Perfectly Secure Message Transmission Revisited

Yvo Desmedt[1,2] and Yongge Wang[3]

[1] Computer Science, Florida State University, Tallahassee
Florida FL 32306-4530, USA, desmedt@cs.fsu.edu
[2] Dept. of Mathematics, Royal Holloway, University of London, UK
[3] Department of Software and Information Systems, University of North Carolina at
Charlotte, 9201 University City Blvd, Charlotte, NC 28223, yonwang@uncc.edu

Abstract. Achieving secure communications in networks has been one of the most important problems in information technology. Dolev, Dwork, Waarts, and Yung have studied secure message transmission in one-way or two-way channels. They only consider the case when all channels are two-way or all channels are one-way. Goldreich, Goldwasser, and Linial, Franklin and Yung, Franklin and Wright, and Wang and Desmedt have studied secure communication and secure computation in multi-recipient (multicast) models. In a "multicast channel" (such as Ethernet), one processor can send the same message—simultaneously and privately—to a fixed subset of processors. In this paper, we shall study necessary and sufficient conditions for achieving secure communications against active adversaries in mixed one-way and two-way channels. We also discuss multicast channels and neighbor network channels.

Keywords: network security, privacy, reliability, network connectivity

1 Introduction

If there is a private and authenticated channel between two parties, then secure communication between them is guaranteed. However, in most cases, many parties are only indirectly connected, as elements of an incomplete network of private and authenticated channels. In other words they need to use intermediate or internal nodes. Achieving participants cooperation in the presence of faults is a major problem in distributed networks. Original work on secure distributed computation assumed a complete graph for secure and reliable communication. Dolev, Dwork, Waarts, and Yung [5] were able to reduce the size of the network graph by providing protocols that achieve private and reliable communication without the need for the parties to start with secret keys. The interplay of network connectivity and secure communication has been studied extensively (see, e.g., [1, 2, 4, 5, 11]). For example, Dolev [4] and Dolev et al. [5] showed that, in the case of k Byzantine faults, reliable communication is achievable only if the system's network is $2k+1$ connected. They also showed that if all the paths are one way, then $3k+1$ connectivity is necessary and sufficient for reliable and private communications. However they did not prove any results for the general case when there are certain number of directed paths in one direction and another number of directed paths in the other direction. While undirected graphs correspond naturally to the case of pairwise two-way channels, directed graphs do not correspond to the case of all-one-way or all-two-way channels considered in [5], but to the mixed case where there are some paths in one direction and some paths in the other direction. In this paper, we will initiate the study in this direction by showing what can be done with a general directed graph. Note that this scenario is important in practice, in particular, when the network is not symmetric. For example, a channel from A to B is cheap and a channel from B to A is expensive but not impossible. Another example is that A has access to more resources than B does.

Goldreich, Goldwasser, and Linial [10], Franklin and Yung [8], Franklin and Wright [7], and Wang and Desmedt [17] have studied secure communication and secure computation in *multi-recipient (multicast)* models. In a "multicast channel" (such as Ethernet), one participant can send the same message—simultaneously and privately—to a fixed subset of participants. Franklin and Yung [8] have given a necessary and sufficient condition for individuals to exchange private messages in multicast models in the presence of passive adversaries (passive gossipers). For the case of active Byzantine adversaries, many results have been presented by Franklin and Wright [7], and, Wang and Desmedt [17]. Note that Goldreich, Goldwasser, and Linial [10] have also studied fault-tolerant computation in the public multicast model (which can be thought of as the largest possible multirecipient channels) in the presence of active Byzantine adversaries. Specifically, Goldreich, et al. [10] have made an investigation of general fault-tolerant distributed computation in the full-information model. In the full information model no restrictions are made on the computational power of the faulty parties or the information available to them. (Namely, the faulty players may be infinitely powerful and there are no private channels connecting pairs of honest players). In particular, they present efficient two-party protocols for fault-tolerant computation of any bivariate function.

There are many examples of multicast channels (see, e.g. [7]), such as an Ethernet bus or a token ring. Another example is a shared cryptographic key. By publishing an encrypted message, a participant initiates a multicast to the subset of participants that is able to decrypt it.

We present our model in Section 2. In Sections 3 and 4, we study secure message transmission over directed graphs. Section 6 is devoted to reliable message transmission over hypergraphs, and Section 7 is devoted to secure message transmission over neighbor networks.

2 Model

We will abstract away the concrete network structures and consider directed graphs. A directed graph is a graph $G(V, E)$ where all edges have directions. For a directed graph $G(V, E)$ and two nodes $A, B \in V$, throughout this paper, n denotes the number of vertex disjoint paths between the two nodes and k denotes the number of faults under the control of the adversary. We write $|S|$ to denote the number of elements in the set S. We write $x \in_R S$ to indicate that x is chosen with respect to the uniform distribution on S. Let $\mathbf{F}$ be a finite field, and let $a, b, c, M \in \mathbf{F}$. We define $\mathrm{auth}(M; a, b) := aM + b$ (following [7, 9, 14, 15]) and $\mathrm{auth}(M; a, b, c) := aM^2 + bM + c$ (following [17]). Note that each authentication key $key = (a, b)$ can be used to authenticate one message M without revealing any information about any component of the authentication key and that each authentication key $key = (a, b, c)$ can be used to authenticate two messages M_1 and M_2 without revealing any information about any component of the authentication key. We will also use a function $\langle \ldots \rangle$ which maps a variable size (we assume that this variable size is bounded by a pre-given bound) ordered subset of $\mathbf{F}$ to an image element in a field extension $\mathbf{F}^*$ of $\mathbf{F}$, and from any image element we can uniquely and efficiently recover the ordered subset. Let k and n be two integers such that $0 \leq k < n \leq 3k + 1$. A $(k + 1)$-out-of-n secret sharing scheme is a probabilistic function S: $\mathbf{F} \rightarrow \mathbf{F}^n$ with the property that for any $M \in \mathbf{F}$ and $\mathrm{S}(M) = (v_1, \ldots, v_n)$, no information of M can be inferred from any k entries of $(v_1, \ldots, v_n)$, and M can be recovered from any $k + 1$ entries of $(v_1, \ldots, v_n)$. The set of all possible $(v_1, \ldots, v_n)$ is called a code and its elements codewords. We say that a $(k + 1)$-out-of-n secret sharing scheme can detect k' errors if given any codeword $(v_1, \ldots, v_n)$ and any tuple $(u_1, \ldots, u_n)$ over $\mathbf{F}$ such that $0 < |\{i : u_i \neq v_i, 1 \leq i \leq n\}| \leq k'$ one can detect that $(u_1, \ldots, u_n)$ is not a codeword. If the code is Maximal Distance Separable, then the maximum value of errors that can be detected is $n - k - 1$ [12]. We say that the $(k + 1)$-out-of-n secret sharing scheme can correct k' errors if from any $\mathrm{S}(M) = (v_1, \ldots, v_n)$ and any tuple $(u_1, \ldots, u_n)$ over $\mathbf{F}$ with $|\{i : u_i \neq v_i, 1 \leq i \leq n\}| \leq k'$ one can recover the secret m. If the code is Maximal Distance Separable, then the maximum value of errors that allows the recovery of the vector $(v_1, \ldots, v_n)$ is $(n - k - 1)/2$ [12]. A $(k+1)$-out-of-n Maximal Distance Separable (MDS) secret sharing scheme is a $(k + 1)$-out-of-n secret sharing scheme with the property that for any $k' \leq (n - k - 1)/2$, one can correct k' errors and simultaneously detect $n - k - k' - 1$ errors (as follows easily by generalizing [12, p. 10]). Maximal Distance Separable (MDS) secret sharing schemes can be constructed from any MDS codes, for example, from Reed-Solomon code [13].

