Secure Message Transmission In Asynchronous Directed Networks

Shashank Agrawal* Abhinav Mehta* Kannan Srinathan*
{shashank.agrawal@research. abhinav_mehta@research. srinathan@}iiit.ac.in

Abstract

We study the problem of information-theoretically secure message transmission (SMT) in asynchronous directed networks. In line with the literature, the distrust and failures of the network is captured via a computationally unbounded Byzantine adversary that may corrupt some subset of nodes. We give a characterization of networks over which SMT from sender **S** to receiver **R** is possible in both the well-known settings, namely *perfect* SMT (PSMT) and *unconditional* SMT (USMT). We distinguish between two variants of USMT: one in which **R** can output an incorrect message (with small probability) and another in which **R** never outputs a wrong message, but may choose to abort (with small probability). We also provide efficient protocols for an important class of networks.

1 Introduction

When dealing with problems in secure distributed computing, it is widely assumed that every pair of participating nodes share a private channel between them. However, in practice, most of the nodes in a communication network are not directly connected to each other. In such a scenario, in the absence of a *physical* private channel between two communicating parties, we would like to simulate a *virtual* one. The study of perfectly secure message transmission(PSMT) and unconditionally secure message transmission(USMT) helps us achieve the fundamental primitive of secure message transmission between two nodes in a network. First introduced in [5], a lot of work has gone into designing efficient protocols and characterizing networks over which PSMT (or USMT) is achievable between two nodes, in a variety of settings [7, 8, 4, 9, 17, 10].

The problem of secure communication assumes relevance in networks where a subset of nodes may collaborate to disrupt communication or learn information about the message being routed through them. This distrust in the network is modelled via a fictitious entity known as *adversary*. In this work we consider an adversary which has unbounded computing power. Hence, our protocols for PSMT and USMT provide (perfect) secrecy in an information-theoretic sense. While the protocols for PSMT also provide perfect reliability w.r.t message delivery, USMT protocols are allowed to make errors with small probability. We make a distinction between two variants of USMT: one in which **R** can output an incorrect message (with small probability) and another in which **R** never outputs a wrong message but may choose to abort (with small probability). We refer to the latter variant as *detecting* USMT in this paper.

We model the underlying communication network as a directed graph. For several real-life networks where a node can communicate with another node but not the other way round, undirected graphs are not a suitable model. For instance, in a sensor network where different nodes have different transmission power, communication links tend to be uni-directional: a node u can hear v but v cannot hear u [18]. Lately, popular problems in distributed computing such as the *black hole search* problem [3] and rendezvous of mobile agents [1] have been attempted in directed graphs.

A wide body of work has focussed on synchronous networks where an upper bound is known *a priori* on the delay in delivery of messages. While synchronous network is an appealing model

*Center for Security, Theory and Algorithmic Research (C-STAR), International Institute of Information Technology, Hyderabad, 500032, India.

to work on, it is hard to achieve synchrony in practice. In a real-life network, messages can be arbitrarily delayed over a channel or may not arrive at all. Hence, protocols which promise guaranteed delivery of messages must cope with such inconsistent behaviour. This motivates the study of secure communication over asynchronous networks where no assumption is made regarding the relative speed of processes running at individual nodes or the delay in delivery of messages.

In [15], Sayeed et. al. initiate the study of PSMT over asynchronous directed networks and give several positive results. In [2], Choudhary et. al. extend their work to USMT. However, in both these cases, the underlying network is abstracted as a set of disjoint wires between the sender **S** and the receiver **R** (or vice versa). This leads to gross under-utilization of resources available in a directed network - it is easy to give examples of directed networks where a protocol for secure communication exists but not according to the extant literature; one such example is given in the following section. Furthermore, both the papers assume the adversary to be t-threshold, i.e., every potentially corrupt subset of nodes has size atmost t. As shown in [9], a t-threshold adversary does not capture all scenarios of distrust and may lead to over-estimation of connectivity requirements of a network.

In this work, we strictly generalize the results of [15, 2] and give the true characterization of general asynchronous directed networks over which PSMT and USMT tolerating non-threshold adversary is possible. Note that it is not easy to give efficient protocols for the general case. However, we provide efficient protocols for an important class of graphs – specifically, if a graph is connected *enough* so that PSMT tolerating t-threshold adversary is possible between every two nodes, we give an efficient protocol to achieve PSMT between any two given nodes.

Desmedt et. al. [4] as well as Choudhary et. al. [2] have shown that if a protocol achieving unconditional reliability (URMT) exists in their respective network models, one can design a protocol that also achieves perfect security (USMT). We show that this holds true in our more general network model as well, where the underlying network is *not* abstracted as a collection of disjoint wires. On a more interesting note, we show that a protocol for *detecting* USMT exists in an asynchronous directed network if and only if a protocol for PSMT exists. As far as we know, this result is the first of its kind in literature connecting two seemingly different problems.

1.1 A Motivating Example

Consider the simple asynchronous network $\mathcal{N}$ shown in figure 1. Here **S** and **R** are the sender and receiver respectively. They are assumed to be non-faulty. One among the rest of the nodes may be Byzantinely corrupt, i.e., the adversary structure is given by $\mathbb{A} = \{\{x\}, \{b_1\}, \{b_2\}, \{b_3\}\}$. Going by the extant literature, if we abstract the network as a collection of disjoint wires, there are only 3 of them between **S** and **R** and hence PSMT is impossible [2]. Surprisingly, we show that a simple protocol Π exists for PSMT in this network. Let m be the secret **S** wants to send. We give an informal description of Π here. In Π, nodes do the following:

- Node **S**: Apply a $(2, 4)$ secret sharing scheme [16] to m and get 4 shares m_0, m_1, m_2, m_3. Send m_0 to node x and m_i to node b_i $(1 \leq i \leq 3)$.
- Node x: Wait for the share m_0 to arrive from **S** and random number ρ from **R**. Send $l = m_0 + \rho$ to each b_i.
- Node b_i: Wait for the share m_i to arrive from **S** and send it to **R**. If some message arrives from node x, forward it to **R**.

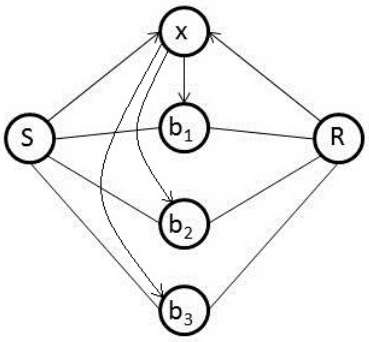

Figure 1:

- Node **R**: Choose a random number ρ and send it to node x. Wait till three consistent shares $m'_\alpha, m'_\beta, m'_\gamma$ ($\{\alpha, \beta, \gamma\} \subset \{0, 1, 2, 3\}$) are obtained. Reconstruct the secret with these shares.

  For $1 \leq i \leq 3$ the share m'_i is obtained from b_i. The share m'_0 is obtained when two concurrent values of l' are received from different b_i's ($m'_0 = l' - \rho$).

As proved in section 4, this protocol provides perfect secrecy and reliability.

1.2 Our Contribution

We make the following contributions: (a) We give the *first* characterization of general asynchronous directed networks over which PSMT tolerating non-threshold Byzantine adversary is possible from **S** to **R**. (b) Under the same network and adversary model, we give characterizations of USMT and *detecting* USMT. (c) We give an efficient protocol for PSMT tolerating threshold adversary between any two nodes (all-pair PSMT) in a network in which PSMT between every pair of nodes is possible.

2 Model and Definitions

We model the underlying asynchronous network as a directed graph $\mathcal{N} = (V, \mathcal{E})$, where V denotes the set of nodes (or players) in the network and $\mathcal{E} \subseteq V \times V$ represents the channels available for communication between nodes. In the following, we use the terms 'graph' and 'network' interchangeably. An important assumption we make in this paper is that all nodes know the topology of the network, i.e., all nodes know the digraph $\mathcal{N}$. Two special non-faulty nodes $\mathbf{S}, \mathbf{R} \in V$ denote the sender and receiver respectively. To distinguish between the message **S** intends to send to **R** through a (PSMT/USMT) protocol and the messages exchanged between nodes during the execution of the protocol, we refer to the former as *secret*. In this work, often a protocol is composed of several sub-protocols. In such a case, the secrets of sub-protocols would be referred to as *sub-secrets*.

