Unconditionally Reliable and Secure Message Transmission in Directed Networks Revisited

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

Abstract

In this paper, we re-visit the problem of *unconditionally reliable message transmission* (URMT) and *unconditionally secure message transmission* (USMT) in a *directed network* under the presence of a threshold adaptive Byzantine adversary, having *unbounded computing power*. Desmedt et.al [5] have given the necessary and sufficient condition for the existence of URMT and USMT protocols in directed networks. Though their protocols are efficient, they are not communication optimal. In this paper, we prove for the first time the lower bound on the communication complexity of URMT and USMT protocols in directed networks. Moreover, we show that our bounds are tight by giving efficient communication optimal URMT and USMT protocols, whose communication complexity satisfies our proven lower bounds.

Keywords: Error Probability, Information Theoretic Security, Byzantine Adversary.

1 Introduction

Consider the following problem: a sender **S** and a receiver **R** are a part of directed synchronous network and are connected by uni-directional vertex disjoint paths/channels (also called as *wires*), which are directed either from **S** to **R** or vice-versa. An adversary $\mathcal{A}_t$ having *unbounded computing power* controls at most t wires between **S** and **R** in Byzantine fashion. **S** intends to communicate a message $M^{\mathbf{S}}$ containing ℓ field elements from a finite field $\mathbb{F}$ to **R**. The challenge is to design a protocol such that after interacting in phases[1] as per the protocol, **R** should output $M^{\mathbf{R}}$ where $M^{\mathbf{R}} = M^{\mathbf{S}}$ with probability at least $1 - \text{poly}(\kappa)2^{-\kappa}$ and κ is the error parameter. Moreover this should hold irrespective of the way adversary controls the t wires. This problem is called *unconditionally reliable message transmission* (URMT)[7, 5]. The problem of *unconditionally secure message transmission* (USMT)[7, 5] has an additional restriction that at the end of the protocol, the adversary should have *no* information about $M^{\mathbf{S}}$ in *information theoretic* sense.

If **S** and **R** are directly connected by a private channel, as assumed in generic secure multiparty computation protocols [2, 13, 3, 9], then reliable and secure communication between them is guaranteed. However this assumption implies that the underlying network is a complete graph, which is impractical! In incomplete networks, where **S** and **R** are NOT directly connected, URMT/USMT protocols help to simulate a reliable/secure link with very high probability. There is another motivation to study USMT protocols. Currently, the security of all existing public key cryptosystems, digital signature schemes, etc are based on *unproven* hardness assumptions of certain number theoretic problems. However with increase in computing speed and advent of new computing paradigm like Quantum computing may render these assumptions to be baseless. In such a scenario, USMT protocols will help to achieve information theoretic security against an all powerful adversary with very high probability.

Existing Literature: In [6] Dolev et.al have shown that PRMT/PSMT between **S** and **R** tolerating $\mathcal{A}_t$ is possible in an undirected network iff there exists $2t + 1$ bidirectional wires between **S** and **R**.

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

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

The problem of PRMT (PSMT) is same as URMT (USMT) except that at the end of the protocol, **R** should correctly output $M^{\mathbf{R}}(=\mathbf{M}^{\mathbf{S}})$ with zero error probability (i.e., $\kappa = 0$). URMT and USMT in the presence of $\mathcal{A}_t$ was first introduced and solved by Franklin et.al [7] in undirected synchronous networks, where they showed that URMT/USMT between **S** and **R** is possible iff there exists $2t+1$ bi-directional wires between **S** and **R**. The problem of URMT and USMT in directed networks was first studied by Desmedt et.al [5]. Modeling the underlying network as a directed graph is well motivated because in practice not every communication channel admits bi-directional communication. For instance, a base-station may communicate to even a far-off hand-held device but the other way round communication may not be possible. Following the approach of Dolev et.al[6], the authors in [5] have abstracted the underlying directed network in the form of directed vertex disjoint paths/wires, which are directed either from **S** to **R** or vice-versa. Under such settings, Desmedt et.al have shown that URMT/USMT tolerating $\mathcal{A}_t$ is possible iff there are total $2t+1$ wires between **S** and **R**, of which at least $t+1$ should be directed from **S** to **R** [5]. Recently, Shankar et.al [10] have studied URMT in *arbitrary directed networks*, where they have given the complete characterization of URMT tolerating $\mathcal{A}_t$ by considering the underling directed network *as a whole*. Their characterization shows that it is inappropriate to model an underlying directed network in the form of directed wires between **S** and **R**. However, it is likely to take exponential time to verify whether a given directed network and $\mathcal{A}_t$ satisfies the conditions given in [10] for the possibility of URMT. Moreover, as a part of their sufficiency condition, the authors in [10] have given an exponential time URMT protocol. These two shortcomings motivate us to relook at the *wire based* characterization of URMT and USMT given by Desmedt et.al, where we can afford to design efficient protocols.

<u>Network Model and Definitions</u>: Even though it is inappropriate to model a directed graphs in the form of directed wires, the characterization of URMT and USMT given by Desmedt et.al is advantageous if the network is densely connected and there are sufficient number of wires between **S** and **R**. For such networks, we can easily check whether URMT/USMT is possible tolerating $\mathcal{A}_t$ (we have to simply verify whether there exists $2t+1$ directed wires between **S** and **R**, of which $t+1$ are directed from **S** to **R** and this can be done efficiently in polynomial time). Moreover, for such networks, Desmedt et.al have also given an efficient, polynomial time URMT/USMT protocol. So the moral is that for enough densely connected digraph, *wire based* abstraction of the network is preferable over the *graph based* one, where the digraph is considered as a whole.

In this paper, we follow the model of Desmedt et.al and abstract the underlying network in the form of a directed graph $G = (V, E)$, where **S** and **R** are two special honest nodes in V. We assume that there are n directed wires $f_1, f_2, \ldots, f_n$ from **S** to **R**, called as *top band* and u directed wires $b_1, b_2, \ldots, b_u$ from **R** to **S**, called as *bottom band.* Moreover, the wires in the *top band* are disjoint from the wires in the *bottom band.* A centralized adversary $\mathcal{A}_t$ with unbounded computing power actively controls at most t wires between **S** and **R**, including the *top* and *bottom band* in a colluded fashion. The adversary is *adaptive*; i.e., it can corrupt wires *dynamically* during the protocol execution and its choice of corrupting a wire depends upon the data seen so far. A wire once under the control of $\mathcal{A}_t$, will remain so for the rest of the protocol. A wire under the control of $\mathcal{A}_t$ may behave arbitrarily and the communication over such wires is fully known and dictated by $\mathcal{A}_t$. We say that a wire is corrupted, if the value(s) sent over the wire is changed arbitrarily by $\mathcal{A}_t$. A wire which is not under the control of $\mathcal{A}_t$ is called *honest.* We assume that $n = max(t+1, 2t-u+1)$ and $n+u = 2t+1$, which is the *minimum* number of wires needed for the existence of URMT/USMT tolerating $\mathcal{A}_t$ [5]. The network is synchronous and a protocol is executed in terms of phases, where a phase denotes a communication either from **S** to **R** or vice-versa.

Our protocols provide *unconditional security*; i.e., *information theoretic security* with an error probability of poly(κ)$2^{-\kappa}$ in reliability, where κ is the error parameter (also known as security parameter). The error probability of our protocols equals the probability to successfully guess an element from the field over which computations are done. So to make the error probability at most poly(κ)$2^{-\kappa}$, all our computations are performed over a finite field $\mathbb{F}$ with $|\mathbb{F}| = \mathrm{GF}(poly(\kappa)2^{\kappa})$. Thus each field element can be represented by $O(\kappa)$ bits. We use $\mathbf{M}^{\mathbf{S}}$ to denote the message, which is a sequence of $\ell \geq 1$ field elements from $\mathbb{F}$, that **S** intends to send to **R**.

<u>Our Contributions</u>: One of the key parameters of any URMT/USMT protocol is its communication complexity, which is the number of field elements (bits) communicated by **S** and **R** in the protocol. Though the USMT protocol of [5] is efficient, it is not optimal in terms of communication complexity. In this paper, we prove the lower bound on the communication complexity of multiphase (more than one phase) URMT(USMT) protocols[2], which reliably(securely) sends a message containing ℓ field elements. Moreover, we show that our bounds are tight by giving efficient, polynomial time communication optimal URMT/USMT protocols which are first of their kind. Specifically, we show that (a) There exists an $O(u)$ phase URMT protocol, which reliably sends $\ell\kappa$ bits by communicating $O(\ell\kappa)$ bits. Thus we get reliability with constant factor overhead in communication complexity. Since any URMT protocol to send $\ell\kappa$ bits needs to communicate $\Omega(\ell\kappa)$ bits, our protocol is communication optimal. (b) If at least one wire in the *bottom band* is un-corrupted, then there exists an $O(u)$ phase USMT protocol which securely sends $\ell\kappa$ bits by communicating $O(\ell\kappa)$ bits. Thus we achieve security with constant factor overhead in communication complexity. It is easy to see that the protocol is communication optimal. (c) If full bottom band is corrupted by $\mathcal{A}_t$, then any multiphase USMT protocol needs to communicate $\Omega\left(\frac{n\ell}{u}\kappa\right)$ bits to securely sends $(\ell\kappa)$ bits. Moreover, we show that the bound is tight by designing an $O(u)$ phase USMT which sends $(\ell\kappa)$ bits by communicating $O\left(\frac{n\ell}{u}\kappa\right)$ bits.

To design our protocols, we use several new techniques, which are of independent interest. For ease of exposition, we assume that if **S** (**R**) is expecting some value(s) in some specific format from **R** (**S**) along a wire and if nothing (or some syntactically incorrect value(s)) comes, then **S** (**R**) substitutes predefined value(s) from $\mathbb{F}$ in the same specific format and continue the protocol. Thus, we separately do not consider the case when nothing or something syntactically incorrect comes along a wire.

Tools Used:

1. Unconditionally Reliable Authentication: It is used to send a message M over a wire such that if the wire is uncorrupted, then **R** correctly gets M and if the wire is corrupted, then **R** does not get M but is able to detect the corruption with very high probability. This is done as follows: Let a non-zero $(a, b) \in_R \mathbb{F}^2$ is *securely* established between **S** and **R** in advance. **S** computes $x = URauth(M; a, b) = aM + b$ and sends (M, x) to **R** over the wire. Let **R** receives (M', x') along the wire. **R** verifies $x' \stackrel{?}{=} URauth(M'; a, b)$. If the test fails then **R** concludes that $M' \neq M$, otherwise $M' = M$. The tuple (a, b) is called *authentication key*. Now similar to the proof of information checking (IC) protocol of [9], the probability that $M' \neq M$, but still **R** fails to detect it is at most $\frac{1}{|\mathbb{F}|}$, which is negligible in our context. Note that a remains information theoretically secure, even if the adversary knows (M, x) by listening the wire.

2. Unconditionally Secure Authentication: Its goal is similar to unconditionally reliable authentication, except that adversary should not get any information about M. This is done as follows: Let $(a, b, c) \in_R \mathbb{F}^3 - \{(0, 0, 0)\}$, which is *securely* established between **S** and **R** in advance. **S** computes $(x, y) = USauth(M; a, b, c) = (M + a, b(M + a) + c)$ and sends (x, y) to **R** over the wire. Let **R** receives (x', y') along the wire. **R** verifies $y' \stackrel{?}{=} bx' + c$. If the test fails then **R** concludes that wire is corrupted, else **R** recovers $x' - a$. It is easy to see that if the adversary knows (x, y), then also M is information theoretic secure. Moreover, if $(x', y') \neq (x, y)$, then except with error probability of at most $\frac{1}{|\mathbb{F}|}$ (which is negligible), **R** will be able to detect it.

3. Unconditional Hashing: Let $(v_1, v_2, \ldots, v_\ell), \ell > 1$ be a random vector from $\mathbb{F}^\ell$ and $k \in \mathbb{F} - \{0\}$. Then we define $hash(k; v_1, v_2, \ldots, v_\ell) = v_1 + v_2k + v_3k^2 + \ldots + v_\ell k^{\ell-1}$ [1]. Here k is called the *hash key*. The probability that two different vectors map to the same hash value for an uniformly chosen hash key is at most $\frac{\ell}{|\mathbb{F}|}$. If $\mathcal{A}_t$ knows only k and $hash(k; v_1, v_2, \ldots, v_\ell)$, then $\ell - 1$ elements in the vector will be information theoretically secure.

4. Extracting Randomness: Suppose **S** and **R** by some means agree on a sequence of n random numbers $x = [x_1\ x_2\ \ldots\ x_n] \in \mathbb{F}^n$ such that $\mathcal{A}_t$ knows $n - f$ components of x, but has no information about the other f components of x. However **S** and **R** do not know which values are known to $\mathcal{A}_t$. The goal of **S** and **R** is to agree on a sequence of f elements $[y_1\ y_2\ \ldots\ y_f] \in \mathbb{F}^f$, such that $\mathcal{A}_t$ has no information about $[y_1\ y_2\ \ldots\ y_f]$. This is done as follows [11]:

[2] Any single phase URMT/USMT protocol in directed network is no different from a single phase URMT(USMT) protocol in undirected networks. So the connectivity requirement for single URMT (USMT) is same for both directed and undirected networks. In [8], the lower bound on the communication complexity of single phase URMT/USMT protocol in undirected networks is proved. The same bound holds in directed networks also.

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 $[y_1\ y_2\ \ldots\ y_f] = [x_1\ x_2\ \ldots\ x_n]V$.

