Secure Communication in Multicast Graphs

Qiushi Yang[⋆] and Yvo Desmedt[⋆⋆]

Department of Computer Science, University College London, UK
{q.yang,y.desmedt}@cs.ucl.ac.uk

Abstract. In this paper we solve the problem of secure communication in multicast graphs, which has been open for over a decade. At Eurocrypt '98, Franklin and Wright initiated the study of secure communication against a Byzantine adversary on multicast channels in a neighbor network setting. Their model requires node-disjoint and neighbor-disjoint paths between a sender and a receiver. This requirement is too strong and hence not necessary in the general multicast graph setting. The research to find the lower and upper bounds on network connectivity for secure communication in multicast graphs has been carried out ever since. However, up until this day, there is no tight bound found for any level of security.

We study this problem from a new direction, i.e., we find the necessary and sufficient conditions (tight lower and upper bounds) for secure communication in the general adversary model with adversary structures, and then apply the results to the threshold model. Our solution uses an extended characterization of the multicast graphs, which is based on our observation on the eavesdropping and separating activities of the Byzantine adversary.

Keywords: secure communication, reliable communication, multicast, privacy, reliability, adversary structure.

1 Introduction

In most communication networks, a *sender* S and a *receiver* R are connected by unreliable and distrusted channels. The distrust of the channels is because of the assumption that there exists an adversary who, with unbounded computational power, can control some nodes on these channels. The interplay of network connectivity and secure communication between S and R has been studied extensively (see, e.g., [2,3,6,4,13]).

Secure communication is based on the problem of *secure message transmission* (SMT) between S and R. The aim of SMT is to enable a message to be transmitted from S to R *privately* (i.e., the adversary does not learn the message) and *reliably* (i.e., R can output the message correctly). In particular, *reliable message transmission* (RMT) is essential for all transmission protocols, and hence it has

[⋆] Part of this work was done while funded by UCL PhD Studentship.

[⋆⋆] Part of this work was done while funded by EPSRC EP/C538285/1, by BT as BT Chair of Information Security, and by RCIS, AIST, Japan.

D.H. Lee and X. Wang (Eds.): ASIACRYPT 2011, LNCS 7073, pp. 538–555, 2011.

been studied exclusively. Normally there are two different measures of security or reliability: *perfect* (i.e., zero probability that the protocol fails to be secure or reliable) and *almost perfect* (i.e., an arbitrarily small probability that the protocol fails to be secure or reliable) [7].

The traditional studies of RMT and SMT consider a *point-to-point* network setting, where a sending node can transmit a message to a receiving node through a channel they choose. In the *threshold model* (t-bounded), the adversary is able to control up to t nodes in a network graph. The result by Dolev et al. [6] shows that $n > 2t$ *node-disjoint* paths are required for RMT and SMT between S and R. In [7], Franklin and Wright showed that the connectivity for almost perfect security can be reduced by using *multicast* channels.

A *multicast* channel allows a sending node to transmit a message to multiple receiving nodes. The study of secure multicast was initiated by Franklin and Yung in [9]. They used hypergraphs to model multicast networks, and studied privacy against a passive adversary (eavesdropper). Goldreich et al. [10] also studied multicast networks, but their work is in the full information model, which is different to the partial broadcast model in which we are interested. At Eurocrypt '98, Franklin and Wright [7] (see also [8]) first studied a *Byzantine* (active) adversary on multicast channels in *neighbor networks* (defined in [9]), in which a message multicast by a node is received—simultaneously and privately—by all its neighbors, where a neighbor is a node that shares a common edge with the sending node.[1] They found that with some properties of the multicast channels, only $n > t$ node-disjoint paths are needed for *almost perfectly* RMT and SMT. However, their setting is based on a strong assumption, that is, all paths between S and R must be *neighbor-disjoint* (i.e., there do not exist two paths that have a common neighbor node). Indeed, such a strong assumption may not be necessary in general multicast networks, and hence they gave the following open problem:

> ... if these n disjoint paths do not have disjoint neighborhood, then an adversary may be able to foil our protocols with $t < n$ faults by using one fault to eavesdrop on two disjoint lines. An obvious direction of further research is to characterize secure communication fully in this more general (multicast graph) setting.

Wang and Desmedt [14] further investigated the problem of secure communication in a more general multicast graph setting. They conjectured that a general connectivity (weaker than $n > t$ neighbor-disjoint) is the upper bound for achieving perfect privacy and almost perfect reliability (see Section 6 for more details). In another study, Desmedt and Wang [4] (see also [15]) extended this result. By using examples, they showed that the previously conjectured connectivity of [14] is not necessary, and they also proposed a lower bound for SMT and conjectured its tightness. Since it is very difficult to apply the threshold model in general

[1] For example, in Fig 1(a) in Section 3, when a message is multicast by node 2, it will be simultaneously received by nodes 1, 3 and 4. A multicast channel does not allow node 2 to send a message to node 1 and 3 without node 4 receiving it.

multicast graphs, up until this day, there has been no result that gives the necessary and sufficient conditions for RMT and SMT in multicast graphs.

Our contributions. We completely solve the problem of secure communication in multicast graphs (neighbor network setting), which has been open and studied for over a decade. We view this problem from a new direction. That is, our solution is based on two basic ideas: (1) a general graph setting can be applied naturally in the *general adversary model* with *adversary structures* (see, e.g., [11,13,5,17]); (2) a threshold corresponds to a special adversary structure. Thus we study multicast graphs in the general adversary model, and then apply the results to the threshold model.

We found that the current adversary structure model is not enough to characterize multicast graphs. Therefore, in Section 3, we give an extended characterization of the multicast graphs, which is based on our observation on the *eavesdropping* and *separating* activities of the adversary on the multicast channels. This characterization gives a clearer view on how the message can be securely transmitted over multicast graphs.

With the new characterization, we give the necessary and sufficient conditions for RMT and SMT respectively in Section 4 and Section 5. Besides proving that our conditions imply the lower bounds on network connectivity, we also provide message transmission protocols to show that these bounds are tight.

Finally in Section 6, we use our results in the general adversary model to find the necessary and sufficient conditions for RMT and SMT in the threshold model. Also by analyzing the previous results, we show how our results explain all the examples and prove all the conjectures in the previous work. Our final result regarding the tight bounds on network connectivity for RMT and SMT in multicast graphs is presented at the end of this paper.

2 Model

We abstract away the concrete network structure and model a *multicast communication neighbor network* by an *undirected graph* $G(V, E)$, whose nodes are the parties in the network and edges are private and authenticated multicast channels. Let $S, R \in V$, the paths between S and R are not necessarily node-disjoint.[2]

Let $\mathbb{F}$ be a *sufficiently large* finite field, we assume that $\mathbb{M} \subseteq \mathbb{F}$ is the message space from which S chooses messages. Let A be a set, we use $|A|$ to denote the number of elements in A, and we write $a \in_R A$ to indicate that a is chosen from A with respect to uniform distribution.

