Unconditionally reliable and secure message transmission in undirected synchronous networks: possibility, feasibility and optimality

Arpita Patra*, Ashish Choudhury and C. Pandu Rangan

Department of Computer Science and Engineering,
Indian Institute of Technology Madras,
Chennai 600036, India
E-mail: arpitapatra10@gmail.com
E-mail: partho31@gmail.com
E-mail: prangan55@gmail.com
*Corresponding author

Kannan Srinathan

Center for Security, Theory and Algorithmic Research,
International Institute of Information Technology,
Gachibowli, Hyderabad, India
E-mail: srinathan@iiit.ac.in

Comment [t1]: Author: Please provide the full mailing address of K. Srinathan.

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 having *unbounded computing power*, who can corrupt some of the nodes in the network in *Byzantine*, *omission*, *fail-stop* and *passive* fashion respectively. We consider two types of adversary, namely *threshold* and *non-threshold*. One of the important conclusions we arrive at from our study is 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.

Reference to this paper should be made as follows: Patra, A., Choudhury, A., Pandu Rangan, C. and Srinathan, K. (xxxx) 'Unconditionally reliable and secure message transmission in undirected synchronous networks: possibility, feasibility and optimality', *Int. J. Applied Cryptography*, Vol. X, No. Y, pp.000–000.

Biographical notes: Arpita Patra is currently a Postdoctoral Researcher in the Department of Computer Science, University of Aarhus, Denmark. She is currently working on secure distributed communication and computation. The work was done when she was a PhD student at the Department of Computer Science and Engineering, IIT Madras, under the supervision of Professor C. Pandu Rangan.

Ashish Choudhury is a Visiting Scientist in the Applied Statistics Unit, Indian Statistical Institute (ISI) Kolkata. He is currently working on secure distributed communication and computation. The work was done when he was a PhD student at the Department of Computer Science and Engineering, IIT Madras, under the supervision of Professor C. Pandu Rangan.

C. Pandu Rangan is currently a Professor at IIT Madras. He is currently working in graph theory, game theory and all aspects of cryptography.

Kannan Srinathan is currently an Assistant Professor at IIIT Hyderabad. He did his PhD at IIT Madras. He is interested in cryptography and all aspects of theoretical computer science.

1 Introduction[1]

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 cryptography), firewalls, virus scanners, etc. However, routers too are vulnerable (Zetter, 2005). Two problems have been identified if a router node is hacked. The hacker can shut down the node or forward incorrect information to the adjacent nodes in the network (Dolev et al., 1993; Hadzilacos, 1984). Hence, there is a need for considering an adversary who can disrupt the network in variety of ways. The problem of *reliable message transmission* (RMT) and *secure message transmission* (SMT) perfectly captures the scenario when a specific node in the network intends to send a message to another *non-adjacent* node with the help of other nodes and edges in the network, some of which may be hacked (corrupted) by an adversary.

Let a sender **S** and a receiver **R** are part of an unreliable connected network, where **S** is connected to **R** through intermediate nodes. To study the cumulative or combined effect of the faults in the network, we assume the existence of an abstract entity called *centralised adversary*. For example, assume that some hackers have taken complete control of say up to t_b nodes in the network and could manipulate the information and computations of these nodes at their will in an arbitrary fashion. In order to study the cumulative effect of the actions of these hackers, we may further assume that the hackers are colluding in an arbitrary fashion and combine all the information available under their control to cause maximum damage. Thus, we arrive at the abstraction called centralised adversary. The centralised adversary can disrupt the communication and computation of some of the intermediate nodes in variety of ways. Moreover, we assume that the adversary has *unbounded computing power*.

In the problem of *RMT*, the sender **S** has a message m, which he wants to *reliably* send to **R**. The goal is to design a protocol, such that after interacting with **S** as per the protocol, **R** should correctly output m. Moreover, this should happen, even if some of the intermediate nodes are under the control of the centralised adversary. The problem of *SMT* has an additional constraint that the adversary should get no information about m what so ever, in information theoretic sense. Security against such a powerful adversary is called information theoretic security or *non-cryptographic security* or *Shannon security*. Notice that if **S** and **R** are connected by a direct edge, then RMT and PSMT is straight forward: **S** simply sends the message to **R**. Thus, the goal of RMT (SMT) protocol is to simulate a direct, virtual, reliable (secure) link between **S** and **R**, who are connected through intermediate nodes, even in the presence of a computationally unbounded centralised adversary. RMT and SMT are well-motivated problems, for it being one of the fundamental primitives used by all fault-tolerant distributed algorithms like Byzantine agreement (Lamport et al., 1982; Lamport, 1983; Feldman and Micali, 1988, 1989), multiparty computation (MPC) (Yao, 1982; Goldreich et al., 1987; Chaum et al., 1988; Ben-Or et al., 1988; Rabin and Ben-Or, 1989; Cramer et al., 1999), etc. All these popular fault-tolerant distributed algorithms assume that the underlying network is a complete graph. When the graph is not complete, we can simulate the effect of the missing links using RMT/SMT protocols. There is another motivation to study SMT 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 ineffective. Hence, it is worthwhile to look for information theoretically SMT schemes.

There are various settings in which RMT and SMT problem has been studied extensively in the past. For example, the underlying network model may be undirected graph (Dolev et al., 1993; Patra et al., 2006; Agarwal et al., 2006; Kurosawa and Suzuki, 2008), directed graph (Patra et al., 2007; Desmedt and Wang, 2003) or hypergraph (Franklin and Yung, 1995; Desmedt and Wang, 2003; Renault and Tomala, 2008). The communication in the network could be synchronous (Dolev et al., 1993; Sayeed and Abu-Amara, 1996) or asynchronous (Sayeed and Abu-Amara, 1995). The faults could be passive, fail-stop, Byzantine or sometimes mixed/hybrid faults (Garay and Perry, 1992). The number of faulty nodes may be bounded by a fixed constant (threshold adversary) (Dolev et al., 1993; Sayeed and Abu-Amara, 1996) or the potential sets of faulty nodes may be described by a collection of subsets of nodes (non-threshold adversary) (Kumar et al., 2002), while the adversary may be mobile (Ostrovsky and Yung, 1991) or adaptive (Dolev et al., 1993; Sayeed and Abu-Amara, 1996). The protocols can be perfect, having no error (Dolev et al., 1993; Kurosawa and Suzuki, 2008) or may be unconditional, having negligible error probability (Franklin and Yung, 1995; Desmedt and Wang, 2003; Renault and Tomala, 2008; Patra et al., 2008; Srinathan et al., 2009). In general, we may use the following parameters to categorise the different settings in which RMT and SMT problem can be studied:

1 underlying network

2 type of communication

3 adversary capacity

4 type of faults

5 type of security.

The taxonomy of settings in which RMT and SMT can be studied is listed in Table 1. For example, one may ask: what is the necessary and sufficient condition for *perfectly* SMT over an *undirected graph* thwarting a *threshold adaptive* adversary? In this way, hundreds of different models/settings can be formulated and many of them are used in practice.

Irrespective of the settings in which RMT and SMT are studied, the following issues are common:

1 *Possibility:* What is the necessary and sufficient condition for the existence of a protocol in a given network?

2 *Feasibility:* Once the existence of a protocol is ensured then does there exist a polynomial time efficient protocol on the given network?

3 *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?

Table 1 The taxonomy of the settings in which RMT/SMT can be studied

Underlying network	*Type of communication*	*Adversary capacity*
Undirected graph	Synchronous	Threshold adaptive
Directed graph	Asynchronous	Threshold mobile
Undirected hypergraph		Non-threshold adaptive
Directed hypergraph		Non-threshold mobile
Types of faults	*Type of security*	
Byzantine	Perfect	
Fail-stop	Unconditional	
Passive		
Mixed		

In this paper, we study the above issues in the context of *unconditional* RMT and SMT in undirected synchronous network. We call *unconditional* RMT and SMT as URMT and USMT respectively. Moreover, we consider two different types of adversary, namely *threshold adaptive mixed* adversary and *non-threshold adaptive mixed* adversary. We now define URMT and USMT. More formal and rigorous definition will appear in Section 2.

1 An RMT protocol is called δ-reliable, for any $0 < \delta < 1/2$, if at the end of the protocol, **R** correctly outputs **S**'s message, except with probability δ. Moreover, this should hold, irrespective of the behaviour of the adversary.

2 An SMT protocol is called ϵ-secure, for any $0 < \epsilon < 1/2$, if at the end of the protocol, the adversary does not get any information about **S**'s message, except with probability ϵ.

3 A message transmission protocol is called (ϵ, δ)-secure, if it is ϵ-secure and δ-reliable.

4 An RMT protocol is called *perfectly reliable* also called as PRMT, if it is 0-reliable.

5 An RMT protocol is called *unconditionally reliable* also called as URMT, if it is δ-reliable. Any URMT protocol is also called as *statistically* RMT protocol, where we want δ to be negligible small.

6 A message transmission protocol is called *perfectly secure*, also called as PSMT, if it is (0, 0)-secure.

7 A message transmission protocol is called *unconditionally secure*, also called as USMT, if it is $(0, \delta)$-secure. Any USMT protocol is also called as *statistically* SMT protocol, where we want δ to be negligible small.

1.1 Motivation of our work

The PRMT and PSMT problem has been studied extensively over the past three decades in both directed and undirected network model, tolerating threshold and non-threshold adversary (see Dolev et al., 1993; Sayeed and Abu-Amara, 1996; Desmedt and Wang, 2003; Srinathan et al., 2004; Narayanan et al., 2006; Kumar et al., 2002; Agarwal et al., 2006; Patra et al., 2006; Srinathan et al., 2007b; Fitzi et al., 2007; Ashwinkumar et al., 2008; Kurosawa and Suzuki, 2008; Patra et al., 2009). The issue of possibility, feasibility and optimality has been completely resolved for PRMT and PSMT in undirected network model, tolerating threshold adversary. Moreover, the issue of possibility has been completely resolved for PRMT and PSMT tolerating non-threshold adversary. However, not too much is known about URMT and USMT.

It is a well-known fact that in several problem domains *randomisation* helps to a great extent in arriving at more efficient and simpler solutions than their deterministic counterpart. The problem domains range from famous number theoretic randomised primality testing algorithms to various distributed computation tasks like verifiable secret sharing (VSS) (Rabin and Ben-Or, 1989; Cramer et al., 1999), MPC (Cramer et al., 1999; Beerliová-Trubíniová and Hirt, 2006; Damgård and Nielsen, 2007) to name a few. In this work, we focus on the effect of randomisation on PRMT and PSMT problems.

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 in undirected synchronous networks. Furthermore, we consider two different types of adversaries, namely *threshold* adaptive mixed adversary and *non-threshold* adaptive mixed adversary.

Our results show that allowance of a small probability of error in the transmission (only in the reliability) significantly improves the existing complexity measures of PRMT and PSMT, namely connectivity requirement, communication complexity and the number of interactions between **S** and **R** during the protocol.

Remark 1 (a note on adversary model): Since, in this paper, we deal with both threshold and non-threshold adversary, for easy understanding, we divide the paper into two parts. The first part deals with threshold adversary while the second part deals with non-threshold adversary.

Remark 2 (a note on the terminology URMT and USMT): In Srinathan et al. (2007a), the authors have used the terms PPRMT and PPSMT for URMT and USMT respectively. The reason for the change of terminology in this paper is as follows: in the literature of secure MPC, protocols with negligible error probability are usually referred as unconditional MPC (Beerliová-Trubíniová and Hirt, 2006, 2008; Damgård and Nielsen, 2007). Since URMT and USMT protocols will be used as a black box in unconditional MPC to simulate a virtual complete network, we prefer to change the terminology from PPRMT and PPSMT to URMT and USMT respectively.

2 Network model and definitions

We now specify the network model and definitions that are used in this paper in the context of threshold adversary. The underlying network is a connected synchronous network represented by an undirected graph where **S** and **R** are two *non-adjacent* nodes of the graph. All the edges in the network are reliable and secure but the nodes can be corrupted.

We assume the presence of a threshold adversary $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$, having *unbounded computing power*, who can corrupt any disjoint set of t_b, t_o, t_f and t_p nodes in the graph (excluding **S** and **R**) in Byzantine, omission, fail-stop and passive fashion respectively. We now formally define these four types of corruptions.

Definition 1 – fail-stop 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, P 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 execution.

Definition 2 – omission corruption: We say that a node P is omission 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 follow the instructions of the protocol honestly. The adversary can eavesdrop the internal data of P but cannot make P to deviate from the proper execution of the protocol. A blocked node P 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 the adversary has full access to the information and internal state of P. But P honestly follows the protocol execution.

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 fail-stop corruption. But it weaker than Byzantine corruption. 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 centralised 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 (Cramer et al., 1999). Thus, he is allowed to *dynamically* corrupt nodes during the protocol execution depending on the data seen so far from the corrupted nodes. So before the protocol execution, it is not known in advance which nodes are going to be influenced by adversary and in what way the nodes will be corrupted by the adversary. Also, once a node is under the control of the adversary in some fashion, then it will remain corrupted in the same fashion throughout the protocol.

Following the approach of Dolev et al. (1993), we abstract away the network and concentrate on solving URMT and USMT problem for a single pair of processors, the sender **S** and the *receiver* **R**, connected by n parallel and synchronous bi-directional channels w_1, w_2, ..., w_n, also known as *wires*. The reason for such an abstraction is as follows: suppose some intermediate node between **S** and **R** is under the control of the adversary. Then all the paths between **S** and **R** which passes through that node are also compromised. Hence, all the paths between **S** and **R** passing through that node can be modelled by a single wire between **S** and **R**. In the worst case, the adversary can compromise an entire wire in certain fashion by controlling a single node on the wire.

Hence, $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ having unbounded computing power can corrupt up to t_b, t_o, t_f and t_p wires in Byzantine, omission, fail-stop and passive fashion respectively. Moreover, we assume that the wires that are under the control of the adversary in Byzantine, omission, fail-stop and passive fashion are mutually disjoint. Any protocol in the network operates as a sequence of *phases*, where a phase is a communication from **S** to **R** or vice-versa.

Throughout this paper, we use m to denote the message that **S** wishes to send to **R**. The message is assumed to be a sequence of ℓ elements from the finite field $\mathbb{F}$ with $\ell \geq 1$. Without loss of generality, we assume that m is selected uniformly and randomly from $\mathbb{F}$. 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 the size of $\mathbb{F}$ as a function of δ so that we could bound the error probability by 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. In any message transmission protocol, **S** selects a message m uniformly and randomly from $\mathbb{F}$ at the beginning. At the end of the protocol, **R** outputs m'. We now give the following definitions:

Definition 5 – 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, 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 recover x by taking majority among the received values.

Definition 6 – PRMT (Dolev et al., 1993): In perfectly reliable message transmission (PRMT) over a sufficiently connected network $N = (V, E)$, tolerating mixed adversary $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$, $S \in V$ intends to transmit a message m which is a sequence of ℓ ($\ell \geq 1$) field elements from a finite field $\mathbb{F}$ to $R \in V$ using some protocol, such that after interacting in phases as per the protocol, the following condition must hold:

- *Perfect reliability*: **R** should correctly output $m' = m$ with probability 1.

Definition 7 – PSMT (Dolev et al., 1993): The problem of perfectly secure message transmission (PSMT) over a sufficiently connected network $\mathcal{N}$ requires perfect reliability of PRMT and the following additional condition:

- *Perfect secrecy*: The message should be hidden from the adversary in information theoretic sense. More formally, let adv(m, r) denote the view of the adversary during the protocol, when the message sent by **S** is m and r is the random coin flips of the adversary. Then, we require that for every two messages m_1, m_2 and every r,

$$\sum_c \left| \Pr\left[adv(m_1, r) = c \right] - \Pr\left[adv(m_2, r) = c \right] \right| = 0.$$

  The probabilities are taken over the coin flips of the honest parties, and the sum is over all possible values of the adversary's view.

Definition 8 – URMT (Franklin and Wright, 2000): The problem of URMT is same as PRMT, except that it should satisfy a weaker notion of perfect reliability, called unconditional reliability or statistical reliability, which is as follows:

- *Unconditional reliability*: **R** should correctly output $m' = m$ with probability at least $1 - \delta$, where $0 < \delta < 1/2$. The probability is over the choice of m, the coin flips of **S** and **R** and the adversary.

Definition 9 – USMT (Franklin and Wright, 2000): USMT requires unconditional reliability property of URMT and perfect secrecy property of PSMT.

Notice that 'unconditional reliability' says that **R** can output a wrong message with small probability δ. We now define a strictly stronger notion of 'unconditional reliability' which we call as 'strong unconditional reliability'. A URMT protocol that achieves 'strong unconditional reliability' always outputs the correct message; otherwise, it fails with output NULL, but it never outputs an incorrect message. Precisely, in an URMT protocol that achieves 'strong unconditional reliability', **R** can detect whether he has correctly received the message sent by **S** or not.

Definition 10 – strong unconditional reliability: **R** should either correctly receive **S**'s message or otherwise output NULL, where the probability of receiving correct message is at least $1 - \delta$, where $0 < \delta < 1/2$.

Definition 11 – strong URMT: Strong URMT satisfies strong unconditional reliability property instead of unconditional reliability.

Definition 12 – strong USMT: Strong USMT requires perfect secrecy of PSMT and should satisfy strong unconditional reliability.

Our single phase URMT and USMT protocols presented in this paper are strong URMT and strong USMT protocols.

Definition 13 – 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 ($\ell \geq 1$) field elements by communicating $O(b)$ field elements, over an n-(**S**, **R**)-connected network. If the lower bound on the communication complexity of any r phase URMT (USMT) protocol to send m over such a network is $\Omega(b)$ field elements, then Π is said to be a communication optimal URMT (USMT) protocol to reliably (securely) send m.

Definition 14 – Reed-Solomon (RS) codes (MacWilliams and Sloane, 1978): For message block $M = (m_1\ m_2\ \ldots\ m_k)$ over $\mathbb{F}$, we define RS polynomial as $P_M(x) = m_1 + m_2x + m_3x^2 + \ldots + m_kx^{k-1}$. Let $\alpha_1, \alpha_2, \ldots, \alpha_L$, $L > k$, denote a sequence of L distinct and fixed elements from $\mathbb{F}$. Then vector $C = (c_1\ c_2\ \ldots\ c_L)$ where $c_i = P_M(\alpha_i)$, $1 \leq i \leq L$ is called the (RS) codeword of size L for the message block M.