Fault in the network is modelled via an unbounded centralized fictitious entity called the adversary that can control a subset of nodes in the network, specified via an adversary structure (defined later), and make them behave in a Byzantine fashion [11]. We assume that the channels available between nodes, i.e. the set $\mathcal{E}$, cannot be corrupted by the adversary (similar to the

secure channels setting). The adversary is *adaptive* and is allowed to dynamically corrupt nodes during protocol execution (and his choice may depend on the data seen so far). It knows the complete protocol specification as well as the topology of the network.

Additionally, since the network is asynchronous, computation proceeds in a sequence of steps controlled by the adversary. In a single step, the adversary activates a node by delivering some message to it, called an *event*, the node then performs local computation, changes its state and sends messages on its outgoing channels. A *schedule* is a finite or infinite sequence of events. See [6] for a detailed description of the asynchronous model.

A non-threshold adversary structure $\mathbb{A}$ is a set of subsets of the node set, i.e., $\mathbb{A} \subseteq \mathcal{P}(V \setminus \{S, R\})$, one of which may be corrupt during an execution. When an upper bound t is known on the number of faulty nodes in a network, the adversary structure contains all t-sized subsets of the node set and is referred to as t-threshold adversary. The adversary structures we consider have the property of *monotonicity*, i.e., whenever $B_1 \in \mathbb{A}$, then $\forall\ B_2$ such that $B_2 \subset B_1$, $B_2 \in \mathbb{A}$. We note that $\mathbb{A}$ can be uniquely represented by listing the elements in its maximal basis $\overline{\mathbb{A}}$ defined as follows.

Definition 1 (Maximal basis of $\mathbb{A}$)**.** *For any monotone adversary structure $\mathbb{A}$, its maximal basis $\overline{\mathbb{A}}$ is defined as $\overline{\mathbb{A}} = \{B \mid B \in \mathbb{A}$ and $\nexists\ X \in \mathbb{A}$ s.t. $B \subset X\}$. Abusing the standard notation, we assume that $\mathbb{A}$ itself is a maximal basis.*

Let the message space be a large finite field $\langle \mathbb{F}, +, \cdot \rangle$. All computations are done in this field. We now give formal definitions of PSMT and USMT. In an execution of a communication protocol, let $\Gamma(m, r)$ denote the view of the adversary when **S** chooses to send the secret m and coin tosses of the adversary are r.

Definition 2 ($\mathbb{A}$-PSMT)**.** *In a graph $\mathcal{N} = (V, \mathcal{E})$ a protocol for transmitting any secret $m \in \mathbb{F}$ from $\boldsymbol{S}$ to $\boldsymbol{R}$ tolerating an adversary structure $\mathbb{A}$ is an $\mathbb{A}$-PSMT protocol if for every Byzantine corruption $B \in \mathbb{A}$ and every schedule $\mathcal{D}$ the following two conditions are satisfied:*

1. ***Resiliency**: $\mathbf{R}$ always terminates with the secret m, $\mathbf{S}$ has chosen to send.*
2. ***Secrecy**: $\forall r, \forall m_0, m_1$ and every possible view c of the adversary it holds that $Pr[\Gamma(m_0, r) = c] = Pr[\Gamma(m_1, r) = c]$ where the probabilities are taken over the coin tosses of honest parties.*

Definition 3 ($(\mathbb{A}, \delta)$-USMT, $(\mathbb{A}, \delta)$-USMT$_\perp$)**.** *Let $\delta < \frac{1}{2}$. In a graph $\mathcal{N} = (V, \mathcal{E})$ a protocol for transmitting any secret $m \in \mathbb{F}$ from $\boldsymbol{S}$ to $\boldsymbol{R}$ tolerating an adversary structure $\mathbb{A}$ is an $(\mathbb{A}, \delta)$-USMT protocol if for every Byzantine corruption $B \in \mathbb{A}$ and every schedule $\mathcal{D}$ the following two conditions are satisfied:*

1. ***Resiliency**: $\forall m\ Pr[\boldsymbol{R}$ outputs $m | \boldsymbol{S}$ has sent $m] \geq (1 - \delta)$ where the probability is taken over the coin tosses of all players.*
2. ***Secrecy**: $\forall r, \forall m_0, m_1$ and every possible view c of the adversary it holds that $Pr[\Gamma(m_0, r) = c] = Pr[\Gamma(m_1, r) = c]$ where the probabilities are taken over the coin tosses of honest parties.*

We call an $(\mathbb{A}, \delta)$-USMT protocol a detecting $(\mathbb{A}, \delta)$-USMT protocol and denote it by $(\mathbb{A}, \delta)$-USMT$_\perp$ if the following stronger resiliency condition is satisfied:

- ***Resiliency**: $\forall m\ Pr[\boldsymbol{R}$ outputs $m | \boldsymbol{S}$ has sent $m] \geq (1 - \delta)$ where the probability is taken over the coin tosses of all players. Otherwise $\mathbf{R}$ outputs $\perp \notin \mathbb{F}$ or does not terminate.*

Definition 4 (Strong path)**.** *A sequence of vertices $v_1, v_2, v_3, ..., v_k$ is said to be a strong path from v_1 to v_k in the network $\mathcal{N} = (V, \mathcal{E})$ if for each $1 \leq i < k$, $(v_i, v_{i+1}) \in \mathcal{E}$. Furthermore, we assume that there vacuously exists a strong path from a node to itself.*

Definition 5 (Weak path)**.** *A sequence of vertices $v_1, v_2, v_3, ..., v_k$ is said to be a weak path from v_1 to v_k in the network $\mathcal{N} = (V, \mathcal{E})$ if for each $1 \leq i < k$, $(v_i, v_{i+1}) \in \mathcal{E}$ or $(v_{i+1}, v_i) \in \mathcal{E}$.*

Definition 6 (Blocked node, Head node)**.** *A node u along a weak path p is called a blocked node if its out-degree along p is 0. A node y along a weak path p is called a head node if it is an intermediate node with out-degree 2 or a terminal node with out-degree 1.*

The head nodes and blocked nodes along a weak path play a special role. A head node generates messages and forwards them to the two (or one) blocked nodes adjacent to it through the intermediate nodes. A blocked node receives messages from its two (or one) adjacent head nodes, performs operations on the messages received and forwards it to another node along a separate path. Hence, we look at a weak path p from **S** to **R**, which is not a strong path, as an alternating sequence of blocked nodes u_i's and head nodes y_i's, i.e, $u_1, y_1, u_2, y_2, \ldots, u_k, y_k$ ($k > 0$) occur along path p in that order. (Here u_1 may be **S** and y_k may be **R**).

Definition 7 (Authentication function)**.** *Let $\mathcal{K} = (K_1, K_2, K_3) \in_R \mathbb{F} \times \mathbb{F} \times \mathbb{F}$ and $m \in \mathbb{F}$. Authentication function χ is defined as $\chi(m; \mathcal{K}) = (\chi_1(m; \mathcal{K}),\ \chi_2(m; \mathcal{K}))$ where $\chi_1(m; \mathcal{K}) = m + K_1$ and $\chi_2(m; \mathcal{K}) = \chi_1(m; \mathcal{K}) \cdot K_2 + K_3$*

Here K_1, K_2, K_3 are usually referred to as *keys*. Using χ_1 we blind the message and using χ_2 we authenticate the blinded message. Suppose a random triplet $\mathcal{K}$ unknown to the adversary is established between two nodes u and v in a network $\mathcal{N}$. The authentication function has the following important properties: (a) Even if u sends $\chi(m; \mathcal{K})$ along a faulty path to v, adversary will not know anything about m. (b) Node v will be able to detect any change in $\chi(m; \mathcal{K})$'s value except with an error probability of atmost $\frac{1}{|\mathbb{F}|}$. (Proofs for the same appear in [14]).