In the *threshold model*, an adversary can control up to t nodes in a graph, and hence control up to t node-disjoint paths. In the *general adversary model*, an adversary is characterized by an adversary structure, which is defined as follows (see [12,11]): Given a party set P, an adversary structure $\mathcal{A}$ on P is a subset

[2] Throughout the paper we consider only the simple paths. A simple path is a path with no repeated nodes.

of 2^P such that for any $A \in 2^P$, if $A \in \mathcal{A}$ and $A \supseteq A'$, then $A' \in \mathcal{A}$. The adversary is able to choose one set $A \in \mathcal{A}$ to control. It is straightforward that the threshold model is a special case of the general adversary model, because a threshold t can be seen as a special adversary structure $\mathcal{A}$ such that any set $A \in 2^P$ that has t parties or less is in $\mathcal{A}$.

In this paper we consider a *Byzantine* adversary who can exhibit an *active* behavior. A Byzantine adversary has unlimited resources and computational power. Not only can the adversary read the traffic through the parties it controls, but it can also decide, whether to deny or to modify the message, whether to follow the protocol or not, etc.

We use the security model given by Franklin and Wright [7]. Let Π be an SMT protocol. S starts with a message m^S drawn from a message space $\mathbb{M}$. At the end of Π, R outputs a message m^R. For any execution of the protocol Π, let adv be the adversary's view of the entire protocol, i.e., the behavior of the faulty nodes, the initial state of the adversary, and the coin flips of the adversary during the execution. We write $adv(m, r)$ to denote the adversary's view when $m^S = m$ and when the coin flips of the adversary are r.

Privacy. Π is ϵ-private if, for any two messages $m_1, m_2 \in \mathbb{M}$ and any r, we have $\sum_c |\Pr[adv(m_1, r) = c] - \Pr[adv(m_2, r) = c]| \leq 2\epsilon$. The probabilities are taken over the coin flips of the honest parties, and the sum is over all possible values of the adversary's view.

Reliability. Π is δ-reliable if, with probability at least $1 - \delta$, R outputs $m^R = m^S$ at the end of the protocol. The probability is over the choice of m^S and the coin flips of all parties.

Security. Π is (ϵ, δ)-secure if it is ϵ-private and δ-reliable.

We say Π is perfectly secure (PSMT) if it is a $(0, 0)$-SMT protocol. In this paper, we also discuss reliability (without requirement for privacy): δ-RMT, 0-RMT, and almost perfect security: (ϵ, δ)-SMT and $(0, \delta)$-SMT. Note that in the rest of the paper, ϵ and δ only appear when studying almost perfect security, thus we let $\epsilon > 0$ and $0 < \delta < \frac{1}{2}$.

We employ the authentication code $\text{auth}(m; a, b) = am + b$ for information-theoretically secure authentication. An authentication key $(a, b) \in_R \mathbb{F}^2$ can be used to authenticate one message m without revealing any information about the key itself.

3 Characterization of Multicast Graphs

In this section we characterize multicast graphs based on the adversary structures. We give an extended characterization which is essential for obtaining the necessary and sufficient conditions in the multicast model. This should give a clearer insight to the problems we are dealing with.

We let P be the set of all paths between S and R in a given graph $G(V, E)$. The adversary chooses a set of nodes $A \in \mathcal{A}$ to control, where $\mathcal{A}$ is an adversary structure on $V \setminus \{S, R\}$. For each path $p \in P$, we define *eavesdropping* and *separating* as follows.

Definition 1. *We say that the adversary can* eavesdrop *on p if it* cannot *control any node on p but can control some neighbors of p.*[3] *Suppose that the adversary can eavesdrop on p and there is an element a to be transmitted between S and R on p. We say that the adversary can* completely eavesdrop *on p if, despite what protocol is executed, the adversary can learn a by eavesdropping.*

Definition 2. *We say that the adversary can* separate *S and R on p if it can control some nodes on p. Suppose that the adversary can separate S and R on p and there are k elements $(a_1, \ldots, a_k) \in \mathbb{F}^k$ to be transmitted on p. We let $(a_1^S, \ldots, a_k^S)$ and $(a_1^R, \ldots, a_k^R)$ be the views of S and R respectively on these k elements at the end of any protocol. We say that the adversary can* completely separate *S and R if, despite what protocol is executed and how large k is, there exists a strategy of the adversary that causes $\forall i \ (1 \leq i \leq k) : a_i^S \neq a_i^R$.*

Next we show two lemmas regarding the eavesdropping and separating activities of the adversary on a single path $p \in P$. We assume that the path p is placed in a *left-to-right* direction, with S at the left end and R at the right end.

Lemma 1. *The adversary can completely eavesdrop on a path $p \in P$ if and only if it can eavesdrop on two adjacent nodes*[4] *on p.*

Proof. We first prove the "if" direction. The privacy problem has been studied by Franklin and Yung in [9]. They showed that private communication on p is possible only if, by removing all the faulty nodes and the hyperedges on which the faulty nodes are, path p remains.[5] Evidently, this necessary condition for privacy is satisfied if and only if the adversary *cannot* eavesdrop on two adjacent nodes on p (See Example 1 following this proof). Thus if the adversary can eavesdrop on two adjacent nodes on p, then it can completely eavesdrop on p.

Next we prove the "only if" direction. We give the following protocol, which allows S to send an element a^S to R with perfect privacy, when the adversary cannot eavesdrop on two adjacent nodes on p. First we assume that including S and R, there are $k+2$ nodes $v_0, \ldots, v_{k+1}$ on p. We let S be node v_0, R be node v_{k+1}, and $v_1, \ldots, v_k$ be the other k nodes from left to right.

Single Path Private Propagation Protocol

1. For each $1 \leq i \leq k+1$, v_i initiates an element $a_i \in_R \mathbb{F}$ and multicasts it. Thus for each $0 \leq i \leq k$, v_i receives element a_{i+1} from its right side neighbor node v_{i+1}.

[3] Obviously, if the adversary *can* control some nodes on p, then it can learn everything passing through those controlled nodes. However, for the purpose of our observation, we do not consider this activity as "eavesdropping", instead, we characterize it as "separating", which we describe in Definition 2.

[4] Two nodes $u, v \in V$ are said to be *adjacent* to one another if there is an edge $\{u, v\} \in E$ between them.

[5] In the threshold model where any t nodes can be the faulty, such connectivity is called the *weak t_{hyper}-connectivity.* We discuss this connectivity in more detail in Section 6.

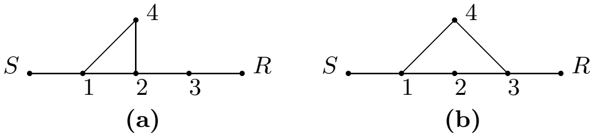

Fig. 1. Eavesdropping activities on a single path p

2. S sets $i := 1$ and multicasts $b_0 = a^S + a_1$. While $i \leq k$, v_i receives element b_{i-1} from its left side neighbor node v_{i-1}, v_i then multicasts $b_i = b_{i-1} - a_i + a_{i+1}$ and sets $i := i + 1$.
3. When $i = k + 1$, R receives element b_k from v_k, R then sets $a^R := b_k - a_{k+1}$. **End.**

Obviously, for each $0 \leq i \leq k$, the element that v_i multicasts is an encrypted ciphertext $b_i = a^S + a_{i+1}$. In order to decrypt a^S, the adversary needs to learn a pair (b_i, a_{i+1}) for some $0 \leq i \leq k$. Since b_i is multicast by v_i and a_{i+1} is multicast by v_{i+1}, the adversary who cannot eavesdrop on two adjacent nodes is not able to learn a^S by eavesdropping. □

Single Path Eavesdropping Examples.