2 Three Phase USMT Protocol of Desmedt et.al [5, 12]

We now briefly recall the three phase USMT protocol of [5] to send a message $m^{\mathbf{S}} \in \mathbb{F}$ from **S** to **R**. We call the protocol as $\Pi^{Existing}$. Though we present the protocol in terms of phases, it was actually presented in terms of rounds in [5], where in each round, either **S** or **R** does some communication through a wire in top or bottom band respectively. But one can easily verify that when expressed in terms of phases, the USMT protocol of [5] takes three phases. The current informal description of the protocol is taken from [12], where as the formal description is taken from [5]. The main goal of recalling the protocol here is to highlight few techniques which are used in the protocol. These techniques are also used in our URMT and USMT protocols. In the protocol, there are following two cases: (a) There exists $t+1$ non-faulty wires in the top band; (b) There exists less that $t+1$ non-faulty wires in the top band, which implies that at least one wire in the bottom band is non-faulty.

Phase I: S to R

1. **S** selects a random polynomial $p(x)$ of degree t over $\mathbb{F}$ such that $p(0) = m^{\mathbf{S}}$ and computes the secret shares $(s_1^{\mathbf{S}}, s_2^{\mathbf{S}}, \ldots, s_n^{\mathbf{S}})$, where $s_i^{\mathbf{S}} = p(i)$, $1 \leq i \leq n$. In order to authenticate each $s_i^{\mathbf{S}}$, **S** selects n random non-zero authentication keys $(a_{i,j}^{\mathbf{S}}, b_{i,j}^{\mathbf{S}}) \in \mathbb{F}^2$, $1 \leq j \leq n$. In addition, corresponding to each wire f_i in *top band*, **S** selects a random non-zero three tuple $(a_i^{\mathbf{S}}, b_i^{\mathbf{S}}, c_i^{\mathbf{S}}) \in \mathbb{F}^3$.
2. **S** sends $\{s_i^{\mathbf{S}}, d_{i,1}^{\mathbf{S}}, d_{i,2}^{\mathbf{S}}, \ldots, d_{i,n}^{\mathbf{S}}\}$ and the three tuple $(a_i^{\mathbf{S}}, b_i^{\mathbf{S}}, c_i^{\mathbf{S}})$ to **R** through wire f_i where $d_{i,j} = URauth(s_i^{\mathbf{S}}; a_{i,j}^{\mathbf{S}}, b_{i,j}^{\mathbf{S}}), 1 \leq j \leq n$. In addition, **S** sends the authentication key $(a_{i,j}^{\mathbf{S}}, b_{i,j}^{\mathbf{S}})$ to **R** through wire f_j, for $1 \leq j \leq n$.

Computation by R at the end of Phase I:

1. Let **R** receives $\{s_i^{\mathbf{R}}, d_{i,1}^{\mathbf{R}}, d_{i,2}^{\mathbf{R}}, \ldots, d_{i,n}^{\mathbf{R}}\}$ and $(a_i^{\mathbf{R}}, b_i^{\mathbf{R}}, c_i^{\mathbf{R}})$ along wire f_i and keys $(a_{i,j}^{\mathbf{R}}, b_{i,j}^{\mathbf{R}})$ along wire f_j.
2. **R** computes $Support_i = |\{j : d_{i,j}^{\mathbf{R}} = URauth(s_i^{\mathbf{R}}; a_{i,j}^{\mathbf{R}}, b_{i,j}^{\mathbf{R}})\}|$. If $Support_i \geq t+1$, then **R** concludes that $s_i^{\mathbf{R}}$ is a valid share. Otherwise, it is an invalid share. If **R** receives $t+1$ valid shares then **R** recovers the secret $m^{\mathbf{R}}$ from these valid shares and terminates. Otherwise, **R** proceeds to execute **Phase II**.

Table 1: Phase I of USMT Protocol $\Pi^{Existing}$[5]

During **Phase I**, **S** constructs $(t+1)$-out-of-n secret shares of $m^{\mathbf{S}}$ and associates one share with one wire in the top band. In order to authenticate the share associated with a wire, **S** selects n pair of random authentication keys. **S** then sends to **R** the share associated with a wire, authenticated with all the n keys. Parallely, **S** sends the authentication keys to **R**, one over each wire. In addition, **S** associates a random three tuple with each wire and sends it to **R**. If there are $t+1$ non-faulty wires in the top band, then at the end of **Phase I**, **R** will get at least $t+1$ correct shares with which he can recover $m^{\mathbf{R}}$. The **Phase I** of $\Pi^{Existing}$ is given in Table 1.

If **R** cannot recover the secret $m^{\mathbf{R}}$ at the end of **Phase I**, then it implies that there is at least one honest wire in the bottom band. In this case, using the wires in the bottom band, **S** and **R** tries to correctly and securely agree on a shared authentication key and encryption key to securely communicate $m^{\mathbf{S}}$ from **S** to **R**. For this, **R** uses the 3-tuples $(a_i^{\mathbf{R}}, b_i^{\mathbf{R}}, c_i^{\mathbf{R}})$ which **R** has received from **S**. Now **R** sends a random non-zero 2-tuple $(d_i^{\mathbf{R}}, e_i^{\mathbf{R}})$ to **S** on each wire in bottom band. In addition, each such 2-tuple is authenticated by u random non-zero keys, so that **S** can verify whether it has correctly received the 2-tuples. Now according to the values that **S** receives from **R**, **S** divides the bottom band into consistent sub-sets $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_k$, where $k \leq u$, such that for each $1 \leq l \leq k$, all the wires in $\mathcal{B}_l$ behave in a "consistent" way. In particular, there exists at least one path set $\mathcal{B}_l$ that behave honestly during **Phase II**. Though **S** cannot determine which path set was honest, **S** will try to use each of them in a separate way and let **R** to determine which path set is honest. The computation and communication by **R** during **Phase II** and the computation by **S** at the end of **Phase II** is shown in Table 2.

Note 1 *In Table 2, $\langle \ldots \rangle$ denotes a function which is used in [5]. The function $\langle \ldots \rangle$ maps a variable size (the variable size is bounded by a pre-defined bound) ordered subset of $\mathbb{F}$ to an image element in a field extension $\mathbb{F}^*$ of $\mathbb{F}$. Moreover, from any image element (in $\mathbb{F}^*$), one can uniquely and efficiently recover the ordered subset (in $\mathbb{F}$).*

Phase II: R to S (if R has not recovered the secret at the end of Phase I)

1. For $1 \leq i \leq n$, **R** chooses a random non-zero $r_i^{\mathbf{R}} \in \mathbb{F}$ and computes $\beta^{\mathbf{R}} = \{(r_1^{\mathbf{R}}, \gamma_1^{\mathbf{R}}), (r_2^{\mathbf{R}}, \gamma_2^{\mathbf{R}}), \ldots, (r_n^{\mathbf{R}}, \gamma_n^{\mathbf{R}})\}$, where $\gamma_j^{\mathbf{R}} = hash(r_j^{\mathbf{R}}; a_j^{\mathbf{R}}, b_j^{\mathbf{R}}, c_j^{\mathbf{R}}), 1 \leq j \leq n$. For each $1 \leq i \leq u$, **R** selects a random non-zero 2-tuple $(d_i^{\mathbf{R}}, e_i^{\mathbf{R}}) \in \mathbb{F}^2$. In order to authenticate $(d_i^{\mathbf{R}}, e_i^{\mathbf{R}})$, **R** selects u random non-zero keys $\{(v_{i,j}^{\mathbf{R}}, w_{i,j}^{\mathbf{R}}) \in \mathbb{F}^2 : 1 \leq j \leq u\}$.
2. For each $1 \leq i \leq u$, **R** sends $\beta^{\mathbf{R}}, (d_i^{\mathbf{R}}, e_i^{\mathbf{R}})$ and $\{\alpha_{i,j}^{\mathbf{R}} : 1 \leq j \leq n\}$, where $\alpha_{i,j}^{\mathbf{R}} = URauth(\langle d_i^{\mathbf{R}}, e_i^{\mathbf{R}} \rangle; v_{i,j}^{\mathbf{R}}, w_{i,j}^{\mathbf{R}}) : 1 \leq j \leq u\}$ to **S** via wire b_i and the keys $(v_{i,j}^{\mathbf{R}}, w_{i,j}^{\mathbf{R}})$ to **S** via wire b_j for each $1 \leq j \leq u$.

Computation by S at the end of Phase II:

1. Let **S** receives $\beta_i^{\mathbf{S}}, (d_i^{\mathbf{S}}, e_i^{\mathbf{S}})$ and $\{\alpha_{i,j}^{\mathbf{S}} : 1 \leq j \leq u\}$ from **R** via wire b_i and $(v_{i,j}^{\mathbf{S}}, w_{i,j}^{\mathbf{S}})$ from **R** via wire b_j for each $1 \leq j \leq u$. **S** divides the bottom band $\{b_1, b_2, \ldots, b_u\}$ into subsets $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_k$, where $k \leq u$, such that for any l, m, p with $1 \leq l \leq k, 1 \leq m, p \leq u$ and $b_m, b_p \in \mathcal{B}_l$, we have: (a) $\beta_m^{\mathbf{S}} = \beta_p^{\mathbf{S}}$; (b) $\alpha_{m,p}^{\mathbf{S}} = URauth(\langle d_m^{\mathbf{S}}, e_m^{\mathbf{S}} \rangle; v_{m,p}^{\mathbf{S}}, w_{m,p}^{\mathbf{S}})$; (c) $\alpha_{p,m}^{\mathbf{S}} = URauth(\langle d_p^{\mathbf{S}}, e_p^{\mathbf{S}} \rangle; v_{p,m}^{\mathbf{S}}, w_{p,m}^{\mathbf{S}})$.
2. For $\mathcal{B}_l$, let $b_m \in \mathcal{B}_l$ and $\beta_m^{\mathbf{S}} = \{(r_{i,l}^{\mathbf{S}}, \gamma_{i,l}^{\mathbf{S}}) : 1 \leq i \leq n\}$. **S** computes the set of wires in top band

$$\mathcal{F}_l = \{i : \gamma_{i,l}^{\mathbf{S}} = hash(r_{i,l}^{\mathbf{S}}; a_i^{\mathbf{S}}, b_i^{\mathbf{S}}, c_i^{\mathbf{S}}), 1 \leq i \leq n\}$$

If $|\mathcal{B}_l| + |\mathcal{F}_l| \leq t$ then **S** decides that $\mathcal{B}_l$ is unacceptable set, otherwise $\mathcal{B}_l$ is acceptable set.

Table 2: **Phase II** and computation by **S** at the end of **Phase II** of Protocol $\Pi^{Existing}$

From the properties of $URauth$ and $hash$, it is easy to check that the following holds: (a) If b_i is an honest wire in the bottom band and $b_i \in \mathcal{B}_l$, then with very high probability, the random 2-tuples that **S** has received along the wires in $\mathcal{B}_l$ are not modified; (b) If b_i is an honest wire in the bottom band and $b_i \in \mathcal{B}_l$, then $\mathcal{B}_l$ is an acceptable set. However, all the acceptable sets look same to **S** and **S** cannot determine whether an acceptable set contains all honest wires or wires controlled by the adversary. In the worst case, the adversary can control the bottom band in such a way that there are at most u $\mathcal{B}_l$'s, with one wire from the bottom band in each $\mathcal{B}_l$. **S** continues the protocol by assuming that each acceptable set is correct. In other words, assuming that all the wires in an acceptable set $\mathcal{B}_l$ are non-faulty, **S** determines which of the random 3-tuples $(a_i^{\mathbf{S}}, b_i^{\mathbf{S}}, c_i^{\mathbf{S}})$, (that it had sent to **R** during **Phase I**) have been correctly received by **R**. Using these "correctly-received-by-**R**" 3-tuples and the random 2-tuples received by **S** via the wires in $\mathcal{B}_l$, **S** computes the authentication key and encryption key to securely send the messages to **R**. If the assumption that $\mathcal{B}_l$ contains only non-faulty wires is valid, then **R** would be able to compute the same authentication and encryption key. Since at least one of the acceptable path set is non-faulty, **R** will be able to decrypt the secret message correctly. The communication by **S** during **Phase III** and message recovery by **R** is shown in Table 3.

Phase III: S to R: For each acceptable set $\mathcal{B}_l$ and the corresponding set $\mathcal{F}_l$, **S** does the following:

- From the wires in $\mathcal{F}_l$ and $\mathcal{B}_l$, **S** computes his version of the keys $\mathcal{C}_l^{\mathbf{S}} = \sum_{f_i \in \mathcal{F}_l} a_i^{\mathbf{S}} + \sum_{b_i \in \mathcal{B}_l} d_i^{\mathbf{S}}$ and $\mathcal{D}_l^{\mathbf{S}} = \sum_{f_i \in \mathcal{F}_l} b_i^{\mathbf{S}} + \sum_{b_i \in \mathcal{B}_l} e_i^{\mathbf{S}}$. **S** then sends $(\psi_l^{\mathbf{S}}, \lambda_l^{\mathbf{S}})$ to **R** over all the wires in $\mathcal{F}_l$, where $\psi_l^{\mathbf{S}} = \langle \mathcal{B}_l, \mathcal{F}_l, m^{\mathbf{S}} + \mathcal{C}_l^{\mathbf{S}} \rangle$ and $\lambda_l^{\mathbf{S}} = URauth(\psi_l^{\mathbf{S}}; \mathcal{C}_l^{\mathbf{S}}, \mathcal{D}_l^{\mathbf{S}})$.

Message Recovery by R: **R** knows that in the worst case, **S** could have sent u 2-tuples over each wire in the top band, corresponding to the case when there are u acceptable sets. Let **R** receives $(\psi_{i,l}^{\mathbf{R}}, \lambda_{i,l}^{\mathbf{R}})$ over wire f_i for $1 \leq i \leq n$ and $1 \leq l \leq u$.