In a message transmission protocol, the sender A starts with a message M^A drawn from a message space $\mathcal{M}$ with respect to a certain probability distribution. At the end of the protocol, the receiver B outputs a message M^B. We consider a synchronous system in which messages are sent via multicast in rounds. During each round of the protocol, each node receives any messages that were multicast for it at the end of the previous round, flips coins and perform local computations, and then possibly multicasts a message. We will also assume that the message space $\mathcal{M}$ is a subset of a finite field $\mathbf{F}$.

We consider two kinds of adversaries. A passive adversary (or gossiper adversary) is an adversary who can only observe the traffic through k internal nodes. An active adversary (or Byzantine adversary) is an adversary with unlimited computational power who can control k internal nodes. That is, an active adversary will not only listen to the traffics through the controlled nodes, but also control the message sent by those controlled nodes. Both kinds of adversaries are assumed to know the complete protocol specification, message space, and the complete structure of the graph. In this paper, we will not consider a dynamic adversary who could change the nodes it controls from round to round, instead we will only consider static adversaries. That is, at the start of the protocol, the adversary chooses the k faulty nodes. An alternative interpretation is that k nodes are static collaborating adversaries.

For any execution of the protocol, let *adv* be the adversary's view of the entire protocol. We write $adv(M, r)$ to denote the adversary's view when $M^A = M$ and when the sequence of coin flips used by the adversary is r.

Definition 1. *(see Franklin and Wright [7])*

1. *Let $\delta < \frac{1}{2}$. A message transmission protocol is* δ-reliable *if, with probability at least $1 - \delta$, B terminates with $M^B = M^A$. The probability is over the choices of M^A and the coin flips of all nodes.*
2. *A message transmission protocol is* reliable *if it is* 0-reliable.
3. *A message transmission protocol is* ε-private *if, for every two messages M_0, M_1 and every r, $\sum_c |\Pr[adv(M_0,r) = c] - \Pr[adv(M_1,r) = c]| \leq 2\varepsilon$. 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.*
4. *A message transmission protocol is* perfectly private *if it is* 0-private.
5. *A message transmission protocol is* (ε, δ)-secure *if it is ε-private and δ-reliable.*
6. *An (ε, δ)-secure message transmission protocol is* efficient *if its round complexity and bit complexity are polynomial in the size of the network, $\log \frac{1}{\varepsilon}$ (if $\varepsilon > 0$) and $\log \frac{1}{\delta}$ (if $\delta > 0$).*

For two nodes A and B in a directed graph such that there are $2k+1$ node disjoint paths from A to B, there is a straightforward reliable message transmission from A to B against a k-active adversary: A sends the message m to B via all the $2k + 1$ paths, and B recovers the message m by a majority vote.

3 $(0, \delta)$-Secure message transmission in directed graphs

Our discussion in this section will be concentrated on directed graphs. Dolev, Dwork, Waarts, and Yung [5] addressed the problem of secure message transmissions in a point-to-point network. In particular, they showed that if all channels from A to B are one-way, then $(3k + 1)$-connectivity is necessary and sufficient for (0,0)-secure message transmissions from A to B against a k-active adversary. They also showed that if all channels between A and B are two-way, then $(2k + 1)$-connectivity is necessary and sufficient for (0,0)-secure message transmissions between A and B against a k-active adversary. In this section we assume that there are only $2k + 1 - u$ directed node disjoint paths from A to B, where $1 \leq u \leq k$. We show that u directed node disjoint paths from B to A are necessary and sufficient to achieve $(0, \delta)$-secure message transmissions from A to B against a k-active adversary.

Franklin and Wright [7] showed that even if all channels between A and B are two way, $2k+1$ channels between A and B are still necessary for $(1-\delta)$-reliable (assuming that $\delta < \frac{1}{2}\left(1-\frac{1}{|\mathbf{F}|}\right)$) message transmission from A to B against a k-active adversary.

Theorem 1. *(Frandlin and Wright [7]) Let $G(V,E)$ be a directed graph, $A, B \in V$, and there are only $2k$ two-way node disjoint paths between A and B in G. Then δ-reliable message transmission from A to B against a k-active adversary is impossible for $\delta < \frac{1}{2}\left(1-\frac{1}{|\mathbf{F}|}\right)$.*

In the following, we first show that if there is no directed path from B to A, then $2k+1$ directed paths from A to B is necessary and sufficient for $(0,\delta)$-secure message transmission from A to B.

Theorem 2. *Let $G(V,E)$ be a directed graph, $A, B \in V$, and $0 < \delta < \frac{1}{2}$. If there is no directed paths from B to A, then the necessary and sufficient condition for $(0,\delta)$-secure message transmission from A to B against a k-active adversary is that there are $2k+1$ directed node disjoint paths from A to B.*

Proof. The necessity is proved in Theorem 1. Let $p_1, \ldots, p_{2k+1}$ be the $2k+1$ directed node disjoint paths from A to B. Let $M^A \in \mathbf{F}$ be the secret that A wants to send to B. A constructs $(k+1)$-out-of-$(2k+1)$ secret shares $(s_1^A, \ldots, s_{2k+1}^A)$ of M^A. The protocol proceeds from round 1 through round $2k+1$. In each round $1 \le i \le 2k+1$, we have the following steps:

Step 1 A chooses $\{(a_{i,j}^A, b_{i,j}^A) \in_R \mathbf{F}^2 : 1 \le j \le 2k+1\}$.

Step 2 A sends $(s_i^A, \text{auth}(s_i^A; a_{i,1}^A, b_{i,1}^A), \ldots, \text{auth}(s_i^A; a_{i,2k+1}^A, b_{i,2k+1}^A))$ to B via path p_i, and sends $(a_{i,j}^A, b_{i,j}^A)$ to B via path p_j for each $1 \le j \le 2k+1$.

Step 3 B receives $(s_i^B, c_{i,1}^B, \ldots, c_{i,2k+1}^B)$ via path p_i, and receives $(a_{i,j}^B, b_{i,j}^B)$ via path p_j for each $1 \le j \le 2k+1$.

Step 4 B computes $t = |\{j : c_{i,j}^B = \text{auth}(s_i^B; a_{i,j}^B, b_{i,j}^B)\}|$. If $t \ge k+1$, then B decides that s_i^B is a valid share. Otherwise B discards s_i^B.

It is easy to check that after the round $2k+1$, with high probability, B will get at least $k+1$ valid shares of s^A. Thus, with high probability, B will recover the secret $M^B = M^A$. It is straightforward that the protocol achieves perfect privacy. Thus the above protocol is a $(0,\delta)$-secure message transmission protocol from A to B against a k-active adversary. Q.E.D.

By Theorem 1, the necessary condition for $(0,\delta)$-secure message transmission from A to B against a k-active adversary is that there are at least $k+1$ node disjoint paths from A to B and there are at least $2k+1$ node disjoint paths in total from A to B and from B to A. In the following, we show that this condition is also sufficient. We first show that the condition is sufficient for $k=1$.