The error correcting and detecting capability of RS codes is given by the following theorem.

Theorem 1 (MacWilliams and Sloane, 1978; Desmedt and Wang, 2003): Let C denote the RS codeword for a message block of size k, where $|C| = L$. Let a receiver receive C' where C' differ from C in at most tb locations. Then, RS decoding can correct up to c Byzantine errors in C' and simultaneously detect additional d Byzantine errors in C' iff $L - k \geq 2c + d$, where $c + d \leq t_b$.

2.1 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 by considering four possible different types of corruption, namely Byzantine, omission, fail-stop and passive. 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'. The last point will be made clear when we will present our results in the subsequent sections.

3 Existing results and our contribution

We now present the existing results for PRMT, PSMT, URMT and USMT in undirected networks, tolerating threshold adaptive adversary.

3.1 Existing literature in threshold adversarial model

RMT and SMT problem was first formulated by Dolev et al. (1993). Specifically, Dolev et al. (1993) presented the first ever characterisation (POSSIBILITY) for PRMT and PSMT on an undirected synchronous network tolerating threshold adaptive Byzantine adversary, $\mathcal{A}_{t_b}$. Dolev et al. (1993) abstracted the network in terms of channels and concentrated on solving PRMT and PSMT problem for a single pair of processors, the *sender* **S** and the *receiver* **R**, connected by n parallel and synchronous bi-directional channels $w_1, w_2, ..., w_n$, also known as *wires*.[2] The existing results for PRMT and PSMT in undirected synchronous networks tolerating threshold adaptive Byzantine $\left(\mathcal{A}_{t_b}\right)$ and mixed $\left(\mathcal{A}_{(t_b,t_o,t_f,t_p)}\right)$ adversary are summarised in Table 2 and Table 3.

Table 2 Connectivity requirement and lower bounds for PRMT and PSMT in undirected networks

Model	*Connectivity requirement between* **S** *and* **R** *(n)*	*Lower bound on communication complexity*
PRMT (Byzantine adversary)	$n \geq 2t_b + 1, \forall r \geq 1$ (Dolev et al., 1993)	$\Omega\left(\frac{n\ell}{n-2t_b}\right)$ for $r = 1, 2$ (Srinathan et al., 2004)
		$\Omega\left(\frac{n\ell}{n-t_b}\right)$ for $r \geq 3$ (Srinathan et al., 2007b)
PSMT (Byzantine adversary)	$n \geq 3t_b + 1$ for $r = 1$ (Dolev et al., 1993)	$\Omega\left(\frac{n\ell}{n-3t_b}\right)$ for $r = 1$ (Fitzi et al., 2007; Srinathan et al., 2007b)
	$n \geq 2t_b + 1$ for $r \geq 2$ (Dolev et al., 1993)	$\Omega\left(\frac{n\ell}{n-2t_b}\right)$ for $r \geq 2$ (Srinathan et al., 2007b)
PRMT (mixed adversary)	$n \geq 2t_b + t_o + t_f + 1, \forall r \geq 1$ (Srinathan, 2006)	$\Omega\left(\frac{n\ell}{n-(2t_b+t_o+t_f)}\right)$ for $r = 1, 2$ (Srinathan, 2006)
		$\Omega\left(\frac{(n-t_f-t_o)\ell}{n-(t_b+t_o+t_f)}\right)$ for $r \geq 3$ (Srinathan, 2006)
PSMT (mixed adversary)	$n \geq 3t_b + 2t_o + t_f + t_p + 1$ for $r = 1$ (Srinathan, 2006)	$\Omega\left(\frac{n\ell}{n-(3t_b+2t_o+t_f+t_p)}\right)$ for $r = 1$ (Srinathan, 2006)
	$n \geq 2t_b + t_o + t_f + t_p + 1$ for $r \geq 2$ (Choudhury et al., 2008)	$\Omega\left(\frac{n\ell}{n-(2t_b+t_o+t_f+t_p)}\right)$ for $r \geq 2$ (Srinathan, 2006)

Note: r denotes number of phases and ℓ denotes the message size in terms of field elements.

Table 3 Protocols with optimum communication complexity

Model	*Communication complexity in terms of field elements*	*Number of phases*	*Remarks*
PRMT (Byzantine adversary)	$O\left(\frac{n\ell}{n-2t_b}\right)$	≤ 2	• $\ell \geq n$; polynomial computation and communication complexity (Srinathan et al., 2004).
	$O\left(\frac{n\ell}{n-t_b}\right)$	3	• $\ell \geq n^2$; polynomial computation and communication complexity (Patra et al., 2006).
PSMT (Byzantine adversary)	$O\left(\frac{n\ell}{n-3t_b}\right)$	1	• $\ell \geq n$; polynomial computation and communication complexity (Fitzi et al., 2007).
	$O\left(\frac{n\ell}{n-2t_b}\right)$	2	• ℓ is exponential; exponential computation and communication complexity (Agarwal et al., 2006).
	$O\left(\frac{n\ell}{n-2t_b}\right)$	3	• $\ell \geq n^2$; polynomial computation and communication complexity (Patra et al., 2006).
	$O\left(\frac{n\ell}{n-2t_b}\right)$	2	• $\ell \geq n^2$; polynomial computation and communication complexity (Kurosawa and Suzuki, 2008).
PRMT (mixed adversary)	$O\left(\frac{n\ell}{n-(2t_b+t_o+t_f)}\right)$	1	• $\ell \geq n$; polynomial computation and communication complexity (Srinathan, 2006).
	$O\left(\frac{(n-t_f-t_o)\ell}{n-(2t_b+t_o+t_f)}\right)$	$O\left(\log\left(\frac{t_f+t_o}{n-(t_f+t_o)}\right)\right)$	• $\ell \geq n^2$; polynomial computation and communication complexity (Ashwinkumar et al., 2008).
PSMT (mixed adversary)	$O\left(\frac{n\ell}{n-(2t_b+t_o+t_f+t_p)}\right)$	4	• $\ell \geq n$; polynomial computation and communication complexity (Choudhury et al., 2008).

Note: ℓ is the message size in terms of field elements and n is the corresponding connectivity requirement from Table 2.

The problem of URMT and USMT in undirected synchronous networks in the presence of threshold adaptive Byzantine adversary $\mathcal{A}_{t_b}$ was first defined and solved by Franklin and Wright (1998).[3] As one of the key results, they have proved that over undirected graphs, URMT (USMT) tolerating $\mathcal{A}_{t_b}$ is possible if and only if PRMT (PSMT) tolerating $\mathcal{A}_{t_b}$ is possible! Subsequent works on URMT and USMT include Franklin and Wright (2000), Wang and Desmedt (2001), and Desmedt and Wang (2003). 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 Kurosawa and Suzuki (2007) have addressed the issue of optimality of single phase USMT in undirected networks tolerating threshold Byzantine adversary. Most recently, Srinathan and Pandu Rangan (2006) and Shankar et al. (2008) have given the characterisation for the possibility of URMT in arbitrary directed graphs tolerating non-threshold and threshold Byzantine adversary respectively. In Srinathan et al. (2009) have given the characterisation for the possibility of USMT in arbitrary directed graphs tolerating non-threshold adversary. However, to the best of our knowledge, no research work has ever simultaneously addressed the issue of possibility, feasibility and optimality of URMT and USMT protocols in any network model tolerating threshold mixed adversary.

3.2 Our contribution in threshold adversarial model

As mentioned earlier, any reliable/secure protocol is analysed by 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 network tolerating threshold Byzantine adversary (Patra et al., 2006; Srinathan et al., 2007b; Agarwal et al., 2006; Kurosawa and Suzuki, 2008). 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 present characterisation, lower bound on communication complexity and protocols that matches the lower bound for URMT and USMT. 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 $\Omega\left(\frac{n\ell}{n-(t_b+t_o+t_f)}\right)$ field elements to reliably transmit (with high probability) ℓ field elements.

  We also design single phase *polynomial* time *communication optimal* URMT protocol whose communication complexity satisfies 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., $t_o = t_f = t_p = 0$).

- Any multiphase URMT protocol, from **S** to **R** over $n \geq 2t_b + t_o + t_f + 1$ wires communicates (ℓ) field elements to reliably transmit (with high probability) ℓ field elements.

An $O\left(\log \frac{t_f + t_o}{n - t_f - t_o}\right)$ phase PRMT protocol which sends ℓ field elements by communicating $O(\ell)$ field elements is presented in Ashwinkumar et al. (2008). The protocol of Ashwinkumar et al. (2008) is also a valid multiphase URMT protocol (since any PRMT protocol is by default a URMT protocol with $\delta = 0$) satisfying the communication complexity lower bound for multiphase URMT. The 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 $\Omega\left(\frac{n\ell}{n - (2t_b + 2t_o + t_{f+t_p})}\right)$ field elements in order to securely send an ℓ-field element message with very high probability.

  We also design *polynomial* time *communication optimal* single phase USMT protocol whose communication complexity satisfies the above lower bound for single phase USMT. This shows that our lower bound is tight.
- 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* four-phase USMT protocol whose communication complexity satisfies the above lower bound for multiphase USMT. This shows that our lower bound is tight.

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. We now tabulate the results on URMT and USMT in Table 4 and Table 5.

Remark 3: 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 summarised in Table 4 and Table 5, 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. In general the field size will have the following form $|\mathbb{F}| = \frac{n^c}{\delta}$ where c is some small constant. Now, we may set δ to be $2^{-\Omega(\kappa)}$ (we may call κ as security parameter). This gives $|\mathbb{F}| = \frac{n^c}{\delta} = n^c 2^{\Omega(\kappa)}$ which implies a single field element from $\mathbb{F}$ can be represented by $O(\log(n) + \kappa)$ bits. For PRMT and PSMT the only restriction on the size of the underlying field is that $|\mathbb{F}| \geq n$. So any field element can be represented by $O(\log(n))$ bits. So, the communication complexity figures presented in terms of field elements in Table 2 and Table 3 can be represented in terms of bits by multiplying $O(\log(n))$. Similarly, the communication complexity figures presented in terms of field elements in Table 4 and Table 5 can be represented in terms of bits by multiplying $O(\log(n) + \kappa)$.

Now, comparing Table 2 with Table 4 and Table 3 with Table 5, we find that allowing a negligible error probability has tremendous effect on reliable and SMT in terms of POSSIBILITY, FEASIBILITY and OPTIMALITY. Many practical scenarios can be shown where no optimal PRMT or PSMT protocol exist but optimal URMT and USMT protocol does exist, thus, showing the power of allowing negligible error probability in the reliability of the protocols (without sacrificing perfect secrecy).

Table 4 Connectivity requirement and lower bound on communication complexity for URMT and USMT

Model	*Connectivity (n)*	*Lower bounds*
URMT (Byzantine adversary)	$n \geq 2t_b + 1$, $\forall r \geq 1$ (Franklin and Wright, 1998)	$\Omega\left(\frac{n\ell}{n-t_b}\right)$ for $r = 1*$
USMT (Byzantine adversary)	$n \geq 2t_b + 1$, $\forall r \geq 1$ (Franklin and Wright, 1998)	$\Omega\left(\frac{n\ell}{n-2t_b}\right)$ for $r = 1*$
		$\Omega\left(\frac{n\ell}{n-t_b}\right)$ 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 $r = 1*$	$\Omega\left(\frac{n\ell}{n-(2t_b+2t_o+t_f+t_p)}\right)$ for $r = 1*$
	$n \geq t_b + \max(t_b, t_p) + t_o + t_f + 1$ for $r \geq 2*$	$\Omega\left(\frac{n\ell}{n-(t_b+t_o+t_f+t_p)}\right)$ for $r \geq 2*$

Notes: 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.

Table 5 Protocols with optimum communication complexity

Model	*Communication complexity*	*Number of phases*	*Remarks*
URMT (Byzantine adversary)	$O\left(\frac{n\ell}{n-t_b}\right)$	1	$\ell \geq n^2*$
USMT (Byzantine adversary)	$O\left(\frac{n\ell}{n-2t_b}\right)$	1	$\ell \geq n*$
	$O(\ell)$	2	$\ell \geq n^2*$
URMT (mixed adversary)	$O\left(\frac{n\ell}{n-(t_b+t_o+t_f)}\right)$	1	$\ell \geq n(t_b + 1)*$
	$O(\ell)$	$O\left(\log\left(\frac{t_f+t_o}{n-(t_f+t_o)}\right)\right)$	$\ell \geq n^2$ (Ashwinkumar et al., 2008)
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 $t_b > t_p*$

Notes: ℓ is the message size in terms of field elements. n denotes respective connectivity requirement specified in Table 4. All the $*$ marked results are presented in this paper.

3.3 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 (Franklin and Wright, 1998; Desmedt and Wang, 2003) use the idea of *information theoretic* authentication schemes and check vectors along with error correcting codes. The check vectors are introduced in Rabin and Ben-Or (1989) for information checking (IC) protocols, which are used to generate IC signatures. The IC signatures can be used as a semi digital signature (Cramer et al., 1999; Rabin and Ben-Or, 1989). 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.

4 URMT in undirected network tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$

In this section, we characterise 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 *asymptotically tight* by designing a

communication optimal single phase URMT protocol 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, the section ends with the comparison of our results on URMT with the existing results for PRMT.

4.1 Characterisation for single phase URMT

The existing characterisation for URMT tolerating threshold adaptive Byzantine adversary $\mathcal{A}_{t_b}$ in undirected network is as follows.

Theorem 2 (Franklin and Wright, 1998): Any $r \geq 1$ phase URMT between **S** and **R** against an adaptive Byzantine adversary $\mathcal{A}_{t_b}$ is possible iff the network is $(2t_b + 1)$-(**S**, **R**)-connected.

The characterisation for URMT tolerating mixed adversary is as follows.

Theorem 3: 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.[5]

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 fail-stop fashion respectively. It follows that $\mathcal{N}'$ is not $(2t_b + 1)$-(**S**, **R**)-connected. Evidently, if Π is a URMT protocol on N, then Π' is a URMT protocol on $\mathcal{N}'$, where Π' is the protocol Π restricted to the nodes in $\mathcal{N}'$. However, from Theorem 2, Π' is non-existent. Thus, Π is impossible too. □

Significance of Theorem 3: Theorem 3 *strictly generalises* Theorem 2 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 2, on this network, any URMT protocol can tolerate at most one Byzantine fault. However, according to Theorem 3, it is possible to tolerate *one additional* faulty node, which can be either omission or fail-stop faulty. Thus, our characterisation shows availability of *more fault tolerance* in comparison to the existing results. This is one of the motivations for studying URMT and USMT in the context of mixed adversary.

Comparison 1 (possibility of PRMT vs. possibility of URMT): From Table 2 (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 3, the same number of wires are 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 RMT.

Though allowing a negligible error does not affect the connectivity requirement of the network for RMT 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.

4.2 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 4: Any single phase URMT protocol, from **S** to **R** over $n \geq 2t_b + t_o + t_f + 1$ wires, communicates $\Omega\left(\frac{n\ell}{n-(t_b+t_o+t_f)}\right)$ 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.[4] 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)

b c_2 is sent and the adversary again corrupts it to $\alpha \circ \perp$.

Thus, **R** can not distinguish between the receipt of c_1 and c_2 with probability greater than $\frac{1}{2}$, which violates the property of URMT (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, $\Omega\left(\frac{n\ell}{(n-(t_b+t_o+t_f))}\log|\mathbb{F}|\right)$ bits need to be sent. Since $\log |\mathbb{F}|$ bits represents one field element from $\mathbb{F}$, communicating ℓ elements from $\mathbb{F}$ requires communicating $\Omega\left(\frac{n\ell}{(n-(t_b+t_o+t_f))}\right)$ field elements. □

Remark 4: 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 4, any URMT protocol to send ℓ field elements from $\mathbb{F}$ need to communicate $\Omega\left(\frac{n\ell}{(n-(t_b+t_o+t_f))}\log|\mathbb{F}|\right)$ bits. Thus, the communication complexity of any single phase URMT protocol is a function of δ as well (since $|\mathbb{F}|$ is a function of δ), though it is not explicitly mentioned in the expression derived in Theorem 4. 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 $\Omega\left(\frac{n\ell}{(n-(2t_b+t_o+t_f))}\right)$ field elements (see Table 2, third row), the same for URMT is $\Omega\left(\frac{n\ell}{(n-(t_b+t_o+t_f))}\right)$ field elements (Theorem 4). Recall that as pointed out in Comparison 1, the connectivity requirement for both PRMT and PSMT is $n \geq 2t_b + t_o + t_f + 1$. Assuming $n = 2t_b + t_o + t_f + 1$, the lower bound for single phase PRMT and URMT become $\Omega(n\ell)$ and $\Omega\left(\frac{n\ell}{t_b}\right)$ field elements respectively. Now if $t_b = \Theta(n)$ then the lower bound for single phase URMT becomes $\Omega(\ell)$ field elements. This implies that for $t_b = \Theta(n)$, communication of ℓ field elements requires transmission of $\Omega(n\ell)$ field elements for PRMT and $\Omega(\ell)$ field elements for URMT. Now, notice that PRMT and URMT tolerating an adaptive Byzantine adversary $\mathcal{A}_{t_b}\left(t_o = t_f = t_p = 0\right)$ requires $n \geq 2t_b + 1$. If $n = 2t_b + 1$, then $t_b = \Theta(n)$ holds. Hence, the conclusion is that in the presence of $\mathcal{A}_{t_b}$ the lower bounds on the communication complexity of any single phase PRMT and URMT are $\Omega(n\ell)$ and $\Omega(\ell)$ field elements respectively.

In the next section, we design a single phase communication optimal URMT protocol. The same protocol when executed in the presence of $\mathcal{A}_{t_b}$ communicates $O(\ell)$ field elements for sending ℓ field elements and thus achieves reliability with constant factor overhead.

