Minimal Connectivity for Unconditionally Secure Message Transmission in Synchronous Directed Networks*

Manan Nayak, Shashank Agrawal, and Kannan Srinathan

Center for Security, Theory and Algorithmic Research (C-STAR),
International Institute of Information Technology, Hyderabad, 500032, India.
{manan.nayak@research.,shashank.agrawal@research.,srinathan@}iiit.ac.in

Abstract. In this paper we give the minimal connectivity required in a synchronous directed network, which is under the influence of a computationally unbounded *Byzantine* adversary that can corrupt a subset of nodes, so that Secure Message Transmission is possible between sender S and receiver R. We also show that secure communication between a pair of nodes in a given synchronous directed network is possible in both directions if and only if reliable communication is possible between them. We assume that in a network, every node is capable of computation and we model the network along the lines of [14].

Keywords: Directed networks, Connectivity, Information-theoretic security

1 Introduction

Achieving reliable and private communication is one of the fundamental problems in distributed computing. Most solutions to the problem of Secure Multi-Party Computation assume that nodes are connected by secure channels ([1],[2], [5],[11]). However, in practice, such a channel may not be present between every pair of nodes. In such a case we need to simulate the channel using a protocol. The problem of point-to-point Secure Message Transmission (SMT) studies the possibility, optimality and feasibility of protocols in which – given a distributed network where a subset of nodes may be faulty, and given a sender node S and a receiver node R – S should be able to send any message m to R such that even if all the faulty players collude with each other, R receives m reliably and the faulty players get no information about m (privacy or secrecy). The general form of this problem is usually denoted by (ϵ, δ)-SMT where ϵ denotes the error in secrecy and δ the error in reliability [4].

The problem of Secure Message Transmission has been studied under various network and corruption models. The case of synchronous directed (unicast) networks under the influence of a computationally unbounded *Byzantine* adversary

* A shorter version appeared in [9].

has been studied in depth by the research community, beginning with the work of Desmedt and Wang [3]. In [3], the authors abstract a directed network as a collection of directed channels between S and R, and find the minimum number of forward and backward channels required in a network, affected by a threshold adversary, for $(0,0)$-SMT and for $(0,\delta)$-SMT. They also give protocols over networks which satisfy the minimum connectivity requirements. Subsequently, Patra et al. [10] and Yang and Desmedt [15] generalize these results to the case of non-threshold adversary.

While the abstraction of a directed network as a collection of directed channels between S and R is suitable for networks where intermediate nodes are routers, who can only forward messages and do not have any computing power of their own, a more general way of modelling the network as digraphs with computationally capable intermediate nodes is proposed in [14]. The main result of [14] is a characterization of directed networks, under the control of a non-threshold *mixed* adversary, over which reliable message transmission (or $(1,\delta)$-SMT using the standard notation) is possible. Subsequently, in [13], the minimal connectivity requirement in a network for $(0,\delta)$-SMT is studied.

Our work is mainly inspired by the following analogous existing result: the minimum connectivity requirement for $(1,\delta)$-SMT in digraphs (characterized in [14]) is strictly *weaker* than that required for $(1,0)$-SMT in digraphs. Similarly, we ask if the minimum connectivity requirement for (ϵ,δ)-SMT in digraphs is strictly *weaker* than that required for $(0,\delta)$-SMT. The existing results appear to hint at a negative answer to the above question. Specifically, it is known that "$(0,\delta)$-SMT if and only if (ϵ,δ)-SMT" if (a) the network is abstracted as a collection of disjoint directed paths between sender and receiver [15] or if (b) the network is modelled as an undirected graph [4].

Notwithstanding, we present a characterization of the possibility of (ϵ,δ)-SMT and find that in the case of digraphs influenced by a non-threshold Byzantine adversary, there exist graphs in which (ϵ,δ)-SMT is possible while no $(0,\delta)$-SMT protocol is known. For instance, consider the network $\mathcal{G}$ given in Figure 1 with adversary structure $\mathbb{A} = \{\{b_1\},\{b_2\}\}$. We show that this digraph satisfies the necessary and sufficient condition for the existence of a (ϵ,δ)-SMT protocol as given in Theorem 5. On the other hand, no $(0,\delta)$-SMT protocol is known over $\mathcal{G}$ ([13]).

Further, to see why if intermediate nodes can compute, the results of [15] are not applicable, again consider the network $\mathcal{G}$ with the same adversary structure $\mathbb{A}$. According to Theorem 6 and Corollary 1 in [15], (ϵ,δ)-SMT from S to R tolerating $\mathbb{A}$ is possible if and only if there exists a path from S to R, or from R to S, avoiding both the nodes b_1 and b_2. Since no such path is present in the network, no protocol exists for (ϵ,δ)-SMT in $\mathcal{G}$ according to [15]. However, if we assume that every node in the network can compute, there does exist an (ϵ,δ)-SMT protocol in $\mathcal{G}$ as shown in Section 4.1.

We would like to emphasize that the main focus of this work is on the (im)possibility and not the feasibility of SMT protocols. The protocols that we give to prove the possibility of SMT are inefficient in both message and

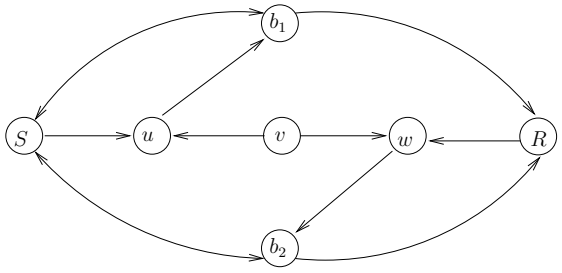

Fig. 1. Network $\mathcal{G}$.

round complexity. Previous results on SMT shed some light on the anomalous behaviour of protocols when "randomness meets directedness" [14, 12], which makes it extremely hard to design worst case efficient protocols.

2 Model and Definitions

Network: The network is modelled as a directed graph $\mathcal{N} = (\mathbb{V}, \mathcal{E})$, where the set of vertices $\mathbb{V}$ represents the set of players and the set of edges $\mathcal{E}$ represents the perfectly secure, point-to-point, directed channels in the network. The network is assumed to be synchronous and any protocol is executed in a sequence of rounds. In each round a player can send messages to its out-neighbours, receive messages sent to it by its in-neighbours in that round and perform computations, in that order. It is assumed that the network topology is known to every player. Throughout the paper we represent the sender node by S and the receiver node by R.

Adversary: Fault in the network is modelled via a computationally unbounded *centralized* adversary that can corrupt a subset of nodes, excluding S and R, in *Byzantine* fashion [8]. This means that the corrupted nodes are in complete control of the adversary and the adversary can make them behave in any arbitrary manner. The adversary is *non-threshold* [6, 7] and is represented by an adversary structure which is the collection of all possible subsets of nodes that can be corrupted by the adversary. More formally, an adversary structure $\mathbb{A}$ is defined as: $\mathbb{A} = \{B_1, B_2, ..., B_n\}$ where $\forall i,\ B_i \subseteq \mathbb{V} \setminus \{S, R\}$. The adversary can choose to corrupt any *one* subset of players from $\mathbb{A}$ and can control their behaviour throughout the execution of the protocol. Note that the adversary is not allowed to change the subset in the middle of an execution. These subsets are also known as *failure patterns* in distributed computing. The adversary structure is monotone which means that if $B_1 \in \mathbb{A}$ then $\forall B_2$ such that $B_2 \subseteq B_1$, $B_2 \in \mathbb{A}$. The players are assumed to have no information about the corrupt subset before the beginning of the protocol. It is assumed that the adversary knows the complete protocol specification and the network topology.

We note that an adversary structure can be uniquely and concisely represented by its maximal basis.

Definition 1 (Maximal basis of $\mathbb{A}$). *: The maximal basis $\overline{\mathbb{A}}$ for an adversary structure $\mathbb{A}$ is defined as: $\overline{\mathbb{A}} = \{B \mid B \in \mathbb{A}$ and $\nexists\, X \in \mathbb{A}$ s.t. $B \subsetneq X\}$*

Throughout this paper we use $\mathbb{A}$ to denote the adversary structure and $\overline{\mathbb{A}}$ to denote its maximal basis. Following [4], the **adversary's view** consists of the messages sent and received and the coin tosses made by the corrupt nodes in each round of the protocol. Random variable $adv(m, r)$ denotes the view of the adversary when S chooses to send m and the coin tosses made by the adversary is r.

Message Space: Let $\mathbb{F}$ be the message space where $< \mathbb{F}, +, * >$ is a large finite field. All the computations are done in this field. The sender S can select any element from $\mathbb{F}$ to send to R. In any message transmission protocol we assume that S starts with a message m_S and R outputs m_R at the end. Throughout the paper we write $|H|$ to denote the cardinality of the set H and $h \in_R H$ denotes that h is uniformly chosen from H.