Theorem 3. *Let $G(V, E)$ be a directed graph, $A, B \in V$. If there are two directed node disjoint paths p_0 and p_1 from A to B, and one directed path q (which is node disjoint from p_0 and p_1) from B to A, then for any $0 < \delta < \frac{1}{2}$, there is a $(0, \delta)$-secure message transmission protocol from A to B against a 1-active adversary.*

Proof. In the following protocol, A $(0, \delta)$-securely transmits a message $M^A \in \mathbf{F}$ to B.

Step 1 A chooses $s_0^A \in_R \mathbf{F}$, $(a_0^A, b_0^A), (a_1^A, b_1^A) \in_R \mathbf{F}^2$, and let $s_1^A = M^A - s_0^A$. For each $i \in \{0, 1\}$, A sends $(s_i^A, (a_i^A, b_i^A), \mathrm{auth}(s_i^A; a_{1-i}^A, b_{1-i}^A))$ to B via path p_i.

Step 2 Assumes that B receives $(s_i^B, (a_i^B, b_i^B), c_i^B)$ via path p_i. B checks whether $c_i^B = \mathrm{auth}(s_i^B; a_{1-i}^B, b_{1-i}^B)$ for $i = 0, 1$. If both equations hold, then B knows that with high probability the adversary was either passive or not on the paths from A to B. B can recover the secret message, sends "OK" to A via the path q, and terminates the protocol. Otherwise, one of equations does not hold and B knows that the adversary was on one of the paths from A to B. In this case, B chooses $(a^B, b^B) \in_R \mathbf{F}^2$, and sends $((a^B, b^B), (s_0^B, (a_0^B, b_0^B), c_0^B), (s_1^B, (a_1^B, b_1^B), c_1^B))$ to A via the path q.

Step 3 If A receives "OK", then A terminates the protocol. Otherwise, from the information A received via path q, A decides which path from A to B is corrupted and recovers B's authentication key (a^A, b^A). A sends $(M^A, \mathrm{auth}(M^A; a^A, b^A))$ to B via the uncorrupted path from A to B.

Step 4 B recovers the message and checks that the authenticator is correct.

Similarly as in the proof of Theorem 2, it can be shown that the above protocol is $(0, \delta)$-secure against a 1-active adversary. Q.E.D.

Theorem 4. *Let $G(V, E)$ be a directed graph, $A, B \in V$, and $k \geq u \geq 1$. If there are $2k + 1 - u$ directed node disjoint paths $p_1, \ldots, p_{2k+1-u}$ from A to B, and u directed node disjoint paths $q_1, \ldots, q_u$ ($q_1, \ldots, q_u$ are node disjoint from $p_1, \ldots, p_{2k+1-u}$) from B to A, then for any $0 < \delta < \frac{1}{2}$, there is an efficient $(0, \delta)$-secure message transmission protocol from A to B against a k-active adversary.*

Before we give an efficient $(0, \delta)$-secure message transmission protocol from A to B. We first demonstrate the underlying idea by giving a non-efficient (exponential in k) $(0, \delta)$-secure message transmission protocol from A to B against a k-active adversary. Let $M^A \in \mathbf{F}$ be the secret that A wants to send to B, and $\mathcal{P}_1, \ldots, \mathcal{P}_t$ be an enumeration of size $k+1$ subsets of $\{p_1, \ldots, p_{2k+1-u}, q_1, \ldots, q_u\}$. The protocol proceeds from round 1 through t. In each round $1 \leq m \leq t$, we have the following steps:

Step 1 For each $p_i \in \mathcal{P}_m$, A chooses $(a_{i,m}^A, b_{i,m}^A) \in_R \mathbf{F}^2$ and sends $(a_{i,m}^A, b_{i,m}^A)$ to B via p_i.

Step 2 For each $p_i \in \mathcal{P}_m$, B receives $(a_{i,m}^B, b_{i,m}^B)$ from A via p_i.

Step 3 For each $q_i \in \mathcal{P}_m$, B chooses $(c^B_{i,m}, d^B_{i,m}) \in_R \mathbf{F}^2$ and sends $(c^B_{i,m}, d^B_{i,m})$ to A via q_i.

Step 4 For each $q_i \in \mathcal{P}_m$, A receives $(c^A_{i,m}, d^A_{i,m})$ from B via q_i.

Step 5 A computes $C^A = \sum_{p_i \in \mathcal{P}_m} a^A_{i,m} + \sum_{q_i \in \mathcal{P}_m} c^A_{i,m}$, $D^A = \sum_{p_i \in \mathcal{P}_m} b^A_{i,m} + \sum_{q_i \in \mathcal{P}_m} d^A_{i,m}$, and sends $(M^A + C^A, \mathrm{auth}(M^A + C^A; C^A, D^A))$ to B via all paths in p_i in $\mathcal{P}_m$.

Step 6 For each $p_i \in \mathcal{P}_m$, B receives $(e^B_{i,m}, f^B_{i,m})$ from A via p_i.

Step 7 If $(e^B_{i,m}, f^B_{i,m}) = (e^B_{j,m}, f^B_{j,m})$ for all $p_i, p_j \in \mathcal{P}_m$, then B goes to Step 8. Otherwise, B goes to round $m + 1$.

Step 8 B computes $C^B = \sum_{p_i \in \mathcal{P}_m} a^B_{i,m} + \sum_{q_i \in \mathcal{P}_m} c^B_{i,m}$, $D^B = \sum_{p_i \in \mathcal{P}_m} b^B_{i,m} + \sum_{q_i \in \mathcal{P}_m} d^B_{i,m}$.

Step 9 If $f^B_{i,m} = \mathrm{auth}(f^B_{i,m}; C^B, D^B)$, then B computes the secret $M^B = f^B_{i,m} - C^B$ and terminates the protocol. Otherwise, B goes to round $m + 1$.

Since there is at least one $\mathcal{P}_m$ such that all paths in $\mathcal{P}_m$ are not corrupted, B receives the correct secret by the end of the protocol with high probability. It is also straightforward to check that the above protocol has perfect secrecy.

Proof. (Proof of Theorem 4) Let $M^A \in \mathbf{F}$ be the secret that A wants to send to B. A constructs $(k+1)$-out-of-$(2k+1-u)$ secret shares $(s^A_1, \ldots, s^A_{2k+1-u})$ of M^A. The protocol proceeds from round 1 through $2k + 2 + u$. For each round $1 \le i \le 2k + 1 - u$, we have the following steps:

Step 1 A chooses $\{(a^A_{i,j}, b^A_{i,j}) \in_R \mathbf{F}^2, : 1 \le j \le 2k + 1 - u\}$.

Step 2 A sends $\{s^A_i, \mathrm{auth}(s^A_i; a^A_{i,1}, b^A_{i,1}), \ldots, \mathrm{auth}(s^A_i; a^A_{i,2k+1-u}, b^A_{i,2k+1-u})\}$ to B via path p_i, and sends $(a^A_{i,j}, b^A_{i,j})$ to B via path p_j for each $1 \le j \le 2k + 1 - u$.

Step 3 B receives $\{s^B_i, d^B_{i,1}, \ldots, d^B_{i,2k+1-u}\}$ via path p_i, and $(a^B_{i,j}, b^B_{i,j})$ via path p_j for each $1 \le j \le 2k + 1 - u$.

Step 4 B computes $t = |\{j : d^B_{i,j} = \mathrm{auth}(s^B_i; a^B_{i,j}, b^B_{i,j})\}|$. If $t \ge k + 1$, then B decides that s^B_i is a valid share. Otherwise B decides that s^B_i is an invalid share.