4.3 Single phase communication optimal URMT protocol 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 $W = \{w_1, w_2, ..., w_n\}$, 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, fail-stop 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 *asymptotically tight*. *URMT_Single_Phase* has a special feature that it achieves reliability with *constant factor* overhead, when executed only in the presence of Byzantine adversary $\mathcal{A}_{t_b}$ (i.e., $t_o = t_f = t_p = 0$). Let δ be a bound on the probability that the protocol may fail 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:

$$m = \left[m_1 \ \ldots \ m_n \ m_{n+1} \ \ldots \ m_{2n} \ \ldots \ m_{t_b n+1} \ m_{t_b n+2} \ \ldots \ m_{t_b n+n} \right].$$

Remark 5: Our single phase protocol *URMT_Single_Phase* is a strong URMT protocol (see Definition 11).

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 jth row, $1 \leq j \leq t_b + 1$ contains the

elements $m_{(j-1)n+1}$, $m_{(j-1)n+2}$, ..., $m_{(j-1)n+n}$. For each column i of A, $1 \le i \le n$ we do the following: we construct the unique t_b degree polynomial $q_i(x)$ passing through the points $(1, m_i)$, $(2, m_{n+i}), \ldots, (t_b+1, m_{t_b n+i})$ where $m_i, m_{n+i}, \ldots, m_{t_b n+i}$ belong to the ith column A. Then $q_i(x)$ is evaluated at $t_b + t_o + t_f$ values of x namely, $x = t_b + 2$, $t_b + 3$, ..., 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

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

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. The elements of C are obtained from A using the above described technique which will be referred subsequently by *extrapolation technique*. Notice that the n values along each column of D lies on a t_b degree polynomial. So for $1 \le i \le n$, each column of D can be viewed as an n length RS codeword for a message block of size $t_b + 1$, consisting of the coefficients of $q_i(x)$. 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 tb 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, ith column, $1 \le i \le n$, 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 (which is just the first $t_b + 1$ rows of D). □

Lemma 3: Modification of t_b elements along any column of D is detectable.

Proof: Recall that in D, the ith column denotes an $L = n = 2t_b + t_o + t_f + 1$ length RS codeword for a block of size $k = t_b + 1$. So by substituting these values, along with $c = 0$ in Theorem 1, the maximum number of errors d that can be detected is $t_b + t_o + t_f$. In other words, the values along ith column lie on a unique t_b degree polynomial $q_i(x)$. Now suppose t_b values along ith 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 ith column (in which at most t_b values has been changed) will not reconstruct any tb degree polynomial. This clearly indicates that t_b values are changed along the column. Hence, the lemma holds. □

We are now ready to describe our single phase URMT protocol called *URMT_Single_Phase*, which is given in Table 6.

Lemma 4: In *URMT_Single_Phase*, if any $w_j \in \mathcal{W} \backslash (\mathcal{F} \cup \mathcal{B})$ is contradicted by at least $(t_b - |\mathcal{B}|) + 1$ wires from the set $\mathcal{W} \backslash (\mathcal{F} \cup \mathcal{B})$, 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 $\mathcal{W} \backslash (\mathcal{F} \cup \mathcal{B})$ 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 *URMT_Single_Phase*, 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 fail-stop corrupted) crash and fail to deliver any 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 will be contradicted by at most $(t_b - |\mathcal{B}|)$ wires and hence will 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 jth 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 jth position. In a similar manner, 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 jth 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 most 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. The proof now follows from Lemma 2. It is easy to see that **R** does not output 'NULL' in this case.

Theorem 5: If *URMT-Single_Phase* is executed over a field $\mathbb{F}$ with $|\mathbb{F}| \geq \frac{n^3}{\delta}$, then *URMT_Single_Phase* is a strong URMT protocol and terminates with message m with probability at least $1 - \delta$.

Proof: Since no two honest wires contradict each other, from Lemma 4, all the wires removed by **R** during step 1 of message recovery are indeed faulty. We now show that if a wire is corrupted and delivered incorrect polynomial, then it will be contradicted by all the honest wires with high probability. This will ensure that the corrupted wire will be removed in step 1 of the message recovery.

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 $p_j(\alpha_i) = p'_j(\alpha_i)$ with a probability of π_{ij}. Since there are only $n - 1$ points at which these two-polynomials intersect and since α_i was selected uniformly at random from $\mathbb{F}$, 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 **F** is chosen such that $|\mathbb{F}| \geq \frac{n^3}{\delta}$, it follows that a Byzantine corrupted wire w_j will not be contradicted by any honest wire with probability at most δ. In other words, a corrupted $p'_j(x) \neq p_j(x)$, received over w_j may 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 correct message m with probability at least $1 - \delta$. □

Theorem 6: Protocol *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 values. 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 7: Protocol *URMT_Single_Phase* is a single phase communication optimal URMT protocol.

Proof: In Theorem 4, 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 6, 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 δ be the maximum probability of **R** outputting 'NULL'. □

From the remarks made in Comparison 2, a communication optimal URMT protocol tolerating $\mathcal{A}_{t_b}$ should achieve message transmission with constant factor overhead. Our *URMT_Single_Phase* is one such communication optimal protocol. So we have the following corollary.

Corollary 1: Protocol *URMT_Single_Phase* when executed in the presence of $\mathcal{A}_{t_b}$, achieves reliability with "constant factor overhead" by sending $\Theta(n^2)$ field elements with a communication complexity of $O(n^2)$ field elements.

Proof: From Theorem 6, *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'. □

Table 6 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 $p_j(x)$, $1 \le j \le n$, each of degree $n - 1$, where $p_j(x)$ is formed using the jth row of D as follows: the coefficient of x^i, $0 \le i \le n - 1$ in $p_j(x)$ is the $(i + 1)$th element of jth row of D.

2 **S** chooses another n *secret*, distinct and random field elements, $\alpha_1, \alpha_2, ..., \alpha_n$, 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 $1 \le i \le n$. Let $v_{ji} = p_i(\alpha_j)$.

Message recovery by **R***:*

1 Let $\mathcal{F}$ denote the set of wires that delivered nothing and let $\mathcal{B}$ denote 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 $\mathcal{W}\backslash(\mathcal{F} \cup \mathcal{B})$, out of which at most $t_b - |\mathcal{B}|$ could be Byzantine corrupted. Let **R** receive $p'_j(x), \alpha'_j$ and v'_{ji}, $1 \le i \le n$ over $w_j \in \mathcal{W}\backslash(\mathcal{F} \cup \mathcal{B})$. We say that w_j *contradicts* w_i if: $v'_{ji} \ne p'_i(\alpha'_j)$ where $w_i, w_j \in \mathcal{W}\backslash(\mathcal{F} \cup \mathcal{B})$. Among all the wires in $\mathcal{W}\backslash(\mathcal{F} \cup \mathcal{B})$, **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 $w_j \in \mathcal{W}\backslash(\mathcal{F} \cup \mathcal{B})$, which is not removed in previous step, **R** fills the jth row of D' in the following manner: coefficient of x^i, $0 \le i \le n - 1$ in $p'_j(x)$ occupies $(i + 1)$th column in the jth row of D'.

3 After doing the above step for each $w_j \in \mathcal{W}\backslash(\mathcal{F} \cup \mathcal{B})$, 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 $i_1, i_2, ..., i_k$, $k \ge t_b + 1$ denote the index of the rows which are inserted by **R** in D'. Let $y^j_{i_1}, y^j_{i_2}, ..., y^j_{i_k}$, $1 \le j \le n$ denote the values along jth, $1 \le j \le n$ column of D'. **R** checks whether the points $(i_1, y^j_{i_1}), (i_2, y^j_{i_2}), ..., (i_k, y^j_{i_k})$ 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. Notice that this check is required only if $k > (t_b + 1)$ as $t_b + 1$ points will always define a t_b degree polynomial.

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).

Remark 6 (a note on message size used in protocol URMT_Single_Phase): In protocol *URMT_Single_Phase*, we have assumed that $n = 2t_b + t_o + t_f + 1$, the minimum number of wires required for single phase URMT. Out of these n wires, at least $t_b + 1$ are honest and will always deliver values to **R**, even if remaining wires simply stop the communication. This is why we selected the message size to be $n(t_b + 1)$. If there are $n > 2t_b + t_o + t_f + 1$ wires, then there will be more honest wires and hence accordingly we can increase the message size in *URMT_Single_Phase*, such that the communication complexity still satisfies the lower bound.

4.4 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 8: Any multiphase URMT protocol between **S** and **R** over $n \ge 2t_b + t_o + t_f + 1$ wires must communicate $\Omega(\ell)$ 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 t least the message. □

Theorem 9: Let **S** and **R** be connected by $n = 2t_b + t_o + t_f + 1$ wires. Then there exists an efficient, polynomial time communication optimal URMT protocol which sends a message containing ℓ field elements by communicating $O(\ell)$ field elements.

Proof: Suppose there exists $n = 2t_b + t_o + t_f + 1$ wires between **S** and **R**. Then from Ashwinkumar et al. (2008), there exists an efficient $O\left(\log \frac{t_f - t_o}{n - t_f - t_o}\right)$ phase PRMT protocol which sends ℓ field elements (for suitably large ℓ) by communicating $O(\ell)$ field elements. The PRMT protocol of Ashwinkumar et al. (2008) is also a valid multiphase URMT (since any PRMT is by default an URMT protocol with $\delta = 0$) which satisfies the communication complexity lower bound for multiphase URMT. □

We do not know whether there exists an 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.

4.5 Comparison of PRMT with URMT

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 alter the connectivity requirement of RMT protocols (see Comparison 1).

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

3 In the presence of $\mathcal{A}_{t_b}$, it is impossible to design any single phase PRMT protocol which achieves reliability with 'constant factor overhead'. That is sending ℓ field elements by communicating $O(\ell)$ field elements is possible (see Comparison 2). The minimum number of phases required by any PRMT protocol to achieve reliability with 'constant factor overhead' is 3 (Patra et al., 2006). 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 phase complexity of RMT.

5 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 $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$. We then prove the lower bound on the communication complexity of any single phase USMT protocol and show that our bound is *asymptotically tight* by designing a *communication optimal* single phase USMT protocol called *USMT_Single_Phase*.

Kurosawa and Suzuki (2007) proved the lower bound on the communication complexity of any single phase USMT protocol tolerating $\mathcal{A}_{t_b}$ and also presented a near optimum single phase USMT protocol whose total communication complexity *approximately* matches the bound given in Kurosawa and Suzuki (2007). But the USMT protocol of Kurosawa and Suzuki (2007) requires exponential (in n) computation. We show that our communication optimal USMT protocol *USMT_Single_Phase* when executed against $\mathcal{A}_{t_b}$, provides a *polynomial time communication optimal* USMT protocol satisfying the lower bound presented in Kurosawa and Suzuki (2007).

Recently in Araki (2008), a *polynomial time* single phase USMT with $n = 3t_b + 1$ (i.e., with non-optimal connectivity) is presented tolerating $\mathcal{A}_{t_b}$, whose communication complexity almost satisfies the lower bound for single phase USMT given in Kurosawa and Suzuki (2007). As a special case of our single phase communication optimal USMT protocol *USMT_Single_Phase*, we show that in the presence of $\mathcal{A}_{t_b}$ (i.e., $t_o = t_f = t_p = 0$), if $3t_b + 1$ wires are available, then protocol *USMT_Single_Phase* achieves security with constant factor overhead; i.e., it securely sends ℓ field elements in a single phase by communicating $O(\ell)$ field elements. This significantly improves the communication complexity of the single phase USMT of Araki (2008) in the same settings.

From Dolev et al. (1993), any single phase PSMT tolerating $\mathcal{A}_{t_b}$ requires $n = 3t_b + 1$ wires between **S** and **R**. Moreover from Fitzi et al. (2007) and Srinathan et al. (2007b), any single phase PSMT over $n = 3t_b + 1$ tolerating $\mathcal{A}_{t_b}$, needs to communicate $\Omega(n\ell)$ field elements to securely send a message containing ℓ field elements. Thus, with $n = 3t_b + 1$ wires in the presence of $\mathcal{A}_{t_b}$, while it is impossible to design any single phase PSMT protocol with constant factor overhead, it is possible to obtain single phase USMT protocol with constant 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 SMT protocols.

5.1 Single phase USMT protocol tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$: characterisation and lower bound on communication complexity

Theorem 10: Any single phase USMT protocol tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ from **S** to **R** over n wires is possible if and only if $n \geq 2t_b + 2t_o + t_f + t_p$. Moreover, any such single phase USMT protocol is required to communicate $\Omega\left(\frac{n\ell}{n-(2t_b+2t_o+t_f+t_p)}\right)$ field elements in order to send a message containing ℓ field elements.

Remark 7: 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 10, 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 10.

Proof: We first prove the lower bound on the communication complexity. Let Π be any single phase USMT over n wires, tolerating $\mathcal{A}_{(t_b,t_f,t_p)}$, which sends a message m containing $\ell \geq 1$ field elements from $\mathbb{F}$. We now define the following notations:

1 $\mathcal{M}$ denotes the message space from where **S** selects the message to be sent. In our context, $\mathcal{M} = \mathbb{F}^{\ell}$

2 $\mathbf{T}_i^m$ denotes the set of all possible transmissions that can occur on wire $W_i \in \{W_1, ..., W_n\}$, when **S** transmits message $m \in M$ using protocol Π

3 for $j \geq i, \mathbf{M}_{i,j}^m \subseteq \mathbf{T}_i^m \times \mathbf{T}_{i+1}^m \times ... \times \mathbf{T}_j^m$ denotes the set of all possible transmissions that can occur over the wires $\{W_i, W_{i+1}, ..., W_j\}$, when **S** transmits message $m \in \mathcal{M}$ using protocol Π

4 $\mathbf{M}_{i,j} = \bigcup_{m \in \mathcal{M}} \mathbf{M}_{i,j}^m$ and $\mathbf{T}_i = \bigcup_{m \in \mathcal{M}} \mathbf{T}_i^m$. We call $\mathbf{T}_i$ as the *capacity* of wire W_i and $\mathbf{M}_{i,j}$ as the capacity of the set of wires $\{W_i, W_{i+1}, ..., W_j\}$.

In protocol Π, one element from the set $\mathbf{T}_i$ is transmitted over each wire W_i, for $i = 1, ..., n$. Moreover, each element of the set $\mathbf{T}_i$ can be represented by $\log |\mathbf{T}_i|$ bits. Thus, if we can find out each $\mathbf{T}_i$, then the lower bound on the communication complexity of Π is $\sum_{i=1}^{n} \log |\mathbf{T}_i|$ bits. In the sequel, we try to compute $\mathbf{T}_i$.

Since Π is a single phase USMT protocol, it implies that the transmission on any set of $t_b + t_o + t_p$ wires is independent of the message. Otherwise, the adversary will also know the secret message by passively listening the contents of these wires (recall that the eavesdropping capability of $\mathcal{A}_{(t_b,t_f,t_p)}$ is at most $t_b + t_o + t_p$). Thus, for any two-messages $m_1, m_2 \in \mathcal{M}$, it must hold that:

$$\mathbf{M}_{t_b+t_o+t_f+1,2tb+2to+t_f+t_p}^{m_1} = \mathbf{M}_{t_b+t_o+t_f+1,2tb+2to+t_f+t_p}^{m_2}.$$

Notice that the relation above must hold for any selection of $t_b + t_o + t_p$ wires. We focussed on the set $\left\{W_{t_b+t_o+t_f+1}, ..., W_{2t_b+2t_o+t_f+t_p}\right\}$ just for simplicity.

Similarly, since Π is a single phase USMT protocol, the data sent over any $(n - (t_b + t_o + t_f))$ wires during the protocol will always have full information about the secret message. This requirement ensures that even if the adversary simply blocks 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. Thus, it must also hold that:

$$\mathbf{M}_{t_b+t_o+t_f,n}^{m_1} \cap \mathbf{M}_{t_b+t_o+t_f,n}^{m_2} = \emptyset.$$

We again stress that the above relation must hold for any selection of $n - (t_b + t_o + t_f)$ wires. We focussed on the set $\left\{W_{t_b+t_o+t_f+1}, ..., W_n\right\}$ just for simplicity. As mentioned earlier, $\mathbf{M}_{t_b+t_o+t_f+1,2t_b+2t_o+t_f+t_p}^{m}$ will be same for all messages m. Thus, in order that the above relation holds, it must hold that $\mathbf{M}_{2t_b+2t_o+t_f+t_p+1,n}^{m}$ is unique for every message m. This implies that:

$$\left|\mathbf{M}_{2t_b+2t_o+t_f+t_p+1,n}\right| = |\mathcal{M}|.$$

From the definition of $\mathbf{T}_i$ and $\mathbf{M}_{i,j}$, we get:

$$\Pi_{i=2t_b+2t_o+t_f+t_p+1}^{n} |\mathbf{T}_i| \geq \left|\mathbf{M}_{2t_b+2t_o+t_f+t_p+1,n}\right| \geq |\mathcal{M}|.$$

Let $g = n - (2t_b + 2t_o + t_f + t_p)$. The above inequality holds for any selection of g wires $\mathcal{D} \subset \{W_1, ..., W_n\}$, where $|\mathcal{D}| = g$; i.e., $\Pi_{W_i \in \mathcal{D}} |\mathbf{T}_i| \geq |\mathcal{M}|$. In particular, it holds for every selection $\mathcal{D}_k = \{W_{kg+1} \bmod n, W_{kg+2} \bmod n, ..., W_{kg+g} \bmod n\}$, with $k \in \{0, ..., n-1\}$.

If we consider all above $\mathcal{D}_k$ sets separately, then each wire is accounted for exactly g times. Thus, the product of the capacities of all $\mathcal{D}_k$ yields the capacity of the full wire set to the gth power, and since each $\mathcal{D}_k$ has capacity at least $|\mathcal{M}|$, we get:

$$|\mathcal{M}|^n \leq \Pi_{k=0}^{n-1} \Pi_{W_j \in \mathcal{D}_k} |\mathbf{T}_j| = \left(\Pi_{i=0}^{n} |\mathbf{T}_i|\right)^g,$$

and therefore,

$$n \log(|\mathcal{M}|) \leq g \sum_{i=1}^{n} \log(|\mathbf{T}_i|).$$

As $\log(|\mathcal{M}|) = \ell \log(|\mathbb{F}|)$, from the above inequality, we ge:t

$$\sum_{i=1}^{n} \log(|\mathbf{T}_i|) \geq \left(\frac{n\ell \log(|\mathbb{F}|)}{g}\right) \geq \left(\frac{n\ell \log(|\mathbb{F}|)}{n - (2t_b + 2t_o + t_f + t_p)}\right).$$

As mentioned earlier, $\sum_{i=1}^{n} \log(|\mathbf{T}_i|)$ denotes the lower bound on the communication complexity of protocol Π in bits. From the above inequality, we find that the lower bound on the communication complexity of protocol Π is $\left(\frac{n\ell \log(|\mathbb{F}|)}{n-(2t_b+2t_o+t_f+t_p)}\right)$ bits. Now each field element from $\mathbb{F}$ can be pre-presented by $\log(|\mathbb{F}|)$ bits. Thus, the lower bound on the communication complexity of protocol Π is $\left(\frac{n\ell}{n-(2t_b+2t_o+t_f+t_p)}\right)$ field elements. This completes the derivation of lower bound on the communication complexity of single phase USMT tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$.