1. For each $1 \leq i \leq n$, **R** computes $\langle \mathcal{B}_{i,l}^{\mathbf{R}}, \mathcal{F}_{i,l}^{\mathbf{R}}, \tau_{i,l}^{\mathbf{R}} \rangle = \psi_{i,l}^{\mathbf{R}}$ (that is, **R** decomposes $\psi_{i,l}^{\mathbf{R}}$). **R** then computes his version of the keys $\mathcal{C}_{i,l}^{\mathbf{R}} = \sum_{f_j \in \mathcal{F}_{i,l}} a_j^{\mathbf{R}} + \sum_{b_j \in \mathcal{B}_{i,l}} d_j^{\mathbf{R}}$ and $\mathcal{D}_{i,l}^{\mathbf{R}} = \sum_{f_j \in \mathcal{F}_{i,l}} b_j^{\mathbf{R}} + \sum_{b_j \in \mathcal{B}_{i,l}} e_j^{\mathbf{R}}$.
2. For $1 \leq i \leq n$, **R** checks whether $\lambda_{i,l}^{\mathbf{R}} \overset{?}{=} URauth(\psi_{i,l}^{\mathbf{R}}; \mathcal{C}_{i,l}^{\mathbf{R}}, \mathcal{D}_{i,l}^{\mathbf{R}})$. If the equation holds then **R** computes the secret $m^{\mathbf{R}} = \tau_{i,l}^{\mathbf{R}} - \mathcal{C}_{i,l}^{\mathbf{R}}$ and terminates.

Table 3: Phase III and Secret Recovery in Protocol $\Pi^{Existing}$

It is easy to check that with very high probability, $m^{\mathbf{R}}$ recovered by **R** is the same as $m^{\mathbf{S}}$. Since, for an acceptable set $\mathcal{B}_l$, $|\mathcal{F}_l| + |\mathcal{B}_l| > t$, the adversary learns no information about $\mathcal{C}_l^{\mathbf{S}}$ or $\mathcal{D}_l^{\mathbf{S}}$ and hence about $m^{\mathbf{S}}$. Thus the protocol achieves perfect privacy.

$M^{\mathbf{S}}(x)$, Lower order $\frac{n}{3}$ coefficients of $M^{\mathbf{S}}(x)$ are elements of $m^{\mathbf{S}}$			
$M^{\mathbf{S}}(1)$	$M^{\mathbf{S}}(2)$	$\dots$	$M^{\mathbf{S}}(n+t)$
$f_1^{\mathbf{S}}(x)$ $f_1^{\mathbf{S}}(0) = M^{\mathbf{S}}(1)$	$f_2^{\mathbf{S}}(x)$ $f_2^{\mathbf{S}}(0) = M^{\mathbf{S}}(2)$	$\dots$ $\dots$	$f_{n+t}^{\mathbf{S}}(x)$ $f_{n+t}^{\mathbf{S}}(0) = M^{\mathbf{S}}(n+t)$
$f_1^{\mathbf{S}}(1)$	$f_2^{\mathbf{S}}(1)$	$\dots$	$f_{n+t}^{\mathbf{S}}(1)$
$f_1^{\mathbf{S}}(2)$	$f_2^{\mathbf{S}}(2)$	$\dots$	$f_{n+t}^{\mathbf{S}}(2)$
$\dots$	$\dots$	$\dots$	$\dots$
$f_1^{\mathbf{S}}(j)$	$f_2^{\mathbf{S}}(j)$	$\dots$	$f_{n+t}^{\mathbf{S}}(j)$
$\dots$	$\dots$	$\dots$	$\dots$
$f_1^{\mathbf{S}}(n)$	$f_2^{\mathbf{S}}(n)$	$\dots$	$f_{n+t}^{\mathbf{S}}(n)$

Table 4: Matrix T as computed by **S**

.

2.1 Modified Version of Desmedt's USMT Protocol

We now present a modified version of protocol $\Pi^{Existing}$, called $\Pi^{Existing}_{modified}$, where all the computation and communication is done in the field $\mathbb{F}$. The purpose of presenting $\Pi^{Existing}_{modified}$ is to introduce certain new techniques, which we have also used in our later protocols. Protocol $\Pi^{Existing}_{modified}$ will be used as a sub-protocol in our final communication optimal URMT and USMT protocols. The protocol securely sends a message $m^{\mathbf{S}} = \{m_1^{\mathbf{S}}\ m_2^{\mathbf{S}}\ \dots\ m_{\frac{n}{3}}^{\mathbf{S}}\}$ containing $\frac{n}{3} = \Theta(n)$ elements from $\mathbb{F}$ by communicating $O(n^3)$ elements from $\mathbb{F}$ with very high probability.

During **Phase I**, **S** selects a random polynomial $M^{\mathbf{S}}(x)$ over $\mathbb{F}$ of degree $n-1+t$ such that the lower order $\frac{n}{3}$ coefficients of $M^{\mathbf{S}}(x)$ are elements of $m^{\mathbf{S}}$. **S** then computes $M^{\mathbf{S}}(1), M^{\mathbf{S}}(2), \dots, M^{\mathbf{S}}(n+t)$. **S** selects $n+t$ random polynomials $f_1^{\mathbf{S}}(x), f_2^{\mathbf{S}}(x), \dots, f_{n+t}^{\mathbf{S}}(x)$ over $\mathbb{F}$, each of degree t, such that $f_i^{\mathbf{S}}(0) = M^{\mathbf{S}}(i), 1 \leq i \leq n+t$. **S** then evaluates each $f_i^{\mathbf{S}}(x)$ at $x = 1, 2, \dots, n$ to form an n tuple $f_i^{\mathbf{S}} = [f_i^{\mathbf{S}}(1)\ f_i^{\mathbf{S}}(2)\ \dots\ f_i^{\mathbf{S}}(n)]$. **S** now constructs an $(n) \times (n+t)$ matrix T where i^{th} column of T contains the n tuple $f_i^{\mathbf{S}}, 1 \leq i \leq n+t$. The matrix T is pictorially shown in Figure. 4. Let $F_j^{\mathbf{S}} = [f_1^{\mathbf{S}}(j)\ f_2^{\mathbf{S}}(j)\ \dots\ f_{n+t}^{\mathbf{S}}(j)]$ denotes the $j^{th}, 1 \leq j \leq n$ row of T. Now the communication by **S** during **Phase I** and the computation by **R** at the end of **Phase I** is expressed in Table 5.

Phase I: S to R: Along wire $f_j, 1 \leq j \leq n$, **S** sends the following to **R**

1. The vector $F_j^{\mathbf{S}}$, a random non-zero *hash key* $\alpha_j^{\mathbf{S}}$ and the n tuple $[v_{1j}^{\mathbf{S}}\ v_{2j}^{\mathbf{S}}\ \dots\ v_{nj}^{\mathbf{S}}]$, where $v_{ij}^{\mathbf{S}} = hash(\alpha_j^{\mathbf{S}}; F_i^{\mathbf{S}}), 1 \leq i \leq n$.
2. A random non-zero $(n+1)$ tuple $(x_{1,j}^{\mathbf{S}}, x_{2,j}^{\mathbf{S}}, \dots, x_{n+1,j}^{\mathbf{S}})$, which is independent of $F_j^{\mathbf{S}}$.

Computation by R at the end of Phase I:

1. Let **R** receives the vector $F_j^{\mathbf{R}}$, *hash key* $\alpha_j^{\mathbf{R}}$, the n tuple $[v_{1j}^{\mathbf{R}}\ v_{2j}^{\mathbf{R}}\ \dots\ v_{nj}^{\mathbf{R}}]$ and the $n+1$ tuple $(x_{1,j}^{\mathbf{R}}, x_{2,j}^{\mathbf{R}}, \dots, x_{n+1,j}^{\mathbf{R}})$ along wire $f_j, 1 \leq j \leq n$.
2. For $1 \leq j \leq n$, **R** computes $Support_j = |\{f_i : v_{ji}^{\mathbf{R}} = hash(\alpha_i^{\mathbf{R}}; F_j^{\mathbf{R}})\}|$. If $Support_j \geq t+1$, then **R** concludes that $F_j^{\mathbf{R}}$ is a valid row of T. Otherwise, **R** concludes that $F_j^{\mathbf{R}}$ is an invalid row of T.
3. If **R** has received $t+1$ valid rows, then **R** reconstructs the secret $m^{\mathbf{R}}$ from them and terminates protocol (see Theorem 1). Otherwise, **R** proceeds to execute **Phase II**.

Table 5: Phase I of the modified protocol $\Pi^{Existing}_{modified}$

Lemma 1 *If $F_j^{\mathbf{R}}$ is a valid row, then with overwhelming probability $F_j^{\mathbf{R}} = F_j^{\mathbf{S}}$*

PROOF: The lemma is true if wire f_j is uncorrupted. If f_j is corrupted, then $F_j^{\mathbf{R}} \neq F_j^{\mathbf{S}}$. In this case, if $F_j^{\mathbf{R}}$ is valid then it implies that $Support_j \geq t+1$. Since there can be at most t corrupted wires in the top band, this implies that there exists an honest wire, say f_i, which correctly delivered the *hash key* $\alpha_i^{\mathbf{R}} = \alpha_i^{\mathbf{S}}$ and hash value $v_{ji}^{\mathbf{R}} = v_{ji}^{\mathbf{S}}$, such that $f_i \in Support_j$. But from the properties of unconditional hashing, it can happen with probability at most $\frac{n-1+t}{|\mathbb{F}|}$, which is negligible in our context. □

Lemma 2 *During* **Phase I**, *at least n coefficients of $M^{\mathbf{S}}(x)$ are information theoretically secure.*

PROOF: We consider the worst case, when $\mathcal{A}_t$ controls at most t wires in the top band. Without loss of generality, let these be the first t wires. So $\mathcal{A}_t$ will know the vectors $F_1^{\mathbf{S}}, F_2^{\mathbf{S}}, \dots, F_t^{\mathbf{S}}$, from which it will come to know t distinct points of the polynomials $f_j^{\mathbf{S}}(x), 1 \leq j \leq n+t$. But each $f_j^{\mathbf{S}}(x)$ is of degree t and so $\mathcal{A}_t$ lacks one point to reconstruct each $f_j^{\mathbf{S}}(x)$. However, $\mathcal{A}_t$ also knows t hash values corresponding to each $F_j^{\mathbf{S}}, 1 \leq j \leq n$. Since the vectors $F_1^{\mathbf{S}}, F_2^{\mathbf{S}}, \ldots, F_t^{\mathbf{S}}$ are already known to $\mathcal{A}_t$, the t hash values corresponding to them does not add anything new to $\mathcal{A}_t$'s view. Moreover, the vectors $F_{t+2}^{\mathbf{S}}, F_{t+3}^{\mathbf{S}}, \ldots, F_n^{\mathbf{S}}$ can be expressed as a linear combination of vectors $F_1^{\mathbf{S}}, F_2^{\mathbf{S}}, \ldots, F_{t+1}^{\mathbf{S}}$. So the t hash values corresponding to $F_{t+2}^{\mathbf{S}}, F_{t+3}^{\mathbf{S}}, \ldots, F_n^{\mathbf{S}}$ can always be expressed as a linear combination of the t hash values corresponding to $F_1^{\mathbf{S}}, F_2^{\mathbf{S}}, \ldots, F_{t+1}^{\mathbf{S}}$, which are known to the adversary. So, out of the t hash values corresponding to each $F_j^{\mathbf{S}}(x), 1 \leq j \leq n$, which are known to $\mathcal{A}_t$, only the t hash values corresponding to $F_{t+1}^{\mathbf{S}}(x)$ add to $\mathcal{A}_t$'s view. But $F_{t+1}^{\mathbf{S}}$ is of length $n+t$. So from the properties of unconditional hashing, $(n+t)-t=n$ coefficients of $F_{t+1}^{\mathbf{S}}$ will be information theoretic secure. This further implies that n coefficients of $M^{\mathbf{S}}(x)$ are information theoretically secure. □

Theorem 1 *If* **R** *gets* $t+1$ *valid rows then* **R** *can securely recover* $m^{\mathbf{S}}$ *with very high probability.*

PROOF: From Lemma 1, with very high probability, each valid row is indeed sent by **S**. If **R** gets $t+1$ valid rows, then from them, **R** gets $t+1$ distinct points on each $f_i^{\mathbf{S}}(x)$. Since each $f_i^{\mathbf{S}}(x)$ is of degree t, using the $t+1$ valid rows, **R** reconstructs each $f_i^{\mathbf{S}}(x)$ and hence $f_i^{\mathbf{S}}(0) = M^{\mathbf{S}}(i)$. Now using the $M^{\mathbf{S}}(i)$'s, **R** interpolates $M^{\mathbf{S}}(x)$ and recovers $m^{\mathbf{S}}$. The security of $m^{\mathbf{S}}$ follows from Lemma 2. □

If **R** does not get $t+1$ valid rows, then **R** concludes that at least one wire in the bottom band is honest. So **R** proceeds to execute **Phase II** as shown in Table 6. **Phase II** is similar to the **Phase II** of protocol $\Pi^{Existing}$, except that $\beta^{\mathbf{R}}$ contains the hashed value of each $n+1$ tuple received from **S**. Moreover, along each wire in the bottom band, **R** now sends an $(n+u)$ tuple and hash it with u random keys. Now as in protocol $\Pi^{Existing}$, depending upon the values received along the wires in the bottom band, **S** divides the bottom band into different subsets. As in the previous protocol, from the properties of hash function, it is straightforward to check that the following holds: (a) If b_i is an honest wire in the bottom band and $b_i \in \mathcal{B}_l$, then with very high probability, the random $(n+u)$-tuples that **S** has received along the wires in $\mathcal{B}_l$ are not modified; (b) If b_i is an honest wire in the bottom band and $b_i \in \mathcal{B}_l$, then $\mathcal{B}_l$ is an acceptable set.