At the end of round $2k + 1 - u$, if B has received $k + 1$ valid shares, then B recovers the secret M^B from these valid shares and terminates the protocol. Otherwise, B proceeds to round $2k + 2 - u$. In round $2k + 2 - u$, we have the following steps:

Step 1 A chooses $\{(a^A_i, b^A_i, c^A_i) \in_R \mathbf{F}^3 : 1 \le i \le 2k + 1 - u\}$, and sends (a^A_i, b^A_i, c^A_i) to B via path p_i for each $i \le 2k + 1 - u$.

Step 2 For each $1 \le i \le 2k + 1 - u$, B receives (a^B_i, b^B_i, c^B_i) on path p_i from A (if no value is received on path p_i, B sets it to a default value).

Step 3 For each $1 \le i \le 2k + 1 - u$, B chooses $r^B_i \in_R \mathbf{F}$ and computes $\beta^B = \{(r^B_i, \mathrm{auth}(r^B_i; a^B_i, b^B_i, c^B_i)) : 1 \le i \le 2k + 1 - u\}$.

In each round $2k+3-u \leq i \leq 2k+2$, we have the following steps:

Step 1 B chooses $(d_i^B, e_i^B) \in_R \mathbf{F}^2$ and $\{(v_{i,j}^B, w_{i,j}^B) \in_R \mathbf{F}^2 : 1 \leq j \leq u\}$.

Step 2 B sends (d_i^B, e_i^B), β^B, and $\{\mathrm{auth}(\langle d_i^B, e_i^B\rangle; v_{i,j}^B, w_{i,j}^B) : 1 \leq j \leq u\}$ to A via path q_i, and $(v_{i,j}^B, w_{i,j}^B)$ to A via path q_j for each $1 \leq j \leq u$.

Step 3 A receives (or substitutes default values) (d_i^A, e_i^A), β_i^A, and $\{\alpha_{i,j}^A : 1 \leq j \leq u\}$ from B via path q_i, and $(v_{i,j}^A, w_{i,j}^A)$ from B via path q_j for each $1 \leq j \leq u$.

According to the values that A has received, A divides the paths set $\{q_1, \ldots, q_u\}$ into subsets $\mathcal{Q}_1, \ldots, \mathcal{Q}_t$ such that for any l, m, n with $1 \leq l \leq t$, $1 \leq m, n \leq u$, and $q_m, q_n \in \mathcal{Q}_l$, we have

1. $\beta_m^A = \beta_n^A$;
2. $\alpha_{m,n}^A = \mathrm{auth}(\langle d_m^B, e_m^B\rangle; v_{m,n}^B, w_{m,n}^B)$;
3. $\alpha_{n,m}^A = \mathrm{auth}(\langle d_n^B, e_n^B\rangle; v_{n,m}^B, w_{n,m}^B)$.

For each $\mathcal{Q}_l$, let $q_m \in \mathcal{Q}_l$ and $\beta_m^A = \{(r_{i,l}^A, \gamma_{i,l}^A) : 1 \leq i \leq 2k+1-u\}$. A computes the number

$$t_l = |\{i : \gamma_{i,l}^A = \mathrm{auth}(r_{i,l}^A; a_i^A, b_i^A, c_i^A), 1 \leq i \leq 2k+1-u\}| + |\mathcal{Q}_l|$$

If $t_l \leq k$, then A decides that $\mathcal{Q}_l$ is an unacceptable set, otherwise, A decides that $\mathcal{Q}_l$ is an acceptable set. Let $\mathcal{Q}_l = \emptyset$ for $t < l \leq u$.

For each round $2k+3 \leq l \leq 2k+2+u$, we have the following steps:

Step 1 If $\mathcal{Q}_l = \emptyset$ or $\mathcal{Q}_l$ is an unacceptable set, then go to round $l+1$.

Step 2 A computes $\mathcal{P}_l = \{p_i : \gamma_{i,l}^A = \mathrm{auth}(r_{i,l}^A; a_i^A, b_i^A, c_i^A), 1 \leq i \leq 2k+1-u\}$, $C_l^A = \sum_{p_i \in \mathcal{P}_l} a_i^A + \sum_{q_i \in \mathcal{Q}_l} d_i^A$, and $D_l^A = \sum_{p_i \in \mathcal{P}_l} b_i^A + \sum_{q_i \in \mathcal{Q}_l} e_i^A$.

Step 3 A sends $(\langle \mathcal{Q}_l, \mathcal{P}_l, M^A + C_l^A\rangle, \mathrm{auth}(\langle \mathcal{Q}_l, \mathcal{P}_l, M^A + C_l^A\rangle; C_l^A, D_l^A))$ to B via all paths $p_i \in \mathcal{P}_l$.

Step 4 B receives $(\alpha_{i,l}^B, \beta_{i,l}^B)$ from path p_i for $1 \leq i \leq 2k+1-u$.

Step 5 For each $1 \leq i \leq 2k+1-u$, B computes $\alpha_{i,l}^B = \langle \mathcal{Q}_{i,l}^B, \mathcal{P}_{i,l}^B, \beta_{i,l}^B\rangle$, $C_{i,l}^B = \sum_{p_j \in \mathcal{P}_{i,l}} a_j^B + \sum_{q_j \in \mathcal{Q}_{i,l}} d_j^B$, and $D_{i,l}^B = \sum_{p_j \in \mathcal{P}_{i,l}} b_j^B + \sum_{q_j \in \mathcal{Q}_{i,l}} e_j^B$.

Step 6 For each $1 \leq i \leq 2k+1-u$, B checks whether $\beta_{i,l}^B = \mathrm{auth}(\alpha_{i,l}^B; C_{i,l}^B, D_{i,l}^B)$. If the equation holds, then B computes the secret $M^B = \beta_{i,l}^B - C_{i,l}^B$.

If B has not got the secret at the end of round $2k+1-u$, then there exists an uncorrupted path q_j from B to A and a paths set $\mathcal{Q}_l$ such that $q_j \in \mathcal{Q}_l$ and the information that A receives from paths in $\mathcal{Q}_l$ are reliable. Thus, at the end of round $2k+2+u$, B will output a secret M^B. It is easy to check that, with high probability, this secret is the same as M^A.

It is straightforward to show that the protocol achieves perfect privacy. Thus it is a $(0, \delta)$-secure message transmission transmission protocol from A to B against a k-active adversary. Q.E.D.

4 $(0,0)$-Secure message transmission in directed graphs

In the previous section, we addressed probabilistic reliable message transmission in directed graphs. In this section, we consider perfectly reliable message transmission in directed graphs. We will show that if there are u directed node disjoint paths from B to A, then a necessary and sufficient condition for $(0,0)$-secure message transmission from A to B against a k-active adversary is that there are $\max\{3k+1-2u, 2k+1\}$ directed node disjoint paths from A to B.

Theorem 5. *Let $G(V,E)$ be a directed graph, $A, B \in V$. Assume that there are u directed node disjoint paths from B to A. Then a necessary condition for $(0,0)$-secure message transmission from A to B against a k-active adversary is that there are $\max\{3k+1-2u, 2k+1\}$ directed node disjoint paths from A to B.*

Proof. If $u = 0$, then by the results in [5], we need $3k+1$ directed node disjoint paths from A to B for $(0,0)$-secure message transmission against a k-active adversary. If $u \geq \lceil \frac{k}{2} \rceil$, then again by the results in [5], we need $2k+1$ directed node disjoint paths from A to B for 0-reliable (that is, perfectly reliable) message transmission from A to B against a k-active adversary. From now on, we assume that $0 < u < \lceil \frac{k}{2} \rceil$.