(a) If the adversary can eavesdrop on two adjacent nodes on path p, then the necessary condition of [9] is not satisfied. For example, in Fig 1(a), the faulty node is node 4 and the hyperedges are

$$(S, \{1\}), (1, \{S, 2, 4\}), (2, \{1, 3, 4\}), (3, \{2, R\}), (4, \{1, 2\}) \text{ and } (R, \{3\}).$$

By removing the hyperedges that node 4 is on, the remaining hyperedges are

$$(S, \{1\}), (3, \{2, R\}) \text{ and } (R, \{3\}).$$

Thus p does not remain because edge $\{1, 2\}$ is removed, and hence the condition of [9] is not satisfied.

(b) If the adversary cannot eavesdrop on two adjacent nodes on path p, then the necessary condition of [9] is satisfied. For example, in Fig 1(b), the faulty node is node 4 and the hyperedges are

$$(S, \{1\}), (1, \{S, 2, 4\}), (2, \{1, 3\}), (3, \{2, 4, R\}), (4, \{1, 3\}) \text{ and } (R, \{3\}).$$

By removing the hyperedges that node 4 is on, the remaining hyperedges are

$$(S, \{1\}), (2, \{1, 3\}) \text{ and } (R, \{3\}).$$

Thus p remains because all edges on p remain, and hence the condition of [9] is satisfied.

The different separating activities were observed by Franklin and Wright in [7], but here we extend their result and upgrade their protocol.

Lemma 2. (following [7]) *The adversary can completely separate S and R on a path $p \in P$ if and only if it can control two or more nodes on p.*

Proof. We refer the proof of the "if" direction to [8].

Next we prove the "only if" direction. We assume that including S and R, there are $k+2$ nodes $v_0, \ldots, v_{k+1}$ on p. We let S be node v_0, R be node v_{k+1}, and $v_1, \ldots, v_k$ be the other k nodes from left to right. We show that with the following protocol, the adversary cannot completely separate S and R when k elements $(a_1, \ldots, a_k)$ are transmitted on p if the adversary can control no more than one node on p.

Single Path Distribution Protocol

1. For each $1 \leq i \leq k$, v_i initiates an element $a_i \in_R \mathbb{F}$ and multicasts it.
2. For each $1 \leq i \leq k$, the nodes on the left side of v_i execute an instance of the Single Path Private Propagation Protocol from v_{i-1} to S in which v_{i-1} sends a_i, and the nodes on the right side of v_i execute an instance of the Single Path Private Propagation Protocol from v_{i+1} to R in which v_{i+1} sends a_i.
3. At the end of the protocol, for each $1 \leq i \leq k$, S receives an element a_i^S and R receives an element a_i^R. If S (or R) receives nothing regarding element a_i for some $1 \leq i \leq k$, then S (or R) sets $a_i^S = 1$ (or $a_i^R = 1$). **End.**

Let v_e $(1 \leq e \leq k)$ be the only faulty node on p. It is straightforward that at the end of the protocol, $a_e^S = a_e^R$, even if v_e does not initiate and multicast any element (in this case $a_e^S = a_e^R = 1$). □

Next, we give the following two lemmas, which are trivial so we omit the proofs.

Lemma 3. *If the adversary can only control one node v on a path $p \in P$, then despite what protocol is executed on p, there exists a strategy of the adversary that causes the views of S and R to be different except for their views on the elements multicast by v.*

Lemma 4. *Given a node v on a path $p \in P$, if the adversary cannot separate S and R on p, completely eavesdrop on p, or control a neighbor of v, then during the execution of the Single Path Distribution Protocol on p, the adversary cannot learn the elements multicast by v.*

Having these lemmas, we now present an *extended characterization* $\zeta_{\mathcal{A}}$ of a multicast graph $G(V,E)$ given an adversary structure $\mathcal{A}$ on $V \setminus \{S,R\}$.

Definition 3. *Given a graph $G(V,E)$, let $\mathcal{A} = \{A_1, \ldots, A_z\}$ be an adversary structure on $V \setminus \{S,R\}$ and P be the set of all paths between S and R. An* Extended Characterization *of G given $\mathcal{A}$ is $\zeta_{\mathcal{A}} = \{\zeta_{A_1}, \ldots, \zeta_{A_z}\}$ where for each $1 \leq i \leq z$, we have $\zeta_{A_i} = (P_i^{(+)}, P_i^{(1)}, P_i^{(*)}, P_i)$ where*

- *$P_i^{(+)}$ is the set of all paths on each of which there are at least two nodes in A_i,*

- $P_i^{(1)}$ *is the set of all paths on each of which there is exactly one node in* A_i*,*
- $P_i^{(*)}$ *is the set of all paths on each of which there is no node in* A_i*, but on each path in* $P_i^{(*)}$*, there are two adjacent nodes that both have neighbors in* A_i*, and*
- $P_i = P_i^{(+)} \cup P_i^{(1)}$ *is the set of all paths on each of which there is at least one node in* A_i*.*

With the extended characterization $\zeta_{\mathcal{A}}$, we know that during the execution of any protocol, by choosing a set $A_i \in \mathcal{A}$ to control, the adversary can separate S and R on P_i, completely separate S and R on $P_i^{(+)}$ and completely eavesdrop on $P_i^{(*)}$.

Given any set $A_i \in \mathcal{A}$, we are particularly interested in the nodes of A_i on the paths of $P_i^{(1)}$. For each path $p \in P_i^{(1)}$, we use $A_i \sqcap p$ to denote the single node $v \in A_i$ that is on path p; i.e., $v = A_i \sqcap p$. Note that this notation is only used for the paths in $P_i^{(1)}$.

Definition 4. *Given a graph* $G(V, E)$ *and an adversary structure* $\mathcal{A}$ *on* $V \setminus \{S, R\}$*, we say that* S *and* R *are* highly $\mathcal{A}$-connected *if for any set* $A_i \in \mathcal{A}$*, we have* $P_i \cup P_i^{(*)} \neq P$*.*

Definition 5. *Given a graph* $G(V, E)$ *and an adversary structure* $\mathcal{A}$ *on* $V \setminus \{S, R\}$*, we say that* S *and* R *are* lowly $2\mathcal{A}$-separated *if there exist two (not necessarily distinct) sets* $A_1, A_2 \in \mathcal{A}$ *such that*

(a) $P_1 \cup P_2 = P$*, and*

(b) $P_1^{(1)} = \emptyset$*, or for each path* $p \in P_1^{(1)}$*, we have that* $p \in P_2 \cup P_2^{(*)}$ *or* $A_1 \sqcap p$ *has a neighbor in* A_2*, and*

(c) $P_2^{(1)} = \emptyset$*, or for each path* $p \in P_2^{(1)}$*, we have that* $p \in P_1 \cup P_1^{(*)}$ *or* $A_2 \sqcap p$ *has a neighbor in* A_1*.*

We say that S *and* R *are* lowly $2\mathcal{A}$-connected *if they are not lowly* $2\mathcal{A}$*-separated.*

Lemma 5. *Given a graph* $G(V, E)$ *and an adversary structure* $\mathcal{A}$ *on* $V \setminus \{S, R\}$*, if* S *and* R *are lowly* $2\mathcal{A}$*-connected, then for any set* $A_i \in \mathcal{A}$*, we have* $P_i \neq P$*.*

Proof. Assume there exits a set $A_i \in \mathcal{A}$ such that $P_i = P$, if we let both the sets A_1, A_2 of Definition 5 be A_i, then it is straightforward that S and R are lowly $2\mathcal{A}$-separated. Thus we have a contradiction. □

4 Reliable Communication

In this section, we discuss reliable communication with no requirement for privacy. We study almost perfect reliability (δ-RMT) in Section 4.1 and perfect reliability (0-RMT) in Section 4.2.