Definition 2 (Reliability). *A message transmission protocol is said to be δ-reliable if the probability that $m_R = m_S$ is at least $(1-\delta)$, where the probability is taken over the random coin tosses of all the players and the random coin tosses of the adversary.*

Definition 3 (Privacy). *Again following [4], a message transmission protocol is said to be ϵ-private if, for every two messages m and $m' \in \mathbb{F}$ and every r, $\sum_c |Pr[adv(m, r) = c] - Pr[adv(m', r) = c]| \leq 2\epsilon$. The probabilities are over the coin tosses of the honest players and the sum is over all possible views of the adversary.*

Definition 4 ((ϵ, δ)-SMT). *A message transmission protocol is said to be (ϵ, δ)-SMT if it is ϵ-private and δ-reliable, where ϵ and δ are negligibly small.*

Definition 5 (δ-URMT). *A message transmission protocol is said to be δ-URMT (Unconditionally Reliable Message Transmission) if it is δ-reliable.*

Definition 6 (δ-URMT$_{FK}$). *We say that a message transmission protocol tolerating adversary structure $\mathbb{A}$ is δ-URMT$_{FK}$ if for all valid Byzantine corruptions of any $B \in \mathbb{A}$, the probability that R outputs $m_R = m_S$ or knows that the set B is faulty is at least $(1 - \delta)$.*[1]

Throughout this paper we use the following terms interchangeably: (a) δ-URMT and URMT (b) δ-URMT$_{FK}$ and URMT$_{FK}$.

It should be noted that protocols with error probabilities greater than $\frac{1}{2}$ or negligibly close to $\frac{1}{2}$ in reliability or secrecy are not interesting. Instead, we would like to have protocols with these error probabilities negligibly small.

Authentication Scheme: Our protocols use the following information theoretically secure authentication code to circumvent the low connectivity in the

[1] FK stands for Fault Knowledge.

graph. Suppose two random keys k_1 and k_2 are privately shared between two parties S and R.[2] Let S send $(m, m*k_1+k_2)$ to R and let R receive (x, y). Then, R can easily check if adversary has tampered with the authenticated message by verifying if $y \stackrel{?}{=} x * k_1 + k_2$. If adversary has altered the messages en-route then with probability at least $1 - \frac{1}{|\mathbb{F}|}$, verification will fail and R will find out (see [11] for proof). In addition to this if one more key k_3 is privately shared and S sends $(m + k_3, (m + k_3) * k_1 + k_2)$ to R, then the message m remains perfectly secret, since $m + k_3$ is independent of m. We use the following notations in the paper: (i) $\chi(m, k_1, k_2) = (m, m * k_1 + k_2)$; (ii) $\zeta(m, k_1, k_2, k_3) = \chi(m+k_3, k_1, k_2) = (m+k_3, (m+k_3)*k_1+k_2)$; where $m, k_1, k_2, k_3 \in \mathbb{F}$. For brevity, we sometimes abuse the notation and write $\zeta(m, K)$ to denote $\zeta(m, k_1, k_2, k_3)$ where $K = (k_1, k_2, k_3)$.

3 URMT

In [14], Srinathan and Pandu Rangan gave the characterization of directed graphs for URMT tolerating mixed adversary (*Byzantine* and *Fail-stop*). In that paper, they prove the following theorem that reduces the problem of URMT tolerating adversary structures of arbitrary size to URMT tolerating two-sized adversary structures.

Theorem 1. *In a digraph $\mathcal{N} = (\mathbb{V}, \mathcal{E})$, a δ-URMT protocol from S to R tolerating an arbitrary adversary structure $\mathbb{A}$ ($|\mathbb{A}| \geq 2$) exists iff δ-URMT protocols tolerating every $\mathcal{A}$ s.t. $\mathcal{A} \subseteq \overline{\mathbb{A}}$ and $|\mathcal{A}| = 2$ exist, where $\delta < \frac{1}{2}$.*

Once the problem is reduced to tolerating two-sized adversary structures only, they give three constructions using which we can add virtual nodes and edges in the graph. Finally a very simple condition remains to be checked in the augmented graph which shows whether or not URMT is possible in the graph.

Since in this paper we are dealing with *Byzantine* adversary only, the three constructions in [14] collapse to a single construction. We now give that construction which shall be used extensively in the characterization for (ϵ, δ)-SMT in Section 4.

Construction of Y: For a given adversary structure $\mathcal{A} = \{B_1, B_2\}$ and a given node $u \in \mathbb{V} \setminus (B_1 \cup B_2)$ we construct the set $Y(u)$ as follows: $Y(u)$ is initialized to $\{u\}$; a node $v \in \mathbb{V} \setminus (B_1 \cup B_2)$ is added to $Y(u)$ if one of the following holds:

1. $\exists\, a \in Y(u)$ s.t. $(v, a) \in \mathcal{E}$
2. $\exists\, b \in Y(u)$ s.t. $(b, v) \in \mathcal{E}$ and $\exists\, a \in Y(u), \exists \alpha \in \{1, 2\}$ s.t. v has a path to a avoiding the set $B_{\overline{\alpha}}$ where $\overline{\alpha} = 3 - \alpha$. This path may contain nodes from B_α (see Figure 2).

The above steps are executed iteratively until no more nodes can be added. Note that nodes in $B_1 \cup B_2$ are never considered.

[2] We take no such assumption in our protocols. The protocols establish keys between parties on their own before using them.

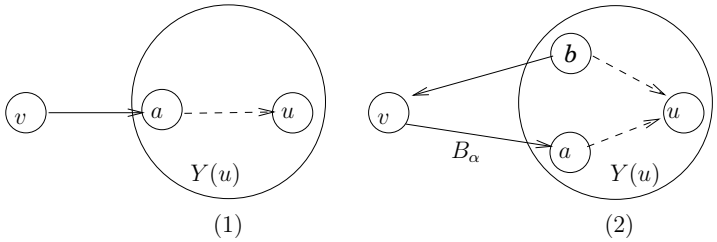

Fig. 2. Constructions for $Y(u)$.

Remark: Unlike [14], where virtual paths with certain properties are added in the graph for the above construction, we used the set notation, Y. Nevertheless, the set notation is equivalent to what is done in [14], i.e. a node v added to $Y(u)$ is equivalent to adding a virtual path from v to u in the graph.

The following two Lemmas act as the basic blocks for the (ϵ, δ)-SMT characterization.

Lemma 1. *If a node $v \in Y(u)$, then v can do δ-URMT$_{FK}$ to u, for any $\delta \geq \frac{1}{|\mathbb{F}|}$, with the additional property that the message sent will remain perfectly secret from the adversary.*

The proof of this lemma appears in [14]. They give a protocol, with the above mentioned properties, that simulates the virtual path added in the graph. Although the message remains perfectly secret throughout the protocol, it is not mentioned explicitly in the proof. Nevertheless, for the sake of completeness, we give the proof of this lemma in Appendix A.1.

Lemma 2. *If a node $v \notin Y(u)$ then there does not exist any δ-URMT protocol with $\delta < \frac{1}{2}$.*

Proof appears in [14]. It assumes that the adversary knows the message that is being transmitted. We can do away with that assumption and show that any URMT protocol with $\delta < \frac{1}{2}(1 - \frac{1}{|\mathbb{F}|})$ does not exist (a similar proof is given in [4]).

The key idea used in the proof is that if $v \notin Y(u)$, then for any message transmission protocol from v to u, the adversary can simulate a copy of the node v (which we call $\overline{v}$) on a message of its own choice in such a way that u can't distinguish between the "actual" v and the "simulated" $\overline{v}$. In this way if v intends to send message m and adversary simulates the node $\overline{v}$ on some message m' such that $m \neq m'$ [3], then u cannot do better than guessing between m and m'.

Finally, using the construction of $Y(R)$ we can restate the main theorem of [14] as follows:

Theorem 2. *In a digraph $\mathcal{N} = (\mathbb{V}, \mathcal{E})$, for $\delta < \frac{1}{2}\big(1 - \frac{1}{|\mathbb{F}|}\big)$, δ-URMT from S to R tolerating two-sized adversary structure $\mathcal{A} = \{B_1, B_2\}$ is possible if and only if*

[3] Adversary can do that with probability $1 - \frac{1}{|\mathbb{F}|}$ by choosing $m' \in_R \mathbb{F}$.

$S \in Y(R)$ and there exist two paths p_1 and p_2 from S to R with path p_α avoiding B_α for $\alpha \in \{1, 2\}$.