Secret Sharing: We use the simple (k, n) *threshold scheme* ($n \geq k$) from [16] to create n shares of a secret where knowledge of any set of atmost $k - 1$ shares reveals no information about the secret. The secret can be efficiently reconstructed using the Berlekamp-Welch (BW) algorithm [12] from any set of shares S of size m (where $k \leq m \leq n$) if it contains at most $\lfloor \frac{m-k}{2} \rfloor$ incorrect shares. Such a set S of shares is said to be consistent.

In the following, we only consider adversary structures of size greater than 1. If the adversary structure is of unit size, say $\mathbb{A} = \{\{B_1\}\}$, the adversary can always fail-stop every node in B_1. Hence, a strong path from **S** to **R** avoiding all nodes in B_1 is necessary to enable **S** to send messages to **R** and therefore it is necessary for any reliable protocol. It is easy to see that this is also sufficient.

3 Characterizing asynchronous networks for $\mathbb{A}$-PSMT

Since working with an adversary structure of arbitrary size can be cumbersome and non-intuitive, we first show that working with an adversary structure of size three is sufficient for the case of PSMT. For the sake of completeness, we first settle the straightforward case of adversary structure of size two.

Theorem 1. *In a directed asynchronous network $\mathcal{N} = (V, \mathcal{E})$, $\{B_1, B_2\}$-PSMT protocol exists if and only if there exists a strong path from **S** to **R** in the network avoiding all nodes in $B_1 \cup B_2$.*

Proof. *Sufficiency*: **S** would send the secret m to **R** along the strong path which avoids every potentially faulty node. **R** will receive this secret both securely and reliably.

Necessity: For synchronous directed networks, with $\mathbb{A} = \{B_1, B_2\}$, it can be proved that a strong path avoiding nodes in $B_1 \cup B_2$ is necessary for PSMT along the lines of the proof given for synchronous undirected networks in Lemma 4.1.2 in [9]. If such a strong path is unavailable in an asynchronous directed network, the adversary has a schedule which will make PSMT impossible. □

From now on, in this section, we only consider adversary structures of size at least 3.

Theorem 2. *In a directed asynchronous network $\mathcal{N} = (V, \mathcal{E})$, $\mathbb{A}$-PSMT protocol exists if and only if for every adversary structure $A \subseteq \mathbb{A}$ such that $|A| = 3$, A-PSMT protocol exists.*

Proof. Necessity is trivial. We give sufficiency proof here. We show how to construct a protocol for an adversary structure $\mathcal{A}$ of size $n > 3$ from protocols for adversary structures of smaller size. Using this technique, starting from protocols for adversary structures of size 3, we would be able to construct a protocol for an adversary structure of arbitrary size inductively.

Consider $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$ and $\mathcal{A}_4$, four $\lceil \frac{3\mathcal{A}}{4} \rceil$-sized subsets of $\mathcal{A}$ such that each element of $\mathcal{A}$ occurs in at least three of the four sets. For $1 \leq i \leq 4$, let $\Pi_{\mathcal{A}_i}$ be the PSMT protocol tolerating $\mathcal{A}_i$. Let $f \in \mathbb{F}$ be the secret **S** intends to send. The PSMT protocol $\Pi_{\mathcal{A}}$ tolerating $\mathcal{A}$ proceeds as follows:

- **S** does a $(2, 4)$ secret sharing of f to obtain four shares f_1, f_2, f_3, f_4.
- The four protocols $\Pi_{\mathcal{A}_1}, \Pi_{\mathcal{A}_2}, \Pi_{\mathcal{A}_3}$ and $\Pi_{\mathcal{A}_4}$ are run in parallel; for $1 \leq i \leq 4$, $\Pi_{\mathcal{A}_i}$ is run on f_i as the sub-secret.
- **R** first waits for any three of the above four protocols to terminate. If the sub-secrets received through these protocols lead to the reconstruction of a unique secret, **R** outputs it. Otherwise, **R** further waits for another protocol to terminate. It now applies the BW algorithm on the sub-secrets obtained through the four protocols and outputs the outcome of the algorithm.

The correctness of this protocol is proved in the following lemma. □

Lemma 3. *The protocol $\Pi_{\mathcal{A}}$ is an $\mathcal{A}$-PSMT protocol.*

Proof. No matter which $B \in \mathcal{A}$ adversary chooses to corrupt, at least three out of the four sets $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$ and $\mathcal{A}_4$ contain B. Hence, at least three out of the protocols $\Pi_{\mathcal{A}_1}, \Pi_{\mathcal{A}_2}, \Pi_{\mathcal{A}_3}$ and $\Pi_{\mathcal{A}_4}$ will be resilient and secure. W.l.o.g assume that $\Pi_{\mathcal{A}_1}, \Pi_{\mathcal{A}_2}, \Pi_{\mathcal{A}_3}$ are those 3 protocols. Hence, for $1 \leq i \leq 3$, protocol $\Pi_{\mathcal{A}_i}$ terminates securely with **R** receiving the sub-secret f_i.

Resiliency: If the protocols $\Pi_{\mathcal{A}_1}$, $\Pi_{\mathcal{A}_2}$ and $\Pi_{\mathcal{A}_3}$ terminate before $\Pi_{\mathcal{A}_4}$ does, it is easy to see that **R** will output f. However, adversary may schedule events in the network such that $\Pi_{\mathcal{A}_4}$ terminates before all of $\Pi_{\mathcal{A}_1}, \Pi_{\mathcal{A}_2}, \Pi_{\mathcal{A}_3}$ do. When $\Pi_{\mathcal{A}_4}$ terminates, if **R** receives f_4 then we know that f will be reconstructed. However, since $\Pi_{\mathcal{A}_4}$ may not be tolerating the corrupt set B, **R** may receive f_4' $(\neq f_4)$. But then **R** will wait for another protocol to terminate. Now, with only one incorrect share out of four, it is easy to see that BW algorithm will output f.

Secrecy: Since adversary knows only f_4 which is a share of f obtained using $(2, 4)$ secret sharing of f, it does not reveal any information about f. □

Having reduced the problem of characterizing PSMT tolerating adversary structure $\mathbb{A}$ to all its 3-sized subsets, we now proceed to give a characterization of directed asynchronous networks tolerating a given 3-sized adversary structure. We prove sufficiency in this section itself but address necessity in a separate subsection.

Theorem 4. *In a directed asynchronous network $\mathcal{N}$, $\{B_1, B_2, B_3\}$-PSMT protocol from $\mathbf{S}$ to $\mathbf{R}$ is possible if and only if for each $\alpha \in \{1,2,3\}$, there exists a weak path q_α avoiding nodes in $B_1 \cup B_2 \cup B_3$ such that every node u along the path q_α has a strong path to $\boldsymbol{R}$ avoiding all nodes in $\bigcup_{\beta \in \{1,2,3\}-\{\alpha\}} B_\beta$. (Paths q_1, q_2, q_3 need not be distinct.)*

Sufficiency. Let f be any field element $\mathbf{S}$ intends to send. In the special case when q_α is a strong path from $\mathbf{S}$ to $\mathbf{R}$, for an $\alpha \in \{1,2,3\}$, $\mathbf{S}$ can trivially send f along q_α. Since q_α does not contain any corrupt node, $\mathbf{R}$ receives f securely and reliably.

For the rest of the cases, we construct a protocol $\Pi_{\{B_1,B_2,B_3\}}$ for $\{B_1, B_2, B_3\}$-PSMT whose correctness is proved in the following Lemma. The protocol $\Pi_{\{B_1,B_2,B_3\}}$ is composed of three sub-protocols Π_1, Π_2, Π_3 which are run in parallel in the network, each one on f as the sub-secret. $\mathbf{R}$ first waits for any two of these three sub-protocols to terminate. If the same sub-secret is recovered from both these protocols, $\mathbf{R}$ outputs it. Otherwise, $\mathbf{R}$ waits for the third protocol to terminate and outputs the majority of the outcome of the three protocols.