4.1 Almost Perfect Reliability

We give the necessary and sufficient condition for δ-RMT in multicast graphs.

Theorem 1. *Given a graph $G(V, E)$ and an adversary structure $\mathcal{A}$ on $V \setminus \{S, R\}$. The necessary and sufficient condition for δ-RMT from S to R is that S and R are lowly $2\mathcal{A}$-connected.*

Next, we use Lemma 7 and Lemma 8 to show the necessity and sufficiency of the condition respectively. Before we present these two lemmas, we first give the following Lemma 6, which is a key ingredient for proving the necessity.

Lemma 6. *If there exists two sets $A_1, A_2 \in \mathcal{A}$ such that $P_1^{(+)} \cup P_2^{(+)} = P$, and $\delta < \frac{1}{2}(1 - \frac{1}{|\mathbb{M}|})$, then δ-RMT from S to R is impossible.*

Proof. This lemma can be easily proven using a similar technique as that in [8, Theorem 5.1] and [5, Theorem 3]. See the full version of this paper [1]. □

Lemma 7. *The condition of Theorem 1 is necessary.*

Proof. It is straightforward that in order to achieve δ-reliability, it is necessary to have $P_i \neq P$ for any $A_i \in \mathcal{A}$; i.e., $P \setminus P_i \neq \emptyset$.

Next we prove the necessity of the condition by contradiction. We assume that S and R are lowly $2\mathcal{A}$-separated (i.e., there exist two sets $A_1, A_2 \in \mathcal{A}$ as they are in Definition 5) and there exists a δ-RMT protocol Π that transmits a message $m \in \mathbb{M}$ from S to R. Without loss of generality, we let $P_1 \cap P_2 = \emptyset$. Now if $P_1^{(1)} = \emptyset$ and $P_2^{(1)} = \emptyset$, then we have $P_1^{(+)} = P_1$ and $P_2^{(+)} = P_2$, and hence $P_1^{(+)} \cup P_2^{(+)} = P$ (following Definition 5(a)), thus due to Lemma 6, δ-RMT is impossible in the case. In the rest of our proof we let $P_1^{(1)} \neq \emptyset$ and/or $P_2^{(1)} \neq \emptyset$.

We make an observation on how protocol Π can achieve δ-reliability. Given a node v on a path $p \in P$, we use $(v \sim p)$ to denote the tuple of the elements that are multicast by v and received (in any way) by both S and R on p, and let $(v \sim p)^S$ and $(v \sim p)^R$ be the views of S and R respectively on $(v \sim p)$.

The strategy of the adversary is to choose an $e \in_R \{1, 2\}$ and control the set A_e. Let $d \in \{1, 2\}$ such that $d \neq e$, then R should be able to recover the actual message from the elements received on P_d. If, despite whether $e = 1$ or $e = 2$, $(v \sim p)^S \neq (v \sim p)^R$ for any v on any $p \in P_e$ (i.e., the views of S and R are completely different on P_e), then following Lemma 6, δ-RMT is impossible. Therefore, there must exist an $e \in \{1, 2\}$ such that $(v \sim p)^S = (v \sim p)^R$ is guaranteed for some v on some $p \in P_e$. We say that the tuple of elements $(v \sim p)$ where $p \in P_e$ such that $(v \sim p)^S = (v \sim p)^R$ *supports* the actual message. Following Lemma 2, the adversary can completely separate S and R on $P_e^{(+)}$ and cause $\forall(p \in P_e^{(+)}, v \text{ on } p) : (v \sim p)^S \neq (v \sim p)^R$. Following Lemma 3, for any path $p \in P_e^{(1)}$ (if $P_e^{(1)} \neq \emptyset$), $(v \sim p)^S = (v \sim p)^R$ can only be guaranteed if $v = A_e \sqcap p$. Therefore, there must exist an $e \in \{1, 2\}$ such that the actual message received on P_d is supported by some $((A_e \sqcap p) \sim p)$ where $p \in P_e^{(1)}$. Next, following Definition 5(b,c), for each path $p \in P_d^{(1)}$ (if $P_d^{(1)} \neq \emptyset$), we have case 1: $p \in P_e \cup P_e^{(*)}$, or case 2: $A_d \sqcap p$ has a neighbor in A_e. In case 1: $p \in P_e \cup P_e^{(*)}$, due to Lemma 1, there is no private transmission on path p whatsoever, so the adversary can learn $((A_d \sqcap p) \sim p)$. In case 2: $A_d \sqcap p$ has a neighbor in A_e, it is trivial that the adversary can learn $((A_d \sqcap p) \sim p)$.

To sum up, we can *conclude* that when the adversary chooses A_e to control, then the actual message, which can be recovered from the elements received on P_d, should be supported by some $((A_e \sqcap p) \sim p)$ where $p \in P_e^{(1)}$ (if $P_e^{(1)} \neq \emptyset$), and the adversary can learn $((A_d \sqcap p) \sim p)$ for each $p \in P_d^{(1)}$ (if $P_d^{(1)} \neq \emptyset$).

Now during the execution of the protocol Π, the adversary corrupts P_e and causes $(v \sim p)^S \neq (v \sim p)^R$ for all nodes v on all paths $p \in P_e$ except for $p \in P_e^{(1)}$ and $v = A_e \sqcap p$. This is possible due to Lemma 2 and Lemma 3. As we *concluded* above, the adversary can always learn $((A_d \sqcap p) \sim p)$ for each $p \in P_d^{(1)}$. Thus on P_e, the adversary simulates the protocol as S sent a message $m' \in \mathbb{M}$, and m' can be supported by $((A_d \sqcap p) \sim p)$, where $p \in P_d^{(1)}$.

Therefore, at the end of the protocol Π, despite whether $e = 1$ or $e = 2$, the view of R always consists of the following:

- on P_1, a message is recovered which can be supported by $((A_2 \sqcap p) \sim p)$ for any $p \in P_2^{(1)}$ (if $P_2^{(1)} \neq \emptyset$), but may not be supported by any other elements received on P_2;
- on P_2, a different message is recovered which can be supported by $((A_1 \sqcap p) \sim p)$ for any $p \in P_1^{(1)}$ (if $P_1^{(1)} \neq \emptyset$), but may not be supported by any other elements received on P_1.

Thus as we showed in Lemma 6, with probability $\delta \geq \frac{1}{2}(1 - \frac{1}{|\mathbb{M}|})$, R recovers the wrong message m'. We have a contradiction, which proves the necessity of the low $2\mathcal{A}$-connectivity. □

Let $P = \{p_1, \ldots, p_n\}$, we first generalize some of Franklin and Wright's protocols in multicast graphs.