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 PSMT and USMT): From Srinathan (2006), single phase PSMT protocol tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ is possible iff there exists $n \geq 3t_b + 2_{\text{to}} + t_f + t_p + 1$ wires between **S** and **R**. But from Theorem 10, we find that single phase USMT tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ is possible iff there exists $n \geq 2t_b + 2t_o + t_f + t_p + 1$ wires between **S** and **R**. This shows that allowing a negligible error probability (only in the reliability), significantly helps in the possibility of single phase SMT protocols.

Comparison 4 (communication complexity of single phase USMT and PSMT): In Srinathan (2006), 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. From Theorem 10, any single phase USMT tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ over $n \geq 2t_b + 2t_o + t_f + t_p + 1$ wires has to communicate $\Omega\left(\frac{n\ell}{n-(2t_b+2t_o+t_f+t_p)}\right)$ field elements to send a message containing ℓ field elements. Let us fix $n = 3t_b + 2t_o + t_f + t_p + 1$ such that both PSMT and USMT is possible [notice that with $n = 2t_b + 2t_o + t_f + t_p + 1$ USMT is possible but PSMT is not possible (Srinathan, 2006)]. With $n = 3t_b + 2t_o + t_f + t_p + 1$, the lower bounds for PSMT and USMT become $\Omega(n\ell)$ and $\Omega\left(\frac{n\ell}{t_b}\right)$ field elements respectively. Specifically, if we consider $\mathcal{A}_{t_b}$ then n must be at least $3t_b + 1$ for PSMT to be possible (notice that USMT requires only $2t_b + 1$ wires tolerating $\mathcal{A}_{t_b}$). With $n = 3t_b + 1$, the lower bounds for PSMT and USMT become $\Omega(n\ell)$ and $\Omega(\ell)$ field elements respectively for now $t_b = \Theta(n)$. Hence, with $n = 3t_b + 1$ while USMT can be achieved with constant factor overhead tolerating $\mathcal{A}_{t_b}$, PSMT can not be achieved. This shows the power of allowing a negligible error probability (only in the reliability) in single phase SMT.

In the sequel, we design a single phase *communication optimal* USMT protocol, whose total communication complexity matches the bound proved in Theorem 10, thus showing that the bound is tight.

5.2 Single phase communication optimal 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 10 is *asymptotically tight*. We require the field size $|\mathbb{F}| \geq \frac{2n^3}{\delta}$, to bound the error probability by δ in *USMT_Single_Phase*. We first briefly recall an algorithm from Srinathan et al. (2004), 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 (Srinathan et al., 2004):

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 (Srinathan et al., 2004): 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 $f \times f$ subdeterminant 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**. We call our method as pad *establishment technique* which is very similar to *extrapolation technique* discussed in Section 4.

Pad establishment technique

Suppose $n = 2t_b + 2t_o + t_f + t_p + 1$. **S** randomly chooses $(t_b + t_o + t_p + 1) \times (n + t_p)$ 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 jth, $1 \leq j \leq t_b + t_o + t_p + 1$ row contains the elements $M_{j1}, M_{j2}, \ldots, M_{j(n+t_p)}$. Now consider the first column of A, containing $M_{11}, M_{21}, \ldots, M_{(t_b,t_o,t_p+1)1}$. **S** constructs the unique $t_b + t_o + t_p$ degree polynomial $q_1(x)$ passing through the points $(1, M_{11})$, $(2, M_{21})$, ..., $(t_b + t_o + t_p + 1, M_{(t_b,t_o,t_p+1)1})$. **S** then evaluates $q_1(x)$ 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 $c_{11}, c_{21}, \ldots, c_{(t_b+to+tf)1}$. **S** repeats the procedure for all the $n + t_p$ columns of A. In general, considering the ith, $1 \leq i \leq n + t_p$ column of A consisting of the elements $M_{1i}, M_{2i}, \ldots, M_{(t_b,t_o,t_p+1)i}$, **S** constructs the unique $t_b + t_o + t_p$ degree polynomial $q_i(x)$ passing through the points

$(1, M_{1i})$, $(2, M_{2i})$, ..., $((t_b + t_o + t_p + 1), M_{(t_b+t_o+t_p+1)i})$. Then $q_i(x)$ is evaluated at $t_b + t_o + t_f$ values of x, namely at $x = t_b + t_o + t_p + 2$, $t_b + t_o + t_p + 3$, ..., n to obtain $c_{1i}, c_{2i}, ..., c_{(t_b+t_o+t_f)i}$. Finally, **S** obtains a rectangular array D of size $n \times (n + t_p)$ containing $n \times (n + t_p)$ elements, where:

$$D = \begin{bmatrix} M_{11} & \cdots & M_{1(n+t_p)} \\ M_{21} & \cdots & M_{2(n+t_p)} \\ \cdots & \cdots & \cdots \\ M_{j1} & \cdots & M_{j(n+t_p)} \\ \cdots & \cdots & \cdots \\ M_{(t_b+t_o+t_p+1)1} & \cdots & M_{(t_b+t_o+t_p+1)(n+t_p)} \\ c_{11} & \cdots & c_{1(n+t_p)} \\ c_{21} & \cdots & c_{2(n+t_p)} \\ \cdots & \cdots & \cdots \\ c_{j1} & \cdots & c_{j(n+t_p)} \\ \cdots & \cdots & \cdots \\ c_{(t_b+t_o+t_f)1} & \cdots & c_{(t_b+t_o+t_f)(n+t_p)} \end{bmatrix} = \begin{bmatrix} A \\ C \end{bmatrix}$$

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 8 (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 constitute the message which **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.

We now prove the properties of D generated using *pad establishment technique*.

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 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* in Table 7. Let the message be denoted by $m = (m_1\ m_2\ \ldots\ m_{t_b+t_o+t_f+t_p+1})$ a and the set of n wires be denoted as $\mathcal{W} = \{w_1, w_2, ..., w_n\}$.

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 other words w_j is Byzantine corrupted.

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

Lemma 11: In the protocol *USMT_Single_Phase*, 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'.

Proof: We consider the worst case, where $t_o + t_f$ wires which are omission and fail-stop corrupted, 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 jth 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 jth position. Like this 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 jth 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. □

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 of the coefficients of all the n polynomials $p_j(x)$, $1 \le j \le n$. **R** then correctly constructs the vector y by applying EXTRAND algorithm to E and recovers m by computing $m = d \oplus y$. □

Theorem 11: 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 $p_i(x)$, $1 \le i \le t_b + t_o + t_p$ 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 $1 \le i \le n$ and $1 \le j \le t_b + t_o + t_p$. 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 $p_i(x)$, $t_b + t_o + t_p + 1 \le i \le n$ 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 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 12: If $|\mathbb{F}| \ge \frac{2n^3}{\delta}$, then protocol *USMT_Single_Phase* is a strong USMT protocol and terminates with the correct message m with probability at least $1 - \delta$.

Proof: From the protocol, it is easy to see that no two honest wires (which has delivered correct values and polynomials) contradict each other. From Lemma 10, all the wires removed by **R** during step 2 of message recovery are indeed faulty. We now 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 $p'_j(x) \ne p_j(x)$ 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 $n - 1 + t_p$) and since α_i was selected uniformly at random from $\mathbb{F}$, we have $\pi_{ij} \le n - 1 + t_p \,|\, \mathbb{F} \,|$ 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 $\sum_{i,j} \pi_{ij} \le \frac{n^2(n-1+t_p)}{|\mathbb{F}|}$. Now $n^2(n - 1 + t_p) < n^2(2n) < 2n^3$. Since $|\mathbb{F}| \ge \frac{2n^3}{\delta}$, it follows that corrupted $p'_j(x) \ne p_j(x)$, 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 correct message with probability at least $1 - \delta$. □

Theorem 13: 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 $(t_b + t_o + t_f + t_p + 1) \log |\mathbb{F}| = \Theta(n \log |\mathbb{F}|)$ bits by communicating $O(n_2 \log |\mathbb{F}|)$ 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. 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 10, 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*. □

Table 7 Single phase USMT protocol

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 $M_{11}, M_{21}, ..., M_{1(n+t_p)}$, $M_{21}, M_{22}, ..., M_{2(n+t_p)}, ..., M_{(t_b+t_o+t_p+1)1}, M_{(t_b+t_o+t_p+1)2}, ..., M_{(t_b+t_o+t_p+1)(n+t_p)}$, 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 $p_j(x)$, $1 \le j \le n$, each of degree $n - 1 + t_p$ where $p_j(x)$ is formed using the jth row of D as follows: the coefficient of x^i, $0 \le i \le n - 1 + t_p$ in $p_j(x)$ is the $(i + 1)$th element of jth row of D.

3 **S** chooses another n *secret* and random field elements, $\alpha_1, \alpha_2, ..., \alpha_n$. Over w_j, **S** sends the following to **R**: the polynomial $p_j(x)$, the secret value α_j and the n tuple $\{p_i(\alpha_j): 1 \le i \le n\}$. 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 $y = [y_1\ y_2\ \cdots\ y_{t_b+t_o+t_f+t_p+1}] = \text{EXTRAND}_{n(n+t_p),t_b+t_o+t_f+t_p+1}(E)$ and broadcasts $d = m \oplus y$ to **R**.

Message recovery by **R**:

1 Let $\mathcal{F}$ denote the set of wires that delivered nothing and let $\mathcal{B}$ denote 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 can not 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** receive $p'_j(x), \alpha'_j$ and the n tuple $\{v'_{ji} : 1 \le i \le n\}$ 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: $v'_{ji} \ne p'_i(\alpha'_j)$, 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 jth row of D' in the following manner: coefficient of x^i, $0 \le i \le n - 1 + t_p$ in $p'_j(x)$ occupies $(i + 1)$th column in the jth row of D'; i.e., the coefficients of $p'_j(x)$ are inserted in jth row of D' such that the coefficient of x^i in $p'_j(x)$ occupies $(i + 1)$th column in the jth 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 $i_1, i2, ..., i_k$, $k \ge t_b + t_o + t_p + 1$ denote the index of the rows which are inserted by **R** in D'. Let $y^j_{i_1}, y^j_{i_2}, ..., y^j_{i_k}$, $1 \le j \le n + t_p$ denote the values along jth, $1 \le j \le n$ column of D'. **R** checks whether the points $(i_1, y^j_{i_1}), (i_2, y^j_{i_2}), ..., (i_k, y^j_{i_k})$ 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. Moreover, if k is exactly equal to $t_b + t_o + t_p + 1$, then the checking will always pass. If the test fails for at least one column of D', then **R** outputs 'NULL' and halts. Otherwise, proceed to the next step.

5 Using the already inserted rows of D', **R** regenerates the complete D correctly (see Lemma 12). **R** now knows all the polynomials $p_i(x)$, $1 \le i \le n$ and hence, the list E, which is the concatenation of rows of D. **R** then computes $y = [y_1\ y_2\ \cdots\ y_{t_b+t_o+t_f+t_p+1}] = \text{EXTRAND}_{n(n+t_p),t_b+t_o+t_f+t_p+1}(D)$ and recovers m by computing $d = m \oplus y$.

5.2.1 Single phase USMT with constant factor overhead tolerating $\mathcal{A}_{t_b}$

From Dolev et al. (1993), any single phase PSMT tolerating $\mathcal{A}_{t_b}$ requires $n = 3t_b + 1$ wires between **S** and **R**. Moreover, from Fitzi et al. (2007) and Srinathan et al. (2007b), 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 over a $3t_b + 1$-(**S**, **R**) connected network. 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 $n = 3t_b + 1$ 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 **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', **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 14: 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, tolerating $\mathcal{A}_{t_b}$.

Proof: Follows from the above discussion. □

Recently in Araki (2008), a single phase USMT protocol with $n = 3t_b + 1$ and tolerating $\mathcal{A}_{t_b}$ is provided. However, the protocol does not provides security with constant factor overhead; i.e., the communication complexity of the protocol is much more than $O(\ell)$. Thus, our single phase USMT when executed with $n = 3t_b + 1$ tolerating $\mathcal{A}_{t_b}$, significantly improves the communication complexity of the USMT protocol of Araki (2008) in the same settings.

5.2.2 Lower bound on communication complexity (Kurosawa and Suzuki, 2007) and our polynomial time single phase communication optimal USMT protocol tolerating $\mathcal{A}_{t_b}$

In Kurosawa and Suzuki (2007), the authors have shown that single phase USMT tolerating $\mathcal{A}_{t_b}$ is possible iff $n \geq 2t_b + 1$. In addition, they have shown that for any single phase USMT protocol with $n = 2t_b + 1$, the following must hold:

$$|\mathcal{X}_i| \geq \frac{|\mathcal{S} - 1|}{\delta} + 1 \tag{1}$$

where $\mathcal{S}$ denotes the set of possible secret messages from which **S** intends to send one element to **R**, $\mathcal{X}_i$ denotes the set of possible data sent through the ith wire in the protocol and $0 < \delta < 1$ 2 is the error probability of the protocol. In any single phase USMT protocol, one element from $\mathcal{X}_i$ is sent through the ith channel. Now each element of $\mathcal{X}_i$ can be represented by $\log(| \mathcal{X}_i |)$ bits. Similarly, each message from $\mathcal{S}$ can be represented by $\log(| \mathcal{S} |)$ bits. Thus, inequality (1) says that any single phase USMT protocol must communicate $\Omega(n \log(| \mathcal{X}_i |))$ bits to securely send $\log(|\mathcal{S}|)$ bits with error probability of at most $0 < \delta < \frac{1}{2}$.

In Kurosawa and Suzuki (2007), the authors have proposed a near optimum single phase USMT protocol whose total communication complexity *approximately* matches the bound given in inequality (1). However, the computation done by **R** in their protocol is exponential in n. We now show that if we execute our single phase USMT protocol *USMT_Single_Phase* against only $\mathcal{A}_{t_b}$ over $n = 2t_b + 1$ wires, then it satisfies the lower bound given in inequality (1). 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. Recall that the field size $|\mathbb{F}|$ must be at least $\frac{2n^3}{\delta}$ for bounding the error probability of *USMT_Single_Phase* by δ. We select $\kappa > 0$ such that $\delta \approx 2^{-\kappa}$ and express the error probability by $2^{-\kappa}$ (instead of δ). So now $|\mathbb{F}| \geq 2n^3 2^{\kappa}$. So a field element can be represented by $O(\log n + \kappa)$ bits. Our protocol securely sends $O((t_b + 1)(\log n + \kappa))$ bits (if $n = 2t_b + 1$, then $t_b = \Theta(n)$) by communicating $O(n^2(\log n + \kappa))$ bits.

We now show that the communication complexity of our protocol (with $n = 2t_b + 1$) satisfies the bound given in inequality (1). In our protocol message space is $\mathbb{F}^{t_b+1}$. So $\mathcal{S} = \mathbb{F}^{t_b+1}$ and thus, $\log(|\mathcal{S}|) = (t_b + 1)\log(|\mathbb{F}|) = (t_b + 1)(\log n + \kappa)$. Substituting $\delta = 2^{-\kappa}$ and value of $\mathcal{S}$ in inequality (1), we get $|\mathcal{X}_i| \geq \frac{|\mathbb{F}^{t_b+1}|-1}{2^{-\kappa}} + 1$ and thus, $\log(|\mathcal{X}i|) \geq \kappa + (t_b + 1)(\log n + \kappa)$. So according to the lower bound given by inequality (1), our protocol must communicate $\Omega(n(t_b + 1)(\log n + \kappa)) = \Omega(n^2(\log n + \kappa))$ bits to securely send $(t_b + 1)(\log n + \kappa) = \Theta(n(\log n + \kappa))$ bits. However, the total communication complexity of our protocol is $\Theta(n^2(\log n + \kappa))$ bits.

5.3 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 in the reliability *significantly* helps in the possibility of single phase SMT protocols (see Comparison 3)
- allowing a negligible error probability in the reliability *significantly* reduces the communication complexity of single phase SMT protocols (see Comparison 4 and Subsection 5.2.1)
- allowing a negligible error probability in the reliability helps in the possibility of single phase SMT protocol tolerating which achieves security with constant factor overhead against $\mathcal{A}_{t_b}$ (see Theorem 14).

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

As mentioned earlier, one of the key parameters of any SMT protocol is the number of phases. In the context of PSMT, it is well known that allowing interaction between **S** and **R** significantly helps in reducing the connectivity requirement and lower bound on communication complexity of PSMT protocols (see Table 2 and Table 3). In this section, we show that same holds for USMT also. Here, we provide the characterisation 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 our lower bound is *asymptotically tight*. Comparing these results with the results for single phase USMT, we find that allowing interaction between **S** and **R** significantly reduces the connectivity requirement of USMT and also helps in reducing the communication complexity of USMT protocols. Finally, comparing our results on multiphase USMT with the results on multiphase PSMT (given in last rows of Table 2 and Table 3), we observe a notable effect of allowing a negligible error probability in reliability of multiphase SMT protocols.

6.1 Characterisation for multiphase USMT protocol tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$

Theorem 15: Multiphase USMT between **S** and **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 $(t_b + \max(t_b, t_p) + t_o + t_f + 1)$-(**S**, **R**)-connected.

Proof:

Necessity: We consider two cases for proving the necessity.

1 Case 1: $tp \leq t_b$: In this case, the necessity condition says that the network should be $(2t_b + t_o + t_f + 1)$-(**S**, **R**)-connected. Since the condition is necessary for URMT (Theorem 3), it is obviously necessary for USMT.