We give a construction for Π_1, and the constructions of Π_2 and Π_3 follow by symmetry. The protocol Π_1 uses the honest weak path q_1. Since q_1 is not a strong path, it can be expressed as $u_1, y_1, u_2, y_2, \ldots, u_{n_1}, y_{n_1}$ $(n_1 \in \mathbb{N})$ where u_i's represent blocked nodes and y_i's represent head nodes. Π_1 proceeds as follows:

1. $\mathbf{S}$ sends f to u_1 along q_1. For $1 \leq j \leq n_1$, node y_j chooses a random key K_j and sends it to u_j and u_{j+1} along q_1 (u_{n_1+1} denotes $\mathbf{R}$).

2. Node u_1 sends $L_1 = f + K_1$ to $\mathbf{R}$ along a strong path avoiding $B_2 \cup B_3$ when it receives f from $\mathbf{S}$ and K_1 from y_1. For $1 < j \leq n_1$, u_j sends $L_j = K_{j-1} + K_j$ to $\mathbf{R}$ along a strong path avoiding $B_2 \cup B_3$ when it receives K_{j-1} from y_{j-1} and K_j from y_j.

3. $\mathbf{R}$ waits until it receives K'_{n_1} from y_{n_1} and for $1 \leq j \leq n_1$, L'_j from u_j. It then does the following:

   for z in n_1 to 2

   $K'_{z-1} = L'_z - K'_z$.

   Output $f'_1 = L'_1 - K'_1$.

This completes the description of Π_1, and hence of $\Pi_{\{B_1,B_2,B_3\}}$. □

Lemma 5. *The protocol $\Pi_{\{B_1,B_2,B_3\}}$ is a $\{B_1, B_2, B_3\}$-PSMT protocol.*

Proof. W.l.o.g let us assume that the set B_1 is corrupt.

Resiliency: Since protocols Π_2 and Π_3 do not involve nodes in B_1, these protocols are bound to terminate (with the correct sub-secret) no matter how the adversary schedules messages in the asynchronous system. Hence, when $\mathbf{R}$ waits for at least two protocols to terminate, it will not wait indefinitely. If Π_2 and Π_3 indeed terminate before Π_1 does, $\mathbf{R}$'s output is correct.

However, there may exist a schedule in the network such that Π_1 terminates before both Π_2 and Π_3 terminate. Say Π_1 and Π_2 terminate before Π_3 does. If the same sub-secret is recovered from Π_1 and Π_2, $\mathbf{R}$'s output is correct as Π_1 terminates with the correct sub-secret. Otherwise,

R waits for Π_3 to terminate. Again, **R** does not have to wait indefinitely. When Π_3 eventually terminates, we know that the majority will be the correct secret.

Secrecy: Since protocols Π_2 and Π_3 do not involve nodes in B_1, none of the messages exchanged during these protocols is available to the adversary. Even in the case of protocol Π_1, adversary sees only $f + K_1, K_1 + K_2, \ldots, K_{n_1-1} + K_{n_1}$ where $K_1, K_2, \ldots, K_{n_1}$ are random numbers. This does not reveal any information about the secret f. □

Notice that for $\Pi_{\{B_1,B_2,B_3\}}$ both communication complexity of the protocol and computation complexity at every node is polynomial in the size of the network. If we denote the size of adversary structure $\mathbb{A}$ by N, from Theorem 2 we can see that $O(N^5)$ sub-protocols (of the same complexity as of $\Pi_{\{B_1,B_2,B_3\}}$) need to be run to achieve $\mathbb{A}$-PSMT. Starting with the output of these $O(N^5)$ protocols [1] , **R** can compute the output of $\mathbb{A}$-PSMT in $O(N^5)$ computational steps. Hence our $\mathbb{A}$-PSMT protocol is efficient in the size of network and the size of adversary structure.

3.1 Necessity

Consider a directed asynchronous network $\mathcal{N} = (V, \mathcal{E})$, with $\mathbf{S}, \mathbf{R} \in V$ as the sender and receiver respectively and three subsets $B_1, B_2, B_3 \subseteq V \setminus \{S, R\}$ comprising the adversary structure $B = \{B_1, B_2, B_3\}$. We show that if $\mathcal{N}$ does not satisfy the conditions of Theorem 4, PSMT tolerating B is impossible in $\mathcal{N}$. Without loss of generality, let us assume that the three sets comprising the adversary structure are disjoint. Let the path q_1 be not present between **S** and **R** in $\mathcal{N}$. (The case where path q_2 or q_3 is not present can be handled analogously.) Hence, every weak path between **S** and **R** avoiding $B_1 \cup B_2 \cup B_3$ has at least one node w such that every strong path from w to **R** passes through nodes in $B_2 \cup B_3$.

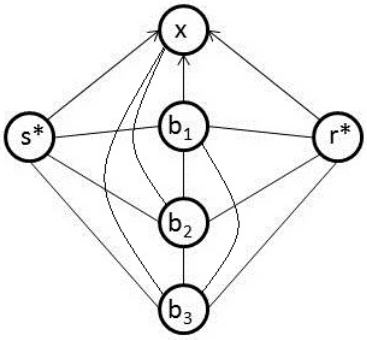

Figure 2:

We first consider the simple digraph $\mathcal{N}^* = (V^*, \mathcal{E}^*)$ shown in figure 2 consisting of six nodes s^*, r^*, b_1, b_2, b_3 and x where s^* is the sender and r^* is the receiver. Note that the graph $\mathcal{N}^*$ does not satisfy the conditions given in Theorem 4 – the edges (x, s^*), (x, r^*) and (x, b_1) are not present in the graph. We show that $\{\{b_1\}, \{b_2\}, \{b_3\}\}$-PSMT from s^* to r^* is impossible in $\mathcal{N}^*$. We then prove that the digraph $\mathcal{N}$ can be partitioned into disjoint sets whose connectivity properties are similar to the connectivity between nodes of the digraph $\mathcal{N}^*$. Now, if PSMT is possible in $\mathcal{N}$, it would also be possible in $\mathcal{N}^*$, which is a contradiction. This implies that the conditions mentioned in Theorem 4 are necessary.

[1] Note that **R** need not wait for all sub-protocols to terminate.

Theorem 6. *In the network $\mathcal{N}^*$ shown in figure 2, $\{\{b_1\},\{b_2\},\{b_3\}\}$-PSMT from s^* to r^* is impossible.*

Proof. W.r.t. to an execution E_i, we define the following: (a) The vector $\vec{C_i} = (c^i_{s^*}, c^i_{r^*}, c^i_{b_1}, c^i_{b_2}, c^i_{b_3}, c^i_x)$ which denotes the coin tosses input to players, where c^i_p denotes the coin tosses of player p. (b) The time instant T_{E_i} at which r^* halts (T_{E_i} may not be finite). (c) The view of a player p $view_p(E)$ which comprises of the internal coin tosses c^i_p of player p and the messages it receives during the execution E_i.

Till time T_{E_1} (defined later), adversary schedules events in the asynchronous network $\mathcal{N}^*$ in the following way: Any execution proceeds in a sequence of time periods. In any time period i, all players except b_3 are activated one by one; when a player p is active, all messages generated for player p in the previous time period $i-1$ are delivered in order.

Assume that a protocol Π^* exists for $\{\{b_1\},\{b_2\},\{b_3\}\}$-PSMT from s^* to r^* in the network $\mathcal{N}^*$. Consider the following four executions of Π^*.