4 (ϵ, δ)-SMT

We now characterize the family of graphs in which (ϵ, δ)-SMT from S to R tolerating an adversary structure $\mathbb{A}$ is possible. As done in Section 3, we again start with the theorem that reduces the adversary structures of arbitrary size to two-sized adversary structures. Similar theorem has been proved in [13] for $(0, \delta)$-SMT.

Theorem 3. *In digraph $\mathcal{N} = (\mathbb{V}, \mathcal{E})$, (ϵ, δ)-SMT tolerating an arbitrary adversary structure $\mathbb{A}$ ($|\mathbb{A}| \geq 2$) is possible if and only if (ϵ, δ)-SMT tolerating $\mathcal{A}$ for all $\mathcal{A} \subseteq \overline{\mathbb{A}}$, such that $|\mathcal{A}| = 2$, is possible, where $\epsilon \leq \frac{1}{648}$ and $\delta \leq \frac{1}{864}$.*

Proof. The *only-if* part is obvious. We prove the *if* part here. Suppose that (ϵ, δ)-SMT protocols tolerating all two-sized subsets of $\overline{\mathbb{A}}$ exist. Let $\overline{\mathbb{A}} = \{B_1, B_2, ..., B_n\}$ and let $\Pi_{i,j}$ be the (ϵ, δ)-SMT protocol tolerating $\{B_i, B_j\}$ where $1 \leq i, j \leq n$. Using these as the subprotocols we construct a (ϵ, δ)-SMT protocol Π tolerating $\overline{\mathbb{A}}$ (which is also the protocol tolerating $\mathbb{A}$).

We show how to construct a (ϵ, δ)-SMT protocol $\Pi'_{i,j,k}$ tolerating $\{B_i, B_j, B_k\}$ using $\Pi_{i,j}, \Pi_{j,k}$ and $\Pi_{k,i}$. The protocol $\Pi'_{i,j,k}$ is a $(6\epsilon, 12\delta)$-SMT protocol as will be shown in Lemmas 3 and 4. Further, in Lemma 5, we will show how this protocol can be used to construct an (ϵ, δ)-SMT protocol $\Pi_{i,j,k}$ (the upper bounds on ϵ and δ become critical here). The key idea used in the construction of $\Pi'_{i,j,k}$ is that each of the subsets B_i, B_j and B_k are tolerated in two of the three protocols which means that no matter which set is corrupt, two of them will be successful. Similar process can be used to construct a protocol $\Pi_{i,j,k,l}$ using protocols $\Pi_{i,j,k}$, $\Pi_{i,j,l}$ and $\Pi_{j,k,l}$. In general for any $\mu > 2$, a μ-sized set H can be divided into three $\lceil \frac{2\mu}{3} \rceil$-sized subsets H_1, H_2 and H_3 such that every element $h \in H$ occurs in at least two of H_1, H_2 and H_3. In this way, ultimately the grand protocol Π that tolerates all the n subsets simultaneously is constructed. It can be easily shown that $poly(n)$ sub-protocols are used to construct the protocol Π.

The protocol $\Pi'_{i,j,k}$ consists of 3 phases where in each phase, protocols $\Pi_{i,j}$, $\Pi_{j,k}$ and $\Pi_{k,i}$ are run in parallel. Phase 2 begins only after the completion of Phase 1 and similarly Phase 3 begins only after the completion of Phase 2.[4] The protocol proceeds in the following steps:

- S chooses 3 set of keys K_1, K_2 and K_3 randomly from $\mathbb{F}^3$ where $K_i = (k_{i1}, k_{i2}, k_{i3})$, $i \in \{1, 2, 3\}$.
- S sends $\zeta(m_S, K_1)$, $\zeta(m_S, K_2)$ and K_3 through the protocol $\Pi_{i,j}$ in phases 1, 2 and 3 respectively. Similarly S sends $\zeta(m_S, K_2), \zeta(m_S, K_3)$ and K_1 through the protocol $\Pi_{j,k}$ and sends $\zeta(m_S, K_3)$, $\zeta(m_S, K_1)$ and K_2 through the protocol $\Pi_{k,i}$ in phases 1, 2 and 3 respectively.

[4] Although Phase 1 and 2 are separated just for better understanding, it is crucial that Phase 3 begins only after Phases 1 and 2 have ended.

– Let R receive $(x_1^{i,j}, y_1^{i,j}), (x_2^{i,j}, y_2^{i,j})$ and K'_3 from $\Pi_{i,j}$ in phases 1, 2 and 3 respectively. Similarly R receives $(x_2^{j,k}, y_2^{j,k}), (x_3^{j,k}, y_3^{j,k})$ and K'_1 from $\Pi_{j,k}$, and $(x_3^{k,i}, y_3^{k,i}), (x_1^{k,i}, y_1^{k,i})$ and K'_2 from $\Pi_{k,i}$ in phases 1, 2 and 3 respectively (see Figure 3) where $K'_i = (k'_{i1}, k'_{i2}, k'_{i3})$.
– R tries to find an $\alpha \in \{i, j, k\}$ such that the messages received through the two protocols tolerating B_α are *consistent* with each other. For instance, the messages received through two protocols tolerating B_i ($\Pi_{i,j}$ and $\Pi_{k,i}$) are *consistent* with each other when $(x_2^{i,j}, y_2^{i,j}) = \chi(x_2^{i,j}, k'_{21}, k'_{22})$ and $(x_3^{k,i}, y_3^{k,i}) = \chi(x_3^{k,i}, k'_{31}, k'_{32})$ and $x_2^{i,j} - k'_{23} = x_3^{k,i} - k'_{33}$.
  - If more than one such α exists, proceed with any one of them. If no such α exists then choose $\alpha \in_R \{i, j, k\}$ and proceed.
  - If α is i then output $x_2^{i,j} - k'_{23}$. Similarly if α is j then output $x_3^{j,k} - k'_{33}$ and if α is k then output $x_1^{k,i} - k'_{13}$.

□

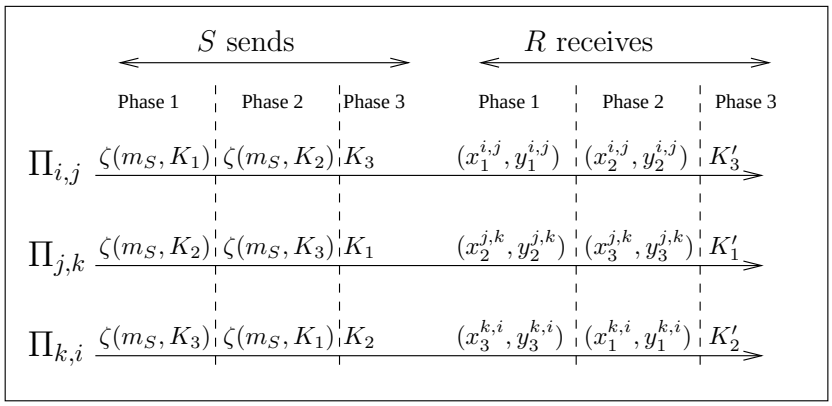

Fig. 3. Protocol $\Pi'_{i,j,k}$

We give proof ideas for the following three lemmas here. Formal proofs of Lemma 3, Lemma 4 and Lemma 5 appear in Appendix A.2, A.3 and A.4 respectively.

Lemma 3. *Protocol $\Pi'_{i,j,k}$ is (12δ)-reliable.*

Proof idea: With probability at least $(1-\delta)^{12}$, R will be able to find a pair of protocols such that the messages received through them are *consistent* and the message that R finally outputs is m_S.

Lemma 4. *Protocol $\Pi'_{i,j,k}$ is (6ϵ)-secure.*

Proof idea: The messages sent through the protocol, that is not tolerating the corrupt set, can be completely revealed to the adversary. In that case there are six messages that are sent along the other two ϵ-secret protocols that are such that m_S remains secret iff these 6 messages remain secret. This in turn shows that $\Pi'_{i,j,k}$ is (6ϵ)-secure.

Lemma 5. *An (ϵ, δ)-SMT protocol $\Pi_{i,j,k}$ can be constructed using a $(6\epsilon, 12\delta)$-SMT protocol $\Pi'_{i,j,k}$.*

Proof idea: To enhance reliability we can repeat the protocol thrice and let R output the majority element. This brings the error in reliability down to $432\delta^2$ but increases the error in secrecy to 18ϵ. Next, to enhance security, any message m is sent by sending f and $m + f$ in separate executions, where $f \in_R \mathbb{F}$. This reduces the error in secrecy to $648\epsilon^2$ but increases the error in reliability to $864\delta^2$. For the given upper bounds on ϵ and δ, the protocol becomes (ϵ, δ)-SMT.

4.1 (ϵ, δ)-SMT Characterization

Following Theorem 3, it is now sufficient to give only a characterization for (ϵ, δ)-SMT tolerating two-sized adversary structures of the form $\mathcal{A} = \{B_1, B_2\}$.