For a contradiction, we assume that there are only $3k-2u$ directed node disjoint paths from A to B, denoted as $p_1, \ldots, p_{3k-2u}$. Let $q_1, \ldots, q_u$ be the directed node disjoint paths from B to A.

Let Π be a $(0,0)$-secure message transmission protocol from A to B. In the following, we will construct a k-active adversary to defeat this protocol. The transcripts distribution $view_{\Pi}^{A}$ of A is drawn from a probability distribution that depends on the message M^A to be transmitted by A, the coin flips R^A of A, the coin flips R^B of B, the coin flips $R^{\mathcal{A}}$ of the adversary (without loss of generality, we assume that the value $R^{\mathcal{A}}$ will determine the choice of faulty paths controlled by the adversary), and the coin flips $R^{\mathcal{T}}$ of all other honest nodes. Without loss of generality, we can assume that the protocol proceeds in steps, where A is silent during even steps and B is silent during odd steps (see [5]).

The strategy of the adversary is as follows. First it uses $R^{\mathcal{A}}$ to choose a value b. If $b = 0$, then it uses $R^{\mathcal{A}}$ again to choose k directed paths $p_{a_1}, \ldots, p_{a_k}$ from A to B and controls the first node on each of these k paths. If $b = 1$, then it uses $R^{\mathcal{A}}$ again to choose $k-u$ directed paths $p_{a_1}, \ldots, p_{a_{k-u}}$ from A to B and controls the first node on each of these $k-u$ paths and the first node on each of the u paths from B to A. It also uses $R^{\mathcal{A}}$ to choose a message $\hat{M}^A \in \mathbf{F}$ according to the same probability distribution from which the actual message M^A was drawn. In the following we describe the protocol the adversary will follow.

- Case $b = 0$. The k paths $p_{a_1}, \ldots, p_{a_k}$ behaves as a passive adversary. That is, it proceeds according to the protocol Π.
- Case $b = 1$. The $k-u$ paths $p_{a_1}, \ldots, p_{a_{k-u}}$ ignores what A sends in each step of the protocol and simulates what A would send to B when A sending $\hat{M}^A$

to B. The u paths from B ignores what B sends in each step of the protocol and simulates what B would send to A when $b = 0$.

In the following, we assume that the tuple $(M^A, R^A, R^B, R^{\mathcal{T}}, R^{\mathcal{A}})$ is fixed, $b = 0$, the protocol halts in l steps, and the view of A is $view^A_{\Pi}(M^A, R^A, R^B, R^{\mathcal{T}}, R^{\mathcal{A}})$. Let $\alpha^A_{i,j}$ be the values that A sends on path p_i in step j and $\boldsymbol{\alpha}^A_i = (\alpha^A_{i,1}, \ldots, \alpha^A_{i,l})$. We can view $\boldsymbol{\alpha}^A_i$ as shares of the message M^A. Similarly, let $\alpha^B_{i,j}$ be the values that B receives on path p_i in step j and $\boldsymbol{\alpha}^B_i = (\alpha^B_{i,1}, \ldots, \alpha^B_{i,l})$.

First, it is straightforward to show that for any k paths $p_{a_1}, \ldots, p_{a_k}$ from A to B, there is an $\hat{R}^{\mathcal{A}}_1$ such that $b = 0$, the adversary controls the paths $p_{a_1}, \ldots, p_{a_k}$, and

$$view^A_{\Pi}(M^A, R^A, R^B, R^{\mathcal{T}}, R^{\mathcal{A}}) = view^A_{\Pi}(M^A, R^A, R^B, R^{\mathcal{T}}, \hat{R}^{\mathcal{A}}_1) \qquad (1)$$

Due to the fact that Π is a perfectly private message transmission protocol, from any k shares from $(\boldsymbol{\alpha}^A_1, \boldsymbol{\alpha}^A_2, \ldots, \boldsymbol{\alpha}^A_{3k-2u})$ one cannot recover the secret message M^A. Thus $(\boldsymbol{\alpha}^A_1, \boldsymbol{\alpha}^A_2, \ldots, \boldsymbol{\alpha}^A_{3k-2u})$ is at least a $(k+1)$-out-of-$(3k-2u)$ secret sharing scheme.

Secondly, for any $k-u$ paths $p_{a_1}, \ldots, p_{a_{k-u}}$ from A to B, there is an $\hat{R}^{\mathcal{A}}_2$ such that $b = 1$, $\hat{M}^A \neq M^A$, the adversary controls the paths $p_{a_1}, \ldots, p_{a_{k-u}}, q_1, \ldots, q_u$, and

$$view^A_{\Pi}(M^A, R^A, R^B, R^{\mathcal{T}}, R^{\mathcal{A}}) = view^A_{\Pi}(M^A, R^A, R^B, R^{\mathcal{T}}, \hat{R}^{\mathcal{A}}_2) \qquad (2)$$

Due to the fact that Π is a perfectly reliable message transmission protocol, any $k-u$ errors in the shares $(\boldsymbol{\alpha}^B_1, \boldsymbol{\alpha}^B_2, \ldots, \boldsymbol{\alpha}^B_{3k-2u})$ can be corrected by B to recover the secret message M^A.

In summary, $(\boldsymbol{\alpha}^A_1, \boldsymbol{\alpha}^A_2, \ldots, \boldsymbol{\alpha}^A_{3k-2u})$ is at least a $(k+1)$-out-of-$(3k-2u)$ secret sharing scheme that can correct $k-u$ errors. By the results in [12], we know that the maximum number of errors that a $(k+1)$-out-of-$(3k-2u)$ secret sharing scheme could correct is

$$\left\lfloor \frac{(3k-2u)-k-1}{2} \right\rfloor = \left\lfloor \frac{2k-2u-1}{2} \right\rfloor = k-u-1.$$

This is a contradiction, which concludes the proof. Q.E.D.

For the sufficient condition, we first show the simple case for $u = 1$.

Theorem 6. *Let $G(V, E)$ be a directed graph, $A, B \in V$, and $k \geq 2$. If there are $3k-1$ directed node disjoint paths $p_1, \ldots, p_{3k-1}$ from A to B and one directed path q from B to A (q is node disjoint from $p_1, \ldots, p_{3k-1}$) then there is a $(0,0)$-secure message transmission protocol from A to B against a k-active adversary.*

Proof. In the following protocol π, A $(0,0)$-securely transmits $M^A \in_R \mathbf{F}$ to B:

Step 1 Both A and B sets $I = 1$.

Step 2 A constructs $(k+1)$-out-of-$(3k-1)$ MDS secret shares $(s^A_{1,I}, ..., s^A_{3k-1,I})$ of M^A. For each $1 \leq i \leq 3k-1$, A sends $s^A_{i,I}$ to B via the path p_i.

Step 3 For each $i \leq 3k-1$, B receives $s^B_{i,I}$ on path p_i. By correcting at most $k-1$ errors, B recovers a value $\hat{M}^B_I$ from these shares. B sends $s^B_{I,I}$ to A via the path q.