- Execution E_1: s^* chooses secret m_1 and the coin tosses of players are $\vec{C_1}$. In this execution node b_3 fail-stops. Let r^* halt at time instant T_{E_1} outputting m_1.
- Execution E_2: s^* chooses secret m_1 and the coin tosses of players are $\vec{C_2} = \vec{C_1}$. In this execution node b_1 is passively corrupt. As the view of r^* in this execution is same as in E_1 (b_3 is never active before T_{E_1}), it halts at time instant $T_{E_2} = T_{E_1}$ outputting m_1. Let the view at b_1 be v.

There must exist coin tosses $\vec{C_3}$ such that $c^3_{b_1} = c^2_{b_1}$ and when s^* chooses to send a secret $m_2(\neq m_1)$ view at node b_1 is v. Else, view v would reveal information about m_1.

- Execution E_3: s^* chooses secret m_2 and the coin tosses of players are $\vec{C_3}$. In this execution node b_1 is passively corrupt. View at node b_1 is v.
- Execution E_4: s^* chooses secret m_2 and the coin tosses of players are $\vec{C_4}$ such that $c^4_{s^*} = c^3_{s^*}$, $c^4_{r^*} = c^2_{r^*}$ and $c^4_{b_1} = c^3_{b_1} = c^2_{b_1}$. In this execution node b_2 is actively corrupt. Node b_2 ignores all messages that it receives, sends to s^* what it sent to s^* in E_3, sends to r^* what it sent to r^* in E_2, sends to b_1 what it sent in E_3 (or E_2) and does not send any message to any other player.

As proved in the following lemma, $view_{r^*}(E_4) = view_{r^*}(E_2)$ implying r^* cannot distinguish between executions E_2 and E_4. Since its output in E_2 is m_1, it outputs m_1 in E_4 too, where s^* chose to send m_2. However, Π^* is a PSMT protocol. We have a contradiction. □

Lemma 7. *Till time T_{E_1}, the following equalities hold: $view_{s^*}(E_4) = view_{s^*}(E_3)$, $view_{b_1}(E_4) = view_{b_1}(E_3) = view_{b_1}(E_2)$ and $view_{r^*}(E_4) = view_{r^*}(E_2)$.*

Proof. Notice that the coin tosses of b_1 in all executions described above is $c^4_{b_1}$. Also, b_1 cannot distinguish between first three executions, it receives and sends same messages in all these executions. We already know that b_3 is never active before T_{E_1}. We give a proof of the lemma by induction on the number of *time-periods*:

Time period 1: The equalities obviously hold for this time-period.

Time period i: Assume that the equalities hold till time-period $i-1$. As $view_{b_1}(E_4) = view_{b_1}(E_3) = view_{b_1}(E_2)$ till time-period $i-1$, messages sent by b_1 in time-period $i-1$ are same in E_4, E_3 and E_2. Similar statements can be made for s^* and r^*. In time-period i, s^* receives

messages from b_1 which are same as sent in E_4. Also, as described above, b_2 sends to s^* what it sent to s^* in E_3. Till time T_{E_1}, as s^* does not receive input from any other node and its coin tosses are same in E_4 and E_3, $view_{s^*}(E_4) = view_{s^*}(E_3)$. Similarly, in time-period i, r^* receives messages from b_1 which are same as sent in E_2. Also, b_2 sends to r^* what it sent to r^* in E_2. Till time T_{E_1}, as r^* does not receive input from any other node and its coin tosses are same in E_4 and E_2, $view_{r^*}(E_4) = view_{r^*}(E_2)$. Similarly the remaining equality also holds. □

Theorem 8. *The set of nodes V in the network $\mathcal{N}$ can be partitioned into 6 disjoint sets $S^*, R^*, B_1' \subseteq B_1, B_2, B_3$ and X' such that $\mathbf{S} \in S^*$, $\mathbf{R} \in R^*$ and $\forall\ 1 \leq i < j \leq 6$ an edge exists between a node of $F[i]$ and a node of $F[j]$ only if $(f(i), f(j)) \in \mathcal{E}^*$ where $F = (S^*, R^*, B_1', B_2, B_3, X')$ and $f = (s^*, r^*, b_1, b_2, b_3, x)$ are two vectors.*

Proof. In the network $\mathcal{N}$, every weak path between $\mathbf{S}$ and $\mathbf{R}$ avoiding $B_1 \cup B_2 \cup B_3$ has at least one node w such that every strong path from w to $\mathbf{R}$ passes through nodes in $B_2 \cup B_3$.

We partition the non-faulty nodes $H = V \setminus \{B_1 \cup B_2 \cup B_3\}$ into 3 disjoint sets. Let $R^* \subset H$ denote the set of all nodes that have a weak path to $\mathbf{R}$ (avoiding $B_1 \cup B_2 \cup B_3$) such that every node w in the weak path has a strong path to $\mathbf{R}$ avoiding nodes in $B_2 \cup B_3$. Divide the rest of non-faulty nodes in two disjoint sets S^* and X. Define $S^* = \{w \in H \setminus R^* \mid w$ has a strong path to $\mathbf{R}$ avoiding nodes in $B_2 \cup B_3\}$. Define $X = H \setminus \{S^* \cup R^*\}$. Clearly, $\mathbf{R} \in R^*$ and $\mathbf{S} \in S^*$ (otherwise even reliable message transmission would not be possible in $\mathcal{N}$). Moreover, if any node $w \in X$ has a strong path to $\mathbf{R}$, it passes through some node in $B_2 \cup B_3$. Otherwise w would belong to S^* itself.

Also, divide the set B_1 into two disjoint sets. Define $B_1^X = \{v \in B_1 | \exists\ u \in X$ such that there is a strong path from u to $v\}$. Let $B_1' = B_1 \setminus B_1^X$. Let us consider the two sets X and B_1^X together as a set X', i.e., $X' = X \cup B_1^X$.

The only edges missing from $\mathcal{N}^*$ are $(x, s^*), (x, r^*), (x, b_1)$ and $(s^*, r^*), (r^*, s^*)$. It easily follows from the definitions above that $\nexists\ (u, v) \in \mathcal{E}$ such that $u \in X'$ and $v \in S^* \cup R^* \cup B_1'$. Also, there cannot exist any directed edge between a node in S^* and a node in R^*. Hence proved. □

Theorem 9. *In the directed asynchronous network $\mathcal{N} = (V, \mathcal{E})$, if $(\{B_1, B_2, B_3\})$-PSMT is possible from $\mathbf{S}$ to $\mathbf{R}$ then $(\{\{b_1\}, \{b_2\}, \{b_3\}\})$-PSMT is possible from s^* to r^* in the network $\mathcal{N}^*$.*

Proof. It is straightforward to prove the above theorem using standard player simulation technique. □

However from Theorem 6 we know that $(\{\{b_1\}, \{b_2\}, \{b_3\}\})$-PSMT is impossible from s^* to r^* in the network $\mathcal{N}^*$. We arrive at a contradiction. Hence, the conditions mentioned in Theorem 4 are necessary.

4 All pairs PSMT

Let $\mathcal{N} = (V, \mathcal{E})$ be a network in which PSMT tolerating t-threshold adversary is possible from any node $x \in V$ to any node $y \in V$. It follows from Theorems 2 4 that there are $3t + 1$ node disjoint weak paths between x and y of which at least $2t + 1$ are strong paths from x to y. For such a network, we give an efficient protocol Π_{uv} for PSMT tolerating t-threshold adversary from node $u \in V$ to another node $v \in V$.