2 Case 2: $t_p > tb$: In this case, the necessity condition says that the network should be $(t_b + t_p + t_o + t_f + 1)$-(**S**, **R**)-connected. This condition is necessary for USMT because if the network is $(t_b + t_p + t_o + t_f)$-(**S**, **R**)-connected, then the adversary may strategise to simply block all message through $(t_b + t_o + t_f)$ vertex disjoint paths and thereby ensure that every value received by **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 (1927) theorem, 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 *USMT_Three_Phase* to securely send a single field element $m \in \mathbb{F}$. The protocol is similar to the USMT protocol of Franklin and Wright (2000) and is given in Table 8.

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 Franklin and Wright (2000)]. Since $n = t_b + \max(t_b, t_p) + t_o + t_f + 1$, there exists at least one wire say wi, 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 *USMT_Three_Phase* is $O(n^2)$ field elements, where the field size $|\mathbb{F}|$ is set appropriately as a function of δ. □

Comparison 5 (possibility of multiphase PSMT and USMT): From Table 2 (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 $n \geq 2t_b$ + to + t_f + t_p + 1 wires between **S** and **R**. From Theorem 15, any r ≥ 2 phase USMT protocol tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ is possible iff there exists $n \geq t_b + \max(t_b, t_p) + t_o + t_f + 1$ wires between **S** and **R**. Therefore, 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 SMT protocol.

The protocol *USMT_Three_Phase* is used to prove the sufficiency of Theorem 15. Using it as a black-box, we will design a communication optimal multiphase USMT protocol. Before that, in the sequel we prove the lower bound on the communication complexity of any multiphase USMT protocol.

6.2 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 $n \geq t_b + \max(t_b, t_p) + t_o + t_f + 1$. 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 will remove it. While proving the lower bound on the communication complexity, we assume that $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ will fully utilise 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, fail-stop and passive fashion respectively, where $b \leq t_b$, $F \leq t_f$ and $P \leq t_b + t_o + t_p$.

Theorem 16: 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 9: 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 of Theorem 16 follows from Lemma 13 and Lemma 14, which are proved below.

Lemma 13: The communication complexity of any multiphase 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 $n - F$ 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 fetection (USMTED). We then prove the equivalence of the communication complexity of USMTED protocol to send message **M** and the share complexity of distributing n shares for **M** such that any set of $n - F$ 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 the communication complexity of any multiphase USMT protocol is at least equal to 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 15: 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 **R** detects it, and aborts.

3 If adversary crashes $F \leq t_f$ wires and does no malicious corruption, then **R** recovers message correctly. Else if adversary either crashes more than t_f wires or do some malicious modifications (or both), then **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 **M** is equivalent to the problem of distributing n shares for **M** such that any set of $n - F$ shares has full information about **M** while any set of P shares has no information about **M**.

Claim 1: Let Π be a USMTED protocol executed over n wires between **S** and **R**. In an execution of Π for sending a message **M**, the data s_i, $1 \leq i \leq n$ sent by the **S** along the wires w_i, $1 \leq i \leq n$, form n shares for **M** such that any set of $n - F$ shares has full information about **M** while any set of P shares has no information about **M**.

Proof: The fact that any set of P shares have no information about **M** follows directly from property 1 and 4 of definition of USMTED. We now show that any set of $n - F$ shares has full information about **M**. The proof is by contradiction. For a set of wires A, let *Message*(**M**, A) denote the set of messages sent along the wires in A during the execution of USMTED to send **M**. Now for any set C of honest wires with $| C | \geq n - F$, *Message*(**M**, C) should uniquely determine the message **M**. Suppose not, then there exists another message **M′** such that $Message(\mathbf{M}, C) = Message(\mathbf{M}', C)$. a 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 **M** and **M′**, there exists an adversary strategy such that view of **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 corruptions take place. □

The above claim also says that the communication complexity of USMTED protocol to send **M** is same as the share complexity (sum of the length of all shares) of distributing n shares for a message **M** such that any set of $n - F$ shares has full information about **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. **S** and **R** are modelled as polynomial time Turing machines with access to a random tape. The number of random bits used by **S** and **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 **S** and **R** respectively. The message **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 **S** and **R** along all the wires.

Table 8 A three-phase USMT protocol

Protocol USMT_Three_Phase – 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 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** receive 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 values of t_b and t_p, n is at least $2t_b + t_o + t_f + 1$ and any information which is broadcast over these many 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 $H = \{w_i \mid u_i = (K\rho_{i1} + \rho_{i2})\}$. 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 $m \oplus \rho$ 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' (which is equal to ρ with very high probability). If z' is the blinded message received, **R** outputs $m = z' \oplus \rho'$.

Definition 16: 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 **S** and the receiver **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 Π, $\mathbf{M}$, r_1, r_2 are obvious from the context, we drop them and denote the passive transcript restricted to a wire w_i by $\mathcal{T}_{wi}$. 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 **S** to compute $\mathcal{T}(\Pi, \mathbf{M}, r_1, r_2)$ by simulating **R** with random tape r_2. Similarly given (M, r_1, r_2) **R** can compute $\mathcal{T}(\Pi, \mathbf{M}, r_1, r_2)$ by simulating **S** with random tape r_1. Note that although **S** and **R** require both r_1, r_2 to generate the transcript, **R** requires only r_2 in order to obtain the message **M** from the transcript $\mathcal{T}(\Pi, \mathbf{M}, r_1, r_2)$. This is clear since **R** does not have access to r_1 during the execution of Π but still can retrieve the message **M** from the messages exchanged.

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

Definition 17: A passive transcript $\mathcal{T}_B$, with $n - F \le |B| \le n$ is said to be a valid fault-free transcript with respect to **R**, if there exists random string r_2 and message **M**, such that USMT protocol Π at **R**, with r_2 as the contents of the random tape and $\mathcal{T}_B$ as the messages exchanged, terminates by outputting the message **M**.

Definition 18: Two transcripts $\mathcal{T}_B$ and $\mathcal{T}'_B$, where $n - F \le |B| \le n$ are said to be adversely close if the two transcripts differ only on a set of wires A such that $|A| \le b + (|B| - (n - F))$. Formally $\left|w_i \mid \mathcal{T}_{w_i} \ne \mathcal{T}'_{w_i}\right| \le b + (|B| - (n-F))$.

We next claim an important property of valid fault free transcripts.

Claim 2: No 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 $\mathbf{M}$, $\mathbf{M}'$, can be adversely close to each other, where $n - F \le B \le n$, irrespective of the value of r_1, r_1', r_2 and r_2'.

Proof: Suppose there exists r_1, r_1', r_2 and r_2' and two different messages $\mathbf{M}$, $\mathbf{M}'$, such that the 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. This implies that there is a set of wires A, where $|A| \le b + (|B| - (n - F))$, 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 $|Y| = (|B| - (n - F))$. If such transcripts exist, then adversary can also generate $\mathcal{T}_B(\Pi, \mathbf{M}, r_1, r_2)$ by simulating **S** with message $\mathbf{M}$ and random coin r_1 and simulating **R** with random coin r_2. In a similar way, he can simulate **S** and **R** and generate $\mathcal{T}_B(\Pi, \mathbf{M}', r_1', r_2')$.

Now consider the following adversary behaviour: in each execution of Π, irrespective of the random coins of **S**, **R** and irrespective of the message selected by **S**, adversary guesses that **S** wants to send $\mathbf{M}$ using randomness r_1, while **R** is using randomness r_2'. Now irrespective of whether adversary's guess is correct or not, adversary blocks the messages over the wires in Y and tries to change the messages along wires in X such that the view of **S** becomes $\mathcal{T}_{B-Y}(\Pi, \mathbf{M}, r_1, r_2)$ while the view of R becomes $\mathcal{T}_{B-Y}(\Pi, \mathbf{M}', r_1', r_2')$.

Notice that if either **S** or **R** (or both) behaves differently, as opposed to adversary's guess then adversary will not be able to generate the above views at **S** and **R**'s end and will be caught. But in an execution of Π, where **S** indeed wants to send $\mathbf{M}$ using randomness r_1, while **R** is using randomness r_2', adversary will be successful in causing $\mathcal{T}_{B-Y}(\Pi, \mathbf{M}, r_1, r_2)$ and $\mathcal{T}_{B-Y}(\Pi, \mathbf{M}', r_1', r_2')$ to be **S** and **R**'s view respectively, at the end of the protocol. In such an execution, **R** will end up outputting $\mathbf{M}' \ne \mathbf{M}$, which violates the property of URMT. This shows a contradiction. □

Table 9 Single phase protocol USMTED

Protocol USMTED

- **S** computes the passive transcript $\mathcal{T}(\Pi, \mathbf{M}, r_1, r_2)$ for some random r_1 and r_2 and sends $\mathcal{T}(\Pi, \mathbf{M}, r_1, r_2, w_i)$ to **R** along w_i.
- If **R** does not receive information through at least $n - F$ wires then **R** outputs ERROR and stop. Otherwise, let **R** receive information over the set of wires $B = \{w_{i1}, w_{i2}, ..., w_i\}$ where $n - F \le |B| \le n$. **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 $M \in \{0, 1\}^{p(n)}$ 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.

Till now, we have shown that a passive transcript over at least $n - F$ correct wires allows **R** to output **M** correctly. We now show how to reduce a multiphase USMT protocol into a single phase USMTED protocol. The USMTED protocol is given in Table 9.

Claim 3: The communication complexity of any multiphase USMT protocol Π to send **M** is at least equal to the communication complexity of USMTED protocol. Moreover protocol USMTED satisfies the properties given in Definition 15.

Proof: Let Π be any multiphase USMT protocol and $\Pi^{passive}$ denotes an execution of Π where the adversary does only eavesdropping and does 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 types of corruptions, 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 15. □

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 $n - F$ 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 $n - F$ shares has full information about the message while any set of P shares has no information about the message is $\Omega\left(\frac{n\ell}{(n-F-P)}\right)$.

Proof: To prove this lemma, we use similar arguments as used in deriving the lower bound on the communication complexity of single phase USMT. We now define the following notations:

1 $\mathcal{M}$ denotes the message space from where the message m is selected. In our context, $\mathcal{M} = \mathbb{F}^{\ell}$.

2 For $i = 1, ..., n$, $\mathbf{X}_i^m$ denotes the set of all possible ith share corresponding to message $m \in \mathcal{M}$.

3 For $j \geq i$, $\mathbf{M}_{i,j}^m \subseteq \mathbf{X}_i^m \times \mathbf{X}_{i+1}^m \times \ldots \times \mathbf{X}_j^m$ denotes the set of all possible $\{i\text{th}, (i+1)\text{th}, ..., j\text{th}\}$ shares, corresponding to message $m \in \mathcal{M}$.

4 $\mathbf{M}_{i,j} = \bigcup_{m \in \mathcal{M}} \mathbf{M}_{i,j}^m$ and $\mathbf{X}_i = \bigcup_{m \in \mathcal{M}} \mathbf{X}_i^m$. We call $\mathbf{X}_i$ as the *capacity* of ith share and $\mathbf{M}_{i,j}$ as the *capacity* of the set of $\{i\text{th}, (i+1)\text{th}, ..., j\text{th}\}$ shares.

To generate n shares for message m, one element from the set $\mathbf{X}_i$ is selected as the ith share, for $i = 1, ..., n$. Moreover, each element of the set $\mathbf{X}_i$ can be represented by $\log |\mathbf{X}_i|$ bits. Thus, if we can find out each $\mathbf{X}_i$, then the share complexity corresponding to m will be $\sum_{i=1}^{n} \log|\mathbf{X}_i|$ bits. In the sequel, we try to compute $\mathbf{X}_i$.

From the properties of share distribution, any set of P shares is independent of the message. Thus, for any two messages $m_1, m_2 \in \mathcal{M}$, it must hold that:

$$\mathbf{M}_{F+1,F+P}^{m_1} = \mathbf{M}_{F+1,F+P}^{m_2}.$$

Notice that the relation above must hold for any selection of P shares. We focussed on the set of $\{(F+1)\text{th}, ..., (F+P)\text{th}\}$ shares just for simplicity. Also, from the properties of share distribution, any set of $n - F$ shares have full information about the message m and uniquely determine m. Thus, it must also hold that:

$$\mathbf{M}_{F+1,n}^{m_1} \cap \mathbf{M}_{F+1,n}^{m_2} = \emptyset.$$

We again stress that the above relation must hold for any selection of $n - F$ shares. We focussed on the set of $\{(F+1)\text{th}, ..., n\text{th}\}$ shares just for simplicity. As mentioned earlier, $\mathbf{M}_{F+1,F+P}^{m}$ will be same for all messages m. Thus, in order that the above relation holds, it must hold that $\mathbf{M}_{F+P+1,n}^{m}$ is unique for every message m. This implies that:

$\left|\mathbf{M}_{F+P+1,n}\right| = |\mathcal{M}|$.

From the definition of $\mathbf{X}_i$ and $\mathbf{M}_{i,j}$, we get:

$$\prod_{i=F+P+1}^{n} \left|\mathbf{X}_i\right| \geq \left|\mathbf{M}_{F+P+1,n}\right| \geq |\mathcal{M}|.$$

Let $g = n - (F + P)$. The above inequality holds for any set of g shares $\mathcal{D}$, where $|\mathcal{D}| = g$; i.e., $\Pi_{i\in\mathcal{D}} |\mathbf{X}_i| \geq |\mathcal{M}|$. In particular, it holds for every selection $\mathcal{D}_k$ of {(kg + 1)th mod n, (kg + 2)th mod n, ..., (kg + g)th mod n} shares, with $k \in \{0, ..., n-1\}$.

If we consider all above $\mathcal{D}_k$ sets separately, then each of the n share is accounted for exactly g times. Thus, the product of the capacities of all $\mathcal{D}_k$ yields the capacity of the full share set to the gth power, and since each $\mathcal{D}_k$ has capacity at least $|\mathcal{M}|$, we get:

$$|\mathcal{M}|^n \leq \prod_{k=0}^{n-1} \prod_{j\in\mathcal{D}_k} \left|\mathbf{X}_j\right| = \left(\prod_{i=1}^{n} \left|\mathbf{X}_j\right|\right)^g,$$

and therefore,

$$n \log\left(|\mathcal{M}|\right) \leq g \sum_{i=1}^{n} \log\left(|\mathbf{X}_i|\right).$$

As $\log(|\mathcal{M}|) = \ell \log(|\mathbb{F}|)$, from the above inequality, we get:

$$\sum_{i=1}^{n} \log\left(|\mathbf{X}_i|\right) \geq \left(\frac{n\ell \log(|\mathbb{F}|)}{g}\right) \geq \left(\frac{n\ell \log(|\mathbb{F}|)}{n-(F+P)}\right).$$

As mentioned earlier, $\sum_{i=1}^{n} \log\left(|\mathbf{X}_i|\right)$ denotes the share complexity in bits of distributing n shares of a message m. From the above inequality, we find that the share complexity is $\Omega\left(\frac{n\ell \log(|\mathbb{F}|)}{n-(F+P)}\right)$ bits. Now each field element from $\mathbb{F}$ can be pre-presented by $\log(|\mathbb{F}|)$ bits. Thus, the share complexity is $\Omega\left(\frac{n\ell}{n-(F+P)}\right)$ field elements. □

Since $P \leq t_b + t_o + t_p$ and $F \leq t_f$,

$$\Omega\left(\frac{n\ell}{n-F-P}\right) = \Omega\left(\frac{n\ell}{n-\left(t_b+t_o+t_f+t_p\right)}\right).$$

Theorem 16 now follows from Lemma 13 and Lemma 14. □

Comparison 6: (lower bound on communication complexity of single phase USMT and PSMT): In Srinathan (2006), it is shown that any multiphase PSMT tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ over $n \geq 2t_b + t_o + t_f + t_p + 1$ wires has to communicate $\Omega\left(\frac{n\ell}{n-(2t_b+t_o+t_f+t_p)}\right)$ field elements to send a message containing ℓ field elements. From Theorem 16, any single phase USMT tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ over $n \geq t_b + \max(t_b, t_p) + t_o + t_f + 1$ wires has to communicate $\Omega\left(\frac{n\ell}{n-(t_b+t_o+t_f+t_p)}\right)$ field elements to send a message containing ℓ field elements. Let us fix $n = 2t_b + t_o + t_f + t_p + 1$ for which both PSMT and USMT is possible. With $n = 2t_b + t_o + t_f + t_p + 1$, the lower bounds for PSMT and USMT become $\Omega(n\ell)$ and $\Omega\left(\frac{n\ell}{t_b}\right)$ field elements respectively. Particularly, if we consider $\mathcal{A}_{t_b}$ then n must be at least $2t_b + 1$ for both PSMT and USMT to be possible. With $n = 2t_b + 1$, the lower bounds for PSMT and USMT become $\Omega(n\ell)$ and $\Omega(\ell)$ field elements respectively for now $t_b = \Theta(n)$. Hence, with $n = 2t_b + 1$ while USMT can be achieved with constant factor overhead tolerating $\mathcal{A}_{t_b}$, PSMT can not be achieved with constant factor overhead tolerating $\mathcal{A}_{t_b}$. This shows the power of allowing a negligible error probability (only in the reliability) in multiphase SMT.

In the sequel, we design a four-phase *communication optimal* USMT protocol, whose total communication complexity matches the bound proved in Theorem 16, thus showing that the bound is *asymptotically tight*. Also our four-phase *communication optimal* USMT protocol has a special property that it can achieve security with constant factor overhead tolerating $\mathcal{A}_{t_b}$.

6.3 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 $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$. The protocol terminates in four-phases and uses the three *phase USMT_Three_Phase* protocol (described in Theorem 15) 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 16 is *asymptotically 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 it securely sends ℓ field elements by communicating $O(\ell)$ field elements if the fault is only of Byzantine type (i.e., $t_o = t_f = t_p = 0$). Thus, it achieves security with 'constant factor overhead' (note that as pointed out in Comparison 6 USMT tolerating $\mathcal{A}_{t_b}$ is possible with communication complexity satisfying constant factor overhead).

Remark 10: Since $n = t_b + \max(t_b, t_p) + t_o + t_f + 1$, we can use *USMT_Three_Phase* 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 (i.e.,

$2t_b + 2t_o + t_f + t_p + 1$) is more than the connectivity requirement for multiphase USMT (i.e., $t_b + \max(t_b, t_p) + t_o + t_f + 1$).