Phase II: R to S (if R has not recovered the secret at the end of Phase I)

1. For each $1 \leq j \leq n$, **R** chooses a random non-zero hash key $r_j^{\mathbf{R}} \in \mathbb{F}$ and computes the set $\beta^{\mathbf{R}} = \{(r_j^{\mathbf{R}}, \gamma_j^{\mathbf{R}}) : 1 \leq j \leq n\}$, where $\gamma_j^{\mathbf{R}} = hash(r_j^{\mathbf{R}}; x_{1,j}^{\mathbf{R}}, x_{2,j}^{\mathbf{R}}, \ldots, x_{n+1,j}^{\mathbf{R}})$.
2. For each $1 \leq j \leq u$, **R** selects a random non-zero $n+u$ tuple $(y_{1,j}^{\mathbf{R}}, y_{2,j}^{\mathbf{R}}, \ldots, y_{n+u,j}^{\mathbf{R}}) \in \mathbb{F}^{n+u}$. In order to hash each such n tuple, **R** selects u random non-zero keys $\{key_{i,j}^{\mathbf{R}} : 1 \leq i \leq u\}$ from $\mathbb{F}$.
3. For each $1 \leq j \leq u$, **R** sends $\beta^{\mathbf{R}}$ and the $n+u$-tuple $(y_{1,j}^{\mathbf{R}}, y_{2,j}^{\mathbf{R}}, \ldots, y_{n+u,j}^{\mathbf{R}})$ to **S** over wire b_j and the 2-tuple $(key_{i,j}^{\mathbf{R}}, \alpha_{i,j}^{\mathbf{R}})$ to **S** over wire $b_i, 1 \leq i \leq u$, where $\alpha_{i,j}^{\mathbf{R}} = hash(key_{i,j}^{\mathbf{R}}; y_{1,j}^{\mathbf{R}}, y_{2,j}^{\mathbf{R}}, \ldots, y_{n+u,j}^{\mathbf{R}})$.

Computation by S at the end of Phase II: For $1 \leq j \leq u$, **S** receives $\beta_j^{\mathbf{S}}$ and the $n+u$-tuple $(y_{1,j}^{\mathbf{S}}, y_{2,j}^{\mathbf{S}}, \ldots, y_{n+u,j}^{\mathbf{S}})$ over wire b_j and the pair $(key_{i,j}^{\mathbf{S}}, \alpha_{i,j}^{\mathbf{S}})$ over $b_i, 1 \leq i \leq u$. **S** then does the following:

1. **S** divides the bottom band $\{b_1, b_2, \ldots, b_u\}$ into subsets $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_k$, where $k \leq u$, such that for any l, m, p with $1 \leq l \leq k, 1 \leq m, p \leq u$ and $b_m, b_p \in \mathcal{B}_l$, we have: (a) $\beta_m^{\mathbf{S}} = \beta_p^{\mathbf{S}}$; (b) $\alpha_{m,p}^{\mathbf{S}} = hash(key_{m,p}^{\mathbf{S}}; y_{1,p}^{\mathbf{S}}, y_{2,p}^{\mathbf{S}}, \ldots, y_{n,p}^{\mathbf{S}})$; (c) $\alpha_{p,m}^{\mathbf{S}} = hash(key_{p,m}^{\mathbf{S}}; y_{1,m}^{\mathbf{S}}, y_{2,m}^{\mathbf{S}}, \ldots, y_{p,m}^{\mathbf{S}})$.
2. For $\mathcal{B}_l$, let $b_m \in \mathcal{B}_l$ and $\beta_m^{\mathbf{S}} = \{(r_{j,l}^{\mathbf{S}}, \gamma_{j,l}^{\mathbf{S}}) : 1 \leq j \leq n\}$. **S** then computes the set

$$\mathcal{F}_l^{\mathbf{S}} = \{j : \gamma_{j,l}^{\mathbf{S}} = hash(r_{j,l}^{\mathbf{S}}; x_{1,j}^{\mathbf{S}}, x_{2,j}^{\mathbf{S}}, \ldots, x_{n+1,j}^{\mathbf{S}}), 1 \leq j \leq n\}$$

   If $|\mathcal{F}_l| + |\mathcal{B}_l| \leq t$ then **S** decides that $\mathcal{B}_l$ is unacceptable set, otherwise $\mathcal{B}_l$ is acceptable set.

Table 6: **Phase II** and computation by **S** at the end of **Phase II** in $\Pi_{modified}^{Existing}$

Before proceeding further, we prove the following important claim.

Claim 1 *Let* f_i *and* b_j *be two honest wire in top and bottom band respectively. Then at the end of* **Phase II**, *at least* n *elements in* $(x_{1,i}^{\mathbf{S}}, x_{2,i}^{\mathbf{S}}, \ldots, x_{n+1,i}^{\mathbf{S}})$ *and* $(y_{1,j}^{\mathbf{R}}, y_{2,j}^{\mathbf{R}}, \ldots, y_{n+u,j}^{\mathbf{R}})$ *are information theoretically secure.*

PROOF: Follows from the properties of hash function. □

Now the steps during **Phase III** and message recovery by **R** are presented in Table 7.

<u>**Phase III: S to R**</u>: For each acceptable set $\mathcal{B}_l$ and corresponding set $\mathcal{P}_l$, **S** does the following:

1. **S** considers the first n elements from the $n+1$ tuples which it had sent over the wires in $\mathcal{P}_l$ during **Phase I** and the first n elements from the $(n+u)$ tuples which **S** had received over the wires in $\mathcal{B}_l$ during **Phase II**. By using them, **S** computes his version of n authentication keys $\mathcal{C}^{\mathbf{S}}_{1,l} = \sum_{f_j \in \mathcal{F}_l} x^{\mathbf{S}}_{1,j} + \sum_{b_j \in \mathcal{B}_l} y^{\mathbf{S}}_{1,j}$, $\mathcal{C}^{\mathbf{S}}_{2,l} = \sum_{f_j \in \mathcal{F}_l} x^{\mathbf{S}}_{2,j} + \sum_{b_j \in \mathcal{B}_l} y^{\mathbf{S}}_{2,j}$, ..., $\mathcal{C}^{\mathbf{S}}_{n,l} = \sum_{f_j \in \mathcal{F}_l} x^{\mathbf{S}}_{n,j} + \sum_{b_j \in \mathcal{B}_l} y^{\mathbf{S}}_{n,j}$.
2. For each element of $m^{\mathbf{S}}$ (recall that $|m^{\mathbf{S}}| = \frac{n}{3}$), **S** takes three elements from the keys computed in the previous step and computes the set $\mathcal{S}^{\mathbf{S}}_l = \{(c^{\mathbf{S}}_{i,l}, d^{\mathbf{S}}_{i,l}) : 1 \leq i \leq \frac{n}{3}\}$ where $(c^{\mathbf{S}}_{i,l}, d^{\mathbf{S}}_{i,l}) = USauth(m^{\mathbf{S}}_i; \mathcal{C}^{\mathbf{S}}_{3i-2}, \mathcal{C}^{\mathbf{S}}_{3i-1}, \mathcal{C}^{\mathbf{S}}_{3i}), 1 \leq i \leq \frac{n}{3}$.
3. **S** sends the set $\mathcal{F}_l, \mathcal{B}_l$ and $\mathcal{S}^{\mathbf{S}}_l$ to **R** over all the wires in the set $\mathcal{F}_l$ and terminates.

<u>**Message Recovery by R**: Let **R** receives the sets $\mathcal{F}^{\mathbf{R}}_{j,l}, \mathcal{B}^{\mathbf{R}}_{j,l}$ and $\mathcal{S}^{\mathbf{R}}_{j,l}$ along wire $f_j, 1 \leq j \leq n$, for $1 \leq l \leq u$. **R** then does the following:</u>

1. If for some $j \in \{1, 2, \ldots, n\}$ and some $l \in \{1, 2, \ldots, u\}$, $|\mathcal{F}^{\mathbf{R}}_{j,l}| + |\mathcal{B}^{\mathbf{R}}_{j,l}| \leq t$, then **R** concludes that wire f_j is corrupted and neglects all the values received along f_j.
2. If f_j is not neglected, then for each $\mathcal{F}^{\mathbf{R}}_{j,l}, \mathcal{B}^{\mathbf{R}}_{j,l}$ and $\mathcal{S}^{\mathbf{R}}_{j,l}$ received along f_j, **R** does the following: let $\mathcal{S}^{\mathbf{R}}_{j,l} = \{(c^{\mathbf{R}}_{j,i,l}, d^{\mathbf{R}}_{j,i,l}) : 1 \leq i \leq \frac{n}{3}\}$. By using the index of the wires in $\mathcal{F}^{\mathbf{R}}_{j,l}$ and $\mathcal{B}^{\mathbf{R}}_{j,l}$, **R** computes his version of authentication keys $\mathcal{C}^{\mathbf{R}}_{j,1,l}, \mathcal{C}^{\mathbf{R}}_{j,2,l}, \ldots, \mathcal{C}^{\mathbf{R}}_{j,n,l}$. Then for each $1 \leq i \leq \frac{n}{3}$, **R** applies the verification process of $USauth$ on $c^{\mathbf{R}}_{j,i,l}, d^{\mathbf{R}}_{j,i,l}, \mathcal{C}^{\mathbf{R}}_{3i-2}, \mathcal{C}^{\mathbf{R}}_{3i-1}$ and $\mathcal{C}^{\mathbf{R}}_{3i}$. If the verification is successful for all $1 \leq i \leq \frac{n}{3}$, then **R** recovers $m^{\mathbf{R}}_i$ from $c^{\mathbf{R}}_{j,i,l}, 1 \leq j \leq \frac{n}{3}$. Finally, **R** concatenates $m^{\mathbf{R}}_1, m^{\mathbf{R}}_2, \ldots, m^{\mathbf{R}}_{\frac{n}{3}}$ to reconstruct the secret $m^{\mathbf{R}}$ and terminates.

Table 7: **Phase III** and Secret Recovery in the modified protocol $\Pi^{Existing}_{modified}$

Theorem 2 *Protocol $\Pi^{existing}_{modified}$ is a three phase USMT protocol which securely sends $\Theta(n\kappa)$ bits by communicating $O(n^3\kappa)$ bits with very high probability.*

Proof: If **R** is able to recover $m^{\mathbf{R}}$ at the end of **Phase I**, then the correctness and secrecy of $\Pi^{existing}_{modified}$ follows from Theorem 1. If **R** is unable to recover $m^{\mathbf{R}}$ at the end of **Phase I**, then the correctness and security of $\Pi^{existing}_{modified}$ follows using the similar argument as the correctness and security of $\Pi^{existing}$.

During **Phase I** and **Phase II**, $O(n^2)$ and $O(nu+u^2)$ field elements are communicated respectively. During **Phase III**, in the worst case, **S** can have u distinct acceptable sets $\mathcal{B}_l$ each of size one and correspondingly only one set $\mathcal{F}_l$ consisting of the entire top band. In this case, on behalf of each $\mathcal{B}_l$, **S** will have to communicate $O(n^2)$ field elements, thus incurring a total communication overhead of $O(n^2u)$. Since $u = O(n)$, the worst case communication complexity of **Phase III** and hence $\Pi^{existing}_{modified}$ is $O(n^3)$. Now each field element can be represented by $O(\kappa)$ bits. So the protocol sends $m^{\mathbf{S}}$ containing $\Theta(n\kappa)$ bits by communicating $O(n^3\kappa)$ bits with very high probability. □

3 Unconditionally Secure One Time Pad Establishment Protocol

We now propose a six phase protocol called Π^{Pad}, which securely establishes a random non-zero one time pad between **S** and **R** with very high probability by communicating $O(n^3)$ field elements. If the entire bottom band is corrupted, then the size of the pad is $\Theta(n^2u)$. Otherwise the size of the pad is $\Theta(n^3)$. We first design a sub-protocol Π which is used in Π^{Pad}.

<u>**Protocol Π**</u>: Suppose **S** and **R** in advance know that full bottom band is corrupted. This implies that at most $t-u$ and at least $t+1$ wires in the top band are corrupted and honest respectively. Under this assumption, we design a sub-protocol Π, which securely establishes an information theoretic secure non-zero random one time pad of size $\Theta(n^2u)$ between **S** and **R** by communicating $O(n^3)$ field elements, with very high probability.