Let f be any field element u intends to send to v securely and reliably. The node u makes $3t + 1$ shares of f, namely $f_1, f_2, \ldots, f_{3t+1}$, using $(t + 1, 3t + 1)$ secret sharing scheme. Now,

sub-protocols $\Gamma_1, \Gamma_2, \ldots, \Gamma_{3t+1}$ (described later) are run in parallel in the network. **R** waits till at least $2t$ of these sub-protocols terminate. It sets a variable e to 0, then runs the following loop:

while (true)

- Wait for another sub-protocol to terminate.
- Let W denote the set of outcomes of the sub-protocols that have terminated so far. Run the BW algorithm on inputs t, e, W [12]. If it returns a polynomial, output its constant term, come out of the loop and halt.
- $e := e + 1$

For $1 \leq i \leq 3t+1$, the sub-protocol Γ_i tries to securely communicate f_i assuming that weak path w_i does not contain any faulty node. If w_i is a strong path from u to v, the protocol Γ_i is simply: send f_i to v along w_i. Otherwise, w_i is a weak path expressed as $u_1, y_1, u_2, y_2, \ldots, u_{n_i}, y_{n_i}$ ($n_i \in \mathbb{N}$) where u_i's represent blocked nodes and y_i's represent head nodes and Γ_i proceeds in the following steps -

1. u sends f_i to u_1 along w_i. For $1 \leq j \leq n_i$, node y_j chooses a random key K_j and sends it to u_j and u_{j+1} along w_i. (u_{n_i+1} denotes v).

2. Node u_1 sends $L_1 = f_i + K'_1$ along $2t+1$ node disjoint strong paths to v when it receives f_i from u and K'_1 from y_1. For $1 < j \leq n_i$, u_j sends $L_j = K'_{j-1} + K'_j$ along $2t+1$ node disjoint strong paths to v when it receives K'_{j-1} from y_{j-1} and K'_j from y_j. (This ensures that for each j, $1 \leq j \leq n_i$, v receives L_j reliably).

3. v waits until it receives K''_{n_i} from y_{n_i} and for each j, $1 < j \leq n_i$, at least $t+1$ concurrent readings of L'_j from u_j. It then runs the following loop:
   for z in n_i to 2
   $$K''_{z-1} = L'_z - K''_z.$$
   Output $f'_i = L'_1 - K''_1$.

This completes the description of Γ_i, and hence of Π_{uv}.

Lemma 10. *The protocol Π_{uv} is an efficient PSMT protocol tolerating t-threshold adversary.*

Proof. Consider a sub-protocol Γ_i where the weak path w_i does not contain any corrupted node. However, the messages sent by blocked nodes in w_i may be accessible to the adversary. In the worst case it may know all of $f_i + K_1, K_1 + K_2, \ldots, K_{n_i-1} + K_{n_i}$. Still this does not reveal any information about f_i. Moreover, every blocked node sends messages to **R** along $2t+1$ vertex disjoint paths. Hence if **R** waits long enough he would receive all these messages reliably and with the extra knowledge of K_{n_i} recover f_i correctly.

Now, we know that at least $2t+1$ weak paths are honest. So at least $2t+1$ Γ_i's would eventually terminate with $f'_i = f_i$. Hence, in every iteration of the while loop the set W supplied to BW algorithm has atmost t errors. So the algorithm produces an output in some iteration and we know that if it does produce an output, it is correct [12]. Therefore Π_{uv} is a reliable protocol. Moreover, at least $2t+1$ shares of f are unknown to the adversary, making Π_{uv} secure.

It is easy to see that the overall communication complexity of the protocol Π_{uv} and the computation complexity at each node is a polynomial in t and the size of network. □

5 Characterizing asynchronous networks for $(\mathbb{A}, \delta)$-USMT

In this section we give a characterization of asynchronous directed networks over which $(\mathbb{A}, \delta)$-USMT is possible. In this variant of USMT, **R** is allowed to output an incorrect message with small probability. In the next section we discuss *detecting* USMT where if **R** outputs a message it must be the correct one. One important difference between the two variants is that while in the former it is sufficient to deal with adversary structure of size two as shown in the following theorem, in the latter we have to be content with three-sized adversary structure. Specifically, the following *reduction* technique cannot be applied to the case where **R** is not allowed to output a wrong message.

Theorem 11. *In a directed asynchronous network $\mathcal{N}$, $(\mathbb{A}, \delta)$-USMT protocol exists if and only if for every adversary structure $A \subseteq \mathbb{A}$ such that $|A| = 2$, (A, δ)-USMT protocol exists.*

Proof. Necessity is trivial. We give sufficiency proof here. We show how to construct a protocol for an adversary structure $\mathcal{A}$ of size $n > 2$ from protocols for adversary structures of smaller size. Using this technique, starting from protocols for adversary structures of size 2, we would be able to construct a protocol for an adversary structure of arbitrary size inductively.

Consider $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$, three $\lceil \frac{2\mathcal{A}}{3} \rceil$-sized subsets of $\mathcal{A}$ such that each element of $\mathcal{A}$ occurs in at least two of the three sets. For $1 \leq i \leq 3$, let $\Pi_{\mathcal{A}_i}$ be the USMT protocol tolerating $\mathcal{A}_i$. Let f be the secret **S** intends to send. The USMT protocol $\Pi_{\mathcal{A}}$ tolerating $\mathcal{A}$ proceeds as follows:

1. For each $\beta \in \{1, 2, 3\}$, **S** chooses a three-tuple $\mathcal{K}_\beta = (K_{\beta,1}, K_{\beta,2}, K_{\beta,3}) \in_R \mathbb{F} \times \mathbb{F} \times \mathbb{F}$ and evaluates $\chi_1(f, \mathcal{K}_\beta) = f + K_{\beta,1}$ and $\chi_2(f, \mathcal{K}_\beta) = (f + K_{\beta,1}) \cdot K_{\beta,2} + K_{\beta,3}$.

2. Seven instances of the protocol $\Pi_{\mathcal{A}_1}$ are run in parallel in the network on the sub-secrets $K_{1,1}$, $K_{1,2}$, $K_{1,3}$, $\chi_1(f; \mathcal{K}_2)$, $\chi_2(f; \mathcal{K}_2)$, $\chi_1(f; \mathcal{K}_3)$ and $\chi_2(f; \mathcal{K}_3)$. Similarly, seven instances each of protocols $\Pi_{\mathcal{A}_2}$ and $\Pi_{\mathcal{A}_3}$ are run in parallel alongside the instances of protocol $\Pi_{\mathcal{A}_1}$. In essence, for each $\beta \in \{1, 2, 3\}$, the tuple of keys $\mathcal{K}_\beta$ is sent through three instances of protocol $\Pi_{\mathcal{A}_\beta}$ and the secret f authenticated with $\mathcal{K}_\beta$ is sent through both the other protocols.

3. **R** waits until for two *distinct* indices $x, y \in \{1, 2, 3\}$ it receives $K'_{x,1}$, $K'_{x,2}$, $K'_{x,3}$, $\chi_1(f, \mathcal{K}_y)'$, $\chi_2(f, \mathcal{K}_y)'$ through the protocol $\Pi_{\mathcal{A}_x}$ and $K'_{y,1}$, $K'_{y,2}$, $K'_{y,3}$, $\chi_1(f, \mathcal{K}_x)'$, $\chi_2(f, \mathcal{K}_x)'$ through the protocol $\Pi_{\mathcal{A}_y}$ such that the following conditions are satisfied:

   - $\chi_2(f, \mathcal{K}_x)' = \chi_1(f, \mathcal{K}_x)' \cdot K'_{x,2} + K'_{x,3}$
   - $\chi_2(f, \mathcal{K}_y)' = \chi_1(f, \mathcal{K}_y)' \cdot K'_{y,2} + K'_{y,3}$
   - $\chi_1(f, \mathcal{K}_x)' - K'_{x,1} = \chi_1(f, \mathcal{K}_y)' - K'_{y,1}$

   In simple words, **R** waits until two authenticated messages pass verification against corresponding keys and the same secret is recovered. **R** outputs $f' = \chi_1(f, \mathcal{K}_x)' - K'_{x,1}$.