Full Distribution Protocol

1. For each $1 \leq j \leq n$, the nodes on path p_j execute an instance of the Single Path Distribution Protocol for each node v_i on p_j to distribute an element $a_{i,j}$. The nodes not on p_j do not multicast anything.
2. At the end of the protocol, on each path p_j $(1 \leq j \leq n)$, S and R receive $a_{i,j}^S$ and $a_{i,j}^R$ respectively as the element initiated by node v_i on p_j. **End.**

Private Propagation Protocol

1. For each $1 \leq j \leq n$, the nodes on path p_j execute an instance of the Single Path Private Propagation Protocol from S to R in which S sends an element a_j^S, and the nodes not on p_j do not multicast anything.
2. At the end of the protocol, on each path p_j $(1 \leq j \leq n)$, R receives a_j^R as the element that S initiated and propagated on p_j. **End.**

Now we present the following protocol, which achieves δ-RMT for a message $m \in \mathbb{M}$ in a graph $G(V, E)$.

Reliable Transmission Protocol

1. The nodes of V execute an instance of the Full Distribution Protocol in which for each $1 \leq j \leq n$, the elements that node v_i on path p_j initiates are $(a_{i,j}, b_{i,j}) \in_R \mathbb{F}^2$. Let $(a^S_{i,j}, b^S_{i,j})$ and $(a^R_{i,j}, b^R_{i,j})$ be what S and R receive respectively regarding $(a_{i,j}, b_{i,j})$.
2. The nodes of V execute an instance of the Private Propagation Protocol from S to R in which S sends the same vector on all paths in P:

   $$(m, \langle \mathrm{auth}(m; a^S_{i,j}, b^S_{i,j}) \rangle),$$

   where $\langle \mathrm{auth}(m; a^S_{i,j}, b^S_{i,j}) \rangle$ is an ordered set of the authenticated m with *all* keys $(a^S_{i,j}, b^S_{i,j})$ that S receives in Step 1. At the end of the instance, R receives a vector $(m_k, \langle u_{i,j,k} \rangle)$ on each path $p_k \in P$.
3. Given the vector $(m_k, \langle u_{i,j,k} \rangle)$ that R receives on p_k, if $\exists (i,j) : u_{i,j,k} = \mathrm{auth}(m_k; a^R_{i,j}, b^R_{i,j})$, then we say that m_k is *qualified* on $(v_i \sim p_j)$. R finds an $A_f \in \mathcal{A}$ that satisfies the following three α-conditions:

   α-1 all vectors received on $P \setminus P_f$ are the same, say vector $(m_l, \langle u_{i,j,l} \rangle)$;

   α-2 $P^{(1)}_f = \emptyset$, or for each $p_j \in P^{(1)}_f$, m_l is qualified on $((A_f \sqcap p_j) \sim p_j)$;

   α-3 $P_f \cup P^{(*)}_f = P$, or for any vector $(m_k, \langle u_{i,j,k} \rangle)$ received on path $p_k \in P_f$ such that $m_k \neq m_l$, we have that m_k is *not* qualified on any $(v_i \sim p_j)$ where $p_j \in P \setminus (P_f \cup P^{(*)}_f)$ and v_i does not have a neighbor in A_f.

   R then outputs the message m_l. **End.**

Lemma 8. *The Reliable Transmission Protocol is a δ-RMT protocol under the condition of Theorem 1.*

Proof. It is straightforward that if the adversary cannot learn some $(a_{i,j}, b_{i,j})$ (initiated by v_i and multicast on p_j) but a *corrupted* m_k is qualified on $(v_i \sim p_j)$, then the Reliable Transmission Protocol fails. We use $\overline{RT}$ to denote the event when the above failure occurs and RT to denote the event otherwise. Let n be the total number of paths between S and R and y be the maximum number of nodes on any path, following the proof of [8, Theorem 3.4], the probability that the protocol fails is $\Pr[\overline{RT}] < \frac{yn^2}{|\mathbb{F}|}$. This probability is negligible in the security parameter (given $\mathbb{F}$ is sufficiently large). Next in our proof, we assume that the above failure does not happen. That is, *we analyze the protocol in the event RT.*

In the following, we first show that R can always find an $A_f \in \mathcal{A}$ that satisfies the three α-conditions, then we prove, by contradiction, that in the event RT, the message output by R is correct.

Now we show that there always exists an A_f that satisfies all three α-conditions, at least when the adversary chooses A_f to control so that P_f is *corrupted.* Since $P_f \neq P$ (following Lemma 5), we immediately have that condition α-1 is satisfied and m_l received on $P \setminus P_f$ is the actual message. If $P^{(1)}_f \neq \emptyset$, then as shown in the proof of Lemma 2, on each $p_j \in P^{(1)}_f$, S and R always have the same view on the key initiated by $A_f \sqcap p_j$. Thus it is clear that m_l is qualified on

$((A_f \sqcap p_j) \sim p_j)$, and hence condition α-2 is satisfied. If $P_f \cup P_f^{(*)} \neq P$, then the adversary cannot learn the key initiated by any node v_i which is on a path $p_j \in P \setminus (P_f \cup P_f^{(*)})$ if v_i does not have a neighbor in A_f. Thus without the above mentioned failure $\overline{RT}$, any faulty message $m_k \neq m_l$ cannot be qualified on such $(v_i \sim p_j)$, and hence condition α-3 is satisfied.

Next, using contradiction, we show that in the event RT, the message m_l that S outputs is the actual message. For contradiction, we assume that m_l is modified by the adversary who chooses a set $A_e \in \mathcal{A}$ to control, and all three α-conditions are satisfied. We now show that the three α-conditions imply the three properties of A_1, A_2 in Definition 5.

- From condition α-1, since all vectors received on $P \setminus P_f$ are modified, we have $P_e \cup P_f = P$ (i.e., corresponding to Definition 5(a)).
- Condition α-2 indicates that either $P_f^{(1)} = \emptyset$, or the adversary can learn the key initiated by node $A_f \sqcap p_j$ on any path $p_j \in P_f^{(1)}$ to make the faulty message m_l qualified on $((A_f \sqcap p_j) \sim p_j)$. Due to Lemma 4, this means that the adversary can separate S and R on p_j, completely eavesdrop on p_j or control a neighbor of $A_f \sqcap p_j$. Thus from condition α-2 we can conclude that $P_f^{(1)} = \emptyset$, or for each path $p_j \in P_f^{(1)}$, we have that $p_j \in P_e \cup P_e^{(*)}$ or $A_f \sqcap p_j$ has a neighbor in A_e (i.e., corresponding to Definition 5(c)).
- Finally, since $P_e \neq P$ and $P_e \cup P_f = P$, there exists at least one path $p_k \in P_f$ such that the message m_k received on p_k is the actual message. Due to condition α-3, there are two cases:

  case 1 $P_f \cup P_f^{(*)} = P$, thus we have $P_e^{(1)} \subseteq P_f \cup P_f^{(*)} = P$;

  case 2 The actual message m_k is *not* qualified on any $(v_i \sim p_j)$ where $p_j \in P \setminus (P_f \cup P_f^{(*)})$ and v_i does not have a neighbor in A_f. This implies that either $p_j \in P_e^{(+)}$, or $p_j \in P_e^{(1)}$ but any v_i on p_j that does not have a neighbor in A_f is not $A_e \sqcap p_j$ (because otherwise the actual message m_k should be qualified on $(v_i \sim p_j)$, due to the proof of Lemma 2). That is, if such $p_j \in P_e^{(1)}$ exists, then all the nodes on p_j that do not have a neighbor in A_f are not $A_e \sqcap p_j$. This implies that $A_e \sqcap p_j$ has a neighbor in A_f.