Let $c = n^2+t-u$. **S** selects $(t+1)\times c$ random non-zero elements from $\mathbb{F}$, denoted by $k^{\mathbf{S}}_{1,1}, k^{\mathbf{S}}_{1,2}, \ldots, k^{\mathbf{S}}_{1,c}$, $k^{\mathbf{S}}_{2,1}, k^{\mathbf{S}}_{2,2}, \ldots, k^{\mathbf{S}}_{2,c}, \ldots, k^{\mathbf{S}}_{t+1,1}, k^{\mathbf{S}}_{t+1,2}, \ldots, k^{\mathbf{S}}_{t+1,c}$. Now using these elements, **S** constructs an $(t+1) \times c$ matrix $A^{\mathbf{S}}$, where the $j^{th}, 1 \leq j \leq t+1$ row of $A^{\mathbf{S}}$ is $[k^{\mathbf{S}}_{j,1}\ k^{\mathbf{S}}_{j,2}\ \ldots\ k^{\mathbf{S}}_{j,i}\ \ldots\ k^{\mathbf{S}}_{j,c}]$. Now consider the $i^{th}, 1 \leq i \leq c$ column of A containing the elements $[k^{\mathbf{S}}_{1,i}\ k^{\mathbf{S}}_{2,i}\ \ldots k^{\mathbf{S}}_{t+1,i}]^T$. **S** forms a t degree polynomial $q_i(x)$ passing through the $t+1$ points $[(1, k^{\mathbf{S}}_{1,i}), (2, k^{\mathbf{S}}_{2,i}), \ldots, (t+1, k^{\mathbf{S}}_{t+1,i})]$. **S** now evaluates $q_i(x)$ at $x = t+2, t+3, \ldots, n$ to obtain $y^{\mathbf{S}}_{t+2,i}, y^{\mathbf{S}}_{t+3,i}, \cdots, y^{\mathbf{S}}_{n,i}$ respectively. Finally, **S** constructs the matrix $B^{\mathbf{S}}$

of size $n \times c$, where the $i^{th}, 1 \leq i \leq c$ column of $B^{\mathbf{S}}$ is $[k^{\mathbf{S}}_{1,i}\ k^{\mathbf{S}}_{2,i}\ \ldots k^{\mathbf{S}}_{t+1,i}\ y^{\mathbf{S}}_{t+2,i}\ y^{\mathbf{S}}_{t+3,i}\ \cdots\ y^{\mathbf{S}}_{n,i}]^T$, the n points on $q_i(x)$ as shown in Table 8. Now using the $j^{th}, 1 \leq j \leq n$ row of $B^{\mathbf{S}}$, **S** forms a $n^2 + t - u - 1$ degree polynomial $F^{\mathbf{S}}_j(x) = k^{\mathbf{S}}_{j,1} + k^{\mathbf{S}}_{j,2}x^1 + k^{\mathbf{S}}_{j,3}x^2 + \ldots + k^{\mathbf{S}}_{j,c}x^{c-1}$. **S** also selects n random and non-zero distinct elements from $\mathbb{F}$, denoted by $\alpha^{\mathbf{S}}_1, \alpha^{\mathbf{S}}_2, \ldots, \alpha^{\mathbf{S}}_n$. Now the communication by **S** during **Phase I** and the computation by **R** at the end of **Phase I** is formally expressed in Table 9.

$k^{\mathbf{S}}_{1,1}$	$k^{\mathbf{S}}_{1,2}$	$\ldots$	$k^{\mathbf{S}}_{1,i}$	$\ldots$	$k^{\mathbf{S}}_{1,c}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$k^{\mathbf{S}}_{j,1}$	$k^{\mathbf{S}}_{j,2}$	$\ldots$	$k^{\mathbf{S}}_{j,i}$	$\ldots$	$k^{\mathbf{S}}_{j,c}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$k^{\mathbf{S}}_{t+1,1}$	$k^{\mathbf{S}}_{t+1,2}$	$\ldots$	$k^{\mathbf{S}}_{t+1,i}$	$\ldots$	$k^{\mathbf{S}}_{t+1,c}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$y^{\mathbf{S}}_{n,1}$	$y^{\mathbf{S}}_{n,2}$	$\ldots$	$y^{\mathbf{S}}_{n,i}$	$\ldots$	$y^{\mathbf{S}}_{n,c}$

Table 8: Matrix $B^{\mathbf{S}}$ as computed by **S**

Computation and Communication by S: Along wire $f_j, 1 \leq j \leq n$, **S** sends to **R** the polynomial $F^{\mathbf{S}}_j(x)$, the random value $\alpha^{\mathbf{S}}_j$ and n tuple $[v^{\mathbf{S}}_{1j}\ v^{\mathbf{S}}_{2j}\ \ldots\ v^{\mathbf{S}}_{nj}]$ where $v^{\mathbf{S}}_{ij} = F^{\mathbf{S}}_i(\alpha^{\mathbf{S}}_j), 1 \leq i \leq n$. Let $\mathcal{V}^{\mathbf{S}}$ denotes the concatenation of the elements in the first $t+1$ rows of $B^{\mathbf{S}}$. **S** computes $\mathcal{P}^{\mathbf{S}} = \mathbf{EXTRAND}_{|\mathcal{V}^{\mathbf{S}}|,(u+1)n^2}(\mathcal{V}^{\mathbf{S}})$. The vector $\mathcal{P}^{\mathbf{S}}$ denotes the information theoretically secure random pad of size $\Theta(n^2u)$ which will be correctly established with **R** with very high probability.

Computation by R at the end of Phase I:

1. Let **R** receives $F^{\mathbf{R}}_j(x)$, the random value $\alpha^{\mathbf{R}}_j$ and the n tuple $[v^{\mathbf{R}}_{1j}\ v^{\mathbf{R}}_{2j}\ \ldots\ v^{\mathbf{R}}_{nj}]$ along wire $f_j, 1 \leq j \leq n$.
2. For $1 \leq j \leq n$, **R** computes $Support_j = |\{i : F^{\mathbf{R}}_j(\alpha^{\mathbf{R}}_i) = v^{\mathbf{R}}_{ji}\}|$. If $Support_j \geq t+1$, then **R** concludes that $F^{\mathbf{R}}_j(x)$ is a valid polynomial. Otherwise, **R** concludes that $F^{\mathbf{R}}_j(x)$ is an invalid polynomial.
3. Since there are at least $t+1$ honest wires in the top band, **R** will get at least $t+1$ valid polynomials. Now using $t+1$ valid polynomials, **R** will construct array $B^{\mathbf{R}}$. From $B^{\mathbf{R}}$, **R** computes $\mathcal{V}^{\mathbf{R}}$, from which it finally computes $\mathcal{P}^{\mathbf{R}}$ and terminates. With very high probability, $\mathcal{P}^{\mathbf{R}} = \mathcal{P}^{\mathbf{S}}$ (see Lemma 3).

Table 9: Protocol Π

Theorem 3 *If the entire bottom band is corrupted, then protocol* Π *securely establishes a random non-zero pad of size* $\Theta(n^2u\kappa)$ *bits by communicating* $O(n^3\kappa)$ *bits.*

Proof: Since protocol Π is similar to the **Phase I** of protocol $\Pi^{existing}_{modified}$, using similar arguments as in Lemma 1, if $F^{\mathbf{R}}_j(x)$ is a valid polynomial, then with overwhelming probability $F^{\mathbf{R}}_j(x) = F^{\mathbf{S}}_j(x)$. Since, there are at least $t+1$ honest wires in the *top band*, the polynomials corresponding to these wires will always be considered as *valid*. So **R** will always get at least $t+1$ valid polynomials. Thus using similar arguments as in Lemma 1, **R** will be able to correctly recover $\mathcal{V}^{\mathbf{S}}$ and $\mathcal{P}^{\mathbf{S}}$ with very high probability. The secrecy of $\mathcal{P}^{\mathbf{S}}$ follows using similar argument as in Lemma 2 and the properties of **EXTRAND**. It is easy to see that $O(n^3)$ field elements and hence $O(n^3\kappa)$ bits are communicated by **S**. □

Six Phase Protocol Π^{Pad}: We now present the protocol Π^{Pad} which uses protocols Π and $\Pi^{Existing}_{modified}$ as black-box. The first two phases of the protocol are given in Table 10.

Before proceeding further, we prove the following claim.

Claim 2 *Let* b_j *and* f_i *be two honest wire in bottom and top band respectively. Then at the end of* **Phase II**, *at least* n^2 *elements in the tuple* $(y^{\mathbf{R}}_{1,j}, y^{\mathbf{R}}_{2,j}, \ldots, y^{\mathbf{R}}_{n^2+1,j})$ *and* $(x^{\mathbf{S}}_{1,i}, x^{\mathbf{S}}_{2,i}, \ldots, x^{\mathbf{S}}_{n^2+t,i})$ *are information theoretically secure.*

Proof: The proof follows from the properties of hash function. □

As in protocol $\Pi^{existing}_{modified}$, from the properties of hash function, it is straightforward to check that the following holds: (a) If f_i is an honest wire in the top band and $f_i \in \mathcal{F}_l$, then with very high probability, the random (n^2+t)-tuples that **R** has received along the wires in $\mathcal{F}_l$ are not modified; (b) If f_i is an honest wire in the top band and $f_i \in \mathcal{F}_l$, then $\mathcal{F}_l$ is an acceptable set.

Phase I: R to S: Corresponding to each wire $b_j, 1 \leq j \leq u$ in the bottom band, **R** selects a random non-zero $n^2 + 1$ tuple $(y^{\mathbf{R}}_{1,j}, y^{\mathbf{R}}_{2,j}, \ldots, y^{\mathbf{R}}_{n^2+1,j})$ and sends it to **S**.

Phase II: S to R:

1. Let **S** receives $(y^{\mathbf{S}}_{1,j}, y^{\mathbf{S}}_{2,j}, \ldots, y^{\mathbf{S}}_{n^2+1,j})$ along wire b_j. Corresponding to each wire $b_j, 1 \leq j \leq u$, **S** selects a random non-zero hash key r_j from $\mathbb{F}$ and computes the set $\beta^{\mathbf{S}} = \{(r^{\mathbf{S}}_j, \gamma^{\mathbf{S}}_i) : 1 \leq j \leq u\}$, where $\gamma^{\mathbf{S}}_j = hash(r^{\mathbf{S}}_j; y^{\mathbf{S}}_{1,j}, y^{\mathbf{S}}_{2,j}, \ldots, y^{\mathbf{S}}_{n^2+1,j})$.
2. **S** associates a random non-zero $n^2 + t$ tuple $(x^{\mathbf{S}}_{1,j}, x^{\mathbf{S}}_{2,j}, \ldots, x^{\mathbf{S}}_{n^2+t,j})$ with wire $f_j, 1 \leq j \leq n$ in the top band. Moreover, in order to hash the tuple, **S** selects n random non-zero keys from $\mathbb{F}$ denotes by $key^{\mathbf{S}}_{i,j}$, for $1 \leq i \leq n$.
3. For each $1 \leq j \leq n$, **S** sends the set $\beta^{\mathbf{S}}$ and the $(n^2 + t)$ tuple $(x^{\mathbf{S}}_{1,j}, x^{\mathbf{S}}_{2,j}, \ldots, x^{\mathbf{S}}_{n^2+t,j}$ to **R** along wire f_j and the 2-tuple $(key^{\mathbf{S}}_{i,j}, \alpha^{\mathbf{S}}_{i,j})$ to **R** along wire $f_i, 1 \leq i \leq n$, where $\alpha^{\mathbf{S}}_{i,j} = hash(key^{\mathbf{S}}_{i,j}; x^{\mathbf{S}}_{1,j}, x^{\mathbf{S}}_{2,j}, \ldots, x^{\mathbf{S}}_{n^2+t,j})$.

Computation by R at the end of Phase II:

1. For each $1 \leq j \leq n$, **R** receives the set $\beta^{\mathbf{R}}_j$ and the $(n^2 + t)$ tuple $(x^{\mathbf{R}}_{1,j}, x^{\mathbf{R}}_{2,j}, \ldots, x^{\mathbf{R}}_{n^2+t,j})$ along wire f_j and the 2-tuple $(key^{\mathbf{R}}_{i,j}, \alpha^{\mathbf{R}}_{i,j})$ along wire $f_i, 1 \leq i \leq n$.
2. **R** divides the top band $\{f_1, f_2, \ldots, f_n\}$ into subsets $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_k$, where $k \leq t + 1$, such that for any l, m, p with $1 \leq l \leq k, 1 \leq m, p \leq n$ and $f_m, f_p \in \mathcal{F}_l$, we have: (a) $\beta^{\mathbf{R}}_m = \beta^{\mathbf{R}}_p$; (b) $\alpha^{\mathbf{R}}_{m,p} = hash(key^{\mathbf{R}}_{m,p}; x^{\mathbf{R}}_{1,p}, x^{\mathbf{R}}_{2,p}, \ldots, x^{\mathbf{R}}_{n^2+t,p})$; (c) $\alpha^{\mathbf{R}}_{p,m} = hash(key^{\mathbf{R}}_{p,m}; x^{\mathbf{R}}_{1,m}, x^{\mathbf{R}}_{2,m}, \ldots, x^{\mathbf{R}}_{n^2+t,m})$.
3. For $\mathcal{F}_l$, let $f_m \in \mathcal{F}_l$ and $\beta^{\mathbf{R}}_m = \{(r^{\mathbf{R}}_{j,l}, \gamma^{\mathbf{R}}_{j,l}) : 1 \leq j \leq u\}$. **R** computes the set

$$\mathcal{B}_l = \{j : \gamma^{\mathbf{R}}_{j,l} = hash(r^{\mathbf{R}}_{j,l}; y^{\mathbf{R}}_{1,j}, y^{\mathbf{R}}_{2,j}, \ldots, y^{\mathbf{S}}_{n^2+1,j}), 1 \leq j \leq u\}$$

If $|\mathcal{F}_l| + |\mathcal{B}_l| \leq t$ then **S** decides that $\mathcal{F}_l$ is unacceptable set, otherwise $\mathcal{F}_l$ is acceptable set.

Table 10: First two phases of Protocol Π^{Pad}

In the worst case, in **R**'s view, there can be at most $t + 1$ acceptable sets because the adversary can control at most t wires in the top band. So there can be t acceptable sets, corresponding to t corrupted wires and one acceptable set corresponding to all the honest wires in the top band. The **Phase III** of the protocol is shown in Table 11.