The following Lemma proves the correctness of $\Pi_{\mathcal{A}}$. Observe that the resiliency of $\Pi_{\mathcal{A}}$ is lower than the resiliency of the protocols using which it has been composed. Nevertheless, the failure probability can be brought down to δ by repeating the protocol $\Pi_{\mathcal{A}}$ sufficiently many times in parallel. □

Lemma 12. *The protocol $\Pi_{\mathcal{A}}$ is an $(\mathcal{A}, \delta')$-USMT protocol where $\delta' = 1 - (1 - \delta)^{10} \cdot (\frac{|\mathbb{F}|-1}{|\mathbb{F}|})^2$.*

Proof. No matter what $B \in \mathcal{A}$ adversary chooses to corrupt, at least two of the three sets $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ contain B. Hence, the instances of at least two out of the three protocols $\Pi_{\mathcal{A}_1}$, $\Pi_{\mathcal{A}_2}$ and $\Pi_{\mathcal{A}_3}$ will be resilient and secure. W.l.o.g assume that $\Pi_{\mathcal{A}_1}$ and $\Pi_{\mathcal{A}_2}$ are those two protocols. Therefore, the instances of protocols $\Pi_{\mathcal{A}_1}$ and $\Pi_{\mathcal{A}_2}$ do not reveal any information about the sub-secrets transmitted through them and with high probability, they terminate with the correct output at $\mathbf{R}$. We first prove that the protocol $\Pi_{\mathcal{A}}$ is perfectly secure.

Secrecy: Note that only the instances of protocol $\Pi_{\mathcal{A}_3}$ can reveal information about the sub-secrets transmitted through them. Hence, in the best case, adversary knows $K_{3,1}$, $K_{3,2}$, $K_{3,3}$, $\chi_1(f;\mathcal{K}_1)$, $\chi_2(f;\mathcal{K}_1)$, $\chi_1(f;\mathcal{K}_2)$ and $\chi_2(f;\mathcal{K}_2)$. However, as it does not know $\mathcal{K}_1$ and $\mathcal{K}_2$, it does not know anything about the secret f.

Resiliency: Let $\psi_1 = (K_{1,1}, K_{1,2}, K_{1,3}, \chi_1(f,\mathcal{K}_2), \chi_2(f,\mathcal{K}_2))$ and $\psi_2 = (K_{2,1}, K_{2,2}, K_{2,3}, \chi_1(f,\mathcal{K}_1), \chi_2(f,\mathcal{K}_1))$ denote part of the sequence of sub-secrets sent through the instances of $\Pi_{\mathcal{A}_1}$ and $\Pi_{\mathcal{A}_2}$ respectively. We have ignored the sub-secrets obtained through the authentication of secret f using keys sent through $\Pi_{\mathcal{A}_3}$. Let ψ represent the combined sequence which has 10 elements. Also, let ζ represent a part of the sequence of sub-secrets sent through instances of $\Pi_{\mathcal{A}_3}$, $\zeta = (\chi_1(f,\mathcal{K}_1), \chi_2(f,\mathcal{K}_1), \chi_1(f,\mathcal{K}_2), \chi_2(f,\mathcal{K}_2))$. We do not consider the keys sent through $\Pi_{\mathcal{A}_3}$.

Suppose all sub-secrets in ψ are received reliably. To minimize the chances of reliable transmission of secret f adversary would tamper with sub-secrets in ζ and schedule events in the network such that all sub-secrets in ζ are received before all sub-secrets in ψ are received. Now, consider the following event E: all sub-secrets in ψ are received reliably and any tampering in the sequence of sub-secrets ζ is detected. In such an event $\mathbf{R}$ recovers $f' = f$. The probability that the protocol $\Pi_{\mathcal{A}}$ produces correct output is at least the probability of the event E. Since adversary does not know $\mathcal{K}_1$ and $\mathcal{K}_2$, we know that $Pr(E) > (1-\delta)^{10}(\frac{|\mathbb{F}|-1}{|\mathbb{F}|})^2$. Therefore, the probability of failure δ' is atmost $1 - P(E)$. □

Having reduced the problem of characterizing USMT tolerating adversary structure $\mathbb{A}$ to all its 2-sized subsets, we now proceed to give a characterization of directed asynchronous networks tolerating a given 2-sized adversary structure.

Theorem 13. *In a directed asynchronous network $\mathcal{N}$, $(\{B_1, B_2\}, \delta)$-USMT protocol from $\mathbf{S}$ to $\mathbf{R}$ is possible if and only if for each $\alpha \in \{1,2\}$, there exists a weak path q_α avoiding nodes in $B_1 \cup B_2$ such that every node u along the path q_α has a strong path to $\boldsymbol{R}$ avoiding all nodes in $B_{\overline{\alpha}}$[2] (Paths q_1, q_2 need not be distinct.)*

Proof. According to Theorem 17 in [13], same characterization holds for *unconditionally reliable message transmission*(URMT) tolerating a 2-sized adversary structure. Hence, the characterization is obviously necessary for USMT. Also, the protocol given for URMT in the Sufficiency proof of Theorem 17 in [13] does not reveal any information about the secret being transmitted. Hence the same protocol can be used for USMT. □

An interesting implication follows: if we are ready to compromise a little on reliability, security is for *free* in directed asynchronous networks.

6 Characterizing asynchronous networks for $(\mathbb{A}, \delta)$-USMT$_\perp$

In this section we characterize $(\mathbb{A}, \delta)$-USMT$_\perp$, where $\mathbf{R}$ must not output an incorrect message. With atmost δ probability $\mathbf{R}$ may output $\perp \notin \mathbb{F}$ or does not terminate. As we have been doing in

[2]We denote $\overline{1} = 2$ and vice-versa.

previous sections, we show that instead of dealing with adversary structures of arbitrary size, it is enough to deal with adversary structures of a constant size. In the case of *detecting* USMT that constant turns out to be 3. For the sake of completeness, we first provide the characterization for adversary structures of size 2.

Theorem 14. *In a directed asynchronous network $\mathcal{N} = (V, \mathcal{E})$, $(\{B_1, B_2\}, \delta)$-USMT$_\perp$ protocol exists if and only if there exists a strong path from $\boldsymbol{S}$ to $\boldsymbol{R}$ in the network avoiding all nodes in $B_1 \cup B_2$.*

Proof. Sufficiency is straightforward. For the necessity proof, please refer Section 5.1 in [13]. □

From now on, in this section, we only consider adversary structures of size at least 3.

Theorem 15. *In a directed asynchronous network $\mathcal{N}$, $(\mathbb{A}, \delta)$-USMT$_\perp$ protocol exists if and only if for every adversary structure $A \subseteq \mathbb{A}$ such that $|A| = 3$, (A, δ)-USMT$_\perp$ protocol exists.*

Proof. Necessity is trivial. We give sufficiency proof here. As has been argued in the proof of Theorem 2, it is enough to show how to construct a protocol for an adversary structure $\mathcal{A}$ of size $n > 3$ from protocols for adversary structures of smaller size.

In a manner similar to the proof of Theorem 2, consider $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$ and $\mathcal{A}_4$, four $\lceil \frac{3\mathcal{A}}{4} \rceil$-sized subsets of $\mathcal{A}$ such that each element of $\mathcal{A}$ occurs in at least three of the four sets. For $1 \leq i \leq 4$, let $\Pi_{\mathcal{A}_i}$ be the USMT protocol tolerating $\mathcal{A}_i$. Let $f \in \mathbb{F}$ be the secret $\mathbf{S}$ intends to send. The USMT protocol $\Pi_{\mathcal{A}}$ tolerating $\mathcal{A}$ proceeds as follows:

- $\mathbf{S}$ does a $(2, 4)$ secret sharing of f to obtain four shares f_1, f_2, f_3, f_4.
- The four protocols $\Pi_{\mathcal{A}_1}, \Pi_{\mathcal{A}_2}, \Pi_{\mathcal{A}_3}$ and $\Pi_{\mathcal{A}_4}$ are run in parallel; for $1 \leq i \leq 4$, $\Pi_{\mathcal{A}_i}$ is run on f_i as the sub-secret.
- $\mathbf{R}$ waits until it receives three consistent shares or at least two $\perp$'s. In the former case it reconstructs f', in the latter it outputs $\perp$.

