **R** checks whether the points $(i_1, y^j_{i_1}), (i_2, y^j_{i_2}), \ldots, (i_k, y^j_{i_k})$ lie on a $t_b + t_o + t_p$ degree polynomial. Note that at this point, each column will have at least $t_b + t_o + t_p + 1$ elements, which are enough to do the checking.
5. If the above test fails for at least one column of D', then **R** outputs "NULL" and halts. Otherwise, using the already inserted rows of D', **R** regenerates the complete D correctly (see Lemma 12). **R** now knows all the polynomials $p_i(x), 1 \leq i \leq n$ and hence the list E, which is the concatenation of rows of D. **R** then computes $y = [y_1\ y_2\ \ldots\ y_{t_b+t_o+t_f+t_p+1}] = \text{EXTRAND}_{n(n+t_p),t_b+t_o+t_f+t_p+1}(D)$ and recovers m by computing $m = d \oplus y$.

Proof: We consider the worst case, where $t_o + t_f$ wires which are omission and failstop controlled, gets crashed and fail to deliver any information to **R**. Thus **R** gets information over $2t_b + t_o + t_p + 1$ wires, of which at most t_b could be Byzantine corrupted. Also, out of these wires, at least $t_b + t_o + t_p + 1$ are honest and correctly delivered the polynomials and values to **R**. So $t_b + t_o + t_p + 1$ rows corresponding to these correct polynomials will be present in D'. This is because an honest wire which has correctly delivered the polynomial can be contradicted by at most $(t_b - |\mathcal{B}|)$ wires. Hence the honest wires will not be removed by **R** during step 2 of message recovery and so the coefficients of the polynomials corresponding to these wires will be present in D'. Now if a wire w_j which has delivered a faulty polynomial $p'_j(x) \neq p_j(x)$ to **R** is not removed during step 2 of message recovery, then the coefficients of $p'_j(x)$ are inserted in the j^{th} row of D'. Since $p_j(x) \neq p'_j(x)$, there will be at least one (there can be more than one) coefficient in $p'_j(x)$, which is different from the corresponding coefficient in $p_j(x)$. Let $p_j(x)$ differs from $p'_j(x)$ in the coefficient of x^i. Then $(i + 1)^{th}$ column of D' differs from the $(i + 1)^{th}$ column of original D at j^{th} position. Also the $(i + 1)^{th}$ column of D' may differ from the $(i + 1)^{th}$ column of original D in at most

t_b locations (including j^{th} location). This is because in the worst case, out of the $2t_b + t_o + t_p + 1$ wires, the adversary may change the polynomials along at most t_b wires (which are Byzantine corrupted), such that the coefficient of x^i in all these changed polynomials differ from their corresponding coefficient of x^i in the original polynomials. So, in the worst case, at most t_b elements of the $(i+1)^{th}$ column of D' can be different from $(i+1)^{th}$ column of D. The proof now follows from Lemma 9. Hence **R** will detect that at atmost t_b of the received polynomials are incorrect and outputs "NULL". □

Lemma 12 *In* **USMT_Single_Phase***, if the test in step 4 of message recovery succeeds for all the $n+t_p$ columns of D', then* **R** *will never output "NULL" and always recovers m correctly.*

Proof: As explained in previous Lemma, at the beginning of step 4, there will be at least $t_b + t_o + t_p + 1$ correct rows present in D'. Now if the test in step 4 succeeds for all the $n + t_p$ columns of D', it implies that all the rows present in D' are same as the corresponding rows in the original D. From Lemma 8, **R** will be able to completely regenerate all the $n + t_p$ columns of original D and hence recover the original array D. Once D is reconstructed, **R** can easily form the list E consisting the coefficients of all the n polynomials $p_j(x), 1 \leq j \leq n$. **R** then correctly constructs the vector y by applying EXTRAND algorithm to E and recovers m by computing $m = d \oplus y$. It is easy to see that **R** does not outputs "NULL" in this case. □

Theorem 9 *In* **USMT_Single_Phase***, the mixed adversary $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ gains no information about the message m in information theoretic sense.*

Proof: The security of the protocol depends upon the security of the one time pad y which is established between **S** and **R**, which in turn depends upon how much information in the array D is information theoretically secure from $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$. From Lemma 8, D can be completely recovered from any $t_b + t_o + t_p + 1$ rows of D. So if $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ can completely recover any $t_b + t_o + t_p + 1$ of the n $p_i(x)$'s, then adversary will know D and hence y. Without loss of generality, assume that $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ passively listen the wires w_1 to $w_{t_b+t_o+t_p}$ (recall that $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ can passively listen the wires which are under its control in passive, omission and Byzantine fashion). Thus the adversary knows the coefficients of $p_i(x), 1 \leq i \leq t_b + t_o + t_p$ and hence the first $t_b + t_o + t_p$ rows of D. Furthermore the adversary receives $(t_b + t_o + t_p)$ distinct points on each of the polynomials $p_1(x)$ to $p_n(x)$. Specifically, adversary know the values $p_i(\alpha_j)$, where $1 \leq i \leq n$ and $1 \leq j \leq t_b + t_o + t_p$. The points on the polynomials $p_1(x)$ to $p_{t_b+t_o+t_p}(x)$ are already known to the adversary (the adversary knows these polynomials) and hence does not add any new information to adversary's view. On the other hand, $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ fall short of $(n+t_p) - (t_b + t_o + t_p) = t_b + t_o + t_f + t_p + 1$ points on each $p_i(x), t_b + t_o + t_p + 1 \leq i \leq n$ to completely interpolate $p_i(x)$.

Now from Lemma 8, all the elements of any column of D can be derived from any $t_b + t_o + t_p + 1$ elements of the same column. So, the last $n - (t_b + t_o + t_p + 1)$ rows of **D** can always be expressed as a linear combination of the first $t_b + t_o + t_p + 1$ rows of D. Thus, the polynomials $p_{t_b+t_o+t_f+t_p+2}(x)$ to $p_n(x)$ *linearly* depends upon the polynomials $p_1(x)$ to $p_{t_b+t_o+t_p+1}(x)$. So the points on the the polynomials $p_{t_b+t_o+t_p+2}(x)$ to $p_n(x)$ are *linear* combinations of the points on the polynomials $p_1(x)$ to $p_{t_b+t_o+t_p+1}(x)$, which are already known to the adversary and hence can be removed from his view. Hence out of the $t_b + t_o + t_p$ points on each of the n polynomials that are known to $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$, only the points on $p_{t_b+t_o+t_p+1}(x)$ adds new information to adversary's view. For the polynomial $p_{t_b+t_o+t_p+1}(x)$, the adversary knows *only* $t_b + t_o + t_p$ points that are sent through the wires w_1 to $w_{t_b+t_o+t_p}$. However, as shown above, from these many points, adversary will fall short of $t_b + t_o + t_f + t_p + 1$ points to completely know $p_{t_b+t_o+t_p+1}(x)$ and hence D. So overall, $t_b + t_o + t_f + t_p + 1$ elements of D are information theoretic secure. The proof now follows from the correctness of the EXTRAND algorithm. □

Theorem 10 *If $|\mathbb{F}| \geq \frac{2n^3}{\delta}$, then protocol* **USMT_Single_Phase** *is a strong USMT protocol and terminates with a non-"NULL" output with probability at least $1 - \delta$.*

PROOF: From the protocol, it is easy to see that no honest wire (which has delivered correct values and polynomials) can contradict another honest wire. From Lemma 10, all the wires removed by **R** during step 2 of message recovery are indeed faulty. We now need to show that if a wire has delivered incorrect

polynomial, then it will be contradicted by all the honest wires with high probability. Let π_{ij} be the probability that a corrupted wire w_j, which has delivered incorrect $p'_j(x) \neq p_j(x)$ will not be contradicted by an honest wire w_i. This means that the adversary can ensure that $p_j(\alpha_i) = p'_j(\alpha_i)$ with a probability of π_{ij}. Since there are only $n-1+t_p$ points at which these two polynomials intersect (the degree of p_j and p'_j is $n-1+t_p$), this allows the adversary to guess the value of α_i with a probability of at least $\frac{\pi_{ij}}{n-1+t_p}$. But since α_i was selected uniformly in $\mathbb{F}$, the probability of guessing it is at most $\frac{1}{|\mathbb{F}|}$. Therefore we have $\pi_{ij} \leq \frac{n-1+t_p}{|\mathbb{F}|}$ for each i, j. Thus the total probability that the adversary can find w_i, w_j such that corrupted wire w_j will not be contradicted by any honest wire w_i is at most $\sum_{i,j} \pi_{ij} \leq \frac{n^2(n-1+t_p)}{|\mathbb{F}|}$. Now $n^2(n-1+t_p) < n^2(2n) < 2n^3$. Since $|\mathbb{F}| \geq \frac{2n^3}{\delta}$, it follows that corrupted $p'_j(x) \neq p_j(x)$, received over a corrupted wire w_j can be included in D' with probability at most δ. However, if such a $p'_j(x)$ is included in D', then from Lemma 11, **R** will detect this and will output "NULL". Thus protocol **USMT_Single_Phase** is a strong USMT protocol and outputs a non-"NULL" output with probability at least $1-\delta$. □