Step 4 A receives $\bar{s}^A_{I,I}$ from q. A distinguishes the following two cases:

1. $\bar{s}^A_{I,I} = s^A_{I,I}$. A reliably sends "path p_I maybe OK", sets $I = I + 1$. If $I > 3k-1$ then goes to Step 6, otherwise, goes to Step 2.
2. $\bar{s}^A_{I,I} \neq s^A_{I,I}$. A reliably sends "path p_I or q is faulty". A constructs k-out-of-$(3k-2)$ MDS secret shares $\{r^A_{i,I} : i \neq I, 1 \leq i \leq 3k-1\}$ of M^A. For each $i \leq 3k-1$ such that $i \neq I$, A sends $r^A_{i,I}$ to B via the path p_i. A terminates the protocol.

Step 5 B distinguishes the following two cases:

1. B reliably receives "path p_I maybe OK". B sets $I = I + 1$. If $I > 3k-1$ then goes to Step 6, otherwise, goes to Step 2.
2. B reliably receives "path p_I or q is faulty". In this case, B also receives $r^B_{i,I}$ from path p_i for each $i \neq I$. By correcting at most $k-1$ errors, B recovers M^B from these shares and terminates the protocol.

Step 6 B checks whether $\hat{M}^B_i = \hat{M}^B_j$ for all $1 \leq i, j \leq 3k-1$. If all these values are equal, then B sets $M^B = \hat{M}^B_1$, sends "stop" to A via q, and terminates the protocol. Otherwise, B sends the shares $(s^B_{1,3k-1}, ..., s^B_{3k-1,3k-1})$ to A via q.

Step 7 A distinguishes the following two cases:

1. A receives $(\bar{s}^A_{1,3k-1}, ..., \bar{s}^A_{3k-1,3k-1})$ on the path q. A computes $\mathcal{P} = \{i : \bar{s}^A_{i,3k-1} \neq s^A_{i,3k-1}\}$, reliably sends $\mathcal{P}$ to B, and terminates the protocol.
2. A receives anything else. A terminates the protocol.

Step 8 B reliable receives $\mathcal{P}$ from A, recovers M^B from the shares $\{s^B_{i,3k-1} : i \notin \mathcal{P}\}$, and terminates the protocol.

Since there could be k faulty paths from A to B, and a $(k+1)$-out-of-$(3k-1)$ MDS secret sharing scheme can correct at most $k-1$ errors and simultaneously detect $k-1$ errors. B may recover an incorrect message $\hat{M}^B_I$ in Step 3. B therefore needs to verify whether it has recovered the correct message in the following steps. Note that if q is faulty, then B must have recovered the correct message in Step 3 for each I. In Step 3, B also sends $s^B_{I,I}$ to A via the path q. This does not violate the perfect privacy property since if there are $k-1$ faulty paths from A to B, then the adversary gets at most k shares including this share, and if there are k faulty paths from A to B, then the adversary does not control the path q and does not get this share.

In Step 4, if $\bar{s}^A_{I,I} = s^A_{I,I}$, then it could be the case that $s^B_{I,I} = s^A_{I,I}$ or it could be the case that the path q is faulty. In any case, we have to continue the protocol further. However, if $\bar{s}^A_{I,I} \neq s^A_{I,I}$, then A is convinced that either p_I or q is faulty. Since a k-out-of-$(3k-2)$ MDS secret sharing scheme could correct

$k-1$ errors and detect $k-1$ errors simultaneously. B will recover the correct message in Step 5.

Now assume that B does not recover the correct message at the end of round $I = 3k-1$. We can distinguish the following two cases:

- B recovers the same message $\hat{M}_I^B$ in all $3k-1$ rounds. If this happens, B is convinced that this uniquely recovered message is the correct message. Note that this follows from the following two arguments: If the path q is faulty, then obviously B has recovered the same correct message in each round. If the path q is non-faulty and we assume that the path p_t $(1 \leq t \leq 3k-1)$ is faulty, then in order for the adversary to avoid being caught by A in round t, p_t must behave nicely in round t, that is, $s_{t,t}^A = s_{t,t}^B$ (otherwise A has identified that q or p_t is faulty in round t). Thus there are at most $k-1$ errors in the shares that B received in round t and B recovers the correct message in round t.
- B recovers different messages in these $3k-1$ rounds. This happens only if q is honest. Thus, B could send the shares it receives in round $3k-1$ to A via path q and A can tell B which shares are incorrect. Thus B could recover the correct message from these non-faulty shares (there are at least $2k-1$ non-faulty shares).

The above arguments show that the protocol π is $(0,0)$-secure against a k-active adversary. Q.E.D.

Theorem 7. *Let $G(V, E)$ be a directed graph, $A, B \in V$, and $k \geq 2$. If there are $n = \max\{3k+1-2u, 2k+1\}$ directed node disjoint paths $p_1, \ldots, p_n$ from A to B and u directed path $q_1, \ldots, q_u$ from B to A ($q_1, \ldots, q_u$ are node disjoint from $p_1, \ldots, p_n$) then there is a $(0,0)$-secure message transmission protocol from A to B against a k-active adversary.*

Proof. For $u = 1$ or $k = 2$, the result is proved in Theorem 6. We prove the theorem by induction. Assume that $u > 1$, $k > 2$, and the theorem holds for $u-1$ and $k-1$. In the following, we show that the Theorem holds for u and k by induction.

Let $\mathcal{H} = \{h_I : h_I = \langle p_{I_1}, \ldots, p_{I_u} \rangle\}$ be the set of all ordered u-subsets of $\{p_1, \ldots, p_n\}$. Then $|\mathcal{H}| = \frac{n!}{(n-u)!}$. In the following protocol π, A $(0,0)$-securely transmits $M^A \in_R \mathbf{F}$ to B:

Step 1 Both A and B set $I = 1$.

Step 2 A constructs $(k+1)$-out-of-n MDS secret shares $(s_{1,I}^A, ..., s_{n,I}^A)$ of M^A. For each $i \leq n$, A sends $s_{i,I}^A$ to B via the path p_i.

Step 3 For each $i \leq n$, B receives (or sets default) $s_{i,I}^B$ on path p_i. By correcting at most $k-u$ errors, B recovers a value $\hat{M}_I^B$ from these shares. For each $i \leq u$, B sends $s_{I_i,I}^B$ to A via the path q_i. Note that we assume $h_I = \langle p_{I_1}, \ldots, p_{I_u} \rangle$ here.

Step 4 For each $i \leq u$, A receives (or sets default) $\bar{s}_{I_i,I}^A$ from q_i. A distinguishes the following two cases:

1. $\bar{s}^A_{I_i,I} = s^A_{I_i,I}$ for all $i \leq u$. A reliably sends "paths in h_I maybe OK", sets $I = I + 1$. If $I > |\mathcal{H}|$ then A goes to Step 6, otherwise, goes to Step 2.
2. $\bar{s}^A_{I_{i_0},I} \neq s^A_{I_{i_0},I}$ for some $i_0 \leq u$. A reliably sends "path $p_{I_{i_0}}$ or q_{i_0} is faulty". A goes to the $(0,0)$-secure message transmission protocol against a $(k-1)$-active adversary on the paths $\{p_i : i \neq I_{i_0}\} \cup \{q_i : i \neq i_0\}$ to transmit M^A to B (here we use induction).