We make use of the set Y defined in Section 3. In addition, we define two more sets Z_1 and Z_2.

Construction of Z_1: For a given adversary structure $\mathcal{A} = \{B_1, B_2\}$ and a given node $u \in \mathbb{V} \setminus (B_1 \cup B_2)$ we construct $Z_1(u)$ as follows: $Z_1(u)$ is initialized to $\{u\}$; a node $v \in \mathbb{V} \setminus (B_1 \cup B_2)$ is added to $Z_1(u)$ if one of the following hold:

1. $\exists\ a \in Z_1(u)$ s.t. $(v, a) \in \mathcal{E}$, or,
2. $\exists\ b \in Z_1(u)$ s.t. b can do $URMT_{FK}$ to v (in other words, $b \in Y(v)$) and v has a path to u avoiding the set B_2. This path may contain nodes from B_1.

The above steps are executed iteratively until no new node can be added. Nodes in $(B_1 \cup B_2)$ are never considered. This completes the construction of $Z_1(u)$. The set $Z_2(u)$ is constructed along similar lines (replacing B_2 with B_1 and vice-versa in step (2) of the iteration).

Figure 4 describes the situations in which S can be added to $Z_1(R)$.

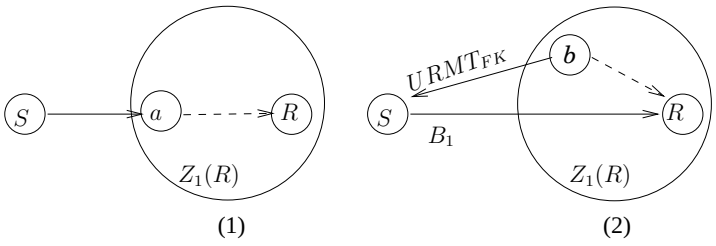

Fig. 4. Constructions for $Z_1(R)$.

Theorem 4. *In a directed network $\mathcal{N} = (\mathbb{V}, \mathcal{E})$, (ϵ, δ)-SMT from S to R tolerating $\mathcal{A} = \{B_1, B_2\}$ is possible if and only if $S \in Y(R) \cap Z_1(R) \cap Z_2(R)$.*

The proof is divided into two parts - Sufficiency (*if* part) and Necessity (*only if* part).

Sufficiency. To prove the sufficiency of the theorem we give a protocol for (ϵ, δ)-SMT from S to R, with $\epsilon \leq \frac{1}{648}$ and $\delta \leq \frac{1}{864}$. The upper bounds on the error probabilities ensure that these protocols, tolerating two-sized adversary structures, can then be used to build the final protocol tolerating the complete adversary structure. The protocol makes use of the 3 properties of S viz. $S \in Y(R)$, $S \in Z_1(R)$ and $S \in Z_2(R)$ in 3 distinct subprotocols and at the end, R, from its view of the entire protocol, outputs m_R such that $m_R = m_S$ with a very high probability and m_S remains secret.

The 3 subprotocols (corresponding to $S \in Y(R)$, $S \in Z_1(R)$ and $S \in Z_2(R)$) are as follows:

1. Subprotocol P_F: S sends m_S to R through the $URMT_{FK}$ protocol.
2. Subprotocol P_1: If S was added to $Z_1(R)$ by:
   - Construction (1), then it simply sends the message m_S to the node a through the honest edge (see Figure 4). The node a then starts another instance of the protocol P_1 to send m_S to R.
   - Construction (2), then b first chooses a random set of keys $K = (k_1, k_2, k_3)$ and sends it to S through $URMT_{FK}$ (see Figure 4). Let S receive K'. Since $S \in Z_1(R)$, S has a path to R that avoids B_2. Let that path be p_2.
     - If S successfully verifies K', it sends $\zeta(m_S, K')$ to R along the path p_2. In addition to this it also sends $m' \in_R \mathbb{F}$ to R along p_2.
     - If the verification fails then S knows the identity of the corrupt set with very high probability. Let $\mathcal{I}_S$ denote the identity knowledge of S. First S chooses $(f_1, f_2) \in_R \mathbb{F}^2$ on its own and sends (f_1, f_2) to R through p_2 (thus, tries to inform R that it didn't receive the keys from b). Next, if $\mathcal{I}_S = B_1$ then S sends $m' \in_R \mathbb{F}$ to R through p_2. But if $\mathcal{I}_S = B_2$ then S sends m_S to R through that path.

     Let R receive (x, y) and m_p. Now b starts new instances of protocol P_1 to send the elements of key K to R.
3. Subprotocol P_2: P_2 is exactly same as P_1 with B_1 replaced with B_2 and vice-versa.

Computation by R: At the end of the subprotocol P_F, R either receives m_S or knows the identity of the corrupt set with probability at least $1 - \frac{1}{|\mathbb{F}|}$. Let $\mathcal{I}_R$ be its identity knowledge. If R receives m' which it is able to verify then it outputs $m_R = m'$ and stops. Otherwise, if $\mathcal{I}_R = B_2$ it ignores all the messages it received from P_2 and does the following computation on the messages received through P_1 (analogous behaviour when $\mathcal{I}_R = B_1$). If S was added to $Z_1(R)$ by Construction (1), then R simply receives m_S (recursively) from node a. In case of Construction (2), it receives a set of keys $K^R = (k_1^R, k_2^R, k_3^R)$ from b. From

S it receives one authenticated message (x, y) and a plain message m_p. If R is able to verify (x, y) with K^R then it outputs $m_R = x - k_3^R$ otherwise it outputs $m_R = m_p$.

We now prove that this protocol is δ-reliable and ϵ-secure such that we can make ϵ and δ arbitrarily small by increasing the size of $\mathbb{F}$.

RELIABILITY: Suppose w.l.o.g. that B_2 is corrupt. At the end of the subprotocol P_F, if R outputs the message m' then $Pr[m_S = m'] \geq 1 - \frac{1}{|\mathbb{F}|}$, else $Pr[\mathcal{I}_R = B_2] \geq 1 - \frac{1}{|\mathbb{F}|}$. Now consider the execution of P_1. We initially assume that the instances of P_1, that are called inside P_1 recursively, finish successfully. In case of Construction (1), R simply receives m_S from node a. In case of Construction (2) since B_2 is corrupt, whatever S sends to R through the path p_2, that avoids B_2, reaches R with perfect reliability and secrecy. We also know that $Pr[K' = K \vee \mathcal{I}_S = B_2] \geq 1 - \frac{1}{|\mathbb{F}|}$, where K' are the keys S receives from b. If $K' = K$ then R will be able to verify (x, y) with the keys K that it receives from b and output $m_S = x - k_3$. Otherwise, when $\mathcal{I}_S = B_2$, S sends $(x, y) = (f_1, f_2) \in_R \mathbb{F}^2$ to R along with m_S. Therefore (x, y) will not verify with the keys K with probability at least $1 - \frac{1}{|\mathbb{F}|}$ and hence R is informed to output $m_p = m_S$. We can easily find the success probability as follows: Let Ev be the event that verification of (x, y) at R fails given that S had sent $(f_1, f_2) \in_R \mathbb{F}^2$.

$$\begin{aligned} Pr[m_R = m_S] &\geq Pr[m' = m_S \vee \mathcal{I}_R = B_2] * Pr[K' = K \vee \mathcal{I}_S = B_2] * Pr[Ev] \\ &\geq \big(1 - \frac{1}{|\mathbb{F}|}\big) * \big(1 - \frac{1}{|\mathbb{F}|}\big) * \big(1 - \frac{1}{|\mathbb{F}|}\big) \\ &\geq \big(1 - \frac{3}{|\mathbb{F}|}\big) \end{aligned}$$

Hence, the protocol is $\frac{3}{|\mathbb{F}|}$-reliable. This argument can be further extended to show that through this protocol even if S sends a set of messages M_S ($|M_S| > 1$), in parallel, the probability that R receives all of them reliably is still at least $1 - \frac{3}{|\mathbb{F}|}$. This can be shown by replacing single messages in the probability expressions by message sets and they shall be considered equal only when all the messages in them are equal. The main reason behind the error probability not increasing is that the fault knowledge ($\mathcal{I}_S$ or $\mathcal{I}_R$), once achieved, can be reused.

The above probabilities are conditioned on the fact that all the messages sent through instances of Protocol P_1 that are invoked recursively inside P_1 itself are received reliably. At most there can be t such recursive calls to the Protocol P_1, where $t = |\mathbb{V}|$, which are all $\frac{3}{|\mathbb{F}|}$-reliable. If we choose $\mathbb{F}$ such that $|\mathbb{F}| \geq 864 * 3t$, we get $\delta = \frac{3t}{|\mathbb{F}|} \leq \frac{1}{864}$.