Theorem 11 USMT_Single_Phase *securely sends $t_b + t_o + t_f + t_p + 1 = \Theta(n)$ field elements by communicating $O(n^2)$ field elements. In terms of bits, the protocol securely sends $(t_b + t_o + t_f + t_p + 1)log|\mathbb{F}| = \Theta(nlog|\mathbb{F}|)$ bits by communicating $O(n^2log|\mathbb{F}|)$ bits. Thus, the protocol is communication optimal.*

Proof: Over each wire, **S** sends a polynomial of degree $n-1+t_p$ and an n tuple, along with a secret value α. Thus the total communication complexity is $n \times (n + t_p + n) = O(n^2)$. Since each field element from field $\mathbb{F}$ can be represented by $\log|\mathbb{F}|$ bits, the communication complexity of the protocol is $O(n^2 log|\mathbb{F}|)$ bits. The protocol securely sends $(t_b + t_o + t_p + t_f + 1) = \Theta(n)$ field elements because if $n = 2t_b + 2t_o + t_f + t_p + 1$, then $t_b + t_o + t_p + t_f + 1 = \Theta(n)$. By substituting $n = 2t_b + 2t_o + t_f + t_p + 1$ and $\ell = \Theta(n)$ in Theorem 8, we get that any single phase USMT protocol need to communicate $\Omega(n^2)$ field elements to securely send $\Theta(n)$ field elements. However, the total communication complexity of our protocol is $O(n^2)$. Hence our protocol is *communication optimal*. □

4.2.1 Single Phase USMT with Constant Factor Overhead Tolerating $\mathcal{A}_{t_b}$

From [8], any single phase PSMT tolerating $\mathcal{A}_{t_b}$ requires $n = 3t_b + 1$ wires between **S** and **R**. Moreover from [11, 36], any single phase PSMT tolerating $\mathcal{A}_{t_b}$ needs to communicate $\Omega(n\ell)$ field elements to securely send a message containing ℓ field elements. We now show that if $n = 3t_b + 1$, then there exists a single phase (strong) USMT protocol with error probability of at most δ, which sends a message containing ℓ field elements by communicating $O(\ell)$ field elements tolerating $\mathcal{A}_{t_b}$. In terms of bits, the protocols securely sends $\ell \log|\mathbb{F}|$ bits by communicating $O(\ell \log|\mathbb{F}|)$ bits, where $|\mathbb{F}|$ is a function of error probability δ. Thus we get security with constant factor overhead in a single phase, with negligible error probability. This is interesting because with $n = 3t+1$ wires, it is impossible to achieve perfect secrecy with constant factor overhead.

If we execute our single phase USMT protocol **USMT_Single_Phase** against only $\mathcal{A}_{t_b}$ over $n = 2t_b+1$ wires (i.e., $t_o = t_f = t_p = 0$), then the protocol securely sends $t_b + 1 = \Theta(n)$ field elements (if $n = 2t_b + 1$, then $t_b = \Theta(n)$) by communicating $O(n^2)$ field elements. However, if $n = 3t_b + 1$, then the same protocol can securely send $\Theta(t_b^2) = \Theta(n^2)$ field elements by communicating $O(n^2)$ field elements. In terms of bits, the USMT protocol will send $\Theta(n^2)\log(|\mathbb{F}|)$ bits by communicating $O(n^2)\log(|\mathbb{F}|)$ bits, where $|\mathbb{F}| \geq \frac{2n^3}{\delta}$. The only change need to be done is in the **Pad Establishment Technique**. Now the array D will be an $(3t_b + 1) \times (3t_b + 1)$ array, where the sub-array A will be of size $(2t_b + 1) \times (3t_b + 1)$ and will consists of $(2t_b + 1) \times (3t_b + 1)$ random elements. The $2t_b + 1$ rows of A will be extrapolated into sub-array C of size $t_b \times (3t_b + 1)$, by fitting $2t_b$ degree polynomials passing through the elements of the individual columns of A. Now in the protocol, **S** will generate a random pad y of length $(t_b + 1) \times (2t_b + 1)$ from the elements of array D and sends a message containing $(t_b + 1) \times (2t_b + 1)$ field elements by using y as an one time pad. The security of y follows from the fact that now $(n - t_b) = 2t_b + 1$ elements along $t_b + 1$ rows of array A will be information theoretically secure from $\mathcal{A}_{t_b}$. The rest of the protocol will remain

same, except that now in D' (array corresponding to D which is reconstructed at $\mathbf{R}$'s end), there will be at least $2t_b + 1$ rows (for $n = 3t_b + 1$, there will be at least $2t_b + 1$ correct and honest wires). To check the validity of the rows inserted in D', $\mathbf{R}$ will check whether the elements along individual columns of D' lie on a $2t_b$ degree polynomial. The rest of the details are same as in protocol **USMT_Single_Phase**. Thus we have the following theorem:

Theorem 12 *If $n = 3t_b + 1$ and $|\mathbb{F}| \geq \frac{2n^3}{\delta}$, then there exists a single phase strong USMT protocol, which securely sends a message containing $\Theta(n^2 \log(|\mathbb{F}|))$ bits by communicating $O(n^2 \log(|\mathbb{F}|))$ bits, with an error probability of at most δ.*

PROOF: Follows from the above discussion. □

4.3 Comparison of Single Phase PSMT with Single phase USMT

The comparison between single phase PSMT and single phase USMT can be listed as follows

- Allowing a negligible error probability only in the reliability, *significantly* helps in the POSSIBILITY of single phase secure message transmission protocols (see Comparison 3).
- Allowing a negligible error probability only in the reliability, *significantly* reduces the communication complexity of single phase secure message transmission protocols (see Comparison 4).

5 Multiphase USMT Tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$

As mentioned earlier, one of the key parameters of any secure message transmission protocol is the number of phases. In the context of PSMT, it is well known that allowing interaction between $\mathbf{S}$ and $\mathbf{R}$ significantly helps in reducing the connectivity requirement and lower bound on communication complexity of PSMT protocols (see Table 3 and Table 4). In this section, we show that same holds for USMT also. In this section, we provide the characterization and lower bound on the communication complexity of any multiphase USMT protocol. We also design a four phase USMT protocol whose total communication complexity matches the proven lower bound, thus showing that the bound is tight. Comparing these results with the results for single phase USMT, we find that allowing interaction between $\mathbf{S}$ and $\mathbf{R}$ significantly helps in the connectivity requirement of USMT and also helps in reducing the communication complexity of USMT protocols.

5.1 Characterization for Multiphase USMT Protocol Tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$

Theorem 13 *Multiphase USMT between $\mathbf{S}$ and $\mathbf{R}$ in an undirected network tolerating a mixed adversary $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ is possible if and only if the network is $(t_b + \max(t_b, t_p) + t_o + t_f + 1)$-($\mathbf{S}$,$\mathbf{R}$)-connected.*

PROOF: *Necessity:* We consider two cases for proving the necessity.

1. **Case 1**: $t_p \leq t_b$: In this case, the necessity condition says that the network should be $(2t_b + t_o + t_f + 1)$-($\mathbf{S}$,$\mathbf{R}$). Since the condition is necessary for URMT (Theorem 2), it is obviously necessary for USMT.
2. **Case 2**: $t_p > t_b$: In this case, the the necessity condition says that the network should be $(t_b + t_p + t_o + t_f + 1)$-($\mathbf{S}$,$\mathbf{R}$)-connected. This condition is necessary for USMT because if the network is $(t_b + t_p + t_o + t_f)$-($\mathbf{S}$,$\mathbf{R}$)-connected, then the adversary may strategize to simply block all message through $(t_b + t_o + t_f)$ vertex disjoint paths and thereby ensure that every value received by $\mathbf{R}$ is also listened by the adversary. This completely rules out the possibility of information-theoretic security.

Sufficiency: Suppose that network is $(t_b+\max(t_b,t_p)+t_o+t_f+1)$-(**S**,**R**)-connected. Then from Menger's theorem [24], there exist at least $n = (t_b + \max(t_b, t_p) + t_o + t_f + 1)$ vertex disjoint paths from **S** to **R**. We model these paths as wires $w_1, w_2, \ldots, w_n$. We now design a three phase USMT protocol called **SECURE** to securely send a single field element $\mathbf{M} \in \mathbb{F}$. The protocol is similar to the USMT protocol of [13].

Protocol SECURE - A Three Phase USMT Protocol

Phase I: S to R

• Along $w_i, 1 \leq i \leq n$, **S** sends to **R** two randomly picked elements ρ_{i1} and ρ_{i2} chosen from $\mathbb{F}$.