The protocol $\Pi_{\mathcal{A}}$ constructed above has failure probability of atmost $3\delta - 3\delta^2 + \delta^3$ as proved in the following lemma. This probability can be reduced to δ by repeating the protocol in parallel sufficiently many times. □

Lemma 16. *The protocol $\Pi_{\mathcal{A}}$ is an $(\mathcal{A}, 3\delta - 3\delta^2 + \delta^3)$-USMT$_\perp$ protocol.*

Proof. In the spirit of the proof of Lemma 3, we can assume w.l.o.g. that protocols $\Pi_{\mathcal{A}_1}$, $\Pi_{\mathcal{A}_2}$ and $\Pi_{\mathcal{A}_3}$ do not reveal any information about f_1, f_2 and f_3 respectively. Moreover, with high probability, for $1 \leq i \leq 3$, $\mathbf{R}$ receives f_i.

Resiliency: To minimize the chances of reliable transmission of f, adversary would schedules events in the network such that $\Pi_{\mathcal{A}_4}$ terminates with some $f_3' (\neq f_3)$ of adversary's choice before other protocols terminate. However, if $\mathbf{R}$ receives correct sub-secrets through the other three protocols, it would be able to reconstruct the secret f. This happens with probability at least $(1 - \delta)^3$.

Secrecy: Since adversary knows only f_4 which is a share of f obtained using $(2, 4)$ secret share of f, it does not reveal any information about f. □

We have reduced the problem of characterizing USMT$_\perp$ tolerating adversary structure $\mathbb{A}$ to tolerating all its 3-sized subsets. For asynchronous directed networks tolerating a 3-sized structure, we now claim that the characterization for USMT$_\perp$ is same as the characterization for PSMT proved in Theorem 4 implying that if a secure protocol satisfying stronger resiliency conditions exists in a network, we may as well design a protocol that achieves perfect resiliency.

Theorem 17. *In a directed asynchronous network $\mathcal{N}$, $(\{B_1, B_2, B_3\}, \delta)$-USMT$_\perp$ protocol from* **S** *to* **R** *is possible if and only if for each $\alpha \in \{1,2,3\}$, there exists a weak path q_α avoiding nodes in $B_1 \cup B_2 \cup B_3$ such that every node u along the path q_α has a strong path to* ***R*** *avoiding all nodes in $\bigcup_{\beta \in \{1,2,3\}-\{\alpha\}} B_\beta$ (Paths q_1, q_2, q_3 need not be distinct.)*

Proof. *Sufficiency*: The protocol $\Pi_{\{B_1,B_2,B_3\}}$ described in the sufficiency proof of Theorem 4 is a PSMT protocol, hence it is obviously a USMT$_\perp$ protocol.

Necessity: The proof here goes along similar lines as the necessity proof for PSMT in 3.1. It is sufficient to show that USMT$_\perp$ is impossible from s^* to r^* in the network $\mathcal{N}^*$ given in figure 2. For a general network $\mathcal{N}$ not satisfying the conditions mentioned above, rest of the proof will follow from Theorem 8,9. Let $\Pi_\perp$ be a $\{\{b_1\}, \{b_2\}, \{b_3\}\}$-USMT$_\perp$ protocol from s^* to r^* in the network $\mathcal{N}^*$. There must exist coin tosses $\vec{C_1'}$ such that when s^* chooses m_1 and b_3 fail-stops, r^* halts with correct output. This gives us an execution E_1' corresponding to execution E_1 in the proof of Theorem 6. We can now construct corresponding executions E_2', E_3' and E_4' and show that lemma 7 holds true on these executions. Hence $\Pi_\perp$ cannot exist. □

7 Conclusion and Open Problems

In this work, we have characterized asynchronous directed networks over which secure communication is possible. We have shown how *wire-based* modelling fails to provide protocols in directed networks even though there exists one. Hence, ours is the first *true* characterization of asynchronous directed networks. We remark that such characterizations have not been given for several popular message transmission problems in synchronous directed networks like PSMT.

We briefly discuss some open problems that are related to our work here: (a) Our work can be extended to the case of directed hypergraphs which are a more suitable model of communication in several practical scenarios. (b) Tolerating threshold adversary, we have given an efficient protocol for PSMT between two nodes in a graph provided that PSMT is possible between all pairs of nodes. Do we have an efficient protocol for PSMT when this is not the case. Or, do exponential lower bounds exist on communication complexity of any protocol. (c) Given a graph, can it be efficiently verified whether PSMT (or USMT) is possible between two nodes tolerating threshold adversary? We conjecture that this problem is not in complexity class **P**. (d) Completely asynchronous network is a worst-case assumption, as completely synchronous is best-case. It is interesting to study the problem of secure communication in partially synchronous networks.

References

[1] J Chalopin, S Das, and P Widmayer. Rendezvous of mobile agents in directed graphs. In N Lynch and A Shvartsman, editors, *Distributed Computing*, volume 6343 of *Lecture Notes in Computer Science*, pages 282–296. Springer Berlin / Heidelberg, 2010.

[2] A Choudhary, A Patra, B V Ashwinkumar, K Srinathan, and C P Rangan. On minimal connectivity requirement for secure message transmission in asynchronous networks. In *ICDCN '09: Proceedings of the 10th International Conference on Distributed Computing and Networking*, pages 148–162, Berlin, Heidelberg, 2009. Springer-Verlag.

[3] J Czyzowicz, S Dobrev, R Královič, S Miklík, and D Pardubská. Black hole search in directed graphs. In S Kutten and J Žerovnik, editors, *Structural Information and Communication Complexity*, volume 5869 of *Lecture Notes in Computer Science*, pages 182–194. Springer Berlin / Heidelberg, 2010.

[4] Y Desmedt and Y Wang. Perfectly secure message transmission revisited. In *EUROCRYPT '02: Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pages 502–517, London, UK, 2002. Springer-Verlag.

[5] D Dolev, C Dwork, O Waarts, and M Yung. Perfectly secure message transmission. *Foundations of Computer Science, Annual IEEE Symposium on*, 0:36–45 vol.1, 1990.

[6] M J Fischer, N A Lynch, and M S Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, 1985.

[7] M Franklin and M Yung. Secure Hypergraphs: Privacy from Partial Broadcast. In *Proceedings of 27th Symposium on Theory of Computing (STOC)*, pages 36–44. ACM Press, 1995.

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

[9] M Kumar, P R Goundan, K Srinathan, and C P Rangan. On perfectly secure communication over arbitrary networks. In *Proceedings of the 21st Symposium on Principles of Distributed Computing (PODC)*, pages 193–202, Monterey, California, USA, July 2002. ACM Press.

[10] K Kurosawa and K Suzuki. Truly efficient 2-round perfectly secure message transmission scheme. *IEEE Trans. Inf. Theor.*, 55(11):5223–5232, 2009.

[11] L Lamport, R Shostak, and M Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, July 1982.

[12] F J MacWilliams and N J A Sloane. *The Theory of Error-Correcting Codes.* North-Holland, Amsterdam, 1977.

[13] A Mehta, S Agrawal, and K Srinathan. Interplay between (im)perfectness, synchrony and connectivity: The case of probabilistic reliable communication. Cryptology ePrint Archive, Report 2010/392, 2010. `http://eprint.iacr.org/`.

[14] T Rabin and M Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *STOC '89: Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 73–85, New York, NY, USA, 1989. ACM.

[15] H M Sayeed and H Abu-Amara. Perfectly secure message transmission in asynchronous networks. In *SPDP '95: Proceedings of the 7th IEEE Symposium on Parallel and Distributeed Processing*, page 100, Washington, DC, USA, 1995. IEEE Computer Society.

[16] A Shamir. How to Share a Secret. *Communications of the ACM*, 22:612–613, 1979.

[17] K Srinathan, P Raghavendra, and C P Rangan. On proactive perfectly secure message transmission. In *2th Australisian Conference on Information Security and Privacy (ACISP 07), Australia,*, July 2007.

[18] Y Wang. Robust key establishment in sensor networks. *SIGMOD Rec.*, 33(1):14–19, 2004.