SECRECY: Suppose w.l.o.g. that B_1 is corrupt. Adversary's view will only consist of the messages sent through the corrupt paths (that contain nodes from B_1). We already know that messages sent through $URMT_{FK}$ remain perfectly secret from the adversary. Hence we will only consider adversary's view as the messages sent by S to R along p_2, that is the path avoiding B_2 in protocol P_1. Now we prove using induction that messages sent from S to R using P_1 remain $\frac{1}{|\mathbb{F}|}$-secret

under the assumption that messages sent by nodes already in $Z_1(R)$ remain $\frac{1}{|\mathbb{F}|}$-secret. In case of Construction (1), S simply sends m_S to a and since a was already in $Z_1(R)$, m_S remains $\frac{1}{|\mathbb{F}|}$-secret when it is sent from a to R. Now we discuss Construction (2). Take the case when adversary alters the keys sent by b to S through URMT$_{FK}$. With probability at least $1 - \frac{1}{|\mathbb{F}|}$, S will find out that B_2 is corrupt and in that case all the messages sent along path p_2 will be independent of m_S. But with probability at most $\frac{1}{|\mathbb{F}|}$, S may get the wrong fault information in which case it will send m_S in plaintext along p_2. In any case, the authenticated message (x, y) conveys no additional information about m_S to the adversary. Hence we consider the view of the adversary as the plain message m_p sent along p_2 and find the error in secrecy.

$$\forall m, r, Pr[adv(m, r) = m] = \frac{1}{|\mathbb{F}|} * 1 + (1 - \frac{1}{|\mathbb{F}|}) * \frac{1}{|\mathbb{F}|} = \frac{2}{|\mathbb{F}|} - \frac{1}{|\mathbb{F}|^2}$$

$$\forall m, m', r, s.t., m \neq m' Pr[adv(m, r) = m'] = (1 - \frac{1}{|\mathbb{F}|}) * \frac{1}{|\mathbb{F}|} = \frac{1}{|\mathbb{F}|} - \frac{1}{|\mathbb{F}|^2}$$

$$\Rightarrow \forall m, m', r \sum_c |Pr[adv(m, r) = c] - Pr[adv(m', r) = c]| \leq \frac{2}{|\mathbb{F}|}$$

where the sum is over all possible views of the adversary, i.e. $c \in \mathbb{F}$. Since the sum is bounded by $\frac{2}{|\mathbb{F}|}$, the protocol is $\frac{1}{|\mathbb{F}|}$-secret.

Now take the case when the adversary does not alter the keys sent by b to S. In that case, (x, y) and m_p sent along p_2 are independent of m_S as long as the keys K remain secret. Hence the secrecy of the protocol completely depends upon the secrecy of K which is sent to R by b. But we know that messages sent by b remain $\frac{1}{|\mathbb{F}|}$-secret and hence, the complete protocol is $\frac{1}{|\mathbb{F}|}$-secret. We already chose $\mathbb{F}$ such that $|\mathbb{F}| \geq 864 * 3t \geq 648$, therefore $\epsilon = \frac{1}{|\mathbb{F}|} \leq \frac{1}{648}$.

Thus we prove the sufficiency.

Necessity. It is obvious that $S \in Y(R)$ is necessary for (ϵ, δ)-SMT because it is necessary for URMT alone from S to R. For the same reason the two paths (not necessarily distinct) avoiding sets B_1 and B_2 respectively are also necessary for (ϵ, δ)-SMT . Now we show that $S \in Z_1(R)$ and $S \in Z_2(R)$ are necessary too. We prove the necessity of $S \in Z_1(R)$ and the proof for the latter is similar.

Lemma 6. *$S \in Z_1(R)$ is necessary for (ϵ, δ)-SMT from S to R.*

Proof. Let $S \notin Z_1$ (in this proof, we simply write Z_1 to denote $Z_1(R)$). We know that S has a path avoiding B_2 to R. Therefore the reason behind S not being in Z_1 is that there is no node in Z_1 that can do $URMT_{FK}$ to S. We now show that there does not exist any (ϵ, δ)-SMT protocol from S to R in that case. Suppose, for contradiction, that there exists such a protocol. We now divide the set of honest nodes *not* in Z_1 into the following sets:

- $X_R = \{x \mid \exists\, a \in Z_1 \, s.t. \, a$ can do $URMT_{FK}$ to $x\}$

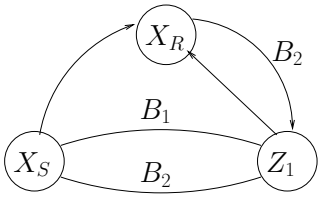

Fig. 5. Connections between the disjoint sets.

– $X_S = \{x \mid x \notin X_R\}$

From the definition of Z_1 and the above sets the following facts are clear: (i) X_S, X_R and Z_1 are disjoint and $X_S \cup X_R \cup Z_1 = \mathbb{V} \setminus (B_1 \cup B_2)$; (ii) $R \in Z_1$; (iii) $S \in X_S$; (iv) $\forall\, u \in Z_1 \cup X_R$, u cannot do $URMT_{FK}$ to any node in X_S; (v) $\forall x \in X_R$, any path from x to Z_1 will have to pass through some node in B_2 otherwise x would be in Z_1. Figure 5 describes the possible connections between the sets. A path p (of a particular kind) from a set H_1 to a set H_2 means that $\exists h_1 \in H_1, \exists h_2 \in H_2, s.t.$ there is a path p (of that kind) from h_1 to h_2. For example the edge from X_R to Z_1 labelled B_2 means that $\exists x \in X_R$, $\exists z \in Z_1$, s.t. there is a path from x to z that passes through some nodes in B_2. Paths with no labels are honest paths.

Note that, given the constraints, these are the best possible connections for the feasibility of the protocol in the graph. For instance there may or may not be an honest path from set X_S to X_R, but we have assumed there is. We shall now give an adversary strategy to prove the impossibility of (ϵ, δ)-SMT in the above graph which will imply that (ϵ, δ)-SMT will be impossible in all the other graphs where $S \notin Z_1$.

The adversary always corrupts one of $\{B_1, B_2\}$. We describe later how it chooses which set to corrupt. The corrupt set B_α behaves as follows:

– It does not send any messages to Z_1, X_R and $B_{\overline{\alpha}}$ and also ignores all the messages it receives from these sets. Here $\overline{\alpha} = 3 - \alpha$.
– It simulates a copy of each node in Z_1 and X_R. Call the simulated sets of nodes $\overline{Z_1}$ and $\overline{X_R}$ respectively. The simulation is carried out as described in [14].

Notice that since Z_1 and X_R can't do $URMT_{FK}$ to X_S, from Lemma 2 we know that the adversary will always be able to successfully simulate $\overline{Z_1}$ and $\overline{X_R}$ and thereby will be able to confuse X_S between the messages it receives from the "actual" and the "simulated" sets. Also note that "X_R can't do $URMT_{FK}$ to X_S" is independent of whether X_S has an honest path to X_R or not.

Observe that one of $\{B_1, B_2\}$ is always corrupt. Let B_α be the corrupt set. The "simulated" sets interact only with B_α and the "actual" sets interact only with $B_{\overline{\alpha}}$. In this way if M_R is the set of messages Z_1 intends to send to X_S, then X_S will receive M_R^1 from B_1 and M_R^2 from B_2.

Consider the case when B_2 is corrupt. In this case: (a) Z_1 will only receive messages from X_S sent along the path avoiding B_2, (b) Z_1 will not receive any

message from X_R, (c) $Pr[M_R^1 = M_R] = 1$ and for $|M_R| \geq 1$, $Pr[M_R^2 = M_R] \leq \frac{1}{|\mathbb{F}|}$.

We now describe how the adversary chooses which set to corrupt. Consider the event E when X_S sends some set of messages M_S along the path containing B_1 (and avoiding B_2) such that m_S can be recovered by R from the knowledge of M_S and M_R^1 only, i.e. without the knowledge of M_R^2. For a given protocol, the adversary strategy depends on $Pr[E]$:

- Case 1: if $Pr[E] \leq \frac{1}{2}$, then corrupt B_2
- Case 2: if $Pr[E] > \frac{1}{2}$, then corrupt B_1.

It is easy to see that with such a strategy a (ϵ, δ)-SMT protocol will not exist with $\delta < \frac{1}{2}\big(1 - \frac{1}{|\mathbb{F}|}\big)$ and $\epsilon < \frac{1}{2}$ simultaneously which means that these error probabilities cannot be made arbitrarily small.