Phase II: R to S

• Suppose **R** receives values in syntactically correct form along $n' \leq n$ wires. **R** neglects the remaining $(n - n')$ wires. Let **R** receives ρ'_{i1} and ρ'_{i2} along wire w_i, where w_i is not neglected by **R**.

• **R** chooses uniformly at random an element $K \in \mathbb{F}$. **R** then broadcasts to **S** the following: identities of the $(n - n')$ wires neglected by him, the random K and the values $(K\rho'_{i1} + \rho'_{i2})$ for all i such that w_i is not neglected by **R**.

Phase III: S to R

• **S** correctly receives the identities of $(n - n')$ wires neglected by **R** during **Phase II** (because irrespective of the value of t_b and t_p, n is at least $2t_b + t_o + t_f + 1$. So any information which is broadcast over n wires will be received correctly). **S** eliminates these wires. **S** also correctly receives K and the values, say $u_i = (K\rho'_{i1} + \rho'_{i2})$ for each i, such that wire w_i is not eliminated by **R**.

• **S** then computes the set H such that $H = \{w_i | u_i = (K\rho_{i1} + \rho_{i2})\}$. Furthermore, **S** computes the secret pad ρ where $\rho = \sum_{w_i \in H} \rho_{i2}$. **S** then broadcasts the set H and the blinded message $\mathbf{M} \oplus \rho$ to **R**, where **M** is the single field element, which **S** wants to send securely to **R**.

Message Recovery by R

• **R** correctly receives H and computes his version of ρ'. If z' is the blinded message received, **R** outputs $\mathbf{M} = z' \oplus \rho'$.

It can be shown that with a probability of at least $\left(1 - \frac{1}{|\mathbb{F}|}\right)$, $\rho' = \rho$ and hence **R** almost always learns the correct message (Proof is similar to that of the correctness of the USMT protocol of [13]). Since $n = t_b + \max(t_b, t_p) + t_o + t_f + 1$, there exists at least one wire say w_i, which is not controlled by the adversary. So, the corresponding ρ_{i2} is unknown to adversary implying information theoretic security for $\rho = \sum_{w_i \in H} \rho_{i2}$ and hence for **M**. It is easy to see that the communication complexity of **SECURE** is $O(n^2)$ field elements, where the field size $|\mathbb{F}|$ is set appropriately as a function of δ. □

Comparison 5 (POSSIBILITY **of Multi Phase USMT and PSMT)** *From Table 3 (last row), any $r \geq 2$ phase PSMT protocol tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ is possible iff there exists $n \geq 2t_b+t_o+t_f+t_p+1$ wires between* **S** *and* **R**. *Comparing this with Theorem 13, we find that except when either $t_b = 0$ or $t_p = 0$, allowing a negligible error probability (only in the reliability), significantly helps in the* POSSIBILITY *of multiphase secure message transmission protocol.*

Comparison 6 (Communication Complexity of Multi Phase USMT and PSMT) *From Table 3 (last row), any $r \geq 2$ phase PSMT protocol tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ needs to communicate $\Omega(\frac{n\ell}{n-(2t_b+t_0+t_f+t_p)})$ field elements to securely send a message containing ℓ field elements, where* **S** *and* **R** *are connected by $n \geq 2t_b+t_o+t_f+t_p+1$ wires. Comparing this with Theorem 13, we find allowing a negligible error probability (only in the reliability), significantly helps in reducing the communication complexity of multiphase secure message transmission protocols.*

The protocol **SECURE** is used to prove the sufficiency of Theorem 13. Using it as a black-box, we will design a more communication efficient multiphase USMT protocol. Before that, we prove the lower bound on the communication complexity of any multiphase USMT protocol in the next section.

5.2 Lower Bound on the Communication Complexity of Multiphase USMT Protocol Tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$

We now prove the lower bound on the communication complexity of any r-phase ($r \geq 2$) USMT protocol which sends ℓ field elements tolerating a mixed adversary $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$. Let $n \geq t_b+\max(t_b,t_p)+t_o+t_f+1$. To prove the lower bound, we use entropy based argument, which is used in [36] for proving the lower bound on the communication complexity of PSMT protocols.

Before proving the lower bound, we briefly recall the capabilities of $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$. A Byzantine corrupted wire is *actively* controlled by the adversary. Thus the adversary fully controls a Byzantine corrupted wire and he can even block such a wire. However, the *most adverse affect* caused by a Byzantine corrupted wire is when the adversary maliciously changes the information passed over such a wire. If the adversary simply blocks a wire which is controlled in Byzantine fashion, then the adversary is not using its true capability. Also, if the adversary blocks a Byzantine controlled wire, instead of maliciously changing the information passing through such a wire, then both **S** and **R** will come to know the identity of the blocked wire and will remove it from the protocol. Similarly, the most adverse affect caused by a omission controlled wire is when the adversary passively listen such a wire. Instead, if the adversary blocks such a wire (omission controlled wire can also be blocked by the adversary), then again both **S** and **R** will come to know the identity of the wire and it will be removed from the protocol. While proving the lower bound on the communication complexity, we assume that $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ will fully utilize its capability. Thus we assume that the adversary either eavesdrop or maliciously change the information passing through the wires which are controlled in Byzantine fashion. Similarly, instead of blocking omission controlled wires, the adversary only eavesdrop such wires. Thus, without loss of generality, we assume that out of the n wires, $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ controls at most b, F and P wires in Byzantine, failstop and passive fashion respectively, where $b \leq t_b, F \leq t_f$ and $P \leq t_b + t_o + t_p$.

Theorem 14 *Any r-phase ($r \geq 2$) USMT protocol which securely sends ℓ field elements in the presence $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ needs to communicate $\Omega\left(\frac{n\ell}{n-(t_b+t_o+t_f+t_p)}\right)$ field elements.*

Remark 7 *In terms of bits, any multiphase USMT protocol must communicate $\Omega\left(\frac{n\ell}{n-(t_b+t_o+t_f+t_p)} \log|\mathbb{F}|\right)$ bits to securely send $\ell \log|\mathbb{F}|$ bits, where $|\mathbb{F}|$ is a function of δ (the probability of error in the reliability). In the next section, we give a concrete communication optimal USMT protocol satisfying this bound and show how to set $|\mathbb{F}|$ as a function of δ.*

Proof: The proof follows from Lemma 13 and Lemma 14, which are proved below.

Lemma 13 *The communication complexity of any multi-phase USMT protocol to send a message against an adversary corrupting up to $b(\leq t_b), F(\leq t_f)$ and $P(\leq t_b + t_o + t_p)$ of the wires in Byzantine, Fail-stop and passive manner respectively is not less than the communication complexity of distributing n shares for the message such that any set of $n - F$ correct shares has full information about the message while any set of P shares has no information about the message.*

To prove the lemma, we begin with defining a weaker version of single-phase USMT called USMT with Error Detection (USMTED). We then prove the equivalence of communication complexity of USMTED protocol to send message **M** and the share complexity of distributing n shares for **M** such that any set of $n-F$ correct shares has full information about **M** while any set of P shares has no information about **M**. To prove the aforementioned statement, we show their equivalence (Claim 1). Finally, we will show that communication complexity of any multiphase USMT protocol is at least as same as the communication complexity of single-phase protocol USMTED (Claim 3). These two equivalence will prove the desired equivalence as stated in this lemma. Note that b, F and P are bounded by t_b, t_f and $t_b+t_o+t_p$ respectively.

Definition 13 *A single phase USMT protocol is called USMTED if it satisfies the following properties:*

1. *If the adversary is passive on P wires then **R** correctly and securely receives the message sent by **S**.*

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

3. *If adversary crashes $F \leq t_f$ wires and does no malicious corruption, then* $\mathbf{R}$ *recovers message correctly. Else if adversary either crashes more than t_f wires or do some malicious modifications (or both), then* $\mathbf{R}$ *detects it and aborts.*

4. *The adversary obtains no information about the transmitted message in information theoretic sense.*

We next show that the properties of USMTED protocol for sending message $\mathbf{M}$ is equivalent to the problem of distributing n shares for $\mathbf{M}$ such that any set of $n - F$ correct shares has full information about $\mathbf{M}$ while any set of P shares has no information about $\mathbf{M}$.

Claim 1 *Let* Π *be a USMTED protocol executed over n wires between* $\mathbf{S}$ *and* $\mathbf{R}$*. 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 the wires $w_i, 1 \leq i \leq n$, form n shares for* $\mathbf{M}$ *such that any set of $n - F$ correct shares has full information about* $\mathbf{M}$ *while any set of P shares has no information.*

Proof: The fact that any set of P shares have no information about $\mathbf{M}$ follows directly from property 1 and 4 of definition of USMTED. We now show that any set of $n - F$ correct 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 USMTED to send $\mathbf{M}$. Now for any set C of honest wires with $|C| \geq n - F$, $Message(\mathbf{M}, C)$ should uniquely determine the message $\mathbf{M}$. Suppose not, then there exists another message $\mathbf{M}'$ such that $Message(\mathbf{M}, C) = Message(\mathbf{M}', C)$. By definition the fail-stop controlled wires can block all the messages sent along the F wires not in C. Thus for two different executions of USMTED to send two distinct message $\mathbf{M}$ and $\mathbf{M}'$, there exists an adversary strategy such that view of $\mathbf{R}$ at the end of two executions is exactly same. This is a contradiction to the property 3 of USMTED protocol Π, which must output the correct message if at most F fail-stop errors and no malicious corruption take place. □

The above claim also says that the communication complexity of USMTED 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 $n - F$ correct shares has full information about $\mathbf{M}$ while any set of P shares has no information about the message. Now we step forward to show that the communication complexity of USMTED protocol is the lower bound on the communication complexity of any multiphase USMT protocol.