Step 5 B distinguishes the following two cases:

1. B reliably receives "paths in h_I maybe OK". B sets $I = I + 1$. If $I > |\mathcal{H}|$ then goes to Step 6, otherwise, goes to Step 2.
2. B reliably receives "path $p_{I_{i_0}}$ or q_{i_0} is faulty". In this case, B goes to the $(0,0)$-secure message transmission protocol against a $(k-1)$-active adversary on the paths $\{p_i : i \neq I_{i_0}\} \cup \{q_i : i \neq i_0\}$ and receives the message M^B.

Step 6 B computes whether $\hat{M}^B_i = \hat{M}^B_j$ for all $i, j \leq |\mathcal{H}|$. If all these values are equal, then B sets $M^B = \hat{M}^B_1$, sends "stop" to A via all paths q_i, and terminates the protocol. Otherwise, B goes to Step 8.

Step 7 If A receives "stop" on all paths $q_1, \ldots, q_u$, then A terminates the protocol, otherwise, A goes to Step 8.

Step 8 A chooses $R^A_1 \in_R \mathbf{F}$, constructs $(k+1)$-out-of-n MDS secret shares $(s^A_1, \ldots, s^A_n)$ of R^A_1, and sends s^A_i to B via path p_i for each $i \leq n$.

Step 9 For each $i \leq n$, B receives (or sets default) s^B_i on path p_i. B distinguishes the following two cases:

1. There are errors in the shares $(s^B_1, \ldots, s^B_n)$. In this case, for each $j \leq u$, B sends $(s^B_1, \ldots, s^B_n)$ to A via the path q_j. Note that a $(k+1)$-out-of-n MDS secret sharing scheme could be used to detect $n - k - 1 \geq k$ errors.
2. There is no error in the shares $(s^B_1, \ldots, s^B_n)$. B recovers the value R^B_1 from these shares and for each $j \leq u$, B sends "OK" to A via path q_j.

Step 10 For each $j \leq u$, A receives (or sets default) $(\bar{s}^A_{1,j}, \ldots, \bar{s}^A_{n,j})$ from the path q_j. A distinguishes the following two cases:

1. $\bar{s}^A_{i_0,j_0} \neq s^A_{i_0}$ for some $i_0 \leq n$ and $j_0 \leq u$. A reliably sends "path p_{i_0} or q_{j_0} is faulty" to B. A goes to the $(0,0)$-secure message transmission protocol against a $(k-1)$-active adversary on the paths $\{p_i : i \neq i_0\} \cup \{q_j : j \neq j_0\}$ to transmit M^A to B (here we use induction again).
2. All other cases. A reliably transmits "continue the protocol" to B and goes to Step 12.

Step 11 B distinguishes the following two cases:

1. B reliably receives "continue the protocol". B goes to Step 12.
2. B reliably receives "path p_{i_0} or q_{j_0} is faulty". In this case, B goes to the $(0,0)$-secure message transmission protocol against a $(k-1)$-active adversary on the paths $\{p_i : i \neq i_0\} \cup \{q_j : j \neq j_0\}$ and receives the message M^B.

Step 12 A computes $R_2^A = M^A - R_1^A$, constructs $(k+1)$-out-of-n MDS secret shares $(s_1^A, \ldots, s_n^A)$ of R_2^A, and sends s_i^A to B via path p_i for each $i \leq n$.

Step 13 For each $i \leq n$, B receives (or sets default) s_i^B on path p_i. B distinguishes the following two cases:

1. There are errors in the shares $(s_1^B, \ldots, s_n^B)$. In this case, for each $j \leq u$, B sends $(s_1^B, \ldots, s_n^B)$ to A via the path q_j.
2. There is no error in the shares $(s_1^B, \ldots, s_n^B)$. B recovers the value R_2^B from these shares, computes the secret $M^B = R_1^B + R_2^B$, and for each $j \leq u$, B sends "OK" to A via path q_j. B terminates the protocol.

Step 14 For each $j \leq u$, A receives (or sets default) $(\bar{s}_{1,j}^A, \ldots, \bar{s}_{u,j}^A)$ from the path q_j. A distinguishes the following two cases:

1. $\bar{s}_{i_0,j_0}^A \neq s_{i_0}^A$ for some $i_0 \leq n$ and $j_0 \leq u$. A reliably sends "path p_{i_0} or q_{j_0} is faulty" to B. A goes to the $(0,0)$-secure message transmission protocol against a $(k-1)$-active adversary on the paths $\{p_i : i \neq i_0\} \cup \{q_j : j \neq j_0\}$ to transmit M^A to B.
2. All other cases. A reliably transmits "the protocol is complete" to B and terminates the protocol.

Step 15 B distinguishes the following two cases:

1. B reliably receives "the protocol is complete". B terminates the protocol.
2. B reliably receives "path p_{i_0} or q_{j_0} is faulty". In this case, B goes to the $(0,0)$-secure message transmission protocol against a $(k-1)$-active adversary on the paths $\{p_i : i \neq i_0\} \cup \{q_j : j \neq j_0\}$ and receives the message M^B.

Since there could be k faulty paths from A to B, and a $(k+1)$-out-of-n MDS secret sharing scheme in Steps 2 and 3 can correct at most $k-u$ errors and simultaneously detect $k-u$ errors. B may recover an incorrect message $\hat{M}_I^B$ in Step 3. B therefore needs to verify whether it has recovered the correct message in the following steps. Note that if all the paths from B to A are faulty, then B must have recovered the correct message in Step 3 for each $I \leq |\mathcal{H}|$. In Step 3, B also sends $s_{I_i,I}^B$ to A via the path q_i. This will not violate the perfect privacy property since if there are t faulty paths from B to A, then the adversary gets $k-t$ shares from the A to B paths and t shares from the B to A paths. That is, the adversary gets at most k shares.

In Step 4, if $\bar{s}_{I_i,I}^A = s_{I_i,I}^A$ for all $i \leq u$, then for each $i \leq u$, it could be the case that $s_{I_i,I}^B = s_{I_i,I}^A$ or it could be the case that the path q_i is faulty. In any case, we have to continue the protocol further. However, if $\bar{s}_{I_{i_0},I}^A \neq s_{I_{i_0},I}^A$ for some $i_0 \leq u$, then A is convinced that either $p_{I_{i_0}}$ or q_{i_0} is faulty. Thus if we delete the two paths $p_{I_{i_0}}$ and q_{i_0}, we have at most $k-1$ unknown faulty paths, $n-1$ paths from A to B, and $u-1$ paths from B to A. Since

$$\begin{aligned}\max\{3(k-1)+1-2(u-1), 2(k-1)+1\} &= \max\{3k-2u-4, 2k-1\} \\ &\leq \max\{3k-2u, 2k\} \\ &= n-1,\end{aligned}$$

there is (by induction) a $(0,0)$-secure message transmission protocol from A to B against a $(k-1)$-active adversary on the paths $\{p_i : i \neq I_{i_0}\} \cup \{q_i : i \neq i_0\}$, B recovers the correct message M^B in Step 5.