In Case 1, B_2 is corrupt and hence R receives messages only from X_S that were sent along the path containing B_1 (and avoiding B_2). Hence R can recover m_S if E happens. In addition to this, even if E does not happen, R may be able to recover the message m_S if B_2 simulates the sets on the message set M_R itself. This means that $Pr[m_R = m_S] \leq Pr[E] * Pr[M_R \neq M_R^2] + 1 * Pr[M_R = M_R^2] \leq \frac{1}{2}\big(1 + \frac{1}{|\mathbb{F}|}\big)$.[5] Therefore $\delta \geq \frac{1}{2}\big(1 - \frac{1}{|\mathbb{F}|}\big)$. In Case 2, B_1 is corrupt and hence if E happens B_1 will always be able to recover m_S from M_S if it knows M_R^1. Since M_R^1 was the set of messages on which it simulated the copy of Z_1, it knows M_R^1. Therefore in this case, since $Pr[E] > \frac{1}{2}$, it gets the message with probability $> \frac{1}{2}$, i.e. $\epsilon > \frac{1}{2}$.

This completes the proof of necessity. □

Combining the result of Theorem 3 and Theorem 4 we can now give the Main Theorem of the paper that gives the complete characterization of directed networks in which (ϵ, δ)-SMT is possible.

Theorem 5. *In a directed network $\mathcal{N} = (\mathbb{V}, \mathcal{E})$, (ϵ, δ)-SMT from S to R tolerating $\mathbb{A}$ is possible if and only if for every $\mathcal{A} = \{B_1, B_2\}$ where $\mathcal{A} \subseteq \overline{\mathbb{A}}$, we have $S \in Y(R) \cap Z_1(R) \cap Z_2(R)$ where $Y(R)$, $Z_1(R)$ and $Z_2(R)$ are defined for a particular $\{B_1, B_2\}$ as described in Section 3 and Section 4.1.*

Proof of the theorem is immediate from Theorem 3 and Theorem 4.

We can now see that (ϵ, δ)-SMT, tolerating $\mathbb{A} = \{\{b_1\}, \{b_2\}\}$, is possible over the network $\mathcal{G}$ in Figure 1 with the help of the above theorem. Notice that there is only one 2-sized subset of $\mathbb{A}$ that needs to be considered, which is $\mathbb{A}$ itself.

We construct sets $Y(R)$, $Z_1(R)$ and $Z_2(R)$ for $B_1 = \{b_1\}$ and $B_2 = \{b_2\}$. S is added to $Y(R)$ through the following steps: w is first added to $Y(R)$ through Construction 2, v is then added through Construction 1, u is then added through Construction 2, and finally, S is added through Construction 1. Hence $S \in Y(R)$. We can follow similar steps to show $R \in Y(S)$. Now, since S has a path to R avoiding b_2 and $R \in Y(S)$, $S \in Z_1(R)$. Similarly, $S \in Z_2(R)$, which further implies that $S \in Y(R) \cap Z_1(R) \cap Z_2(R)$. Thus, (ϵ, δ)-SMT is possible from S to R.

[5] If $|M_R| = 0$ then $Pr[m_R = m_S] \leq Pr[E]$.

5 Concluding Remarks

From the above characterization it follows that URMT between two nodes u and v in both the directions is necessary and sufficient for (ϵ, δ)-SMT between them. URMT between u and v implies that for any given adversary structure $\mathcal{A} = \{B_1, B_2\}$ ($\mathcal{A} \subseteq \overline{\mathbb{A}}$), the following holds:

1. $v \in Y(u)$ and $u \in Y(v)$
2. u has path p_1 and p_2 to v with p_α avoiding nodes from B_α
3. v has paths q_1 and q_2 to u with q_α avoiding nodes from B_α.

(1) and (2) $\Rightarrow u \in Y(v) \cap Z_1(v) \cap Z_2(v)$; (1) and (3) $\Rightarrow v \in Y(u) \cap Z_1(u) \cap Z_2(u)$. Therefore (ϵ, δ)-SMT between u and v is possible in both directions.

This is in line with the existing results in literature, e.g. in both directed and undirected graphs, Perfectly Reliable Message Transmission (PRMT) between two nodes in both directions implies Perfectly Secure Message Transmission (PSMT) between them.

We leave it as an open problem to devise worst case efficient protocols or to characterize graphs over which efficient protocols for (ϵ, δ)-SMT exist.

Acknowledgements. The first author thanks Microsoft Research India and Indian Association for Research in Computing Science (IARCS) for their generous support towards travel expenses.

References

[1] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness Theorems for Non-cryptographic Fault-tolerant Distributed Computation. In *Proceedings of the 20th Symposium on Theory of Computing (STOC)*, pages 1–10. ACM Press, 1988.

[2] D. Chaum, C. Crepeau, and I. Damgard. Multi-party Unconditionally Secure Protocols. In *Proceedings of 20th Symposium on Theory of Computing (STOC)*, pages 11–19. ACM Press, 1988.

[3] Y. Desmedt and Y. Wang. Perfectly Secure Message Transmission Revisited. In *Proceedings of Advances in Cryptology EUROCRYPT '02*, volume 2332 of *Lecture Notes in Computer Science (LNCS)*, pages 502–517. Springer-Verlag, 2002.

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

[5] O. Goldreich, S. Micali, and A. Wigderson. How to Play any Mental Game. In *Proceedings of the 19th Symposium on Theory of Computing (STOC)*, pages 218–229. ACM Press, 1987.

[6] M. Hirt and U. Maurer. Complete Characterization of Adversaries Tolerable in Secure Multi-party Computation. In *Proceedings of the 16th Symposium on Principles of Distributed Computing (PODC)*, pages 25–34. ACM Press, August 1997.

[7] M.V.N.A. Kumar, P. R. Goundan, K. Srinathan, and C. Pandu 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.

[8] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.
[9] Manan Nayak, Shashank Agrawal, and Kannan Srinathan. Minimal connectivity for unconditionally secure message transmission in synchronous directed networks. In *ICITS*, pages 32–51, 2011.
[10] Arpita Patra, Bhavani Shankar, Ashish Choudhary, K. Srinathan, and C. Rangan. Perfectly secure message transmission in directed networks tolerating threshold and non threshold adversary. In Feng Bao, San Ling, Tatsuaki Okamoto, Huaxiong Wang, and Chaoping Xing, editors, *Cryptology and Network Security*, volume 4856 of *Lecture Notes in Computer Science*, pages 80–101. Springer Berlin / Heidelberg, 2007.
[11] T. Rabin and M. Ben-Or. Verifiable Secret Sharing and Multiparty Protocols with Honest Majority. In *Proceedings of the 21st Symposium on Theory of Computing (STOC)*, pages 73–85. ACM Press, 1989.
[12] Bhavani Shankar, Prasant Gopal, Kannan Srinathan, and C. Pandu Rangan. Unconditionally reliable message transmission in directed networks. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1048–1055, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.
[13] Kannan Srinathan, Arpita Patra, Ashish Choudhary, and C. Pandu Rangan. Unconditionally secure message transmission in arbitrary directed synchronous networks tolerating generalized mixed adversary. In *ASIACCS '09: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pages 171–182, New York, NY, USA, 2009. ACM.
[14] Kannan Srinathan and C. Pandu Rangan. Possibility and complexity of probabilistic reliable communications in directed networks. In *Proceedings of 25th ACM Symposium on Principles of Distributed Computing (PODC'06)*, 2006.
[15] Qiushi Yang and Yvo Desmedt. Cryptanalysis of secure message transmission protocols with feedback. In Kaoru Kurosawa, editor, *Information Theoretic Security*, volume 5973 of *Lecture Notes in Computer Science*, pages 159–176. Springer Berlin / Heidelberg, 2010.

A Appendix

A.1 Proof of Lemma 1

We give a proof by induction on the iteration at which a node is added. We denote by Π_v the URMT$_{FK}$ protocol which is run in the network to enable v to send a message to u.

Base Step: The first node added to $Y(u)$ is u which can obviously send any message reliably and securely to itself. Hence, Π_u is trivial.

Induction Step: Assume that $k-1$ nodes $v_1, v_2, \ldots, v_{k-1}$ ($v_1 = u$) have been added to $Y(u)$ in that order. At the k-*th* iteration, node v_k is added. Let m_k be the message v_k intends to send. Protocol Π_{v_k} proceeds as follows. If v_k was added to $Y(u)$ by:

– Construction (1), then there exists a node $v_i \in Y(u)$ ($1 \leq i \leq k-1$) s.t. v_k has an honest path p to v_i. First, v_k sends message m_k to v_i along path p,

which v_i receives reliably and securely. Now, protocol Π_{v_i} is run on message m_k in the network. As Π_{v_i} is a URMT$_{FK}$ protocol with perfect secrecy, so is Π_{v_k}.