Before that we take a closer look at the execution of any multi-phase USMT protocol. $\mathbf{S}$ 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 USMT protocol Π is the concatenation of all the messages sent by $\mathbf{S}$ and $\mathbf{R}$ along all the wires.

Definition 14 *A passive transcript* $\mathcal{T}(\Pi, \mathbf{M}, r_1, r_2)$ *is a transcript for the execution of the multiphase USMT 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 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, $\mathcal{T}_B$ denote the passive transcript restricted to the 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 random tape r_1. Note that although $\mathbf{S}$ and $\mathbf{R}$ require both r_1, r_2 to generate the transcript, $\mathbf{R}$ requires only r_2 in order to obtain the message $\mathbf{M}$ from the transcript $\mathcal{T}(\Pi, \mathbf{M}, r_1, r_2)$. 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.

We next define a special type of passive transcript and prove its properties.

Definition 15 *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 USMT 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 16 *Two 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, where $|A| \leq b + (|B| - (n - F))$, such that the two transcripts differ only on messages sent along the wires in A. Without loss of generality, assume that the 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 block 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 block the messages over the wires in 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 $\mathbf{R}$'s random tape. In both the scenarios $\mathbf{S}$ wanted to send message $\mathbf{M}$. But the message received by receiver $\mathbf{R}$ in the second case is an incorrect message $\mathbf{M}'$. Thus, with only probability $1/2$, $\mathbf{R}$ will output the correct message $\mathbf{M}$. This is a contradiction because Π is a USMT protocol. □

Till now, we have shown that a transcript over at least $n - F$ correct wires allows $\mathbf{R}$ to output $\mathbf{M}$ correctly. We now show how to reduce a multiphase USMT protocol into a single phase USMTED protocol.

Protocol USMTED

• $\mathbf{S}$ 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 $\mathbf{R}$ along w_i.

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

Output ERROR.

Claim 3 *The Communication complexity of any multiphase USMT protocol* Π *to send* $\mathbf{M}$ *is at least as same as the communication complexity of* **USMTED** *protocol. Moreover protocol* **USMTED** *satisfies the properties given in Definition 13.*

PROOF: Let Π be any multiphase USMT protocol and $\Pi^{passive}$ denotes an execution of Π where the adversary does only eavesdropping and do no other type of corruption during the complete execution. It is easy to see that the communication complexity of $\Pi^{passive}$ is trivially a lower bound on the communication complexity of any multiphase USMT protocol (where the adversary may do other type of corruption,

in addition to eavesdropping). We now show that the communication complexity of $\Pi^{passive}$ is same as the communication complexity of **USMTED** protocol. Once we do this, then the communication complexity of **USMTED** protocol is a trivial lower bound on the communication complexity of any multiphase USMT protocol.

In **USMTED**, **S** assumes its random tape to contain r_1 and **R**'s random tape to contain r_2. **S** also assumes that in Π, the adversary will only do eavesdropping and no other type of corruption and generates the passive transcript $\mathcal{T}(\Pi, \mathbf{M}, r_1, r_2)$. As explained earlier, **S** can do so by simulating **R**, assuming the content of **R**'s random tape to be r_2. However, note that **R** neither knows **M**, nor r_1, r_2, which **S** has used for generating $\mathcal{T}$. **S** then communicates $\mathcal{T}$ to **R**, by sending the components of $\mathcal{T}$ restricted to wire w_i, along w_i. It is easy to see that the cost of communicating such a transcript by **USMTED** is same as the communication complexity of $\Pi^{passive}$.

The messages sent along wire w_i in **USMTED** protocol is the concatenation of the messages that would have been exchanged between **S** and **R** along w_i in $\Pi^{passive}$. Since $\Pi^{passive}$ is a special type of execution of USMT protocol Π, by the secrecy property of Π, the adversary cannot obtain any information about the message **M** by passively listening $P \leq t_b + t_o + t_p$ wires in **USMTED** protocol. From Claim 2, 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**. Thus protocol **USMTED** satisfies the properties given in Definition 13. □

Claim 1, along with Claim 3 completes the proof of Lemma 13. We now prove the share complexity of distributing n shares for a message such that any set of $n - F$ correct shares has full information while any set of P shares has no information about the message

Lemma 14 *The share-complexity (that is the sum of length of all shares) of distributing n shares for a message of size ℓ field elements from $\mathbb{F}$ such that any set of $n - F$ correct shares has full information about the message while any set of P shares has no information about the message is $\Omega(\frac{n\ell}{(n-F-P)})$.*

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 secret **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 [5]. Since **M** is a value drawn uniformly at random from $\mathbb{F}^\ell$, we have $H(\mathbf{M}) = \ell$. Since any set B consisting of $n - F$ correct shares has full information about **M**, we have $H(\mathbf{M}|X_B) = 0$. Consider any subset $A \subset B$ such that $|A| = P$. Since any set of P shares has no information about **M**, we have $H(\mathbf{M}|X_A) = H(\mathbf{M})$. From the chain rule of the entropy [5], 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 $X_1 = \mathbf{M}|X_A$ and $X_2 = X_{B-A}$, we get

$$H(\mathbf{M}|X_A, X_{B-A}) \quad = \quad H(X_{B-A}) + H(\mathbf{M}|X_A|X_{B-A})$$

From the properties of joint entropy [5], 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, $H(\mathbf{M}|X_A, X_{B-A}) \geq H(\mathbf{M}|X_A)$. Substituting in the above equation, we get

$$\begin{aligned} H(\mathbf{M}|X_A) &\leq H(\mathbf{M}|X_A|X_{B-A}) + H(X_{B-A}) \\ &\leq 0 + H(X_{B-A}) \text{ because } \mathbf{M} \text{ can be known completely from} X_A \text{ and } X_{B-A} \end{aligned}$$

Consequently, $H(\mathbf{M}) \leq H(X_{B-A})$ because $H(\mathbf{M}|X_A) = H(\mathbf{M})$. Since $|B| = n - F$ and $|A| = P$, we get $|B - A| = n - F - P$. So for any set C of size $|B - A| = n - F - P$,

$$H(X_C) \geq H(\mathbf{M}) \Rightarrow \sum_{i \in C} H(X_i) \geq H(\mathbf{M})$$

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

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

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

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

This implies that $\sum_{i=1}^{n} H(X_i) \geq \frac{n\ell}{n-F-P}$ because $H(\mathbf{M}) = \ell$. But $\sum_{i=1}^{n} H(X_i)$ denotes the share complexity of $\mathbf{M}$. Thus the share-complexity for any $\mathbf{M} \in \mathbb{F}^{\ell}$ is $\Omega\left(\frac{n\ell}{n-F-P}\right)$. □

Since $P \leq t_b + t_o + t_p$ and $F \leq t_f$, $\Omega\left(\frac{n\ell}{n-F-P}\right) = \Omega\left(\frac{n\ell}{n-(t_b+t_o+t_f+t_p)}\right)$. Theorem 14 now follows from Lemma 13 and Lemma 14. □

5.3 Upper Bound on the Communication Complexity of MultiPhase USMT Protocol Tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$

Here we design a *communication optimal* multiphase USMT protocol called **USMT_Mixed** tolerating mixed adversary. The protocol terminates in four phases and uses the three phase **SECURE** protocol (described in Theorem 13) as a black-box. If $t_p \geq t_b$, then the protocol securely sends n^2 field elements by communicating $O(n^3)$ field elements and if $t_b > t_p$, then $(t_b - t_p)n^2$ field elements by communicating $O(n^3)$ field elements where $n = t_b + \max(t_b, t_p) + t_o + t_f + 1$. This shows that the lower bound proved in Theorem 14 is tight. In the protocol, depending upon whether $t_b \leq t_p$ or $t_p < t_b$, the field size $|\mathbb{F}|$ is set to at least $\frac{3n^2}{\delta}$ or $\frac{4n^4(t_b-t_p)}{\delta t_b}$ respectively, where δ is the error probability of the protocol. Our four phase USMT protocol has a special property that when executed *only* under the presence of Byzantine adversary (i.e., $t_o = t_f = t_p = 0$), it securely sends ℓ field elements by communicating $O(\ell)$ field elements. Thus it achieves security with "constant factor overhead".