Theorem 4 *If the entire bottom band is corrupted then Π^{Pad} securely establishes a random non-zero pad of size $\Theta(n^2\kappa)$ bits between **S** and **R** with very high probability. Otherwise, it establishes a random non-zero pad of size $\Theta(n^3\kappa)$ bits between **S** and **R** with very high probability. In either case, the protocol terminates in six phases and communicates $O(n^3\kappa)$ bits.*

Proof: Follows from the protocol description and properties of hash and **EXTRAND**. □

4 URMT with Constant Factor Overhead

Let $u \leq t$ and $n = \max(2t - u + 1, t + 1)$. Then we present an URMT protocol called Π^{URMT} which sends a message $m^{\mathbf{S}}$ containing ℓ field elements by communicating $O(\ell)$ field elements with very high probability, where $\ell = (t - \frac{u}{2} + 1)n^2 = \Theta(n^3)$. The total communication complexity of the protocol is $O(n^3)$ field elements and the protocol terminates in $O(u)$ phases. The principle behind the protocol is to create a win-win situation as follows: if the adversary corrupts at most $t - \frac{u}{2}$ wires in the top band, then **R** recovers the message from the information which it receives from the honest wires in the top band. On the other hand, if more than $t - \frac{u}{2}$ wires are corrupted in the top band, then majority wires in the bottom band will be honest and so both **S** and **R** comes to know about the identity of corrupted wires in the top band by using the honest wires in the bottom band. Now using this information, **S** can re-send $m^{\mathbf{S}}$ so that **R** can recover it correctly.

As a part of pre-processing step, **S** and **R** securely establishes $\Theta(n)$ random non-zero elements from $\mathbb{F}$ with each other in advance with very high probability by executing the three phase protocol $\Pi^{existing}_{modified}$. Let the set of these elements be denoted by $\mathcal{K}$. The elements in $\mathcal{K}$ will be used by **S** and **R** as authentication and hash keys to reliably exchange the outcome of certain steps during the execution of the protocol Π^{URMT}. Note that elements in $\mathcal{K}$ need not be distinct, but they are randomly selected from $\mathbb{F}$. We assume that initially all the elements in $\mathcal{K}$ are marked as "unused". Each time **S** (**R**)

<u>**Phase III: R to S**</u>: For each acceptable set $\mathcal{F}_l$ and corresponding set $\mathcal{B}_l$, **R** does the following:

1. **R** concatenates the first n^2 elements from (n^2+1) and (n^2+t) tuples, which it had sent and received over the wires in $\mathcal{B}_l$ and $\mathcal{F}_l$ respectively. Let $\mathcal{V}_l^{\mathbf{R}}$ denotes the resultant vector.
2. Corresponding to vector $\mathcal{V}_l^{\mathbf{R}}$, **R** selects a random non-zero hash key $\mathcal{K}_l^{\mathbf{R}}$ from $\mathbb{F}$. **R** then computes the 2-tuple $(\mathcal{K}_l^{\mathbf{R}}, \gamma_l^{\mathbf{R}} = hash(\mathcal{K}_l^{\mathbf{R}}; \mathcal{V}_l^{\mathbf{R}}))$. **R** then sends $\mathcal{B}_l, \mathcal{F}_l$ and the 2-tuple $(\mathcal{K}_l^{\mathbf{R}}, \gamma_l^{\mathbf{R}})$ to **S** through all the wires in $\mathcal{B}_l$.

<u>**Computation by S at the end of Phase III**</u>: Now using the hash value(s) received from **R**, **S** tries to find whether there exists at least one uncorrupted wire in the bottom band. For this, **S** does the following:

1. Let **S** receives the index set $\mathcal{F}_{j,l}^{\mathbf{S}}$ and $\mathcal{B}_{j,l}^{\mathbf{S}}$ and the 2-tuple $(\mathcal{K}_{j,l}^{\mathbf{S}}, \gamma_{j,l}^{\mathbf{S}})$ along wire $b_j, 1 \leq j \leq u$ for $1 \leq l \leq t+1$. If for some $j \leq u$ and some $l \leq t+1$, $|\mathcal{F}_{j,l}^{\mathbf{S}}| + |\mathcal{B}_{j,l}^{\mathbf{S}}| \leq t$, then **S** concludes that wire b_j is corrupted and neglects all the values received along b_j.
2. If $\mathcal{F}_{j,l}^{\mathbf{S}}$, $\mathcal{B}_{j,l}^{\mathbf{S}}$ and the tuple $(\mathcal{K}_{j,l}^{\mathbf{S}}, \gamma_{j,l}^{\mathbf{S}})$ is not neglected in the previous step (i.e., b_j is not discarded), then after knowing the index of the wires in $\mathcal{F}_{j,l}^{\mathbf{S}}$ and $\mathcal{B}_{j,l}^{\mathbf{S}}$, **S** computes his version of the vector $\mathcal{V}_{j,l}^{\mathbf{S}}$. Here $\mathcal{V}_{j,l}^{\mathbf{S}}$ denotes the concatenation of first n^2 values from the (n^2+1) and (n^2+t) tuples, which **S** had received and sent over the wires in $\mathcal{B}_{j,l}^{\mathbf{S}}$ and $\mathcal{F}_{j,l}^{\mathbf{S}}$ respectively. **S** now checks $\gamma_{j,l}^{\mathbf{S}} \overset{?}{=} hash(\mathcal{K}_{j,l}^{\mathbf{S}}; \mathcal{V}_{j,l}^{\mathbf{S}})$.
3. If the test in the last step succeeds for some $l \leq t+1$ and $j \leq u$, then **S** concludes that the tuples that are exchanged along the wires in $\mathcal{B}_{j,l}^{\mathbf{S}}$ and $\mathcal{F}_{j,l}^{\mathbf{S}}$ are correctly established between **S** and **R**. **S** now applies **EXTRAND** to $\mathcal{V}_{j,l}^{\mathbf{S}}$ to generate a vector $\mathcal{P}_1^{\mathbf{S}}$ of size tn^2. Finally **S** terminates the protocol by sending a special predefined "success" value from $\mathbb{F}$, along with the index of the wires in the set $\mathcal{B}_{j,l}^{\mathbf{S}}$ and $\mathcal{F}_{j,l}^{\mathbf{S}}$ to **R** by executing the protocol $\Pi_{modified}^{existing}$. **R** securely (and hence correctly) receives these indexes with very high probability and computes his version of $\mathcal{P}_1^{\mathbf{R}}$ and terminates. Since $\Pi_{modified}^{existing}$ takes three phases, the protocol will terminate at the end of **Phase VI**.
4. If the test in step 3 fails for all l and j, then **S** concludes that entire bottom band is corrupted. In this case, **S** sends a special "failure" value from $\mathbb{F}$ to **R** by executing the three phase $\Pi_{modified}^{existing}$ protocol. Parallely, **S** establishes a secure pad $\mathcal{P}_2^{\mathbf{S}}$ of size $\Theta(n^2u)$ with **R** by executing single phase Protocol Π. At the end of $\Pi_{modified}^{existing}$, **R** will know that the entire bottom band is corrupted. Parallely at the end of Π, **R** will output $\mathcal{P}_2^{\mathbf{R}}$, with which very high probability is same as $\mathcal{P}_2^{\mathbf{S}}$. Since $\Pi_{modified}^{existing}$ takes three phases, the protocol will terminate at the end of **Phase VI**.

Table 11: **Phase III** in Protocol Π^{Pad}

needs a key(s) for hashing or authentication, then the first "unused" element(s) from $\mathcal{K}$ is/are selected as key(s). In order to do the verification, **R** (**S**) also uses the same element(s) from $\mathcal{K}$ as keys. Once the verification is done, the element(s) is/are marked as "used" Thus we can view $\mathcal{K}$ as a global set, which is parallely used and updated by both **S** and **R**.

Let $m^{\mathbf{S}} = [m_{1,1}^{\mathbf{S}}\ m_{1,2}^{\mathbf{S}}\ \ldots\ m_{1,n^2}^{\mathbf{S}}\ m_{2,1}^{\mathbf{S}}\ m_{2,2}^{\mathbf{S}}\ \ldots\ m_{2,n^2}^{\mathbf{S}}\ \ldots\ m_{t-\frac{u}{2}+1,1}^{\mathbf{S}}\ m_{t-\frac{u}{2}+1,2}^{\mathbf{S}}\ \ldots\ m_{t-\frac{u}{2}+1,n^2}^{\mathbf{S}}]$ be the message. **S** constructs an array $B^{\mathbf{S}}$ of size $n \times n^2$ from $m^{\mathbf{S}}$ in the same way as in protocol Π with the following modifications: **S** first constructs the array $A^{\mathbf{S}}$ of size $(t - \frac{u}{2} + 1) \times n^2$ from $m^{\mathbf{S}}$, where the $j^{th}, 1 \leq j \leq (t - \frac{u}{2} + 1)$ row of $A^{\mathbf{S}}$ is $[m_{j,1}^{\mathbf{S}}\ m_{j,2}^{\mathbf{S}}\ \ldots\ m_{j,n^2}^{\mathbf{S}}]$. By considering the elements in individual columns as distinct points, **S** interpolates the unique $(t - \frac{u}{2})$ degree polynomial passing through them. **S** then further evaluates the interpolated polynomials at additional $(t - \frac{u}{2})$ values of x and gets the array $B^{\mathbf{S}}$. Now by considering the elements along $j^{th}, 1 \leq j \leq n$ row of $B^{\mathbf{S}}$ as coefficients, **S** constructs $F_j^{\mathbf{S}}(x)$ of degree $n^2 - 1$. First two phases of Π^{URMT} is shown in Table 12.

<u>**Phase I: S to R**</u>: Along wire $f_j, 1 \leq j \leq n$, **S** sends to **R** the polynomial $F_j^{\mathbf{S}}(x)$, a random non-zero value $\alpha_j^{\mathbf{S}}$ and n tuple $[v_{1j}^{\mathbf{S}}\ v_{2j}^{\mathbf{S}}\ \ldots\ v_{nj}^{\mathbf{S}}]$ where $v_{ij}^{\mathbf{S}} = F_i^{\mathbf{S}}(\alpha_j^{\mathbf{S}}), 1 \leq i \leq n$.

<u>**Phase II: R to S**</u>

1. Let **R** receives $F_j^{\mathbf{R}}(x)$, the value $\alpha_j^{\mathbf{R}}$ and the n tuple $[v_{1j}^{\mathbf{R}}\ v_{2j}^{\mathbf{R}}\ \ldots\ v_{nj}^{\mathbf{R}}]$ along wire $f_j, 1 \leq j \leq n$.
2. For $1 \leq j \leq n$, **R** computes $Support_j = |\{i : F_j^{\mathbf{R}}(\alpha_i^{\mathbf{R}}) = v_{ji}^{\mathbf{R}}\}|$. Let $\mathcal{P}^{\mathbf{R}}$ denotes the set of wires f_j, such that $Support_j \geq (t - \frac{u}{2} + 1)$. In addition, **R** constructs a directed graph $\mathcal{G}^{\mathbf{R}} = (\mathcal{V}^{\mathbf{R}}, \mathcal{E}^{\mathbf{R}})$, called *conflict graph*, where $\mathcal{V}^{\mathbf{R}} = \{f_1, f_2, \ldots, f_n\}$ and arc $(f_i, f_j) \in \mathcal{E}^{\mathbf{R}}$ if $F_i^{\mathbf{R}}(\alpha_j^{\mathbf{R}}) \neq v_{ij}^{\mathbf{R}}$.
3. Corresponding to graph $\mathcal{G}^{\mathbf{R}}$, **R** constructs a conflict list $\mathcal{Y}^{\mathbf{R}}$ of five tuples where for each arc $(f_i, f_j) \in \mathcal{E}^{\mathbf{R}}$, there exists a five tuple $(f_i, f_j, \alpha_j^{\mathbf{R}}, F_i^{\mathbf{R}}(\alpha_j^{\mathbf{R}}), v_{ij}^{\mathbf{R}})$ in $\mathcal{Y}^{\mathbf{R}}$. **R** sends $\mathcal{Y}^{\mathbf{R}}$ to **S** through bottom band.

Table 12: First two phases of protocol Π^{URMT}

Before proceeding further, we prove the following claim.

Claim 3 *Let f_i be a wire which has delivered incorrect $F_i^{\mathbf{R}}(x) \neq F_i^{\mathbf{S}}(x)$ to* **R** *and f_j be an honest wire. Then with very high probability $(f_i, f_j) \in \mathcal{E}^{\mathbf{R}}$.*

PROOF: Since f_j is honest, it correctly delivers $\alpha_j^{\mathbf{R}} = \alpha_j^{\mathbf{S}}$ and $v_{ij}^{\mathbf{R}} = v_{ij}^{\mathbf{S}} = F_i^{\mathbf{S}}(\alpha_j^{\mathbf{S}})$ to **R**. If $F_i^{\mathbf{R}}(x) \neq F_i^{\mathbf{S}}(x)$, then in order that $(f_i, f_j) \notin \mathcal{E}^{\mathbf{R}}$, $F_i^{\mathbf{R}}(\alpha_j^{\mathbf{R}}) = v_{ij}^{\mathbf{R}} = F_i^{\mathbf{S}}(\alpha_j^{\mathbf{R}})$. But this can happen with probability at most $\frac{n^2-1}{|\mathbb{F}|}$ because $\alpha_j^{\mathbf{S}}$ is randomly selected from $\mathbb{F}$ and $F_i^{\mathbf{R}}(x) \neq F_i^{\mathbf{S}}(x)$ can have same value at atmost n^2-1 such α's as both are of degree n^2-1. Since $|\mathbb{F}| = \text{poly}(\kappa)2^{\kappa}$, this probability is negligible. □

Now **S** considers the conflict list which it receives identically through at least $\frac{u}{2}+1$ wires. If **S** does not receives any conflict list identically through at least $\frac{u}{2}+1$ wires, then **S** concludes that at least $\frac{u}{2}+1$ wires are corrupted in the bottom band, which further implies that at most $t-\frac{u}{2}-1$ wires are corrupted in the top band. In this case, the protocol proceeds as shown in Table 13.