– Construction (2), then there exist two nodes $v_i, v_j \in Y(u)$ $(1 \leq i, j \leq k-1)$ s.t. v_i has an honest path p_1 to v_k and v_k has a path p_2 passing through at most one of B_1 and B_2 to v_j. Let p_2 pass through B_α. Protocol Π_{v_k} proceeds in the following sequence of steps:
  1. Node v_i chooses three random keys k_1, k_2, $k_3 \in_R \mathbb{F}$ and sends them along path p_1 to v_k which v_k receives reliably and securely.
  2. Node v_k sends $\zeta(m_k, k_1, k_2, k_3)$ to v_j along p_2.
  3. Let v_j receive (f_1, f_2) along p_2. If v_j does not receive two field elements along p_2, it picks two elements $f_1, f_2 \in_R \mathbb{F}$ on its own[6]. Now, protocol Π_{v_j} is run twice in the network, first on message f_1, then on message f_2.
  4. Protocol Π_{v_i} is run thrice in the network, first on message k_1, then on message k_2 and then on k_3.

Since both Π_{v_j} and Π_{v_i} are URMT$_{FK}$ protocols with perfect security, any tampering of the messages sent through either of them is detected with probability at least $(1 - \frac{1}{|\mathbb{F}|})$. If there is no tampering of the message sent through these protocols then u receives f_1, f_2 and keys k_1, k_2, k_3 reliably and securely. Therefore u will be able to recover the message m_k or detect any tampering by B_α $(\alpha \in \{1, 2\})$ on path p_2 with at least $(1 - \frac{1}{|\mathbb{F}|})$ probability (due to the property of the authentication code). Hence, Π_{v_k} is a URMT$_{FK}$ protocol. Also, since adversary does not know k_1, k_2 and k_3, it gets no information about m from $\zeta(m, k_1, k_2, k_3)$.

A.2 Proof of Lemma 3

Each one of protocols $\Pi_{i,j}$, $\Pi_{j,k}$ and $\Pi_{k,i}$ are (ϵ, δ)-SMT protocols. Let us suppose w.l.o.g. that B_i is corrupt. It means that protocols $\Pi_{i,j}$ and $\Pi_{k,i}$ will be δ-reliable and ϵ-secret. Therefore, with probability at least $(1 - \delta)^{10}$, R will receive the following 10 elements sent through $\Pi_{i,j}$ and $\Pi_{k,i}$ reliably: $\zeta(m_S, K_2)$, K_3, $\zeta(m_S, K_3)$ and K_2.[7] In that case protocols $\Pi_{i,j}$ and $\Pi_{k,i}$ will be consistent with each other and $m_R = x_2^{i,j} - k'_{23}$ will be equal to m_S if R chooses α to be i. But α can have other possible values also. The corrupt set B_i can read and alter all the messages that are sent through the protocol $\Pi_{j,k}$. Suppose, in Phase 3, it modifies K_1 to K'_1 such that a different message, $m' \neq m_S$, is recovered when $\zeta(m_S, K_1)$ is unlocked using K'_1. Then it also must modify at least one of $\zeta(m_S, K_2)$ (in Phase 2) and $\zeta(m_S, K_3)$ (in Phase 3) such that the verification passes and the message recovered is m'. But the probability that both these verifications fail (if altered) is at least $(1 - \frac{1}{|\mathbb{F}|})^2$, since adversary does not know the keys K_2 and K_3 during Phase 1 and 2 (this is why it is crucial that Phase

[6] This is an attempt to inform R that path p_2 (and thus, B_α) is corrupt.

[7] Recall that $\zeta(m_S, K)$ consists of 2 field elements and K consists of 3 field elements for any $K \in \{K_1, K_2\}$

3 begins only after the completion of Phases 1 and 2). Hence the probability that R chooses i as α, given that it received the 10 elements reliably, is at least $(1 - \frac{1}{|\mathbb{F}|})^2$.

$$\Rightarrow Pr[m_R = m_S] \geq (1-\delta)^{10} * (1 - \frac{1}{|\mathbb{F}|})^2$$

$$\Rightarrow Pr[m_R = m_S] \geq (1-\delta)^{12}, \text{ choose } \mathbb{F} \text{ such that } \delta \geq \frac{1}{|\mathbb{F}|}$$

$$\Rightarrow Pr[m_R = m_S] \geq 1 - 12\delta, \text{ since } \delta \leq \frac{1}{864}$$

Note that there is another way in which B_i can always pass the verifications, i.e. by not altering any messages. But it won't affect this probability because in that case m_R will always be equal to m_S no matter what value of α is chosen. Hence the protocol $\Pi_{i,j,k}$ is (12δ)-reliable.

A.3 Proof of Lemma 4

Suppose w.l.o.g. that B_i is corrupt. Therefore $\Pi_{j,k}$ will fail completely and hence $\zeta(m_S, K_2)$, $\zeta(m_S, K_3)$ and K_1 will be revealed to the adversary. But we see that these messages convey no information about m_S to the adversary because of the following reasons:

- due to the property of ζ function if K is not known to the adversary then $\zeta(m_S, K)$ is independent of m_S.
- since K_1 was randomly chosen by S it has no relation with the message m_S.

Now notice that among all the messages sent to R through the protocols $\Pi_{i,j}$ and $\Pi_{k,i}$ only six contain "useful" information for the adversary viz.: $\zeta(m_S, K_1)$ and k_{33} sent through $\Pi_{i,j}$ and $\zeta(m_S, K_1)$ and k_{23} sent through $\Pi_{k,i}$. $\zeta(m_S, K_2)$ and $\zeta(m_S, K_3)$ are not useful because they are already revealed to the adversary. Only the third element of the keys K_2 and K_3, i.e. k_{23} and k_{33}, are useful because if they are known, even if adversary knows the other two elements it gains no extra information about the message m_S. Also, without the third element the other two elements give absolutely no information about m_S. For example, even if adversary knows $\zeta(m_S, K_2)$, k_{21} and k_{22} it has no information about m_S. On the other hand if it knows $\zeta(m_S, K_2)$ and k_{23}, it knows m_S completely.

Therefore there are 6 elements sent through $\Pi_{i,j}$ and $\Pi_{k,i}$ that need to be kept secret from the adversary. Let $\{a_i \,|\, 1 \leq i \leq 6\}$ be the variables representing these six elements. It can be clearly seen that if any a_i is revealed to the adversary, m_S will be revealed. For example, if the first element of $\zeta(m_S, K_1)$, i.e. $m_S + k_{13}$ sent through $\Pi_{i,j}$ is revealed then it can find out m_S since it already knows k_{13}. Similarly, it can find out m_S using any of the other 5 "useful" elements. In other words once $\zeta(m_S, K_2)$, $\zeta(m_S, K_3)$ and K_1 are revealed to the adversary (that means once they are fixed), for a given message m_S the values of all the a_i's are fixed. Also, we send all these elements through some ϵ-secret protocol. Suppose protocol P_i was used to send a_i. To find the secrecy factor of the entire protocol

$\Pi'_{i,j,k}$ we look at it as a series of 6 protocols $(P_1, P_2, ..., P_6)$. Therefore, we need to find an upper bound on the expression

$$X = \sum_c |Pr[adv(m_0, r) = c] - Pr[adv(m_1, r) = c]|, \forall m_0, m_1 \in \mathbb{F}, \forall r$$

where r denotes all the coin tosses of the adversary in the six executions combined, i.e. $r = (r_1, r_2, \ldots r_6)$. The sum is over all possible views of the adversary for the execution of the six protocols. In other words $c \in \mathbb{C} = C_1 \times C_2 \cdots \times C_6$ where C_i is the set of all possible views of the adversary for an execution of protocol P_i.

We now define the following notation for readability. $p_i(m, r, c)$ is the probability that adversary's view is c when the message sent was m and its coin tosses were r in an execution of P_i. Notice that these probabilities are over the coin tosses of honest players and hence all the six p_i's are independent of each other.

Let a_i^b be the value fixed for a_i when $m = m_b$, $b \in \{1, 2\}$. Hence we can rewrite the expression X as:

$$X = \sum_{(c_1, c_2, \ldots c_6) \in \mathbb{C}} |\prod_{i=1}^{6} p_i(a_i^0, r_i, c_i) - \prod_{i=1}^{6} p_i(a_i^1, r_i, c_i)|$$

Now we list out some properties of $p_i(m, c, r)$ which will help us in evaluating the above expression:

- $\forall m, r, \sum_{c \in C_i} p_i(m, r, c) = 1$ where $1 \leq i \leq 6$.
- $\forall m_1, m_2, r, \sum_{c \in C_i} |p_i(m_1, r, c) - p_i(m_2, r, c)| \leq 2\epsilon$ since all the protocols are ϵ-secure.