  It is easy to conclude that in either case, $P_e^{(1)} = \emptyset$, or for each path $p_j \in P_e^{(1)}$, we have $p_j \in P_f \cup P_f^{(*)}$ or $A_e \sqcap p_j$ has a neighbor in A_f (i.e., corresponding to Definition 5(b)).

To sum up, A_e, A_f are as A_1, A_2 in Definition 5. This means S and R are lowly $2\mathcal{A}$-separated, which contradicts the condition of Theorem 1.

Therefore, at the end of the Reliable Transmission Protocol, R can recover $m_l = m$ with an arbitrarily small probability of failure (i.e., $\Pr[\overline{RT}] < \frac{yn^2}{|\mathbb{F}|}$). Thus the Reliable Transmission Protocol is a δ-RMT protocol. □

4.2 Perfect Reliability

Here we study 0-RMT in multicast graphs. Similar to the result in [7], we show that the necessary and sufficient condition for 0-RMT in the multicast setting is the same as that in the point-to-point setting. The following theorem can be easily proven following some previous results in [8,5].

Theorem 2. *Given a graph $G(V, E)$ and an adversary structure $\mathcal{A}$ on $V \setminus \{S, R\}$. The necessary and sufficient condition for 0-RMT from S to R is that $P_i \cup P_j \neq P$ for any two sets $A_i, A_j \in \mathcal{A}$.*

Proof. See the full version of this paper [1]. □

5 Secure Communication

In this section we take the problem of achieving privacy into consideration. We study almost perfect security in Section 5.1; i.e., we discuss both (ϵ, δ)-SMT and $(0, \delta)$-SMT. In Section 5.2, we study $(0, 0)$-SMT that enables perfect security.

5.1 Almost Perfect Security

First we give the necessary and sufficient condition for (ϵ, δ)-SMT in multicast graphs. Unlike the setting in [7] in which the conditions for both δ-RMT and (ϵ, δ)-SMT are the same (i.e., $n > t$), in multicast graphs, (ϵ, δ)-SMT requires stronger connectivity than that for δ-RMT.

Theorem 3. *Given a graph $G(V, E)$ and an adversary structure $\mathcal{A}$ on $V \setminus \{S, R\}$. The necessary and sufficient condition for (ϵ, δ)-SMT from S to R is that S and R are highly $\mathcal{A}$-connected and lowly $2\mathcal{A}$-connected.*

Proof. We first prove the necessity of the condition. It is straightforward that the high $\mathcal{A}$-connectivity, i.e., $P_i \cup P_i^{(*)} \neq P$, is necessary for achieving ϵ-privacy, because otherwise there is no private transmission between S and R on any path in P. Moreover, as proven in Lemma 7, the low $2\mathcal{A}$-connectivity is necessary for achieving δ-reliability. Thus the condition is necessary for (ϵ, δ)-SMT.

Next we show that the condition is sufficient. Let $P = \{p_1, \ldots, p_n\}$, we give the following protocol (similar to [8,15]) for S to send a message $m \in \mathbb{M}$ to R.

Private Transmission Protocol

1. The nodes of V execute an instance of the Private Propagation Protocol from S to R in which for each $1 \leq j \leq n$, S sends a pair $(a_j^S, b_j^S) \in_R \mathbb{F}$ on path $p_j \in P$. At the end of the instance, R receives a pair (a_j^R, b_j^R) on each path $p_j \in P$.
2. R chooses an element $r^R \in_R \mathbb{F}$ and for each $1 \leq j \leq n$, computes $s_j^R = \text{auth}(r^R; a_j^R, b_j^R)$. The nodes of V executes an instance of the Reliable Transmission Protocol from R to S in which R sends a vector $(r^R, s_1^R, \ldots, s_n^R)$. At the end of the instance, S outputs a vector $(r^S, s_1^S, \ldots, s_n^S)$.

3. S computes an index set $I = \{j | s_j^S = \text{auth}(r^S; a_j^S, b_j^S)\}$ and an encryption key $key = \sum_{j \in I} a_j^S$, and encrypts the message $c = m + key$. The nodes of V executes an instance of the Reliable Transmission Protocol from S to R in which S sends a vector (I, c). At the end of the instance, R outputs a vector (I', c').
4. R computes a decryption key $key' = \sum_{j \in I'} a_j^R$ and decrypts the message $m' = c' - key'$. **End.**

First we show that this protocol achieves ϵ-privacy. Suppose that the adversary chooses a set A_e to control. Since $P_e \cup P_e^{(*)} \neq P$, there exists a path $p_d \in P \setminus (P_e \cup P_e^{(*)})$. As shown in the proof of Lemma 1, the adversary cannot learn (a_d^S, b_d^S) in Step 1. Because $p_d \notin P_e$, we have $(a_d^R, b_d^R) = (a_d^S, b_d^S)$. Let RT denote the event that the instance of the Reliable Transmission Protocol in Step 2 succeeds and $\overline{RT}$ denote the event otherwise. In the event RT, $r^S = r^R$ and for each $1 \leq j \leq n$, we have $s_j^S = s_j^R$. This implies that $d \in I$. The adversary who cannot learn a_d^S by eavesdropping or by decoding s_d^R will not be able to compute key to decrypt m. That is, for any two messages $m_1, m_2 \in \mathbb{M}$ and any coin flips r, using the adversary's view adv, we have the following:

$$\textstyle\sum_c |\Pr[adv(m_1, r) = c | RT] - \Pr[adv(m_2, r) = c | RT]| = 0 \tag{1}$$

$$\textstyle\sum_c |\Pr[adv(m_1, r) = c | \overline{RT}] - \Pr[adv(m_2, r) = c | \overline{RT}]| \leq | +1| + | -1| = 2 \tag{2}$$

Let $\Pr[\overline{RT}] = \epsilon$, which is arbitrarily small as we discussed in the proof of Lemma 8, by combining Eq. 1 and Eq. 2, we have the following:

$$\textstyle\sum_c |\Pr[adv(m_1, r) = c] - \Pr[adv(m_2, r) = c]| \leq 0 \cdot \Pr[RT] + 2 \cdot \Pr[\overline{RT}] = 2\epsilon.$$

Thus the Private Transmission Protocol achieves ϵ-privacy.

Next we show that the protocol achieves δ-reliability. Let δ_1 be the probability that the instance of the Reliable Transmission Protocol in Step 2 fails and δ_2 be the probability that the instance in Step 3 fails. As we showed in the proof of Lemma 8, δ_1 and δ_2 are negligible in the security parameter. Let δ_3 be the probability that both the above mentioned instances succeed, but R outputs $m' \neq m$. This can only happen if there exists at least one $j \in I$ such that $a_j^S \neq a_j^R$. Since both reliable protocols succeed, the fact $j \in I$ implies $\text{auth}(r^R; a_j^S, b_j^S) = \text{auth}(r^R; a_j^R, b_j^R)$. That is,

$$a_j^S r^R + b_j^S = a_j^R r^R + b_j^R \Rightarrow r^R = \frac{b_j^R - b_j^S}{a_j^S - a_j^R} \in \mathbb{F}, \tag{3}$$

where $a_j^S \neq a_j^R$. Since r^R is chosen with respect to the uniform distribution, if the adversary modifies (a_j^S, b_j^S) to (a_j^R, b_j^R) on path p_j in Step 1, then the probability that Eq. 3 is fulfilled is $\frac{1}{|\mathbb{F}|}$. Since the adversary can corrupt $|P_e|$ paths, it is straightforward that $\delta_3 = \frac{|P_e|}{|\mathbb{F}|} < \frac{n}{|\mathbb{F}|}$, which is much smaller than δ_1 and δ_2. Thus the final probability that the protocol fails to be reliable is

$$\delta = \delta_1 + (1 - \delta_1)\delta_2 + (1 - (\delta_1 + (1 - \delta_1)\delta_2))\delta_3 < \delta_1 + \delta_2 + \delta_3.$$

To sum up, the Private Transmission Protocol is an (ϵ, δ)-SMT protocol. □

Note that the condition of Theorem 3 can be seen as it consists of two parts, with the high $\mathcal{A}$-connectivity enables private communication and the low $2\mathcal{A}$-connectivity enables δ-reliable communication. These two types of connectivity are *independent*. Indeed, with some examples in Section 6, we can show that they do not imply each other.