Remark 8 *Since $n = t_b + \max(t_b, t_p) + t_o + t_f + 1$, we can use* **SECURE** *protocol as a black-box in the four phase USMT protocol. We cannot use any single phase USMT protocol as a black-box because the connectivity requirement for single phase USMT is much more than $t_b + \max(t_b, t_p) + t_o + t_f + 1$. We require the* **SECURE** *protocol to securely send (with very high probability) certain values in our four phase USMT protocol.*

Theorem 15 *By setting $|\mathbb{F}| \geq \frac{3n^2}{\delta}$ (if $t_p \geq t_b$) or $|\mathbb{F}| \geq \frac{4n^4(t_b-t_p)}{\delta t_b}$ (if $t_b > t_p$), protocol* **USMT_Mixed** *reliably transmits the message $\mathbf{M}$ with probability at least $1-\delta$.*

Proof. For ease of understanding, we first prove the theorem when $t_b > t_p$. So $|\mathbb{F}| \geq \frac{4n^4(t_b-t_p)}{\delta t_b}$. It is evident from the protocol construction that the theorem holds if the following are true:

1. For all $1 \leq i \leq n$, $\rho_i' = \rho_i$ with probability $\geq (1 - \frac{\delta}{4})$.
2. For all $1 \leq i \leq n$, $y_i' = y_i$ with probability $\geq (1 - \frac{\delta}{4})$.
3. If the wire w_i were indeed Byzantine corrupt (i.e., the n^2 tuple sent over w_i is changed by the adversary), then $w_i \in L_{fault}$ with probability $\geq (1 - \frac{\delta}{4})$.
4. The protocol **URMT_Single_Phase** successfully sends the vector d with probability $\geq (1 - \frac{\delta}{4})$.

Protocol USMT_Mixed

A Communication Optimal 4-Phase USMT Protocol Tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$

The message $\mathbf{M}$ is a sequence of n^2 field elements if $t_b \leq t_p$, otherwise it is a sequence of $(t_b - t_p)n^2$ field elements.

Phase I ($\mathbf{R}$ to $\mathbf{S}$)

- $\mathbf{R}$ selects at random n^3 elements, r_{ij}, $1 \leq i \leq n, 1 \leq j \leq n^2$ from field $\mathbb{F}$. $\mathbf{R}$ also randomly selects $\rho_1, \rho_2, \ldots \rho_n$ from $\mathbb{F}$.
- $\mathbf{R}$ computes $y_i = \sum_{j=1}^{n^2} \rho_i^j r_{ij}, 1 \leq i \leq n$. Note that ρ_i^j is j^{th} power of ρ_i.
- $\mathbf{R}$ sends to $\mathbf{S}$ over $w_i, 1 \leq i \leq n$, the n^2 field elements $r_{ij}, 1 \leq j \leq n^2$. $\mathbf{R}$ also sends $\rho_i, y_i, 1 \leq i \leq n$ to $\mathbf{S}$ using $2n$ parallel invocations of the three phase **SECURE** protocol (described in Theorem 13) as there are total $2n$ elements to send. Hence **Phase I, II** and **Phase III** are used to do $2n$ parallel executions of **SECURE** protocol.

Phase IV ($\mathbf{S}$ to $\mathbf{R}$)

- Let $\mathbf{S}$ receives $r'_{ij}, 1 \leq j \leq n^2$ along wire w_i. $\mathbf{S}$ adds w_i to a list $L_{erasure}$, if $\mathbf{S}$ does not receive any information over w_i.
- Let $\mathbf{S}$ receives ρ'_i and $y'_i, 1 \leq i \leq n$ after the $2n$ parallel executions of the three phase **SECURE** protocol initiated by $\mathbf{R}$. For each i, such that $w_i \notin L_{erasure}$, $\mathbf{S}$ verifies whether $y'_i \stackrel{?}{=} \sum_{j=1}^{n^2} \rho_i'^j r'_{ij}$. If $y'_i \neq \sum_{j=1}^{n^2} \rho_i'^j r'_{ij}$, then $\mathbf{S}$ adds wire w_i to the set of faulty wires, denoted by L_{faulty}. $\mathbf{S}$ sets $L_{honest} = \mathcal{W} \setminus (L_{faulty} \cup L_{erasure})$. If $t_p \geq t_b$, then $\mathbf{S}$ computes a random pad $Z = (z_1, z_2, \ldots, z_{n^2})$ of size n^2 field elements from the $n^2|L_{honest}|$ field elements which are received over the wires in L_{honest} as follows:

  $$Z = EXTRAND_{n^2|L_{honest}|,n^2}(r'_{ij}|w_i \in L_{honest})$$

  . However, if $t_b > t_p$, then $\mathbf{S}$ computes a random pad Z of length $(t_b - t_p)n^2$ as follows:

  $$Z = EXTRAND_{n^2|L_{honest}|,(t_b-t_p)n^2}(r'_{ij}|w_i \in L_{honest})$$

  .
- $\mathbf{S}$ computes $d = \mathbf{M} \oplus Z$. If $t_p \geq t_b$ then d is of size n^2, so $\mathbf{S}$ broadcasts d to $\mathbf{R}$. On the other hand, if $t_b > t_p$ then d consists of $(t_b - t_p)n^2$ field elements and $\mathbf{S}$ reliably sends d to $\mathbf{R}$ by invoking $\frac{(t_b-t_p)}{t_b} * n$ parallel executions of single phase **URMT_Single_Phase** protocol (This is possible because n is at least $2t_b + t_o + t_f + 1$, which is sufficient for single phase URMT. Since **URMT_Single_Phase** protocol reliably sends nt_b field elements, vector d consisting of $(t_b - t_p)n^2$ field elements can be communicated by $\mathbf{S}$ by invoking the single phase URMT protocol $\frac{(t_b-t_p)}{t_b} * n$ times). $\mathbf{S}$ also broadcasts the set L_{faulty} and $L_{erasure}$ to $\mathbf{R}$.

Message recovery by R. $\mathbf{R}$ correctly receives L_{faulty} and $L_{erasure}$ and sets $L_{honest} = \mathcal{W} \setminus (L_{faulty} \cup L_{erasure})$. $\mathbf{R}$ correctly receives d with certainty (probability one) when $t_p \geq t_b$ and with high probability when $t_b > t_p$. If $t_b \leq t_p$, then $\mathbf{R}$ computes $Z^{\mathbf{R}} = (z_1, z_2, \ldots, z_{n^2})$ of size n^2 field elements as follows:

$$Z^{\mathbf{R}} = EXTRAND_{n^2|L_{honest}|,n^2}(r_{ij}|w_i \in L_{honest}).$$

If $t_b > t_p$, then $\mathbf{R}$ computes $Z^{\mathbf{R}}$ of size $(t_b - t_p)n^2$ field elements as follows:

$$Z^{\mathbf{R}} = EXTRAND_{n^2|L_{honest}|,(t_b-t_p)n^2}(r_{ij}|w_i \in L_{honest}).$$

Once $Z^{\mathbf{R}}$ is computed, $\mathbf{R}$ recovers $\mathbf{M}$ by computing $\mathbf{M} = Z^{\mathbf{R}} \oplus d$.

The error probability of the protocol depends upon the error probability of the first four events. If each of the above are true, then the protocol's failure probability is bounded by δ. We now prove that each of the above four conditions are true.

Claim 4 *In* **USMT_Mixed**, *for all* $1 \leq i \leq n$, $\rho'_i = \rho_i$ *with probability* $\geq (1 - \frac{\delta}{4})$.

PROOF: In **USMT_Mixed**, ρ_i's are sent using n parallel execution of the three phase **SECURE** protocol. From the proof of Theorem 13, the error probability of a single execution of **SECURE** protocol is at most $\frac{1}{|\mathbb{F}|}$. Hence the total error probability of n parallel executions of **SECURE** to communicate ρ_i, $1 \leq i \leq n$ is at most $\frac{n}{|\mathbb{F}|}$. If $|\mathbb{F}| \geq \frac{4n}{\delta}$, then the total error probability of n parallel executions of **SECURE** is at most $\frac{\delta}{4}$. Since, $|\mathbb{F}| \geq \frac{4n^4(t_b-t_p)}{\delta t_b} > \frac{4n}{\delta}$, the claim holds. □

Claim 5 *In* **USMT_Mixed**, *for all* $1 \leq i \leq n$, $y'_i = y_i$ *with probability* $\geq (1 - \frac{\delta}{4})$.