Phase III: S to R: By selecting two elements from $\mathcal{K}$ as authentication keys, **S** authenticates an unique special predetermined signal "terminate" and sends to **R**. **R** receives the signal correctly with very high probability and concludes that at most $t-\frac{u}{2}$ wires have delivered incorrect values during **Phase I**. So by using the polynomials received along the first $t-\frac{u}{2}+1$ wires in $\mathcal{P}^{\mathbf{R}}$ during **Phase I**, **R** constructs the array $B^{\mathbf{R}}$. From $B^{\mathbf{R}}$, **R** recovers $m^{\mathbf{R}}$ and terminates.

Table 13: Execution of Π^{URMT} if **S** does not receives $\frac{u}{2}+1$ identical conflict lists

The correctness of the protocol in this execution sequence is proved in Lemma 3.

Lemma 3 *If* **S** *does not receives the same conflict list through at least $\frac{u}{2}+1$ wires then with very high probability,* **R** *correctly recovers $m^{\mathbf{S}}$ from the polynomials delivered by the wires in $\mathcal{P}^{\mathbf{R}}$.*

PROOF: If **S** does not receive the same conflict list through at least $\frac{u}{2}+1$ wires then it implies that at least $\frac{u}{2}+1$ wires in the bottom band are corrupted which further implies that at most $t-\frac{u}{2}-1$ wires in the top band are corrupted. So, with very high probability, the wires in the set $\mathcal{P}^{\mathbf{R}}$ have correctly delivered the polynomials during **Phase I**. This is because if some wire f_j in the top band has delivered $F_j^{\mathbf{R}}(x) \neq F_j^{\mathbf{S}}(x)$ during **Phase I**, then f_j can be supported by at most $t-\frac{u}{2}-1$ wires in the top band (which are corrupted) and with very high probability, f_j will be contradicted by all the honest wires in the top band, implying $Support_j = t-\frac{u}{2}-1$ which further implies that $f_j \notin \mathcal{P}^{\mathbf{R}}$. Since there are at least $(2t-u+1)-(t-\frac{u}{2}-1) = t-\frac{u}{2}+2$ wires in the top band, these wires will always be in $\mathcal{P}^{\mathbf{R}}$. Now by using the polynomials received over the wires in $\mathcal{P}^{\mathbf{R}}$, **R** can correctly reconstruct the array $B^{\mathbf{S}}$ and hence $A^{\mathbf{S}}$. This is because, any $(t-\frac{u}{2}+1)$ correct polynomials are enough to reconstruct $B^{\mathbf{S}}$. □

If at the end of **Phase III**, **S** receives the same conflict list, say $\mathcal{Y}^{\mathbf{S}}$ through at least $\frac{u}{2}+1$ wires, then **S** does the following: let the five tuples in $\mathcal{Y}^{\mathbf{S}}$ be of the form $(f_i, f_j, \alpha_j'^{\mathbf{R}}, F_i'^{\mathbf{R}}(\alpha_j'^{\mathbf{R}}), v_{ij}'^{\mathbf{R}})$. For each such four tuple, **S** checks $\alpha_j'^{\mathbf{R}} \overset{?}{=} \alpha_j^{\mathbf{S}}$ and $v_{ij}^{\mathbf{S}} \overset{?}{=} v_{ij}'^{\mathbf{R}}$. If any of these test fails then **S** concludes that wire f_j has delivered incorrect values to **R** during **Phase I** and adds f_j to a list $L_{fault}^{\mathbf{S}}$. On the other hand, if both the test passes then **S** checks $F_i^{\mathbf{S}}(\alpha_j^{\mathbf{S}}) \overset{?}{=} F_i'^{\mathbf{R}}(\alpha_j'^{\mathbf{R}})$. If the test fails then **S** concludes that wire f_i has delivered incorrect $F_i'^{\mathbf{R}}(x) \neq F_i^{\mathbf{S}}(x)$ to **R** during **Phase I** and adds f_i to $L_{fault}^{\mathbf{S}}$. Note that **S** does not know whether $\mathcal{Y}^{\mathbf{S}}$ is a genuine conflict list and is indeed sent by **R**. But still **S** computes $L_{fault}^{\mathbf{S}}$.

S now finds the cardinality of list $L_{fault}^{\mathbf{S}}$. Now there are two possible cases. If $|L_{fault}^{\mathbf{S}}| \leq (t-\frac{u}{2})$, then **S** concludes that at least $t-\frac{u}{2}+1$ wires have delivered correct polynomial during **Phase I**. **S** then performs the same computation as shown in Table 13. The correctness of the protocol in this execution sequence is proved in Lemma 4.

Lemma 4 *If $|L_{fault}^{\mathbf{S}}| \leq (t-\frac{u}{2})$, then with very high probability,* **R** *can correctly recover $m^{\mathbf{S}}$ from the polynomials delivered by the wires in $\mathcal{P}^{\mathbf{R}}$.*

PROOF: If $\mathcal{Y}^{\mathbf{S}} \neq \mathcal{Y}^{\mathbf{R}}$, then the lemma follows using the same argument as in the proof of Lemma 3. This is because now at least $\frac{u}{2}+1$ wires in the bottom band are corrupted. We now consider the case when $\mathcal{Y}^{\mathbf{S}} = \mathcal{Y}^{\mathbf{R}}$. From Claim 3, if a corrupted wire f_i has delivered incorrect $F_i^{\mathbf{R}}(x) \neq F_i^{\mathbf{S}}(x)$ to **R** and f_j is an honest wire, then with very high probability $(f_i, f_j) \in \mathcal{E}^{\mathbf{R}}$. Correspondingly, there will be a 5-tuple present in $\mathcal{Y}^{\mathbf{R}}$ and hence in $\mathcal{Y}^{\mathbf{S}}$. From this 5-tuple, **S** will easily find out that **R** has received incorrect polynomial over f_j. Thus, if $\mathcal{Y}^{\mathbf{S}} = \mathcal{Y}^{\mathbf{R}}$, then **S** will come to know the exact identity of all the corrupted wires which delivered incorrect polynomials during **Phase I** and they will be present in list $L^{\mathbf{S}}_{fault}$. Since $|L^{\mathbf{S}}_{fault}| \leq (t - \frac{u}{2})$, this implies that at most $t - \frac{u}{2}$ polynomials were delivered incorrectly and hence at least $t - \frac{u}{2} + 1$ polynomials were delivered correctly which will be present in $\mathcal{P}^{\mathbf{R}}$. The rest of the proof now follows using similar argument as in Lemma 3. □

If $|L^{\mathbf{S}}_{fault}| \geq (t - \frac{u}{2} + 1)$, then **S** further communicates with **R** to find whether $\mathcal{Y}^{\mathbf{S}}$ was indeed sent by **R**. For this, **S** and **R** executes the steps as shown in Table 14.

Phase III: S to R: **S** selects $2|L^{\mathbf{S}}_{fault}|$ elements from the set $\mathcal{K}$ as authentications keys and using them authenticates each element of $L^{\mathbf{S}}_{fault}$ by using *URauth* function. Let $L^{\mathbf{S}}_{fault_{auth}}$ denotes the set of corresponding authenticated values. **S** then sends $(\mathcal{Y}^{\mathbf{S}}, L^{\mathbf{S}}_{fault}, L^{\mathbf{S}}_{fault_{auth}})$ to **R** through top band.

Phase IV: R to S: Let **R** receives $(\mathcal{Y}^{\mathbf{R}}_j, L^{\mathbf{R}}_{fault_j}, L^{\mathbf{R}}_{fault_{j,auth}})$ from **S** along wire $f_j, 1 \leq j \leq n$. From these values, **R** now tries to find out whether **S** has correctly received the original $\mathcal{Y}^{\mathbf{R}}$ over more that $\frac{u}{2} + 1$ wires during **Phase I**, and if yes, then the corresponding $L^{\mathbf{S}}_{fault}$. For this, **R** does the following:

1. For each $1 \leq j \leq n$, **R** checks $\mathcal{Y}^{\mathbf{R}}_j \stackrel{?}{=} \mathcal{Y}^{\mathbf{R}}$ and $|L^{\mathbf{R}}_{fault_j}| \geq (t - \frac{u}{2} + 1)$. In any of the test fails, then **R** neglects all the values received along f_j. Otherwise, **R** applies the *URauth* function to each element of $L^{\mathbf{R}}_{fault_j}$ by using the same keys from $\mathcal{K}$, which were used by **S** to authenticate $L^{\mathbf{S}}_{fault}$ and computes the set $L'^{\mathbf{R}}_{fault_{j,auth}}$. **R** then checks $L'^{\mathbf{R}}_{fault_{j,auth}} \stackrel{?}{=} L^{\mathbf{R}}_{fault_{j,auth}}$. If the test fails then again **R** discards the values received along f_j.
2. If as a result of previous step, **R** has discarded the values along all the wires in the top band, then **R** concludes that **S** has not received original $\mathcal{Y}^{\mathbf{R}}$ over more that $\frac{u}{2} + 1$ wires during **Phase I**, which further implies that at most $t - \frac{u}{2} - 1$ wires were corrupted in the top band during **Phase I**. So **R** recovers $m^{\mathbf{R}}$ by using the polynomials received over the first $t - \frac{u}{2} + 1$ wires in $\mathcal{P}^{\mathbf{R}}$ during **Phase I**. Moreover, by selecting next two "unused" elements k_1, k_2 from $\mathcal{K}$ as authentication keys, **R** computes $response_1 = URauth(\text{"}terminate\text{"}; k_1, k_2)$ where "terminate" is an unique pre-defined special element from $\mathbb{F}$. **R** then send the tuple $(\text{"}terminate\text{"}, response_1)$ to **S** through the bottom band and terminates.
3. If during step 1, there exists a $j \in \{1, 2, \ldots, n\}$ such that $\mathcal{Y}^{\mathbf{R}}_j = \mathcal{Y}^{\mathbf{R}}$, $|L^{\mathbf{R}}_{fault_j}| \geq (t - \frac{u}{2} + 1)$ and $L'^{\mathbf{R}}_{fault_{j,auth}} = L^{\mathbf{R}}_{fault_{j,auth}}$, then **R** concludes that **S** has correctly received original $\mathcal{Y}^{\mathbf{R}}$ over more that $\frac{u}{2} + 1$ wires during **Phase I** and $L^{\mathbf{R}}_{fault_j}$ is the corresponding L_{fault} sent by **S**. So **R** removes the wires in $L^{\mathbf{R}}_{fault_j}$ from his view for further computation and communication. Note that if there are more than one such j (whose probability is negligible), then **R** arbitrarily selects one. Now by selecting k_1, k_2 from $\mathcal{K}$ as authentication keys, **R** computes $response_2 = URauth(\text{"}continue\text{"}; k_1, k_2)$ where "continue" is an unique pre-defined special element from $\mathbb{F}$. **R** then send the tuple $(\text{"}continue\text{"}, response_2)$ to **S** through the bottom band.

Computation by S at the end of Phase IV: **S** checks whether it is getting any 2-tuple identically over at least $\frac{u}{2} + 1$ wires. If not, then **S** concludes that **R** has recovered $m^{\mathbf{R}}$ and terminates. On the other hand, if **S** receives a 2-tuple say $(x^{\mathbf{S}}_1, y^{\mathbf{S}}_1)$ over $\frac{u}{2} + 1$ wires, then **S** verifies $y^{\mathbf{S}}_1 \stackrel{?}{=} URauth(x^{\mathbf{S}}_1; k_1, k_2)$. If the test fails, then **S** again concludes that **R** has recovered $m^{\mathbf{R}}$ and terminates. On the other hand, if the test succeeds then **S** further checks $x^{\mathbf{S}}_1 \stackrel{?}{=} \text{"}terminate\text{"}$. If yes, then **S** again concludes that **R** has recovered $m^{\mathbf{R}}$ and terminates. If no then **S** concludes that $\mathcal{Y}^{\mathbf{S}}$ was indeed sent by **R**.

Table 14: Execution of Π^{URMT} if $|L^{\mathbf{S}}_{fault}| \geq (t - \frac{u}{2} + 1)$

Before proceeding further, we prove the following lemma.

Lemma 5 *If* $|L^{\mathbf{S}}_{fault}| \geq (t - \frac{u}{2} + 1)$*, then at the end of* **Phase III** *in Table 14 one of the following will happen:*

1. *If* **S** *has "not" received the original* $\mathcal{Y}^{\mathbf{R}}$ *over more that* $\frac{u}{2} + 1$ *wires during* **Phase I***, then with very high probability* **R** *will be able to detect this. Moreover* **R** *will be able to correctly recover* $m^{\mathbf{R}}$ *by using the polynomials received over the wires in* $\mathcal{P}^{\mathbf{R}}$ *with very high probability.*

2. *If* **S** *has received the original* $\mathcal{Y}^{\mathbf{R}}$ *over more that* $\frac{u}{2} + 1$ *wires during* **Phase I***, then* **R** *will be able to detect this. Moreover, with very high probability,* **R** *will correctly receive* $L^{\mathbf{S}}_{fault}$*, from which it will come to know the identity of at least* $|L^{\mathbf{S}}_{fault}|$ *corrupted wires in the top band.*

Proof: Follows from the protocol description and properties of URauth function. □

If at the end of **Phase IV** in Table 14, **S** recovers "continue" signal from **R** then **S** removes the wires in $L^{\mathbf{S}}_{fault}$ from further computation and communication. **S** now knows that in both **S** and **R**'s view, there are $n - |L^{\mathbf{S}}_{fault}|$ wires in the top band, of which at most $t - |L^{\mathbf{S}}_{fault}|$ could be corrupted. Since $|L^{\mathbf{S}}_{fault}| \geq (t - \frac{u}{2} + 1)$, in **S** and **R**'s view, there are at most $t - \frac{u}{2}$ wires in the top band, of which at most $\frac{u}{2} - 1$ could be corrupted. Moreover, both **S** and **R** now knows that there exists at least $\frac{u}{2} + 1$ honest wires in the bottom band. **S** now proceeds to re-send $m^{\mathbf{S}}$. For this, out of the $t - \frac{u}{2}$ in their view, both **S** and **R** considers only the first $\frac{u}{2}$ wires. Without loss of generality, let these be the wires $f_1, f_2, \ldots, f_{\frac{u}{2}}$. Now both **S** and **R** knows that at least one wire among these $\frac{u}{2}$ wires is honest. **S** now re-sends $m^{\mathbf{S}}$ by executing the steps given in Table 15. This will take $\Theta(u)$ phases.