Theorem 17: 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 securely transmits the message 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 $\geq \left(1 - \frac{\delta}{4}\right)$

2 for all $1 \leq i \leq n$, $y_i' = y_i$ with probability $\geq \left(1 - \frac{\delta}{4}\right)$

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 $\geq \left(1 - \frac{\delta}{4}\right)$

4 the protocol *URMT_Single_Phase* successfully sends the vector d with probability $\geq \left(1 - \frac{\delta}{4}\right)$.

The error probability of the protocol depends upon the error probability of the above four events. If each of the above are true, then our 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 $\geq \left(1 - \frac{\delta}{4}\right)$.

Proof: In *USMT_Mixed*, $1 \leq i \leq n$, ρ_i's are sent using n parallel execution of the three phase protocol *USMT_Three_Phase*. From the proof of Theorem 15, the error probability of a single execution of *USMT_Three_Phase* protocol is at most $\frac{1}{|\mathbb{F}|}$. Hence, the total error probability of n parallel executions of *USMT_Three_Phase* 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 *USMT_Three_Phase* 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 $\geq \left(1 - \frac{\delta}{4}\right)$.

Proof: Similar to the proof of the previous claim (i.e., Claim 4). □

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, $\rho_i' = \rho_i$ and $y_i' = y_i$ then $w_i \in L_{fault}$ with probability $\geq \left(1 - \frac{\delta}{4}\right)$.

Proof: From the security argument of *USMT_Three_Phase* 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 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 $n^2 |\mathbb{F}|$, the protocol fails to detect the faulty wire. In order to bound this error probability by $\frac{\delta}{4}$, 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 executed in parallel $\frac{n(t_b - t_p)}{t_b}$ 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 5, if $|\mathbb{F}| = \frac{n^3}{\delta'}$ then the error probability of *URMT_Single_Phase* is at most δ'. So in order to bound the error probability of *URMT_Single_Phase* by $\delta' \leq \frac{\delta t_b}{4n(t_b - t_p)}$, we require $|\mathbb{F}| \geq \frac{4n^4(t_b - t_p)}{\delta t_b}$. which is true. Hence, the claim follows. □

Thus, Theorem 17 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 17 is at most $\frac{\delta}{3}$. This is because 4th event does not occur, as d is broadcasted in this case during Phase IV, instead of sending it using single phase URMT. It is easy to check that by setting $|\mathbb{F}| \geq \frac{3n^2}{\delta}$, the theorem holds for $t_b \leq t_p$. □

Table 10 A four-phase communication optimal USMT protocol

Protocol USMT_Mixed

A communication optimal 4-phase USMT protocol tolerating $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$

The message m is a sequence of n^2 field elements if $t_b \le t_p$, otherwise the message is a sequence of $(t_b - t_p)n^2$ field elements.

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

- **R** selects at random n^3 elements, r_{ij}, $1 \le i \le n$, $1 \le j \le n^2$ from field $\mathbb{F}$. **R** also randomly selects $\rho_1, \rho_2, ..., \rho_n$ from $\mathbb{F}$.
- **R** computes $y_i = \sum_{j=1}^{n^2} \rho_i^j r_{ij}$, $1 \le i \le n$. Note that ρ_i^j is jth power of ρ_i.
- **R** sends to **S** over w_i, $1 \le i \le n$, the n^2 field elements r_{ij}, $1 \le j \le n^2$. **R** also sends ρ_i, y_i, $1 \le i \le n$ to **S** using $2n$ parallel invocations of the three phase *USMT_Three_Phase* protocol (described in Theorem 15) as there are total 2n elements to send. Hence, Phase I, II and Phase III are used to run $2n$ parallel executions of *USMT_Three_Phase* protocol.

*Phase IV (***S** *to* **R***)*

- Let **S** receive r'_{ij}, $1 \le j \le n^2$ along wire w_i. **S** adds w_i to a list $L_{erasure}$, if **S** does not receive any information over w_i.
- Let **S** receive ρ'_i and y'_i, $1 \le i \le n$ after the $2n$ parallel executions of the three phase *USMT_Three_Phase* protocol initiated by **R**.

  For each i, such that $w_i \notin L_{erasure}$, **S** verifies whether $y'_i \stackrel{?}{=} \sum_{j=1}^{n^2} \rho_i'^j r'_{ij}$. If $y'_i \ne \sum_{j=1}^{n^2} \rho_i'^j r'_{ij}$, then **S** adds wire w_i to the set of faulty wires, denoted by L_{faulty}. **S** sets $L_{honest} = \mathcal{W} \setminus (L_{faulty} \cup L_{erasure})$. If $t_p \ge t_b$, then **S** computes a random pad $Z = (z_1, z_2, ..., z_{n^2})$ of size n^2 field elements from the $n^2 \mid L_{honest} \mid$ field elements which are received over the wires in L_{honest} as follows:

$$Z = EXTRAND_{n^2|L_{honest}|,n^2}\left(r'_{ij} \mid w_i \in L_{honest}, 1 \le j \le n^2\right)$$

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

$$Z = EXTRAND_{n^2|L_{honest}|,(t_b-t_p)n^2}\left(r'_{ij} \mid w_i \in L_{honest}, 1 \le j \le n^2\right)$$

- **S** computes $d = m \oplus Z$. If $t_p \ge t_b$ then d is of size n^2, so **S** broadcasts d to **R**. On the other hand, if $t_b > t_p$ then d consists of $(t_b - t_p)n^2$ field elements. In this case, **S** reliably sends d to **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 **S** by invoking the single phase URMT protocol $\frac{(t_b - t_p)}{t_b} * n$ times). **S** also broadcasts the set L_{faulty} and $L_{erasure}$ to **R**.

Message recovery by **R**.

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

$$Z^{\mathbf{R}} = EXTRAND_{n^2|L_{honest}|,n^2}\left(r'_{ij} \mid w_i \in L_{honest}, 1 \le j \le n^2\right).$$

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

$$Z^{\mathbf{R}} = EXTRAND_{n^2|L_{honest}|,(t_b-t_p)n^2}\left(r'_{ij} \mid w_i \in L_{honest}, 1 \le j \le n^2\right).$$

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

Remark 11: From Theorem 17, the field size should be either $\frac{3n^2}{\delta}$ (when $t_b \le t_p$) or $\frac{4n^4(t_b - t_p)}{\delta t_b}$ (when $t_b > t_p$). However, in *USMT_Mixed*, during Phase I, **R** needs to select $n^3 + n$ random field elements from $\mathbb{F}$. So, we will set the field size as $\max\left(n^3 + n, \frac{3n^2}{\delta}\right)$ when $t_b \le t_p$ and $\frac{4n^4(t_b - t_p)}{\delta t_b}$ when $t_b > t_p$.

Theorem 18: In *USMT_Mixed*, the adversary learns no information about the message m in information theoretic sense.

Proof: First note that all the n ρ_i's and y_i's are information theoretically secure from the security of *USMT_Three_Phase* protocol. 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. Now, the proof follows from the correctness of EXTRAND 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. Now, the proof follows from the correctness of EXTRAND and working of the protocol. □

Theorem 19: 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 *USMT_Three_Phase* protocol, each having a communication complexity of $O(n^2)$ field elements (see Theorem 15). This incurs total communication cost 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 6), 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 20: 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 $n = t_b + \max(t_b, t_p) + t_o + t_f + 1$. From Theorem 16, if $t_b \geq t_p$ (in this case $n = 2t_b + t_o + t_f + 1$), 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, $n = t_b + t_p + t_o + t_f + 1$), 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 in the presence of Byzantine adversary, $\mathcal{A}_{t_b}$ (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 complexity 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 tolerating $\mathcal{A}_{t_b}$, 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 tolerating $\mathcal{A}_{t_b}$. We design one such protocol in the next section.

Remark 12 (note on the message size used in protocol USMT_Mixed): In protocol *USMT_Mixed*, we have considered $n = t_b + \max(t_b, t_p) + t_o + t_f + 1$, the minimum connectivity required for any multiphase USMT protocol. If $t_p \geq t_b$, then this implies that $n = t_b + t_p + t_o + t_f + 1$ and so there will at least one honest wire, which will not be under the control of the adversary. So the n^2 random values sent over the honest wire will be unknown to the adversary and so it can be used as an information theoretic secure pad to blind a message of size n^2. However, if $n > t_b + \max(t_b, t_p) + t_o + t_f + 1$, then there will be more honest wires and hence, we can establish a pad of size larger than n^2 to send a larger size message. For example, consider the following settings: $t_b = t_p - 1$, $t_o = t_f = 0$ and $n = 2(t_b + t_p)$. It is easy to see that in these settings, multiphase USMT is possible. Moreover, there will be at least $(t_b + t_p)$ honest wires. So the n^2 values sent over these wires will be unknown to the adversary. So if run protocol *USMT_Mixed* over such a setting then we establish an information theoretic secure pad of size $(t_b + t_p)n^2 = \Theta(n^3)$, instead of n^2. As a result, we can send a message of size $\Theta(n^3)$ by communicating $O(n^3)$, which from Theorem 16 satisfies the lower bound and hence will be communication optimal. Thus, our protocol will be communication optimal for all connectivity. If the number of wires is more than $n = t_b + \max(t_b, t_p) + t_o + t_f + 1$, then we have to accordingly increase the message size and run the protocol.

6.4 Two-phase USMT with constant factor overhead tolerating $\mathcal{A}_{t_b}$

The connectivity requirement for any multiphase USMT tolerating only Byzantine adversary $\mathcal{A}_{t_b}$ is $n \geq 2t_b + 1$ (by substituting $t_o = t_f = t_p = 0$ in Theorem 15). 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 tolerating $\mathcal{A}_{t_b}$. Thus, we get security with 'constant factor' overhead in two_phases. 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 4.3, which reliably sends $n(t_b + 1) = \Theta(n^2)$ field elements by communicating $O(n^2)$ field elements, against $\mathcal{A}_{t_b}$, where **S** and **R** are connected by $n = 2t_b + 1$ wires (by substituting $t_o = t_f = t_p = 0$ in protocol *URMT_Single_Phase*).

2 Protocol *USMT_Single_Phase*: Described in the section 5.2, 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*).

We now prove the correctness of protocol *USMT_Byzantine*.

Theorem 21: In protocol USMT Byzantine if $|\mathbb{F}| \frac{16n^3}{\delta}$ 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, tolerating $\mathcal{A}_{t_b}$.

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 $\geq \left(1 - \frac{\delta}{4}\right)$

2 for all $1 \leq \mathrm{i} \leq \mathrm{n}$, $y_i' = y_i$ with probability $\geq \left(1 - \frac{\delta}{4}\right)$

3 if the wire w_i were indeed corrupt, then $w_i \in L_{faulty}$ with probability $\geq \left(1 - \frac{\delta}{4}\right)$

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 that 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 $\geq \left(1 - \frac{\delta}{4}\right)$.

Proof: From Theorem 12, 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 execute *USMT_Single_Phase* in parallel 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 $\geq \left(1 - \frac{\delta}{4}\right)$.

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, $\rho_i' = \rho_i$ and $y_i' = y_i$ then $w_i \in L_{faulty}$ with probability $\geq \left(1 - \frac{\delta}{4}\right)$.

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 $\frac{16n^3}{\delta} > \frac{4n}{\delta}$, the claim holds. □

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

Table 11 A two-phase USMT protocol tolerating only Byzantine corruption

Protocol USMT_Byzantine: a two-phase USMT protocol tolerating $\mathcal{A}_{t_b}$

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

- **R** selects at random n^2 random elements, say r_{ij}, $1 \le i, j \le n$, 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 jth 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 $1 \le i \le n$ 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}, $1 \le j \le n$ along the wire w_i, $1 \le i \le n$. Also let **S** receive ρ'_i and y'_i, $1 \le i \le n$ after the parallel execution of single phase USMT protocol *USMT_Single_Phase* initiated by **R**.
- For each i, **S** verifies whether $y'_i \stackrel{?}{=} \sum_{j=1}^{n^2} \rho_i'^j r'_{ij}$. 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:

$$Z = EXTRAND_{n|L_{honest}|, n(t_b+1)}\left(r'_{ij} \,|\, w_i \in L_{honest}, 1 \le j \le n\right)$$

- **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:

$$Z^{\mathbf{R}} = EXTRAND_{n|L_{honest}|, n(t_b+1)}\left(r_{ij} \,|\, w_i \in L_{honest}, 1 \le j \le n\right)$$

- R recovers the message by computing $m = Z^{\mathbf{R}} \oplus d$.

Proof: As mentioned earlier, *URMT_Single_Phase* fails with probability δ, if $|\mathbb{F}| \ge \frac{n^3}{\delta}$ (see Theorem5). So in order that *URMT_Single_Phase* fails with probability of at most $\frac{\delta}{4}$, we require $|\mathbb{F}| \ge \frac{4n^3}{\delta}$. Since $|\mathbb{F}| \ge \frac{16n^3}{\delta}$, which in turn is greater that $\frac{4n^3}{\delta}$, the claim is true. □

Theorem 22: 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 each of these $t_b + 1$ wires. Now, the proof follows from the correctness of EXTRAND algorithm. □

Theorem 23: The communication complexity of *USMT_Byzantine* is $O(n^2)$ field elements.

Proof: During Phase I, **R** sends $n2$ 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 6 requires communicating $O(n^2)$ field elements. Thus, the total communication complexity of the protocol is $O(n^2)$ field elements. □

Theorem 24: Protocol *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. Hence, it is a communication optimal protocol. Moreover, it is phase optimal because from Theorem 10, by substituting $t_o = t_f = t_p = 0$, we find that any single phase USMT requires a communication complexity 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. □

6.5 Comparison of multiphase PSMT with multiphase USMT

1 Allowing a negligible error probability only in the reliability, *significantly* helps in the possibility of multiphase SMT protocols (see Comparison 5).

2 Allowing a negligible error probability only in the reliability, *significantly* helps in reducing the lower bound on communication complexity of multiphase SMT protocols (see Comparison 6).

3 It is impossible to design any PSMT protocol, irrespective of the number of phases, which achieves security with constant factor overhead; i.e., securely sending ℓ field elements by communicating $O(\ell)$ field elements tolerating $\mathcal{A}_{t_b}$ (see Table 2, second row) in a $(2t_b + 1)$-(**S**, **R**) connected network. 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 SMT protocol, which achieves security with constant factor overhead.

7 Non-threshold adversary settings

Till last section, we have considered threshold adversary settings, where the corruption done by the adversary is bounded by a threshold. We now consider more general adversary settings, namely non-threshold adversary settings. Informally, a non-threshold adversary is represented by a collection of 4-tuples of the form (B, O, F, E), where B, O, F and E denotes the set of nodes which can be potentially corrupted in Byzantine, omission, fail-stop and passive fashion respectively. During the protocol execution, the adversary can choose any such 4-tuple from the collection for corruption.

Over the past few decades, non-threshold adversary has been considered in the context of many distributed computing protocols such as MPC (Hirt and Maurer, 2000; Cramer et al., 200b; Beerliová-Trubíniová et al., 2008; Hirt et al., 2008), VSS (Gennaro, 1996; Cramer et al., 200a), Byzantine agreement (Fitzi and Maurer, 1998; Altmann et al., 1999). Non-threshold adversary in the context of PRMT and PSMT was first studied in Kumar et al. (2002), where the authors have considered undirected networks and only Byzantine corruption. In Patra et al. (2007), the authors have given the necessary and sufficient condition for the existence of PSMT in directed networks tolerating non-threshold Byzantine adversary. In Srinathan and Pandu Rangan (2006), and Srinathan et al. (2008b), the authors have given the necessary and sufficient condition for the existence of URMT in an *arbitrary directed graph* tolerating a non-threshold mixed adversary. Recently, in Srinathan et al. (2009), the authors have given the complete characterisation of USMT in *arbitrary directed networks* tolerating a non-threshold mixed adversary.

Modelling the adversary by a threshold helps in easy characterisation of PSMT. It also helps in analysing protocols and proving lower bound on the communication complexity (Srinathan et al., 2004). However, as mentioned in Kumar et al. (2002), modelling the (dis)trust in the network as a threshold adversary does not capture all possible scenarios. Moreover, the threshold model may lead to a gross overestimation of the connectivity requirement of the underlying network [see Kumar et al. (2002), for an example]. The necessary and sufficient condition for URMT in *undirected networks* tolerating non-threshold mixed adversary can be derived from the characterisation of URMT in *arbitrary directed networks* tolerating non-threshold mixed adversary, as given (Srinathan and Pandu Rangan, 2006; Patra et al., 2007) because undirected networks are a special case of arbitrary directed networks. However, the characterisation of URMT for arbitrary directed networks, as given in Srinathan and Pandu Rangan (2006), and Patra et al. (2007) is indirect and highly non-intuitive. Moreover, it is likely to take exponential time to verify whether a given directed network and a non-threshold adversary satisfies the conditions given in Srinathan and Pandu Rangan (2006), and Patra et al. (2007) for the possibility of URMT. So it is desirable to have a direct and simple characterisation of URMT in undirected networks tolerating non-threshold adversary. Similarly, the characterisation for USMT in *arbitrary directed network* tolerating non-threshold adversary given in Srinathan et al. (2009) is indirect and highly non-intuitive. Furthermore, it is likely to take exponential time to verify whether a given directed network and a non-threshold adversary satisfy the conditions given in Srinathan et al. (2009) for the possibility of USMT. So instead of deriving the characterisation for USMT in *undirected networks* from the characterisation of USMT in *arbitrary directed networks*, it is desirable to have a simple and direct characterisation of USMT in undirected networks tolerating non-threshold adversary. So, we now proceed to give a direct and simple characterisation of URMT and USMT in undirected networks tolerating non-threshold mixed adversary. Before that, we present few definitions.