PROOF: Similar to the proof of the previous claim. □

Claim 6 *In* **USMT_Mixed**, *if wire w_i is corrupted (i.e., at least one of the value $r_{ij}, 1 \leq j \leq n^2$ is changed by the adversary) and for all i, $\rho'_i = \rho_i$ then $w_i \in L_{fault}$ with probability $\geq (1 - \frac{\delta}{4})$.*

PROOF. From the security argument of **SECURE** protocol, the adversary gains no information about ρ_i, y_i for all $1 \leq i \leq n$. Assume that the adversary has changed the n^2 tuple over some wire w_i. Thus, at least one of the n^2 r'_{ij}'s received by **S** over w_i is different from the corresponding original r_{ij}. Moreover, assume that w_i is not marked as faulty by **S**. This implies that $y_i = \sum_{j=1}^{n^2} \rho_i^j r_{ij} = \sum_{j=1}^{n^2} \rho_i^j r'_{ij} = y'_i$. As inferred by the expression, y_i and y'_i are the y-values (evaluated at $x = \rho_i$) of the polynomials of degree n^2 constructed using $r_{ij},\ 1 \leq j \leq n^2$ and $r'_{ij},\ 1 \leq j \leq n^2$ as coefficients respectively. Since the two polynomials (constructed using r_{ij}'s and r'_{ij}'s as coefficients) are of degree n^2, there can be at most n^2 such ρ_i's, at which the two polynomials can have the same value. So, if the adversary can correctly guess one of these n^2 ρ_i's, then w_i will not be marked as faulty by **S**. However, ρ_i is chosen uniformly by **R** from $\mathbb{F}$. Thus, with probability at most $\frac{n^2}{|\mathbb{F}|}$, the protocol fails to detect the faulty wire. In order that this error probability is at most $\frac{\delta}{4}$, we require $|\mathbb{F}|$ to be at least $\frac{4n^2}{\delta}$. Since, $|\mathbb{F}| \geq \frac{4n^4(t_b - t_p)}{\delta t_b} > \frac{4n^2}{\delta}$, the claim holds. □

Claim 7 *In* **USMT_Mixed**, *the single phase URMT protocol* **URMT_Single_Phase** *which is parallely executed $\frac{n(t_b - t_p)}{t_b}$ times to reliably send d, fails with probability at most $\frac{\delta}{4}$.*

PROOF: In **USMT_Mixed**, if $t_b > t_p$, then d is sent during **Phase IV** using $\frac{n(t_b - t_p)}{t_b}$ parallel executions of **URMT_Single_Phase** protocol. If δ' is the failure probability of a single execution of **URMT_Single_Phase**, then the total failure probability to send d is at most $\frac{n(t_b - t_p)\delta'}{t_b}$. To obtain $\frac{n(t_b - t_p)\delta'}{t_b} \leq \frac{\delta}{4}$, we require $\delta' \leq \frac{\delta t_b}{4n(t_b - t_p)}$. Now from Theorem 4, if $|\mathbb{F}| = \frac{n^3}{\delta'}$ then the error probability of **URMT_Single_Phase** is at most δ'. So in order that the error probability of **URMT_Single_Phase** is at most $\delta' \leq \frac{\delta t_b}{4n(t_b - t_p)}$, we require $|\mathbb{F}| \geq \frac{4n^4(t_b - t_p)}{\delta t_b}$, which is true. Hence the claim follows. □

Thus Theorem 15 is true if $t_b > t_p$ and $|\mathbb{F}| \geq \frac{4n^4(t_b - t_p)}{\delta t_b}$. If $t_p \geq t_b$, then **USMT_Mixed** will have an error probability of at most δ, if the error probability of each of first three events mentioned in Theorem 15 is at most $\frac{\delta}{3}$. This is because 4^{th} event does not occur, as d is broadcasted in this case during **Phase IV**, instead of sending it by single phase URMT. It is easy to check that by setting $|\mathbb{F}| \geq \frac{3n^2}{\delta}$, the theorem holds for $t_b \leq t_p$. □

Remark 9 *From Theorem 15, the field size should be either $\frac{3n^2}{\delta}$ or $\frac{4n^4(t_b - t_p)}{\delta t_b}$. However, in* **USMT_Mixed**, *during* **Phase I**, **R** *needs to select n^3 random field elements from $\mathbb{F}$. So depending upon δ, we will set the field size as $max(n^3, \frac{3n^2}{\delta})$. Setting field size like this will not affect the working of the protocol.*

Theorem 16 *In* **USMT_Mixed**, *the adversary learns no information about the message* **M** *in information theoretic sense.*

PROOF: To begin with, we first note than from the security of **SECURE** protocol, all the n ρ_i's and y_i's are information theoretically secure. The proof is now divided into the following two cases:

1. **Case I: If $t_p \geq t_b$**: In this case, $n = t_b + t_p + t_o + t_f + 1$. In the worst case, the adversary can passively listen the contents over $t_b + t_o + t_p$ wires and block t_f wires. So there will be only one honest wire w_i and hence the adversary will have no information about the n^2 random elements sent over w_i. In this case, **S** generates a random pad of length n^2 and sends **M** containing n^2 field elements, using this pad. The proof follows from the correctness of EXTRAND algorithm and working of the protocol.

2. **Case II: If $t_b > t_p$**: In this case, $n = 2t_b + t_o + t_f + 1$. In the worst case, the adversary can passively listen the contents of at most $t_b + t_p + t_o$ wires and block t_f wires. So there are at least $(t_b - t_p)$ wires which

are not under the control of the adversary and hence the adversary will have no information about the n^2 random elements sent over these wires. In this case, **S** generates a random pad of length $(t_b - t_p)n^2$ and sends **M** containing $(t_b - t_p)n^2$ field elements, using this pad. The proof now follows from the correctness of EXTRAND algorithm and working of the protocol. □

Theorem 17 *The communication complexity of* **USMT_Mixed** *is $O(n^3)$ field elements.*

PROOF: During **Phase I**, **R** sends n^2 random field elements over each of the n wires causing a communication complexity of $O(n^3)$ field elements. **R** also invokes $2n$ parallel executions of **SECURE** protocol, each having a communication complexity of $O(n^2)$ field elements (see Theorem 13). This incurs total communication overhead of $O(n^3)$ field elements. During **Phase IV**, **S** sends d to **R**. If $t_p \geq t_b$, then d will consist of n^2 field elements and hence broadcasting it to **R** incurs a communication complexity of $O(n^3)$. On the other hand, if $t_b > t_p$, d consist of $(t_b - t_p)n^2$ field elements. In this case, **S** will send d by invoking $\frac{(t_b - t_p)}{t_b} * n$ parallel executions of single phase URMT protocol. Since, each execution of the single phase URMT protocol has a communication complexity of $O(n^2)$ field elements (see Theorem 5), total communication complexity for sending d is $O\left(\frac{(t_b - t_p) * n^3}{t_b}\right)$, which is $O(n^3)$. Thus, overall communication complexity of **USMT_Mixed** is $O(n^3)$ field elements. □

Theorem 18 **USMT_Mixed** *is a four phase communication optimal USMT protocol tolerating $\mathcal{A}_{(t_b, t_o, t_f, t_p)}$.*

PROOF: **USMT_Mixed** sends $(t_b - t_p)n^2 \log |\mathbb{F}|$ bits (if $t_b > t_p$) or $n^2 \log |\mathbb{F}|$ bits (if $t_b \leq t_p$), by communicating $O(n^3 \log |\mathbb{F}|)$ bits, where $|\mathbb{F}|$ is either $\frac{4n^4(t_b - t_p)}{\delta t_b}$ (if $t_b > t_p$) or $\frac{3n^2}{\delta}$ (if $t_p \geq t_b$) and $n = t_b + \max(t_b, t_p) + t_o + t_f + 1$. From Theorem 14, if $t_b \geq t_p$ (in this case n will be $2t_b + t_o + t_f + 1$), then any four phase USMT protocol needs to communicate $\Omega(n^3 \log |\mathbb{F}|)$ bits to securely send $(t_b - t_p)n^2 \log |\mathbb{F}|$ bits. Similarly, if $t_p \geq t_b$ (in this case, n will be $t_b + t_p + t_o + t_f + 1$), then any four phase USMT protocol need to communicate $\Omega(n^3 \log |\mathbb{F}|)$ bits in order to securely send $n^2 \log |\mathbb{F}|$ bits. Since total communication complexity of **USMT_Mixed** in both cases is $O(n^3 \log |\mathbb{F}|)$ bits, our protocol is *communication optimal.* □

Corollary 2 *If protocol* **USMT_Mixed** *is executed only under the presence of Byzantine adversary (i.e., $t_o = t_f = t_p = 0$), then it achieves security with "constant factor overhead" in four phases by securely sending $\Theta(n^3)$ field elements with a communication overhead of $O(n^3)$ field elements.*

PROOF: In **USMT_Mixed**, if $t_o = t_p = t_f = 0$, then it sends $t_b n^2 = \Theta(n^3)$ field elements in four phases by communicating $O(n^3)$ field elements (if $t_o = t_f = t_p = 0$, then $n = 2t_b + 1$ and so $t_b = \Theta(n)$). Thus we get *secrecy* with *constant* factor overhead in four phases when **USMT_Mixed** is executed under the presence of *only* Byzantine adversary. □

According to Corollary 2, protocol **USMT_Mixed** is able to securely send a message with constant factor overhead in four phases against a t_b active Byzantine adversary, where the size of the message is $n^2 t_b$. However, it is possible to design a two phase USMT protocol, which achieves security with constant factor overhead under the presence of Byzantine adversary. We design one such protocol in the next section.