In [16], Yang and Desmedt proved that reducing the requirement for privacy does not weaken the minimal connectivity. In the following theorem, we show that the condition for (ϵ, δ)-SMT is also necessary and sufficient for $(0, \delta)$-SMT.

Theorem 4. *Given a graph $G(V, E)$ and an adversary structure $\mathcal{A}$ on $V \setminus \{S, R\}$. The necessary and sufficient condition for $(0, \delta)$-SMT from S to R is that S and R are highly $\mathcal{A}$-connected and lowly $2\mathcal{A}$-connected.*

Proof. It is straightforward that the condition is necessary. Next we show that the condition is sufficient by slightly amending the Private Transmission Protocol to the following protocol which achieves perfect privacy.

Perfectly Private Transmission Protocol

1. Same as Step 1 in the Private Transmission Protocol.
2. R chooses an element $r^R \in_R \mathbb{F}$ and for each $1 \leq j \leq n$, computes $s_j^R = \text{auth}(r^R; a_j^R, b_j^R)$. The nodes of V executes an instance of the Reliable Transmission Protocol from R to S in which R sends a vector $(r^R, s_1^R, \ldots, s_n^R)$. At the end of the instance, S distinguishes the following two cases:
   Case 1 If there exist two sets $A_{f_1}, A_{f_2} \in \mathcal{A}$ that satisfy all three α-conditions of the Reliable Transmission Protocol, and the two vectors (both regarding the vector $(r^R, s_1^R, \ldots, s_n^R)$) that S receives respectively on $P \setminus P_{f_1}$ and $P \setminus P_{f_2}$ are different, then S terminates the protocol.
   Case 2 Otherwise, S outputs a vector $(r^S, s_1^S, \ldots, s_n^S)$ and goes to Step 3.
3. Same as Step 3 in the Private Transmission Protocol.
4. Same as Step 4 in the Private Transmission Protocol. **End.**

Now we show that this protocol achieves 0-privacy. Following the proof of Theorem 3, the privacy of the message transmission can only be breached in the event $\overline{RT}$. It is clear that the instance of the Reliable Transmission Protocol in Step 2 allows S to distinguish the events RT and $\overline{RT}$. As we showed in the proof of Lemma 8, in the event RT, only the correct vector can be output after the Reliable Transmission Protocol. This means if two different vectors can be output, then the event $\overline{RT}$ occurs. Thus in Step 2, Case 1 indicates $\overline{RT}$ and Case 2 indicates RT. In the event $\overline{RT}$, S terminates the protocol so the adversary learns nothing about the message. Thus the protocol achieves 0-privacy.[6] Next, using a similar proof as that for Theorem 3, we can prove that the Perfectly Private Transmission Protocol is also δ-reliable, which concludes the proof. □

[6] A more formal proof is available in the full version of this paper [1].

5.2 Perfect Security

In [6], Dolev et al. showed that if σ is the maximum number of channels that a *listening* (passive) adversary can control and ρ is the maximum number of channels that a *disrupting* (active) adversary can control, then there must be at least $\max\{\sigma + \rho + 1, 2\rho + 1\}$ channels between S and R for PSMT (i.e., $(0,0)$-SMT). This setting can be generalized in our model as follows: given an adversary structure $\mathcal{A} = \{A_1, \ldots, A_z\}$, then $\{P_1 \cup P_1^{(*)}, \ldots, P_z \cup P_z^{(*)}\}$ consists of the subsets of paths a listening adversary can control and $\{P_1, \ldots, P_z\}$ consists of the subsets of paths a disrupting adversary can control. Thus we give the following theorem for $(0,0)$-SMT in multicast graphs.

Theorem 5. *Given a graph $G(V,E)$ and an adversary structure $\mathcal{A}$ on $V \setminus \{S,R\}$. The necessary and sufficient condition for (0,0)-SMT from S to R is that*
$(P_i \cup P_i^{(*)}) \cup P_j \neq P$ for any $A_i, A_j \in \mathcal{A}$.

Proof. See the full version of this paper [1]. □

6 Corresponding Threshold Model

In this section we use our results in the general adversary model to find the necessary and sufficient conditions for RMT and SMT in the threshold model. Because a threshold is a special case of an adversary structure, we re-define the threshold model in the adversary structure context.

Definition 6. *Given a graph $G(V,E)$, a threshold t is an adversary structure $\mathcal{A}^T \subseteq 2^{V \setminus \{S,R\}}$ such that $\forall (A \subseteq V \setminus \{S,R\}, |A| \leq t) : A \in \mathcal{A}^T$. Furthermore,*

- *we say that S and R are $t_{\zeta\text{-}private}$-*connected *if they are highly $\mathcal{A}^T$-connected;*
- *we say that S and R are $t_{\zeta\text{-}reliable}$-*connected *if they are lowly $2\mathcal{A}^T$-connected.*

It is easy to show that our results correspond to Franklin and Wright's [7] if the multicast graph only consists of n *node-disjoint* and *neighbor-disjoint* paths. For more details see the full version of this paper [1].

Next we discuss the connectivity in the general multicast graph setting with some previous results. In [4], Desmedt and Wang looked at four different types of connectivity. With slight changes, we show them in our model as follows.

- *t-connectivity.* For any $A \in \mathcal{A}^T$, after removing all nodes in A from G, there remains a path between S and R.
- *weak t_{hyper}-connectivity.* For any $A \in \mathcal{A}^T$, after removing from the hypergraph $H_G(V, E_H)$ all nodes in A and all hyperedges on each of which there is at least one node in A, there remains a path between S and R (see [9]).
- *$t_{neighbor}$-connectivity.* For any $A \in \mathcal{A}^T$, after removing all nodes in A and all their neighbors from G, there remains a path between S and R.
- *weak (n,t)-connectivity.* There are n node-disjoint paths $p_1, \ldots, p_n$ between S and R, and for any $A \in \mathcal{A}^T$, after removing all nodes in A and all their neighbors from G, there remains a path p_i $(1 \leq i \leq n)$ between S and R.