7.1 Model and definitions

A *non-threshold* adversary is represented by an adversary structure which is an enumeration of all the possible snapshots of faults in the network. A single snapshot can be described by an ordered quadruple (B, O, F, E), where B, O, F, $E \subseteq P$, which means that the nodes in the set B, O, F and E can be corrupted in Byzantine, omission, fail-stop and passive fashion respectively. Thus, an adversary structure is a collection of such quadruples. The adversary structure is monotone in the sense that if $(B_1, O_1, F_1, E_1) \in \mathbb{A}$, then $\forall(B_2, O_2, F_2, E_2)$ such that $B_2 \subseteq B_1$, $O_2 \subseteq O_1$, $F_2 \subseteq F_1$ and $F_2 \subseteq F_1$, we have $(B_2, O_2, F_2, E_2) \in \mathbb{A}$. Throughout the

execution of a protocol, the adversary can corrupt nodes from any *one* element (quadruple) of $\mathbb{A}$ in Byzantine, omission, fail-stop and passive fashion respectively. Moreover, **S** and **R** have no information about the quadruple before the beginning of the protocol. It is easy to see that a threshold adversary $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$ is a special type of **A**, where each (B, O, F, E) in A has the following form: $|B| \leq t_b$, $|O| \leq t_o$, $|F| \leq t_f$ and $|E| \leq t_p$. We note that **A** can be uniquely represented by listing the elements in its maximal basis $\overline{\mathbb{A}}$ which we define below.

Definition 19 (maximal basis of **A***):* For any monotone adversary structure **A**, its maximal basis $\overline{\mathbb{A}}$ is defined as $\overline{\mathbb{A}} = \{(B,O,F,E) \mid (B,O,F,E)\} \in \mathbb{A}$, and $\nexists (W, X, Y, Z) \in \mathbf{A}$ such that $(W, X, Y, Z) \neq (B, O, F, E)$ where $W \supseteq B, X \supseteq O, Y \supseteq F$ and $Z \supseteq E\}$.

7.2 URMT in undirected networks tolerating non-threshold adversary

We now characterise URMT in an undirected graph $\mathcal{N}$ tolerating an arbitrary non-threshold adversary **A**. Unlike $\mathcal{A}_{(t_b,t_o,t_f,t_p)}$, working out a direct characterisation of URMT tolerating entire **A** is highly complex and non-intuitive. Rather it is easy to think of a characterisation tolerating small sized subsets from **A**. We now state the following important lemma:

Theorem 25: URMT in an undirected network $\mathcal{N}$ tolerating a non-threshold adversary **A** is possible iff URMT is possible in $\mathcal{N}$ tolerating any $\mathcal{A} \subseteq \mathbf{A}$ with maximal basis $\overline{\mathcal{A}}$ of size two.

Proof: The only-if direction is obvious. For the if-direction, we now show that if an URMT protocol exists while tolerating every monotone subset $\mathcal{A} \subseteq \mathbf{A}$ such that $|\overline{\mathcal{A}}| = 2$, then one can construct an URMT protocol that tolerates **A**. We prove this by induction. Suppose that every monotone subset $\mathcal{A}$ of **A**, such that $|\overline{\mathcal{A}}| = 2$, is tolerable. Then, to show that every monotone subset $\mathcal{A}$ of **A**, such that $|\overline{\mathcal{A}}| = 3$ is also tolerable, we argue as follows: for any subset $\mathcal{A} \subseteq \mathbf{A}$ with $|\overline{\mathcal{A}}| = 3$, there exist three subsets, each of size two, such that any element in $\overline{\mathcal{A}}$ belongs to exactly two of them. Specifically, we may choose to divide $\overline{\mathcal{A}} = \{x_1, x_2, x_3\}$ (where each x_i is an ordered quadruple (B_i, O_i, F_i, E_i)) into $\mathcal{A}_1 = \{x_1, x_2\}$, $\mathcal{A}_2 = \{x_2, x_3\}$ and $\mathcal{A}_3 = \{x_1, x_3\}$. Now by our assumption, we have URMT protocols, say Π_1, Π_2 and Π_3 to tolerate $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ respectively. We now show how to design URMT protocol Π to send a message m, tolerating $\overline{\mathcal{A}}$. From Theorem 3, by substituting $t_b = 1$ and $t_o = t_f = t_p = 0$, we find that URMT is achievable over three wires, out of which one could be Byzantine corrupted. Let *URMT_Single* be such a single phase URMT protocol which runs over three wires w_1, w_2 and w_3, of which one could be Byzantine corrupted. Moreover, let *URMT_Single* transmits α_i over w_i for $1 \leq i \leq 3$, to send message m. We now run the sub-protocols Π_1, Π_2 and Π_3 in parallel for transmitting α_1, α_2 and α_3 respectively. Since every element of $\overline{\mathcal{A}}$ belongs to at least two of the three $\mathcal{A}_i$'s, **R** gets the correct information in at least two of the three sub-protocols with very high probability. **R** can now output m performing the same computation, as done in *URMT_Single* tolerating 1-active Byzantine adversary. The correctness of this URMT protocol tolerating $\overline{\mathcal{A}}$ follows from the correctness of the single phase URMT tolerating 1-active adaptive Byzantine adversary. Therefore, we can conclude that URMT is possible tolerating any subset $\mathcal{A}$ of **A**, such that $|\overline{\mathcal{A}}| = 3$.

Applying the same procedure, we find that if URMT is possible tolerating any subset $\mathcal{A}$ of **A**, such that $|\overline{\mathcal{A}}| = 3$ then it is also possible to design an URMT protocol tolerating any subset $\mathcal{A}$ of **A**, such that $|\overline{\mathcal{A}}| = 4$. This is because any $\overline{\mathcal{A}} = \{x_1, x_2, x_3, x_4\}$ (where each x_i is an ordered quadruple (B_i, O_i, F_i, E_i)) can be divided into three subsets, each of size three, such that every element in $\overline{\mathcal{A}}$ occurs in at least two of the subsets. More formally, we can divide $\overline{\mathcal{A}}$ into $\mathcal{A}_1 = \{x_1, x_2, x_3\}$, $\mathcal{A}_2 = \{x_2, x_3, x_4\}$ and $\mathcal{A}_3 = \{x_1, x_3, x_4\}$. Now as in the previous case, we can run three URMT protocols (as shown above, these protocols exists) in parallel, transmitting α_1, α_2 and α_3 tolerating the adversary structures $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ respectively. Since every element of $\mathcal{A}$ belongs to at least two of the three $\mathcal{A}_i$'s, **R** gets the correct information in at least two of the three sub-protocols and hence recovers the message by performing same computation as in single phase URMT tolerating 1-active adaptive Byzantine adversary.

In general, any $\mathcal{A} \subseteq \mathbf{A}$ whose maximal basis $|\overline{\mathcal{A}}|$ is of size $\mu > 3$, can be divided into three subsets each of size $\left\lceil \frac{2\mu}{3} \right\rceil$, such that every element of $\overline{\mathcal{A}}$ occurs in at least two of the subsets. The rest now follows from induction. □

Remark 13: The protocol given as a part of sufficiency proof in Theorem 25 is an inductive protocol and is exponential in the size of **A**. We leave the issue of designing efficient URMT protocol tolerating **A** as an open problem.

Theorem 25 shows that in order to get a complete characterisation of URMT tolerating the entire adversary structure **A**, it is enough if we characterise URMT tolerating every $\mathcal{A} \subseteq \mathbf{A}$ with maximal basis $\overline{\mathcal{A}}$ of size two. We do the same in next theorem.

Theorem 26: URMT between **S** and **R** in an undirected graph $\mathcal{N} = (V, E)$ tolerating a non-threshold adversary with

maximal basis $\bar{\mathcal{A}} = \{(B_1, O_1, F_1, E_1), (B_2, O_2, F_2, E_2)\}$ is possible iff both the following conditions are satisfied:

1 for each $i \in \{1, 2\}$, there exists a path from **S** to **R** in the network induced by $\mathcal{N}$ on the vertices $(V \setminus (B_i \cup O_i \cup F_i))$

2 there exists a path from **S** to **R** in the network induced by $\mathcal{N}$ on $(V \setminus (B_1 \cup B_2 \cup ((O_1 \cup F_1) \cap (O_2 \cup F_2)))$.

Proof:

Necessity: The necessity of the first condition is obvious, since otherwise the adversary can simply block the nodes in $(B_i \cup O_i \cup F_i)$, causing the receiver to be isolated from the sender and thus preventing any communication from **S** to **R**. Suppose that the second condition is not necessary. Since the nodes in $((O_1 \cup F_1) \cap (O_2 \cup F_2))$ can be deemed as non-existent (since they are 'guaranteed' to be corrupt in the worst-case), we note that a URMT protocol over a network that does not satisfy the second condition can be used to design an URMT protocol in an undirected network where **S** and **R** are connected by two wires, any one of which is potentially Byzantine corruptible. But from Theorem 3 such an URMT protocol is impossible, thus showing a contradiction. We now proceed to prove the sufficiency condition.

Sufficiency: Suppose the conditions of theorem are satisfied. Then, there exists three paths (not necessarily distinct) p_a, p_b and p_c from **S** to **R**, such that pa avoids nodes from $(B_1 \cup O_1 \cup F_1)$, p_b avoids nodes from $(B_2 \cup O_2 \cup F_2)$, while pc avoids nodes from $(B_1 \cup B_2 \cup ((O_1 \cup F_1) \cap (O_2 \cup F_2)))$. We now design an URMT protocol. To transmit a message m, **S** sends m along the paths p_a, p_b and p_c. Each intermediate node u along these paths forwards the message that it received to the corresponding neighbour. If nothing is received by the time something should have been received (since the network is synchronous, strict time-out conditions are feasible) then it forwards a new message namely 'Null-from-u' to its neighbour. **R** recovers m as follows: If **R** receives a valid message x along the path p_c, then $x = \mathbf{m}$, since the path pc cannot be Byzantine corrupt. If a 'Null-from-u' message is received along p_c, then if u's previous node in path p_c belongs to $(O_1 \cup F_1)$, i.e., $predecessor(u) \in (O_1 \cup F_1)$ then **R** outputs the message that is (guaranteed to be) received along path p_b. Else if $predecessor(u) \in (O_2 \cup F_2)$ then **R** outputs the message that is (guaranteed to be) received along path p_a. However, if nothing is received along the path p_c, then if **R**'s previous node in path p_c belongs to $(O_1 \cup F_1)$, i.e., $predecessor(\mathbf{R}) \in (O_1 \cup F_1)$ then **R** outputs the message that is (guaranteed to be) received along path p_b. Else if $predecessor(\mathbf{R}) \in (O_2 \cup F_2)$ then **R** outputs the message that is (guaranteed to be) received along path p_a. It is easy to see that **R** will correctly output m at the end of the protocol. This completes the proof of Theorem 26. □

7.3 USMT in undirected networks tolerating non-threshold adversary

We now give the necessary and sufficient condition for USMT in undirected networks tolerating a non-threshold adversary structure. As in the case of URMT, we first show that USMT tolerating the entire adversary structure is possible iff USMT is possible tolerating every subset of the adversary structure with maximal basis of size two.

Theorem 27: USMT in a digraph $\mathcal{N}$ tolerating a non-threshold adversary **A** is possible iff USMT is possible in $\mathcal{N}$ tolerating any $\mathcal{A} \subseteq \mathbf{A}$ with maximal basis $\bar{\mathcal{A}}$ of size two.

Proof: The proof is similar to the proof of Theorem 25. The only-if direction is obvious. For the if-direction, we now show that if an USMT protocol exists while tolerating every monotone subset $\mathcal{A} \subseteq \mathbf{A}$ such that $|\bar{\mathcal{A}}| = 2$, then one can construct an USMT protocol that tolerates A. Suppose that every monotone subset $\mathcal{A}$ of **A**, such that $|\bar{\mathcal{A}}| = 2$, is tolerable. Then, to show that every monotone subset $\mathcal{A}$ of **A**, such that $|\bar{\mathcal{A}}| = 3$ is also tolerable, we argue as follows: for any subset $\mathcal{A} \subseteq \mathbf{A}$ with $|\bar{\mathcal{A}}| = 3$, there exist three subsets, each of size two, such that any element in $\bar{\mathcal{A}}$ belongs to exactly two of them. Specifically, we may choose to divide $\bar{\mathcal{A}} = \{x_1, x_2, x_3\}$ (where each xi is an ordered quadruple (B_i, O_i, F_i, E_i)) into $\mathcal{A}_1 = \{x_1, x_2\}$, $\mathcal{A}_2 = \{x_2, x_3\}$ and $\mathcal{A}_3 = \{x_1, x_3\}$. Now by our assumption, we have USMT protocols, say Π_1, Π_2 and Π_3 to tolerate $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ respectively. We now show how to design USMT protocol Π to send a message m, tolerating $\bar{\mathcal{A}}$.

From Theorem 10, USMT is achievable over three wires, out of which one could be Byzantine corrupted. Let *USMT_Single* be such a single phase USMT protocol which runs over three wires w_1, w_2 and w_3, of which one could be Byzantine corrupted. Moreover, let *USMT_Single* transmits α_i over ch_i for $1 \le i \le 3$, to send message m. We now run the sub-protocols Π_1, Π_2 and Π_3 in parallel for transmitting α_1, α_2 and α_3 respectively. Since every element of $\bar{\mathcal{A}}$ belongs to at least two of the three $\mathcal{A}_i$'s, **R** gets the correct information in at least two of the three sub-protocols with very high probability. **R** can now output m performing the same computation, as done in *USMT_Single* tolerating 1-active Byzantine adversary. The correctness and secrecy of this USMT protocol tolerating $\bar{\mathcal{A}}$ follows from the correctness and secrecy of the single phase USMT tolerating 1-active adaptive Byzantine adversary. Therefore we can conclude that USMT is possible tolerating any subset $\mathcal{A}$ of **A**, such that $|\bar{\mathcal{A}}| = 3$.

Applying the same procedure, we find that if USMT is possible tolerating any subset $\mathcal{A}$ of **A**, such that $|\bar{\mathcal{A}}|=3$ then it is also possible to design an USMT protocol tolerating any subset $\mathcal{A}$ of **A**, such that $|\bar{\mathcal{A}}|=4.$. This is because any $\bar{\mathcal{A}}=\{x_1,x_2,x_3,x_4\}$ (where each x_i is an ordered quadruple (B_i, O_i, F_i, E_i)) can be divided into three subsets, each of size three, such that every element in $\bar{\mathcal{A}}$ occurs in at least two of the subsets. More formally, we can divide $\bar{\mathcal{A}}$ into $\mathcal{A}_1 = \{x_1, x_2, x_3\}$, $\mathcal{A}_2 = \{x_2, x_3, x_4\}$ and $\mathcal{A}_3 = \{x_1, x_3, x_4\}$. Now as in the previous case, we can run three USMT protocols (as shown above, these protocols exists) in parallel, transmitting α_1, α_2 and α_3 tolerating the adversary structures $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ respectively. Since every element of $\mathcal{A}$ belongs to at least two of the three $\mathcal{A}_i$'s, **R** gets the correct information in at least two of the three sub-protocols and hence recovers the message by performing same computation as in single phase USMT tolerating 1-active adaptive Byzantine adversary.

In general, any $\mathcal{A} \subseteq$ **A** whose maximal basis $|\bar{\mathcal{A}}|$ is of size $\mu > 3$, can be divided into three subsets each of size $\left\lceil \frac{2\mu}{3} \right\rceil$, such that every element of $\bar{\mathcal{A}}$ occurs in at least two of the subsets. The rest now follows from induction. □

The above theorem shows that in order to get a complete characterisation of USMT tolerating the entire adversary structure **A**, it is enough if we characterise USMT tolerating every $\mathcal{A} \subseteq$ **A** with maximal basis $\bar{\mathcal{A}}$ of size two. We do the same in next theorem.

Theorem 28: USMT between **S** and **R** in an undirected network $\mathcal{N}$ = (V, E) tolerating a non-threshold adaptive adversary with maximal basis $\bar{\mathcal{A}}=\{(B_1,O_1,E_1,F_1,H_1),(B_2,O_2,E_2,F_2,H_2)\}$ is possible iff the network $\mathcal{N}$ is such that URMT between **S** and **R** is possible tolerating $\bar{\mathcal{A}}$ and for each $i \in \{1, 2\}$, the removal of the nodes from ($B_i \cup O_i \cup E_i \cup F_i$) does not disconnect **S** and **R**).

Proof:

Necessity: The necessity of URMT is obvious. Also, if there exists an $i \in \{1, 2\}$ such that ($B_i \cup O_i \cup E_i \cup F_i$) disconnects **S** and **R**s, then the adversary can ensure that he reads all the data that **R** receives from **S** (by blocking nodes in F_i and passively corrupting the rest of the cut set). Thus, any secure communication from **S** to **R** will be impossible. This completes the necessity proof. We now proceed to prove the sufficiency condition.

Sufficiency: Suppose the conditions of the theorem are true. This implies that there are two, not necessarily distinct paths, from **S** to **R**, say p_1 and p_2, such that:

Path	*Remarks*
p_1	The path p_1 does not contain nodes from ($B_1 \cup O_1 \cup E_1 \cup F_1$).
p_2	The path p_2 does not contain nodes from ($B_2 \cup O_2 \cup E_2 \cup F_2$).

Now consider the following USMT protocol: **S** chooses six random keys K_{11}, K_{12}, K_{13}, K_{21}, K_{22} and K_{23} and sends K_{i1}, K_{i2} and K_{i3} along the path p_i, for $1 \le i \le 2$. Now, either **R** receives all the six keys (three of which could be corrupted) or he knows whether the first set or the second set in the adversary structure is corrupt. In the latter case, **R** sends to **S** using URMT protocol[6] the identity $\alpha \in \{1, 2\}$ of the corrupted set; once α is agreed upon, **S** forwards the message along the path p_α (which is honest if α is received correctly). In the former case, **R** sends using URMT to **S** the values $\rho_1 = K_{11}K_{22} + K_{23}$ and $\rho_2 = K_{21}K_{12} + K_{13}$. Next **S** verifies if the values ρ_i are correct or not. With high probability, **S** can detect corruption (if any) and inform **R** (using URMT protocol) the identity $\alpha \in \{0, 1, 2\}$ of the corrupt wire (here $\alpha = 0$ represents no corruption detected). Furthermore, if $\alpha \neq 0$, S sends to **R** the message m through the path p_α or else if $\alpha = 0$, **S** sends $m \oplus K_{13} \oplus K_{23}$ to **R** via URMT protocol. Finally, **R** recovers the message. The correctness and secrecy of the protocol is obvious. □

8 Conclusions and open problems

We have studied the problem of URMT and USMT in the presence of mixed adversary. Existing URMT and USMT protocols deals with only 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, in both threshold and non-threshold settings. We have given the complete characterisation of single phase and multiphase URMT protocols in undirected networks tolerating threshold mixed adversary. We have proved the lower bound on the communication complexity of any single phase and multiphase URMT protocol. Moreover, we have shown that our bounds are *asymptotically tight* by designing *communication optimal* protocols. Similarly, we have given complete characterisation 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 multiphase USMT protocol. Moreover, we have shown that our bounds are asymptotically *tight* by designing *communication optimal* protocols. Finally, we have given the complete characterisation of URMT and USMT protocol tolerating non-threshold adversary. The paper shows that allowing a negligible error probability has strong effect in the *possibility*, *feasibility* and *optimality* of reliable and SMT protocols.