5.4 Two Phase USMT with Constant Factor Overhead Tolerating $\mathcal{A}_{t_b}$

The connectivity requirement for any multiphase tolerating only a t_b-active Byzantine adversary is $n \geq 2t_b + 1$ (by substituting $t_o = t_f = t_p = 0$ in Theorem 13). We now design a two phase USMT protocol called **USMT_Byzantine**, where **S** and **R** are connected by $n = 2t_b + 1$ wires. The protocol securely sends $n(t_b+1) = \Theta(n^2)$ field elements by communicating $O(n^2)$ field elements against a t_b active Byzantine adversary. Thus we get security with "constant factor" overhead in two phases. Thus the protocol is both communication and phase optimal. We denote the message by $m = (m_1\ m_2\ \ldots\ m_{n(t_b+1)})$. In our protocol, we use following two protocols as black-box.

1. Protocol **URMT_Single_Phase**: Described in section 3.3, which reliably sends $n(t_b+1) = \Theta(n^2)$ field elements by communicating $O(n^2)$ field elements, against a t_b-active Byzantine adversary, where **S** and **R** are connected by $n = 2t_b + 1$ wires (by substituting $t_o = t_f = t_p = o$ in protocol **URMT_Single_Phase**).

2. Protocol **USMT_Single_Phase**: Described in the section 4.2, which securely sends (t_b+1) field elements by communicating $O(n^2)$ field elements against a t_b-active Byzantine adversary, where **S** and **R** are connected by $n = 2t_b+1$ wires (by substituting $t_o = t_f = t_p = 0$ in **USMT_Single_Phase**).

Protocol USMT_Byzantine: A Two Phase USMT Protocol Tolerating t_b-active Byzantine Adversary

Phase I (**R** to **S**)

- **R** selects at random n^2 random elements, say r_{ij}, $1 \leq i, j \leq n$, which are independent of each other and m from the finite field $\mathbb{F}$. **R** also randomly selects $\rho_1, \rho_2, \ldots \rho_n$ from $\mathbb{F}$ and computes $y_i = \sum_{j=1}^{n} \rho_i^j r_{ij}$. Note that ρ_i^j is j^{th} power of ρ_i.
- Through wire w_i, **R** sends the n field elements $r_{i1}, r_{i2}, \ldots r_{in}$ to **S**. **R** also securely sends ρ_i, y_i for all $1 \leq i \leq n$ to **S**, using four parallel invocations of the single phase **USMT_Single_Phase** protocol (by considering $t_o = t_f = t_p = 0$ and $n = 2t_b + 1$).

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

- Let **S** receive the values $r'_{ij}, 1 \leq j \leq n$ along the wire $w_i, 1 \leq i \leq n$. Also let **S** receive ρ'_i and $y'_i, 1 \leq i \leq n$ after the parallel execution of single phase USMT protocol initiated by **R**.
- For each i, **S** verifies whether $y'_i \stackrel{?}{=} \sum_{j=1}^{n} \rho_i'^j r'_{ij}$. If the test fails, then **S** adds wire w_i to the set of faulty wires, denoted by L_{faulty}.
- **S** sets $L_{honest} = \mathcal{W} \setminus L_{faulty}$. Now, **S** computes a random pad $Z = (z_1, z_2, \ldots, z_{n(t_b+1)})$ of size $n(t_b+1)$ field elements as follows:
$$Z = EXTRAND_{n|L_{honest}|, n(t_b+1)}(r'_{ij} | w_i \in L_{honest})$$
- **S** computes $d = m \oplus Z$ and reliably sends d to **R** using the single phase **URMT_Single_Phase** protocol. **S** also broadcasts the set L_{faulty} to **R**.

Message recovery by R.

- **R** correctly receives the set L_{faulty} (by taking the majority of the sets received along the wires) and sets $L_{honest} = \mathcal{W} \setminus L_{faulty}$. **R** also correctly (probably) receive the vector d (from the correctness of **URMT_Single_Phase**).
- **R** computes the pad $Z^{\mathbf{R}} = (z_1^{\mathbf{R}}, z_2^{\mathbf{R}}, \ldots, z_{n(t_b+1)}^{\mathbf{R}})$ of size $n(t_b+1)$ field elements as follows:
$$Z^{\mathbf{R}} = EXTRAND_{n|L_{honest}|, n(t_b+1)}(r_{ij} | w_i \in L_{honest})$$
- **R** recovers the message by computing $m = Z^{\mathbf{R}} \oplus d$.

We now prove the correctness of protocol **USMT_Byzantine**.

Theorem 19 *In protocol* **USMT_Byzantine** *if* $|\mathbb{F}| \geq \frac{16n^3}{\delta}$ *then the protocol securely transmits a message containing* $n(t_b+1)$ *field elements from* **S** *to* **R** *with an error probability of at most* δ.

Proof: It is evident from the protocol construction that the theorem holds if the following are true:

1. For all $1 \leq i \leq n$, $\rho'_i = \rho_i$ with probability $\geq (1 - \frac{\delta}{4})$.
2. For all $1 \leq i \leq n$, $y'_i = y_i$ with probability $\geq (1 - \frac{\delta}{4})$.
3. If the wire w_i were indeed corrupt, then $w_i \in L_{fault}$ with probability $\geq (1 - \frac{\delta}{4})$.
4. The protocol **URMT_Single_Phase** fails to send the vector d with probability at most $\frac{\delta}{4}$.
5. The adversary learns no (additional) information about the transmitted message m in information theoretic sense.

The error probability of the protocol depends upon the error probability of the first four events. It is clear that if each of the four events are true, then the protocol's failure probability is at most δ. We now prove each of the four events are true.

Claim 8 *In* **USMT_Byzantine**, *for all* $1 \leq i \leq n$, $\rho'_i = \rho_i$ *with probability* $\geq (1 - \frac{\delta}{4})$.

Proof: From Theorem 10, we know that if $|\mathbb{F}| = \frac{2n^3}{\delta'}$, then **USMT_Single_Phase** securely sends $(t_b + 1)$ field elements (by substituting $t_o = t_f = t_p = 0$ in **USMT_Single_Phase**) with an error probability of at most δ'. In our protocol, **R** securely transmits $n = (2t_b + 1)$ ρ_i's using the single phase USMT protocol. Therefore, **R** needs to parallely execute **USMT_Single_Phase** twice in order to securely send $2t_b + 1$ ρ_i's (first execution for the first $t_b + 1$ ρ_i's and second for the remaining t_b ρ_i's). So if the error probability δ' of each of the two executions is at most $\frac{\delta}{8}$, then the total error probability of two parallel executions of the single phase USMT protocol will be at most $\frac{\delta}{4}$. If we want the error probability of **USMT_Single_Phase** to be at most $\frac{\delta}{8}$, then we require $|\mathbb{F}| \geq \frac{16n^3}{\delta}$. Since $|\mathbb{F}| \geq \frac{16n^3}{\delta}$, the claim is true. □

Claim 9 *In* **USMT_Byzantine**, *for all* $1 \leq i \leq n$, $y'_i = y_i$ *with probability* $\geq (1 - \frac{\delta}{4})$.

Proof: Similar to the proof of the above claim. □

Claim 10 *In* **USMT_Byzantine**, *if wire* w_i *is corrupted (i.e., at least one of the value* $r_{ij}, 1 \leq j \leq n$ *is changed by the adversary) and for all* i, $\rho'_i = \rho_i$ *then* $w_i \in L_{fault}$ *with probability* $\geq (1 - \frac{\delta}{4})$.

Proof. From the security of **USMT_Single_Phase** protocol, the adversary gains no information about ρ_i, y_i for all $1 \leq i \leq n$. Assume that adversary has changed the n tuple over some wire w_i and it is not marked as faulty by **S**. This implies that $y_i = \sum_{j=1}^{n} \rho_i^j r_{ij} = \sum_{j=1}^{n} \rho_i^j r'_{ij} = y'_i$. As inferred by the expression, y_i and y'_i are the y-values (evaluated at $x = \rho_i$) of the polynomials of degree n constructed using r_{ij}, $1 \leq j \leq n$ and r'_{ij}, $1 \leq j \leq n$ as coefficients. Since the two polynomials are of degree n, there are at most n points of intersection between the two. The value ρ_i is chosen uniformly by **R** from $\mathbb{F}$. Thus, with probability at most $\frac{n}{|\mathbb{F}|}$, the protocol fails to detect a faulty wire. In order that this error probability is at most $\frac{\delta}{4}$, we require field size to be at least $\frac{4n}{\delta}$. Since, $|\mathbb{F}| \geq \frac{16n^3}{\delta}$ (which in turn is $> \frac{4n}{\delta}$), the claim holds. □

Claim 11 *The* **URMT_Single_Phase** *protocol to reliably send the vector* d *fails with probability of at most* $\frac{\delta}{4}$.