Using the result of Lemma 7 it can be easily shown that $X \leq 12.\epsilon$. Hence the protocol $\Pi'_{i,j,k}$ is (6ϵ)-secure.

Corollary 1. *If an ϵ-secret protocol is repeated k number of times then the error in secrecy increases at most by a factor of k. (In that case all the useful elements, a_i's are m itself).*

Lemma 7. *Given n pairs of vectors u_i and v_i of size l_i, i.e. $u_i = (u_{i1}, u_{i2}, \ldots u_{il_i})$ and $v_i = (v_{i1}, v_{i2}, \ldots v_{il_i})$, $i \in \{1, 2, \ldots, n\}$. Also given that $\forall i \in \{1, 2, \ldots, n\}$:*

1. $\sum_{j=1}^{l_i} u_{ij} \leq 1$ *and* $\sum_{j=1}^{l_i} v_{ij} \leq 1$
2. $\sum_{j=1}^{l_i} |u_{ij} - v_{ij}| \leq 2\epsilon$

Then $\sum_{k_1, k_2, \ldots k_n} |\prod_{i=1}^{n} u_{ik_i} - \prod_{i=1}^{n} v_{ik_i}| \leq 2n.\epsilon$, *where k_i varies from 1 to l_i.*

Proof (By Induction). Let T_n denote the sum in the expression and let $P(n)$ denote the above inequality. In other words:

$$P(n) \Rightarrow (T_n \leq 2n.\epsilon)$$

We know that $P(1)$ is true since it is given that $\sum_{k_1=1}^{l_1} |u_{1k_1} - v_{1k_1}| \leq 2\epsilon$. Suppose $P(n-1)$ is true. Therefore we have: $T_{n-1} \leq 2(n-1).\epsilon$.
Now,

$$T_n = \sum_{k_1,k_2,\ldots k_n} |\prod_{i=1}^{n} u_{ik_i} - \prod_{i=1}^{n} v_{ik_i}|$$

$$\Rightarrow T_n = \sum_{k_1,k_2,\ldots k_n} |\Big(\prod_{i=1}^{n-1} u_{ik_i} - \prod_{i=1}^{n-1} v_{ik_i}\Big).u_{nk_n} + \Big(\prod_{i=1}^{n-1} v_{ik_i}\Big)(u_{nk_n} - v_{nk_n})|$$

$$\Rightarrow T_n \leq \sum_{k_1,k_2,\ldots k_n} |\Big(\prod_{i=1}^{n-1} u_{ik_i} - \prod_{i=1}^{n-1} v_{ik_i}\Big).u_{nk_n}| + \sum_{k_1,k_2,\ldots k_n} |\Big(\prod_{i=1}^{n-1} v_{ik_i}\Big)(u_{nk_n} - v_{nk_n})|$$

$$\Rightarrow T_n \leq T_{n-1}.1 + 1.2\epsilon$$

$$\Rightarrow T_n \leq 2(n-1).\epsilon + 2\epsilon$$

$$\Rightarrow T_n \leq 2n.\epsilon$$

□

A.4 Proof of Lemma 5

Consider the protocol $\Pi''_{i,j,k}$ in which a message m is sent by sending it thrice through $\Pi'_{i,j,k}$. R outputs the majority element (if it exists). From Corollary 1 it is clear that the error in secrecy of $\Pi''_{i,j,k}$ increases to 18ϵ since it was 6ϵ for $\Pi'_{i,j,k}$. Now we find its error in reliability.

$$\begin{aligned} Pr[R \text{ outputs } m] &= 1 - Pr[R \text{ doesn't receive m in at least 2 executions}] \\ &\geq 1 - 3(12\delta)^2 \end{aligned}$$

Hence the error in reliability for $\Pi''_{i,j,k}$ is $432\delta^2$.

Now consider the protocol $\Pi_{i,j,k}$ which sends m by choosing $f \in_R \mathbb{F}$ and sending $m+f$ and f in two separate executions of $\Pi''_{i,j,k}$. It can be clearly seen that the error in reliability gets doubled because both these messages need to be received reliably. Hence error in reliability for $\Pi_{i,j,k}$ is $864\delta^2$. Now we find its error in secrecy. We claim that it gets squared, i.e. from 18ϵ it becomes $2.(18\epsilon)^2 = 648\epsilon^2$. Intuitively, the error in secrecy decreases because now any one share doesn't contain any information about the original message m and hence the message is revealed only when both $m+f$ and f are revealed. We prove formally in Lemma 8 that the error in secrecy of $\Pi_{i,j,k}$ becomes $648\epsilon^2$.

Since $\epsilon \leq \frac{1}{648}$ and $\delta \leq \frac{1}{864}$, $648\epsilon^2 \leq \epsilon$ and $864\delta^2 \leq \delta$. Hence, $\Pi_{i,j,k}$ is an (ϵ, δ)-SMT protocol.

Lemma 8. *Let P be a protocol that is ϵ-secret. If a protocol P' is such that S sends m by sending $m + f$ and f where $f \in_R \mathbb{F}$ through protocol P then the protocol P' is $2\epsilon^2$-secret.*

Proof. Protocol P is ϵ-secret, therefore we have:

$$\sum_c |p(m_0, r, c) - p(m_1, r, c)| \leq 2\epsilon$$

where $p(m, r, c)$ is the probability that adversary's view is c when the message sent was m and adversary's coin tosses are r for the execution of protocol P. Define $p'(m, r, c)$ similarly for P'. Notice that P' is nothing but two executions of P on different messages and hence adversary's view is denoted by an ordered-pair (c_1, c_2), where c_i is its view in the i^{th} execution, $i \in \{1, 2\}$. Similarly its coin tosses are denoted by (r_1, r_2). So to find the secrecy factor of P' we find an upper bound on the following expression:

$$X = \sum_{c_1, c_2} |p'(m_0, (r_1, r_2), (c_1, c_2)) - p'(m_1, (r_1, r_2), (c_1, c_2))|$$

We can write $p'(m, (r_1, r_2), (c_1, c_2))$ as the following summation:

$$p'(m, r, (c_1, c_2)) = \sum_{f \in \mathbb{F}} (\frac{1}{|\mathbb{F}|}) * p(m + f, r_1, c_1) * p(f, r_2, c_2) \tag{1}$$

Using (1) we can rewrite X as:

$$X = (\frac{1}{|\mathbb{F}|}) \sum_{c_1, c_2} |\sum_{f \in \mathbb{F}} \big[\big(p(m_0 + f, r_1, c_1) - p(m_1 + f, r_1, c_1)\big) * p(f, r_2, c_2)\big]| \tag{2}$$

Since $\forall m_1, m_2, r, c \sum_{f \in \mathbb{F}} p(m_1 + f, r, c) = \sum_{f \in \mathbb{F}} p(m_2 + f, r, c)$, we have:

$$\forall m_1, m_2, r, c, \sum_{f \in \mathbb{F}} (p(m_1 + f, r, c) - p(m_2 + f, r, c)) = 0 \tag{3}$$

We know from (3) that for any $f' \in \mathbb{F}$:

$$\sum_{f \in \mathbb{F}} [p(m_0 + f, r_1, c_1) - p(m_1 + f, r_1, c_1)] * p(f', r_2, c_2) = 0 \tag{4}$$

By (2) and (4), we get:

$$X = (\frac{1}{|\mathbb{F}|}) \sum_{c_1, c_2} |\sum_{f \in \mathbb{F}} \big[\big(p(m_0 + f, r_1, c_1) - p(m_1 + f, r_1, c_1)\big) * \big(p(f, r_2, c_2) - p(f', r_2, c_2)\big)\big]|$$

Using the triangular inequality:

$$X \leq (\frac{1}{|\mathbb{F}|}) \sum_{c_1, c_2} \sum_{f \in \mathbb{F}} \big[|\big(p(m_0 + f, r_1, c_1) - p(m_1 + f, r_1, c_1)\big)| * |\big(p(f, r_2, c_2) - p(f', r_2, c_2)\big)|\big]$$

Reversing the order of summation we get:

$$X = (\frac{1}{|\mathbb{F}|}) \sum_{f \in \mathbb{F}} \sum_{c_1} \sum_{c_2} \left[|\big(p(m_0+f, r_1, c_1) - p(m_1+f, r_1, c_1)\big)| * |\big(p(f, r_2, c_2) - p(f', r_2, c_2)\big)| \right]$$

$$X \leq (\frac{1}{|\mathbb{F}|}) * |\mathbb{F}| * (2\epsilon) * (2\epsilon)$$

$$\Rightarrow X \leq 2.(2\epsilon^2)$$

Hence P' is $2\epsilon^2$-secret. □