S divides $m^{\mathbf{S}}$ into blocks $B^{\mathbf{S}}_1, B^{\mathbf{S}}_2, \ldots, B^{\mathbf{S}}_{\frac{u}{2}}$, each of size $\frac{|m^{\mathbf{S}}|}{\frac{u}{2}}$. Moreover **S** and **R** initializes variables $wc^{\mathbf{S}} = 1, bc^{\mathbf{S}} = 1$ and $wc^{\mathbf{R}} = 1, bc^{\mathbf{R}} = 1$ respectively. **S** and **R** now executes the following steps:

1. While ($wc^{\mathbf{S}} \leq \frac{u}{2} - 1$) and (all the blocks of $m^{\mathbf{S}}$ are not sent) do

   (a) **S** sends the block $B^{\mathbf{S}}_{bc^{\mathbf{S}}}$ to **R** *only* over wire $f_{wc^{\mathbf{S}}}$ in the top band.

   (b) Let **R** receives $B^{\mathbf{R}}_{bc^{\mathbf{R}}}$ along wire $f_{wc^{\mathbf{R}}}$. Now by selecting k_{bc} from the set $\mathcal{K}$ as hash key, **R** computes $x^{\mathbf{R}}_{bc} = hash(k_{bc}; B^{\mathbf{R}}_{bc^{\mathbf{R}}})$ and sends $x^{\mathbf{R}}_{bc}$ to **S** through the bottom band.

   (c) **S** correctly receives $x^{\mathbf{R}}_{bc}$ through at least $\frac{u}{2} + 1$ wires (recall that in this case majority wires in bottom band are honest) and verifies $x^{\mathbf{R}}_{bc} \overset{?}{=} hash(k_{bc}; B^{\mathbf{S}}_{bc^{\mathbf{S}}})$. If the test fails then **S** concludes that wire $f_{wc^{\mathbf{S}}}$ has delivered incorrect $B^{\mathbf{S}}_{bc^{\mathbf{S}}}$ to **R**. So **S** increments $wc^{\mathbf{S}}$ by one. Moreover, **S** authenticates an unique pre-defined special "increment-wire" element from $\mathbb{F}$ by using two keys from the set $\mathcal{K}$ and sends it to **R** through the top band. **R** correctly receives the signal with very high probability and accordingly increments $wc^{\mathbf{R}}$ by one.

   On the other hand, if the test succeeds then **S** concludes that wire $f_{wc^{\mathbf{S}}}$ has delivered correct $B^{\mathbf{S}}_{bc^{\mathbf{S}}}$ to **R**. So **S** increments $bc^{\mathbf{S}}$ by one. Moreover, **S** authenticates an unique pre-defined special "increment-block" value from $\mathbb{F}$ by using two keys from the set $\mathcal{K}$ and sends it to **R** through the top band. **R** correctly receives the signal with very high probability and accordingly increments $bc^{\mathbf{R}}$ by one.

2. If all the blocks of $m^{\mathbf{S}}$ are sent then both **S** and **R** terminates. Otherwise **S** concatenates all the remaining blocks of $m^{\mathbf{S}}$ and sends to **R** through wire $f_{\frac{u}{2}}$ and terminates. **R** correctly receives these blocks and terminates.

Table 15: Execution of Π^{URMT} to re-send $m^{\mathbf{S}}$

Lemma 6 *If the original conflict list $\mathcal{Y}^{\mathbf{R}}$ is correctly received by* **S** *over more than $\frac{u}{2} + 1$ wires during* **Phase II** *and if the corresponding $|L^{\mathbf{S}}_{fault}| \geq (t - \frac{u}{2} + 1)$, then with very high probability,* **S** *will be able to correctly re-send $m^{\mathbf{S}}$ by executing the steps in Table 15. Moreover this requires a communication overhead of $O(|m^{\mathbf{S}}|)$ field elements.*

PROOF: Follows from the protocol description and the properties of hash function. □

We thus have the following theorem:

Theorem 5 *If $m^{\mathbf{S}}$ is a message containing ℓ field elements where $\ell \geq (t - \frac{u}{2} + 1)n^2$, then there exists an $O(u)$ phase URMT protocol which reliably sends $m^{\mathbf{S}}$ with very high probability by communicating $O(\ell)$ field elements. In terms of bits, the protocol sends $\ell\kappa$ bits by communicating $O(\ell\kappa)$ bits.*

Remark 1 *In protocol Π^{URMT}, we have assumed that $u \leq t$. If $u > t$, then we can modify the protocol to reliably send a message containing $(\frac{u}{2} + 1)n^2 = \Theta(n^3)$ field elements with a communication overhead of $O(n^3)$ field elements.*

5 Upper Bound on the Communication Complexity of USMT

We now design an $O(u)$ phase USMT protocol called Π^{USMT}, which sends a message $\mathbf{M}^{\mathbf{S}}$ containing ℓ field elements by communicating $O(n^3)$ field elements with very high probability. If the full bottom band is corrupted then $\ell = \Theta(n^2 u)$, otherwise $\ell = \Theta(n^3)$. The protocol uses Π^{Pad} and Π^{URMT} as black-box. The protocol is given in Table 16.

1. Depending upon whether the full bottom band is corrupted or not, **S** and **R** securely establishes a random non-zero one time pad Pad of length $\Theta(n^2u)$ or $\Theta(n^3)$ with very high probability by executing the protocol Π^{Pad}.
2. If Pad is of length $\Theta(n^2u)$, then **S** selects a secret message $\mathbf{M^S}$ of length $\Theta(n^2u)$. **S** then computes $C = \mathbf{M^S} \oplus Pad$ and reliably sends C to **R** with very high probability by executing the protocol Π^{URMT}. **R** correctly receives C with very high probability and recovers $\mathbf{M^R} = C \oplus Pad$. On the other hand, if Pad is of length $\Theta(n^3)$, then **S** and **R** does the same computation, except that $\mathbf{M^S}$ and C (and hence $\mathbf{M^R}$) will be of length $\Theta(n^3)$.

Table 16: An $O(u)$ phase USMT protocol Π^{USMT}

Theorem 6 *Protocol Π^{USMT} is an $O(u)$ phase USMT protocol with a communication complexity of $O(n^3)$ field elements. In terms of bits, the protocol sends either $\Theta(n^2u\kappa)$ or $\Theta(n^3\kappa)$ bits by communicating $O(n^3\kappa)$ bits.*

6 Lower Bound on the Communication Complexity of USMT

An obvious lower bound on the communication complexity of URMT protocols to send a message containing ℓ field elements is $\Omega(\ell)$. Since, we have already shown that this bound is tight by designing the URMT protocol Π^{URMT}, we need not have to prove the lower bound for URMT protocols. Similarly, if at least one wire in the bottom band is uncorrupted, then $\Omega(\ell)$ is a trivial lower bound on the communication complexity of any USMT protocol which securely sends ℓ field elements. Again, since we have already shown that this bound is tight by designing protocol Π^{USMT} (which securely sends ℓ field elements by communicating ℓ field elements if there exists at least one uncorrupted wire in the bottom band), we need not have to prove the lower bound for this case. We now prove the lower bound on the communication complexity of USMT protocols where the entire bottom band is corrupted.

Theorem 7 *Suppose there exists $u \leq t$ wires in the bottom band and $n = max(2t - u + 1, t + 1)$ wires in the top band. Moreover, the entire bottom band is corrupted. Then any multiphase USMT protocol to send a message $M^{\mathbf{S}}$ containing ℓ field elements from $\mathbb{F}$, needs to communicate $\Omega(\frac{n\ell}{u})$ field elements. In terms of bits, the protocol needs to communicate $\Omega(\frac{n\ell}{u}\kappa)$ bits to send $\ell\kappa$ bits.*

PROOF: Note that if $u > t$, then at least one wire in the bottom band is uncorrupted and so the lower bound of $\Omega(\ell)$ holds. So, we consider the case where $u \leq t$. Suppose both **S** and **R** in advance knows that the entire bottom band is corrupted. Under this assumption, any multiphase USMT protocol virtually reduces to a single phase USMT protocol, where **S** is connected to **R** by $n = 2t - u + 1$ wires, of which at most $t - u$ are corrupted. Since perfect secrecy is required in USMT, the data sent along the n wires in any single phase USMT protocol must be such that data on any set of $(t - u)$ wires has no information about the secret message, otherwise the adversary will also know the secret message by passively listening the contents of these wires. Similarly, the data sent over any $(n - (t - u))$ honest wires during the protocol has full information about the secret message. The latter requirement ensures that even if the adversary simply blocks/corrupts all the data that he can, the secret message is not lost and therefore the receiver's ability to recover the message is not completely ruled out.

Let X_i denotes the i^{th} share of some valid distribution scheme and let m denote the secret message containing ℓ field elements chosen from $\mathbb{F}^\ell$. For any subset $A \subseteq \{1, 2 \ldots n\}$ let X_A denote the set of variables $\{X_i | i \in A\}$. Then the secret m and the shares X_i are random variables. For a random variable X, let $H(X)$ denote its entropy [4]. Roughly speaking, entropy quantifies the information contained in a message, usually in bits or symbols. Since m is drawn uniformly at random from $\mathbb{F}^\ell$, we have $H(m) = \ell$. Since in any single phase USMT protocol, the data sent along any set B consisting of $(n - (t - u))$ honest wires have full information about m, we have $H(m|X_B) = 0$.

Consider any subset $A \subset B$ such that $|A| = (t - u)$. Since the data sent along the wires in A is insufficient to retrieve any information about the message m we get $H(m|X_A) = H(m)$. From the chain rule of the entropy [4], for any two random variable X_1, X_2, we have $H(X_1, X_2) = H(X_2) + H(X_1|X_2)$. Here $H(X_1, X_2)$ denotes the joint entropy of X_1, X_2. Informally, the joint entropy measures how much entropy is contained in a joint system of two random variables. Similarly, $H(X_1|X_2)$ denotes conditional entropy of X_1 on X_2. Informally, it quantifies the remaining entropy (i.e. uncertainty) of X_1 given that the value of a second random variable X_2 is known. Substituting $X_1 = m|X_A$ and $X_2 = X_{B-A}$, we get $H(m|X_A, X_{B-A}) = H(X_{B-A}) + H(m|X_A|X_{B-A})$. From the properties of joint entropy [4], for any two variables X_1, X_2, we have $H(X_1, X_2) \geq H(X_1)$ and $H(X_1, X_2) \geq H(X_2)$. Thus, $H(m|X_A, X_{B-A}) \geq H(m|X_A)$. Thus we get

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

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

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

Summing the above equation over all possible sets of size $n - 2(t-u)$ we get

$$\sum_C \sum_{i \in C} H(X_i) \geq \binom{n}{n-2(t-u)} H(m)$$

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

$$\begin{aligned} \binom{n-1}{n-2(t-u)-1} \sum_{i=1}^{n} H(X_i) &\geq \binom{n}{n-2(t-u)} H(m.) \\ \text{Thus } \sum_{i=1}^{n} H(X_i) &\geq \frac{n}{n-2(t-u)} \ell \text{.since } H(m) = \ell \end{aligned}$$

Since $\sum_{i=1}^{n} H(X_i)$ defines the information content over n wires, which is sent during any single phase USMT protocol, the lower bound on the communication complexity of any single phase USMT protocol is $\Omega\left(\frac{n\ell}{n-2(t-u)}\right) = \Omega\left(\frac{n\ell}{u}\right)$. This completes the theorem. □

The lower bound proved in Theorem 7 is tight. Specifically, if the entire bottom band is corrupted, then the USMT protocol Π^{USMT} sends ℓ field elements by communicating $O(\frac{n\ell}{u})$ field elements where $\ell = \Theta(n^2 u)$.

7 Conclusion and Open Problems

In this paper we have proved the lower bound on the communication complexity of URMT and USMT protocols in directed networks. Moreover, we have shown that our bounds are tight by designing communication optimal URMT and USMT protocol, which are first of their kind. It would be interesting to reduce the phase complexity of our URMT and USMT protocols.

References

[1] Z. Beerliová-Trubíniová and M. Hirt. Efficient multi-party computation with dispute control. In *Proc. of TCC*, pages 305–328, 2006.

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

[3] D. Chaum, C. Crpeau, and I. Damgård. Multiparty unconditionally secure protocols (extended abstract). In *Proc. of FOCS 1988*, pages 11–19, 1988.

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

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

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

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

[8] Arpita Patra, Ashish Choudhary, Kannan Srinathan, and C. Pandu Rangan. Unconditionally reliable and secure message transmission in undirected synchronous networks: Possibility, feasibility and optimality. Cryptology ePrint Archive, Report 2008/141, 2008.

[9] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *STOC*, pages 73–85, 1989.

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

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

[12] Y. Wang and Y. Desmedt. Perfectly secure message transmission revisited. IEEE Transactions on Information Theory. Manuscript. Available at www.sis.uncc.edu/∼yonwang/.

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