Proof: As mentioned earlier, **URMT_Single_Phase** fails with probability δ, if $|\mathbb{F}| \geq \frac{n^3}{\delta}$ (see Theorem4). So in order that **URMT_Single_Phase** fails with probability of at most $\frac{\delta}{4}$, we require $|\mathbb{F}| \geq \frac{4n^3}{\delta}$. Since $|\mathbb{F}| \geq \frac{16n^3}{\delta}$, which in turn is greater that $\frac{4n^3}{\delta}$, the claim is true. □

Theorem 20 *In protocol* **USMT_Byzantine**, *the adversary learns no information about the transmitted message* m.

Proof. From the security of **USMT_Single_Phase**, (by substituting $t_o = t_f = t_p = 0$), we know that the adversary gains no information about the ρ_i's and y_i's. In the worst case, the adversary can passively listen the contents of at most t_b wires. So there will be at least $t_b + 1$ wires, which are not under the control of the adversary. Hence the adversary will have no information about the n random elements sent over these $t_b + 1$ wires. The proof follows from the correctness of EXTRAND algorithm. □

Theorem 21 *The communication complexity of* **USMT_Byzantine** *is* $O(n^2)$ *field elements.*

Proof: During **Phase I**, **R** sends n^2 random field elements to **S**. In addition, **R** also invokes four parallel executions of the single phase USMT protocol (two for sending ρ_i's and two for sending y_i's). This involves a communication complexity of $O(n^2)$ field elements. So communication complexity of **Phase I** is $O(n^2)$ field elements. During **Phase II**, **S** sends the vector d by executing **URMT_Single_Phase** protocol, which from Theorem 5 requires communicating $O(n^2)$ field elements. Thus the total communication complexity of the protocol is $O(n^2)$ field elements. □

Theorem 22 USMT_Byzantine *is a communication optimal two phase USMT protocol tolerating Byzantine adversary.*

PROOF: **USMT_Byzantine** sends $n(t_b+1)\log|\mathbb{F}| = \Theta(n^2\log|\mathbb{F}|)$ bits (for $n = 2t_b+1, t_b = \Theta(n)$) by communicating $O(n^2\log|\mathbb{F}|)$ bits. Thus the extra overhead in obtaining security is "constant". Hence it is a communication optimal protocol. Moreover it is phase optimal because from Theorem 8, by substituting $t_o = t_f = t_p = 0$, we find that any single phase USMT requires a communication overhead of $O(n^3\log(|\mathbb{F}|))$ bits to securely send $n(t_b+1)\log|\mathbb{F}| = \Theta(n^2\log|\mathbb{F}|)$ bits. □

5.5 Comparison of MultiPhase USMT with MultiPhase PSMT

1. Allowing a negligible error probability only in the reliability, *significantly* helps in the POSSIBILITY of multiphase secure message transmission protocols (see Comparison 5).

2. Allowing a negligible error probability only in the reliability, *significantly* helps in reducing the communication complexity of multiphase secure message transmission protocols (see Comparison 6).

3. It is impossible to design any PSMT protocol, which irrespective of the number of phases, achieves security with constant factor overhead; i.e., securely sending ℓ field elements by communicating $O(\ell)$ field elements tolerating only a Byzantine adversary (see Table 3, last row, by substituting $t_o = t_f = t_p = 0$). However, there exists a two phase USMT protocol which securely sends ℓ field elements by communicating $O(\ell)$ field elements, thus achieving security with constant factor overhead (Protocol **USMT_Byzantine**). Thus allowing a negligible error probability in the reliability without sacrificing the security, helps to design a two phase secure message transmission protocol, which achieves security with constant factor overhead.

6 Conclusion and Open Problems

We have studied the problem of URMT and USMT in the presence of mixed adversary. Existing URMT and USMT protocols deals only with Byzantine adversary. Moreover, the protocols are not optimal in terms of communication complexity. In this paper, we initiated the study of URMT and USMT tolerating mixed adversary. We have given the complete characterization of single phase and multiphase URMT protocols in undirected networks tolerating mixed adversary. We have proved the lower bound on the communication complexity of any single phase and multi phase URMT protocol. Moreover, we have shown that our bounds are *tight*. Similarly, we have given complete characterization of single phase and multiphase USMT protocols in undirected networks tolerating mixed adversary. We have proved the lower bound on the communication complexity of any single phase and multi phase USMT protocol. Moreover, we have shown that our bounds are *tight*. The paper shows that allowing a negligible error probability has strong effect in the *possibility*, *feasibility* and *optimality* of reliable and secure message transmission protocols.

Our protocols achieve communication optimality for sufficiently long messages. The next obvious and interesting problem is to design communication optimal protocols for messages of any length. Another interesting problem is to find the minimum number of phases required by any URMT protocol which achieves reliability with *constant factor overhead* under the presence of mixed adversary; i.e., sending ℓ field elements with a communicating overhead of $O(\ell)$ field elements.

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] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *STOC*, pages 1–10, 1988.

[3] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *STOC*, pages 11–19, 1988.

[4] Ashish Choudhary, Arpita Patra, AshwinKumar B. V, Kannan Srinathan, and C. Pandu Rangan. Constant phase bit optimal protocols for perfectly reliable and secure message transmission tolerating static and mobile mixed adversary. 2008. Manuscript.

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

[6] 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.

[7] 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.

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

[9] Paul Feldman and Silvio Micali. Optimal algorithms for Byzantine Agreement. In *STOC*, pages 148–161, 1988.

[10] Paul Feldman and Silvio Micali. An optimal probabilistic algorithm for synchronous Byzantine Agreement. In *ICALP*, pages 341–378, 1989.

[11] Matthias Fitzi, Matthew K. Franklin, Juan A. Garay, and S. Harsha Vardhan. Towards optimal and efficient perfectly secure message transmission. In *TCC*, pages 311–322, 2007.

[12] M. Franklin and R. N. Wright. Secure communication in minimal connectivity models. In *Proc of EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science (LNCS)*, pages 346–360. Springer-Verlag, 1998.

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

[14] M. Franklin and M. Yung. Secure hypergraphs: Privacy from partial broadcast. In *Proc. of 27th Ann. Symposium on Theory of Computing*, pages 36–44, 1995.

[15] J. A. Garay and K. J. Perry. A continuum of failure models for distributed computing. In *Proc. of 6th WDAG*, pages 153–165, 1992.

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

[17] R. Guerraoui and L. Rodrigues. *Introduction to Reliable Distributed Programming*. Springer Verlag, 2006.

[18] V. Hadzilacos. *Issues of Fault Tolerance in Concurrent Computations*. PhD thesis, Harvard University, Cambridge, Massachusetts, 1984.

[19] 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.

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

[21] K. Kurosawa and K. Suzuki. Almost secure (1-round, n-channel) message transmission scheme. Cryptology ePrint Archive, Report 2007/076, 2007.

[22] Leslie Lamport. The weak Byzantine generals problem. *J. ACM*, 30(3):668–676, 1983.

[23] Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

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

[25] W. Ogata, K. Kurosawa, and D. Stinson. Optimum secret sharing scheme secure against cheating. *SIAM Journal of Discrete Math*, 20(1), 2006.

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

[27] 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.

[28] Arpita Patra, Bhavani Shankar, Ashish Choudhary, K. Srinathan, and C. Pandu Rangan. Perfectly secure message transmission in directed networks tolerating threshold and non threshold adversary. In *CANS*, pages 80–101, 2007.

[29] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proc. of twenty-first annual ACM symposium on Theory of computing*, pages 73–85. ACM Press, 1989.

[30] H. Sayeed and H. Abu-Amara. Perfectly secure message transmission in asynchronous networks. In *Proc. of Seventh IEEE Symposium on Parallel and Distributed Processing*, 1995.

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

[32] B. Shanker, P. Gopal, K. Srinathan, and C. Pandu Rangan. Unconditional reliable message transmision in directed networks. In Proc. of SODA 2008.

[33] K. Srinathan. *Secure Distributed Communication.* PhD thesis, Indian Institute of Technology Madras, 2006.

[34] 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.

[35] K. Srinathan and C. Pandu Rangan. Possibility and complexity of probabilistic reliable communication in directed networks. In *Proc. of 25th PODC*, pages 265–274. ACM Press, 2006.

[36] Kannan 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.

[37] AshwinKumar B. V, Arpita Patra, Ashish Choudhary, Kannan Srinathan, and C. Pandu Rangan. On tradeoff between network connectivity, phase complexity and communication complexity of reliable communication tolerating mixed adversary. 2008. Manuscript.

[38] Y. Wang and Y. Desmedt. Secure communication in multicast channels: The answer to Franklin and Wright's question. *Journal of Cryptology*, 14(2):121–135, 2001.

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

[40] K. Zetter. Cisco security hole a whopper. http://www.wired.com/news/privacy/0,1848,68328,00.html?tw.