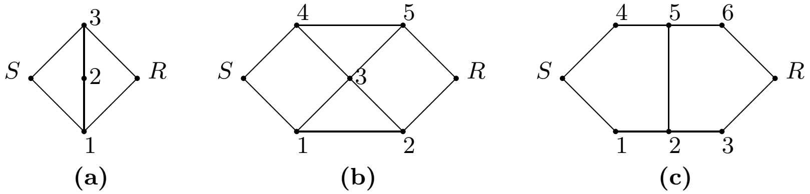

Fig. 2. Private and reliable connectivity

As we showed in the proof of Lemma 1, Franklin and Yung's weak t_{hyper}-connectivity [9] in a hypergraph H_G is essentially our $t_{\zeta\text{-}private}$-connectivity in a multicast graph G. Thus we use the $t_{\zeta\text{-}private}$-connectivity to replace the weak t_{hyper}-connectivity in the rest of the paper for a simpler presentation. Desmedt and Wang [4] showed that the following implications are strict:

$$\text{weak } (n,t)\text{-connectivity} \Rightarrow t_{neighbor}\text{-connectivity} \Rightarrow t_{\zeta\text{-}private}\text{-connectivity} \\ \Rightarrow t\text{-connectivity}.$$

In [14], Wang and Desmedt claimed that the weak (n,t)-connectivity is sufficient for $(0,\delta)$-SMT. Since weak (n,t)-connectivity $\Rightarrow t_{\zeta\text{-}private}$-connectivity, it is clear that 0-privacy can be achieved. However, δ-reliability is only achievable under their condition if weak (n,t)-connectivity $\Rightarrow t_{\zeta\text{-}reliable}$-connectivity. In [14], there is not a proper proof showing this implication. Thus their claim is only a conjecture. We leave this as an open problem.

Later study by Desmedt and Wang [4] showed that the conjectured upper bound, i.e., the weak (n,t)-connectivity, is not necessary for $(0,\delta)$-SMT, by showing an example, as Fig. 2(a), in which S and R are not weakly $(2,1)$-connected but $(0,\delta)$-SMT is possible. We observe that their protocol (appeared in [15]) is actually an (ϵ,δ)-SMT protocol but the claim is correct, because S and R are obviously $1_{\zeta\text{-}private}$-connected and $1_{\zeta\text{-}reliable}$-connected in Fig. 2(a). They also showed that the weak t_{hyper}-connectivity (i.e., the $t_{\zeta\text{-}private}$-connectivity) is the lower bound for $(0,\delta)$-SMT but not necessary for δ-RMT, as in Fig. 2(b) where S and R are not $1_{\zeta\text{-}private}$-connected but δ-RMT is possible. This claim is obvious under our condition because S and R are clearly $1_{\zeta\text{-}reliable}$-connected. Finally they conjectured that the weak t_{hyper}-connectivity (i.e., the $t_{\zeta\text{-}private}$-connectivity) is not sufficient for $(0,\delta)$-SMT, by asking whether $(0,\delta)$-SMT is possible in Fig. 2(c) such that S and R are $1_{\zeta\text{-}private}$-connected. Our condition proves their conjecture. Indeed, not only is $(0,\delta)$-SMT impossible in Fig. 2(c), but δ-RMT is also impossible, because S and R are not $1_{\zeta\text{-}reliable}$-connected. Therefore, our result explains all the examples and proves all the conjectures in the previous work.

Note that the examples of Fig. 2(b) and Fig. 2(c) also show that the $t_{\zeta\text{-}private}$-connectivity (or, the high $\mathcal{A}$-connectivity) and the $t_{\zeta\text{-}reliable}$-connectivity (or, the low $2\mathcal{A}$-connectivity) *do not imply each other*, because in Fig. 2(b), S and R are $1_{\zeta\text{-}reliable}$-connected but not $1_{\zeta\text{-}private}$-connected, and in Fig. 2(c), they are $1_{\zeta\text{-}private}$-connected but not $1_{\zeta\text{-}reliable}$-connected.

At the end, we present the following corollary as the final result of this paper.

Corollary 1. *Given a graph $G(V,E)$ and an adversary who can control up to t nodes in $V \setminus \{S,R\}$.*

- *δ-RMT is possible if and only if S and R are $t_{\zeta\text{-}reliable}$-connected in G.*
- *0-RMT is possible if and only if S and R are $2t$-connected in G.*
- *(ϵ, δ)-SMT or $(0, \delta)$-SMT is possible if and only if S and R are $t_{\zeta\text{-}private}$-connected and $t_{\zeta\text{-}reliable}$-connected in G.*
- *(0,0)-SMT is possible if and only if S and R are $(t_{\zeta\text{-}private} + t)$-connected in G. The $(t_{\zeta\text{-}private} + t)$-connectivity means that for any $A_i, A_j \in \mathcal{A}^T$, we have $(P_i \cup P_i^{(*)}) \cup P_j \neq P$.*

References

1. The full version of this paper will be available on the authors' web pages
2. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computing. In: Proc. ACM STOC 1988, pp. 1–10 (1988)
3. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols. In: Proc. ACM STOC 1988, pp. 11–19 (1988)
4. Desmedt, Y., Wang, Y.: Perfectly Secure Message Transmission Revisited. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 502–517. Springer, Heidelberg (2002)
5. Desmedt, Y., Wang, Y., Burmester, M.: A Complete Characterization of Tolerable Adversary Structures for Secure Point-to-Point Transmissions without Feedback. In: Deng, X., Du, D.-Z. (eds.) ISAAC 2005. LNCS, vol. 3827, pp. 277–287. Springer, Heidelberg (2005)
6. Dolev, D., Dwork, C., Waarts, O., Yung, M.: Perfectly secure message transmission. J. ACM 40(1), 17–47 (1993)
7. Franklin, M.K., Wright, R.: Secure Communication in Minimal Connectivity Models. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 346–360. Springer, Heidelberg (1998)
8. Franklin, M.K., Wright, R.: Secure communication in minimal connectivity models. J. Cryptology 13(1), 9–30 (2000)
9. Franklin, M.K., Yung, M.: Secure hypergraphs: Privacy from partial broadcast. In: Proc. ACM STOC 1995, pp. 36–44 (1995)
10. Goldreich, O., Goldwasser, S., Linial, N.: Fault-tolerant computation in the full information model. SIAM J. Comput. 27(2), 506–544 (1998)
11. Hirt, M., Maurer, U.M.: Player simulation and general adversary structures in perfect multiparty computation. J. Cryptology 13(1), 31–60 (2000)
12. Ito, M., Saito, A., Nishizeki, T.: Secret sharing schemes realizing general access structure. In: Proc. IEEE Globecom 1987, pp. 99–102 (1987)
13. Kumar, M., Goundan, P., Srinathan, K., Rangan, C.P.: On perfectly secure communication over arbitrary networks. In: Proc. ACM PODC 2002, pp. 293–202 (2002)
14. Wang, Y., Desmedt, Y.G.: Secure Communication in Broadcast Channels: The Answer to Franklin and Wright's Question. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 446–458. Springer, Heidelberg (1999)
15. Wang, Y., Desmedt, Y.: Perfectly secure message transmission revisited. IEEE Transaction on Information Theory 54(6), 2582–2595 (2008)
16. Yang, Q., Desmedt, Y.: Cryptanalysis of Secure Message Transmission Protocols with Feedback. In: Kurosawa, K. (ed.) Information Theoretic Security. LNCS, vol. 5973, pp. 159–176. Springer, Heidelberg (2010)
17. Yang, Q., Desmedt, Y.: General Perfectly Secure Message Transmission using Linear Codes. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 448–465. Springer, Heidelberg (2010)