Few questions remain unanswered in the paper which are as follows:

1 Our communication optimal URMT and USMT protocols against threshold adversary achieve communication optimality for sufficiently long messages. The next obvious and interesting problem is to design communication optimal protocols for messages of any length.

2 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.

3 We have only given the necessary and sufficient condition for the presence of URMT and USMT against non-threshold adversary. It is an interesting open problem to further improve the protocols in terms of communication complexity and phase complexity.

4 In the definition of USMT, we have assumed that there is no error is secrecy; i.e., secrecy is perfect. It would be interesting to explore the settings in which negligible error probability is allowed in secrecy as well. That is solving the issues of possibility, feasibility and optimality for (ϵ, δ)-secure protocols are an interesting direction. For partial results, the readers are referred to Franklin and Wright (2000) and Wang and Desmedt (2001).

Acknowledgements

Financial support from Microsoft Research India and Infosys Technology India is acknowledged. Work supported by Project No. CSE/05-06/DITX/CPAN on Protocols for Secure Communication and Computation, sponsored by Department of Information Technology, Government of India.

References

Agarwal, S., Cramer, R. and de Haan, R. (2006) 'Asymptotically optimal two-round perfectly secure message transmission', in Dwork, C. (Ed.): *Proc. of Advances in Cryptology: CRYPTO 2006*, LNCS 4117, pp.394–408, Springer-Verlag.

Altmann, B., Fitzi, M. and Maurer, U.M. (1999) 'Byzantine agreement secure against general adversaries in the dual failure model', in Jayanti, P. (Ed.): *Distributed Computing, 13th International Symposium, Proceedings, Lecture Notes in Computer Science*, 27–29 September, Vol. 1693, pp.123–137, Springer, Bratislava, Slavak Republic.

Araki, T. (2008) 'Almost secure 1-round message transmission scheme with polynomial-time message decryption', in Safavi-Naini, R. (Ed.): *Information Theoretic Security, Third International Conference, ICITS 2008, Proceedings*, *Lecture Notes in Computer Science*, 10–13 August, Vol. 5155, pp.2–13, Springer, Calgary, Canada.

Ashwinkumar, B.V., Patra, A., Choudhury, A., Srinathan, K. and Pandu Rangan, C. (2008) 'On tradeoff between network connectivity, phase complexity and communication complexity of reliable communication tolerating mixed adversary', in Bazzi, R.A. and Patt-Shamir, B. (Eds.): *Proceedings of the Twenty-Seventh Annual ACM Symposium on Principles of Distributed Computing, PODC 2008*, 18–21 August, pp.115–124, ACM, Toronto, Canada.

Beerliová-Trubíniová, Z. and Hirt, M. (2006) 'Efficient multiparty computation with dispute control', in Halevi, S. and Rabin, T. (Eds.): *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, Proceedings*, *Lecture Notes in Computer Science*, 4–7 March, Vol. 3876, pp.305–328, Springer, New York, NY, USA.

Beerliová-Trubíniová, Z. and Hirt, M. (2008) 'Perfectly-secure MPC with linear communication complexity', in Canetti, R. (Ed.): *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, Lecture Notes in Computer Science*, 19–21 March, Vol. 4948, pp.213–230, Springer, New York, USA.

Beerliová-Trubíniová, Z., Fitzi, M., Hirt, M., Maurer, U.M. and Zikas, V. (2008) 'MPC vs. SFE: perfect security in a unified corruption model', in Canetti, R. (Ed.): *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, Lecture Notes in Computer Science*, 19–21 March, Vol. 4948, pp.231–250, Springer, New York, USA.

Ben-Or, M., Goldwasser, S. and Wigderson, A. (1988) 'Completeness theorems for non-cryptographic fault-tolerant distributed computation', in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, 2–4 May, pp.1–10, ACM, Chicago, Illinois, USA.

Chaum, D., Crépeau, C. and Damgård, I. (1988) 'Multiparty unconditionally secure p(extended abstract)', in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, 2–4 May, pp.11–19, ACM, Chicago, Illinois, USA.

Choudhury, A., Patra, A., Ashwinkumar, B.V., Srinathan, K. and Pandu Rangan, C. (2008) 'Perfectly reliable and secure communication tolerating static and mobile mixed adversary', in Safavi-Naini, R. (Ed.): *Information Theoretic Security, Third International Conference, ICITS 2008, Proceedings, Lecture Notes in Computer Science*, 10–13 August, Vol. 5155, pp.137–155, Springer, Calgary, Canada.

Cramer, R., Damgård, I. and Dziembowski, S. (2000a) 'On the complexity of verifiable secret sharing and multiparty computation', in *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, 21–23 May, pp.325–334, ACM, Portland, OR, USA.

Cramer, R., Damgård, I. and Maurer, U.M. (2000b) 'General secure multi-party computation from any linear secret-sharing scheme', in Preneel, B. (Ed.): *Advances in Cryptology – EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Proceeding, Lecture Notes in Computer Science*, 14–18 May, Vol. 1807, pp.316–334, Springer, Bruges, Belgium.

Cramer, R., Damgård, I., Dziembowski, S., Hirt, M. and Rabin, T. (1999) 'Efficient multiparty computations secure against an adaptive adversary', in Stern, J. (Ed.): *Advances in Cryptology – EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Proceeding, Lecture Notes in Computer Science*, 2–6 May, Vol. 1592, pp.311–326, Springer, Prague, Czech Republic.

Damgård, I. and Nielsen, J.B. (2007) 'Scalable and unconditionally secure multiparty computation', in Menezes, A. (Ed.): *Advances in Cryptology – CRYPTO 2007, 27th Annual International Cryptology Conference, Proceedings, Lecture Notes in Computer Science*, 19–23 August, Vol. 4622, pp.572–590, Springer, Santa Barbara, CA, USA.

Desmedt, Y. and Wang, Y. (2003) 'Perfectly secure message transmission revisited', in Biham, E. (Ed.): *Advances in Cryptology – EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Lecture Notes in Computer Science*, 4–8 May, Vol. 2656, pp.502–517, Springer, Warsaw, Poland.

Dolev, D., Dwork, C., Waarts, O. and Yung, M. (1993) 'Perfectly secure message transmission', *JACM*, Vol. 40, No. 1, pp.17–47.

Feldman, P. and Micali, S. (1988) 'Optimal algorithms for Byzantine agreement', in *STOC*, pp.148–161, ACM.

Feldman, P. and Micali, S. (1989) 'An optimal probabilistic algorithm for synchronous Byzantine agreement', in Ausiello, G., Dezani-Ciancaglini, M. and Rocca, S.R.D. (eds.): *Automata, Languages and Programming, 16th International Colloquium, ICALP89, Proceedings, Lecture Notes in Computer Science*, 11–15 July, Vol. 372, pp.341–378, Springer, Stresa, Italy.

Fitzi, M. and Maurer, U.M. (1998) 'Efficient Byzantine agreement secure against general adversaries', in Kutten, S. (Ed.): *Distributed Computing, 12th International Symposium, DISC '98, Proceedings, Lecture Notes in Computer Science*, 24–26 September, Vol. 1499, pp.134–148, Springer, Andros, Greece.

Franklin, M. and Wright, R.N. (1998) 'Secure communication in minimal connectivity models', in Nyberg, K. (Ed.): *Advances in Cryptology – EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Proceeding, Lecture Notes in Computer Science*, 31 May–4 June, Vol. 1403, pp.346–360, Springer, Espoo, Finland.

Franklin, M. and Wright, R.N. (2000) 'Secure communication in minimal connectivity models', *Journal of Cryptology*, Vol. 13, No. 1, pp.9–30.

Fitzi, M., Franklin, M.K., Garay, J.A. and Harsha Vardhan, S. (2007) 'Towards optimal and efficient perfectly secure message transmission', in Vadhan, S.P. (Ed.): *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007*, *Proceedings, Lecture Notes in Computer Science*, 21–24 February, Vol. 4392, pp.311–322, Springer, Amsterdam, The Netherlands.

Franklin, M. and Yung, M. (1995) 'Secure hypergraphs: privacy from partial broadcast', in *Proc. of 27th Ann. Symposium on Theory of Computing*, pp.36–44.

Garay, J.A. and Perry, K.J. (1992) 'A continuum of failure models for distributed computing', in Segall, A. and Zaks, S. (Eds.): *Distributed Algorithms, 6th International Workshop, WDAG '92, Proceedings, Lecture Notes in Computer Science*, 2–4 November, Vol. 647, pp.153–165, Springer, Haifa, Israel.

Gennaro, R. (1996) 'Theory and practice of verifiable secret sharing', PhD thesis, MIT.

Goldreich, O., Micali, S. and Wigderson, A. (1987) 'How to play any mental game', in *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pp.218–229, ACM, New York, USA.

Hadzilacos, V. (1984) 'Issues of fault tolerance in concurrent computations', PhD thesis, Harvard University, Cambridge, Massachusetts.

Hirt, M. and Maurer, U.M. (2000) 'Player simulation and general adversary structures in perfect multiparty computation', *J. Cryptology*, Vol. 13, No. 1, pp.31–60.

Hirt, M., Maurer, U.M. and Zikas, V. (2008) 'MPC vs. SFE: unconditional and computational security', in Pieprzyk, J. (Ed.): *Advances in Cryptology – ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings, Lecture Notes in Computer Science*, 7–11 December, Vol. 5350, pp.1–18, Springer, Melbourne, Australia.

Kumar, M.V.N.A., Goundan, P.R., Srinathan, K. and Pandu Rangan, C. (2002) 'On perfectly secure communication over arbitrary networks', in *PODC 2002, Proceedings of the Twenty-First Annual ACM Symposium on Principles of Distributed Computing*, 21–24 July, pp.193–202, ACM, Monterey, California, USA.

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

Kurosawa, K. and Suzuki, K. (2008) 'Truly efficient 2-round perfectly secure message transmission scheme', in Smart, N.P. (Ed.): *Advances in Cryptology – EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic* Techniques, Proceedings, Lecture Notes in Computer Science, 13–17 April, Vol. 4965, pp.324–340, Springer, Istanbul, Turkey.

Lamport, L. (1983) 'The weak Byzantine generals problem', *J. ACM*, Vol. 30, No. 3, pp.668–676.

Lamport, L., Shostak, R.E. and Pease, M.C. (1982) 'The Byzantine generals problem', *ACM Trans. Program. Lang. Syst*., Vol. 4, No. 3, pp.382–401.

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

Menger, K. (1927) 'Zur allgemeinen kurventheorie', *Fundamenta Mathematicae*, Vol. 10, pp.96–115.

Narayanan, A., Srinathan, K. and Pandu Rangan, C. (2006) 'Perfectly reliable message transmission', *Information Processing Letters*, Vol. 11, No. 46, pp.1–6.

Ostrovsky, R. and Yung, M. (1991) 'How to withstand mobile virus attacks', in *Proceedings of the Tenth Annual ACM Symposium on Princiles of Distributed Computing*, 19–21 August, pp.51–61, ACM Press, Montreal, Quebec, Canada.

Patra, A., Choudhury, A. and Pandu Rangan, C. (2008) 'Unconditionally reliable and secure message transmission in directed networks revisited', in Ostrovsky, R., Prisco, R.D. and Visconti, I. (Eds.): *Security and Cryptography for Networks, 6th International Conference, SCN 2008, Proceedings, Lecture Notes in Computer Science*, 10–12 September, Vol. 5229, pp.309–326, Springer, Amalfi, Italy.

Patra, A., Choudhury, A. and Pandu Rangan, C. (2009) 'Perfectly secure message transmission in directed networks revisited', to appear in *Proc. of PODC 2009*.

Patra, A., Choudhury, A., Srinathan, K. and Pandu Rangan, C. (2006) 'Constant phase bit optimal protocols for perfectly reliable and secure message transmission', in Barua, R. and Lange, T. (Eds.): *Progress in Cryptology – INDOCRYPT 2006, 7th International Conference on Cryptology in India, Proceedings, Lecture Notes in Computer Science*, 11–13 December, Vol. 4329, pp.221–235, Springer, Kolkata, India.

Patra, A., Shankar, B., Choudhury, A., Srinathan, K. and Pandu Rangan, C. (2007) 'Perfectly secure message transmission in directed networks tolerating threshold and non threshold adversary', in Bao, F., Ling, S., Okamoto, T., Wang, H. and Xing, C. (Eds.): *Cryptology and Network Security, 6th International Conference, CANS 2007, Proceedings, Lecture Notes in Computer Science*, 8–10 December, Vol. 4856, pp.80–101, Springer, Singapore.

Rabin, T. and Ben-Or, M. (1989) 'Verifiable secret sharing and multiparty protocols with honest majority (extended abstract)', in *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, 14–17 May, pp.73–85, ACM, Seattle, Washington, USA.

Renault, J. and Tomala, T. (2008) 'Probabilistic reliability and privacy of communication using multicast in general neighbor networks', *J. Cryptology*, Vol. 21, No. 2, pp.250–279.

Sayeed, H. and Abu-Amara, H. (1995) 'Perfectly secure message transmission in asynchronous networks', in *Proceedings of 7th IEEE Symposium on Parallel and Distributed Processing*, IEEE.

Sayeed, H. and Abu-Amara, H. (1996) 'Efficient perfectly secure message transmission in synchronous networks', *Information and Computation*, Vol. 126, No. 1, pp.53–61.

Shanker, B., Gopal, P., Srinathan, K. and Pandu Rangan, C. (2008) 'Unconditional reliable message transmission in directed networks', in Teng, S. (Ed.): *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008*, 20–22 January, pp.1048–1055, SIAM, San Francisco, California, USA.

Srinathan, K. (2006) 'Secure distributed communication', PhD thesis, Indian Institute of Technology Madras.

Srinathan, K. and Pandu Rangan, C. (2006) 'Possibility and complexity of probabilistic reliable communication in directed networks', in Ruppert, E. and Malkhi, D. (Eds.): *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006*, 23–26 July, pp.265–274, ACM Press, Denver, CO, USA.

Srinathan, K., Narayanan, A. and Pandu Rangan, C. (2004) 'Optimal perfectly secure message transmission', in Franklin, M.K. (Ed.): *Advances in Cryptology – CRYPTO 2004, 24th Annual International Cryptology Conference, Proceedings, Lecture Notes in Computer Science*, 15–19 August, Vol. 3152, pp.545–561, Springer, Santa Barbara, California, USA.

Srinathan, K., Patra, A., Choudhury, A. and Pandu Rangan, C. (2007a) 'Probabilistic perfectly reliable and secure message transmission – possibility, feasibility and optimality', in Srinathan, K., Pandu Rangan, C. and Yung, M. (Eds.): *Progress in Cryptology – INDOCRYPT 2007, 8th International Conference on Cryptology in India, Proceedings, Lecture Notes in Computer Science*, 9–13 December, Vol. 4859, pp.101–122, Springer, Chennai, India.

Srinathan, K., Prasad, N.R. and Pandu Rangan, C. (2007b) 'On the optimal communication complexity of multiphase protocols for perfect communication', in *Proceedings of 2007 IEEE Symposium on Security and Privacy (S&P 2007)*, 20–23 May, pp.311–320, IEEE Computer Society, Oakland, California, USA.

Srinathan, K., Choudhury, A., Patra, A. and Pandu Rangan, C. (2008a) 'Efficient single phase unconditionally secure message transmission with optimum communication complexity', in Bazzi, R.A. and Patt-Shamir, B. (Eds.): *Proceedings of the Twenty-Seventh Annual ACM Symposium on Principles of Distributed Computing, PODC 2008*, 18–21 August, p.457, ACM, Toronto, Canada.

Srinathan, K., Patra, A., Choudhury, A. and Pandu Rangan, C. (2008b) 'Unconditionally reliable message transmission in directed hypergraphs', in Franklin, M.K., Hui, L.C.K. and Wong, D.S. (Eds.): *Cryptology and Network Security, 7th International Conference, CANS 2008, Proceedings, Lecture Notes in Computer Science*, 2–4 December, Vol. 5339, pp.285–303, Springer, Hong Kong, China.

Srinathan, K., Patra, A., Choudhury, A. and Pandu Rangan, C. (2009) 'Unconditionally secure message transmission in arbitrary directed synchronous networks tolerating generalized mixed adversary', in Li, W., Susilo, W., Tupakula, U.K., Safavi-Naini, R. and Varadharajan, V. (Eds.): *Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2009*, 10–12 March, pp.171–182, ACM, Sydney, Australia.

Wang, Y. and Desmedt, Y. (2001) 'Secure communication in multicast channels: the answer to Franklin and Wright's question', *J. Cryptology*, Vol. 14, No. 2, pp.121–135.

Yao, A.C. (1982) 'Protocols for secure computations', in *Proceedings of 23rd Annual Symposium on Foundations of Computer Science*, 3–5 November, pp.160–164, IEEE Computer Scciety, Chicago, Illinois.

Zetter, K. (2005) 'Cisco security hole a whopper', available at http://www.wired.com/politics/security/news/2005/07/68328.

Notes

1 Few results of this paper appeared in Srinathan et al. (2007a, 2008a).

2 The approach of abstracting the network as a collection of n wires is justified using Menger's (1927) theorem which states that a graph is c – (**S**, **R**)-connected iff **S** and **R** are connected by at least c vertex disjoint paths.

3 Franklin and Wright (1998) termed URMT (USMT) as almost perfectly reliable (secure) message transmission i.e., APRMT (APSMT).

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

5 All the protocols which uses same set of possible values to send along all the wires are said to satisfy symmetry property. Suppose, however, that there 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 ith 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.

6 Note that in an undirected graph, possibility of URMT from **S** to **R** entails the possibility of URMT from **R** to **S** as well.