Assume that B does not recover the correct message at the beginning of Step 6. If B recovers the same value $\hat{M}_I^B$ in all the $|\mathcal{H}|$ rounds between Step 2 and Step 5, then B is convinced that this uniquely recovered value is the correct message. Note that this follows from the following arguments:

- All paths $q_1, \ldots, q_u$ from B to A are faulty. In this case B obviously has recovered the correct message in each round.
- There is non-faulty path from B to A. In this case, let $t \geq 1$, $q_{i_1}, \ldots, q_{i_t}$ be a list of all non-faulty paths from B to A, $p_{j_1}, \ldots, p_{j_t}$ be faulty, and $h_I = \langle p_{I_1}, \ldots, p_{I_u} \rangle$ with $I_{i_1} = j_1, \ldots, I_{i_t} = j_t$. If $(s^B_{1,I}, ..., s^B_{n,I})$ is the shares that B receives in Step 3 of round I, then $s^B_{j_1,I} = s^A_{j_1,I}$, ..., $s^B_{j_t,I} = s^A_{j_t,I}$ (otherwise A identifies that some q_i or p_j is faulty in round I). That is, there are at most $k-u$ errors in the shares $(s^B_{1,I}, ..., s^B_{n,I})$, and B recovers the correct secret message in round I.

Now assume that B does not recover the correct message at the beginning of Step 6 and B recovers different values in these $|\mathcal{H}|$ rounds between Step 2 and Step 5. If this happens, then there must be non-faulty paths from B to A. In this case, both A and B continues the protocol from Step 8. During Step 8 and Step 15, A tries to send R_1^A and R_2^A to B using the $(k+1)$-out-of-n MDS secret sharing scheme. Since there are at most k faulty paths, and a $(k+1)$-out-of-n MDS secret sharing scheme could be used to correct 0 error and simultaneously detect at least k errors, any error in these shares could be detected by B in Steps 9 and 13. Since there are non-faulty paths from B to A, any errors in these shares will be reported back A via the non-faulty B to A paths. Thus A initiates a $(0,0)$-secure message transmission protocol against a $(k-1)$-active adversary on the paths $\{p_i : i \neq i_0\} \cup \{q_j : j \neq j_0\}$ in Step 10 or Step 14 and B will receive the secret. If any error occurs in these cases, B reports either the shares of R_1^A or the shares of R_2^A (but not both) to A via the B to A paths. Thus we have achieved the perfect privacy here. On the other hand, if there is no error in these shares of R_1^A and R_2^A, then B recovers the correct secret $M^B = R_1^B + R_2^B$.

We therefore proved that the protocol π is $(0,0)$-secure against a k-active adversary. Q.E.D.

In Theorem 7, we have the restriction that $k \geq 2$. In the following we show a sufficient condition which is applicable to $k = 1$.

Theorem 8. *Let $G(V,E)$ be a directed graph, $A, B \in V$. If there are $3k$ directed node disjoint paths $p_1, \ldots, p_{3k}$ from A to B and one directed path q from B to A (q is node disjoint from $p_1, \ldots, p_{3k}$) then there is a $(0,0)$-secure message transmission protocol from A to B against a k-active adversary.*

Proof. In the following protocol π, A $(0,0)$-securely transmits $M^A \in_R \mathbf{F}$ to B.

Step 1 A constructs $(k+1)$-out-of-$3k$ MDS secret shares $v^A = (s_1^A, ..., s_{3k}^A)$ of M^A. For each $1 \le i \le 3k$, A sends s_i to B via the path p_i.

Step 2 Let $v^B = (s_1^B, ..., s_{3k}^B)$ be the shares B receives. If B finds that there are at most $k-1$ errors, B recovers M^B from the shares, sends "stop" to A via the path q, and terminates the protocol. Otherwise there are k errors. In this case B sends v^B back to A via the path q (note that q is an honest path in this case).

Step 3 A distinguishes the following two cases:

1. A receives $\bar{v}^A = (\bar{s}_1^A, ..., \bar{s}_{3k}^A)$ from the path q. A reliably sends $\mathcal{P} = \{i : s_i^A \ne \bar{s}_i^A\}$ to B.
2. A received "stop" or anything else via q. A terminates the protocol.

Step 4 B reliably receives $\mathcal{P}$ from A. B recovers M^B from the shares $\{s_i^B : i \notin \mathcal{P}\}$ and terminates the protocol (note that $|\{s_i^B : i \notin \mathcal{P}\}| = 2k$).

Note that if B sends v^B to A in Step 2 then k paths from A to B are corrupted and the path q is honest. Thus the adversary will not learn v^B. If the adversary controls the path q, then it may change the message "stop" to something else. In this case, A will not be able to identify the corrupted paths from A to B. However, since B has already recovered the key, B will just ignore the next received message. It is straightforward to show that the protocol is $(0,0)$-secure. Q.E.D.

5 Efficient $(0,0)$-secure message transmission in directed graphs

In the previous section, we proved a necessary and sufficient condition for $(0,0)$-secure message transmission from A to B. Our protocols in these proofs are not efficient (exponential in k). In this section, we show that if there are totally $3k+1$ paths between A and B, then there are efficient (linear in u) $(0,0)$-secure message transmission protocols from A to B.

Theorem 9. *Let $G(V,E)$ be a directed graph, $A, B \in V$ and $k \ge u$. If there are $n = 3k+1-u$ directed node disjoint paths $p_1, \ldots, p_n$ from A to B and u directed node disjoint paths $q_1, \ldots, q_u$ from B to A ($q_1, \ldots, q_u$ are node disjoint from $p_1, \ldots, p_n$) then there is an efficient $(0,0)$-secure message transmission protocol from A to B against a k-active adversary.*

Proof. If we replace the steps between Step 1 and Step 7 of the protocol π in the proof of Theorem 7 with the following steps:

Step 1 A constructs $(k+1)$-out-of-n MDS secret shares $(s_1^A, ..., s_n^A)$ of M^A. For each $i \le n$, A sends s_i^A to B via the path p_i.

Step 2 For each $i \le n$, B receives (or sets default) s_i^B on path p_i. If there are at most $k-u$ errors in the shares $(s_1^B, ..., s_n^B)$, then B recovers the secret message M^B from these shares by correcting the errors, sends "stop" to A via all paths q_i, and terminates the protocol. Otherwise, B sends "continue the protocol" to A via all paths q_i and goes to Step 8 of the protocol π.

Step 3 If A receives "stop" on all paths $q_1, \ldots, q_u$, then A terminates the protocol, otherwise, A goes to Step 8 of the protocol π.

Note that a $(k+1)$-out-of-n MDS secret sharing scheme could be used to detect k errors and simultaneously correct $k-u$ errors. Thus if all the paths $q_1, \ldots, q_u$ are controlled by the adversary, then B recovers the secret message M^B in Step 2. If at least one path from B to A is not controlled by the adversary, then the protocol π in the proof of Theorem 7 starting from Step 8 will let B to recover the secret message M^B. Here we should also note that the induction initiated in Step 10 or Step 14 of the protocol π works since $3k+1-u-1 = 3k-u > 3(k-1)+1-(u-1)$. It is straightforward that the protocol will terminates in at most $11u$ steps. Q.E.D.

In the previous theorems, including Theorem 9, we have the restriction that the directed paths from B to A are all node disjoint from the directed paths from A to B. In the following theorem we partially remove this restriction.

Theorem 10. *Let $G(V, E)$ be a directed graph, $A, B \in V$. Assume that there are $n = 3k + 1 - u \geq 2k + 1$ (which implies $k \geq u$) directed node disjoint paths $p_1, \ldots, p_n$ from A to B and u node disjoint directed paths $q_1, \ldots, q_u$ from B to A. If $3k + 1 - 2u$ paths among these $3k + 1 - u$ paths from A to B are node disjoint from the u paths from B to A, then there is an efficient $(0, 0)$-secure message transmission protocol from A to B against a k-active adversary.*

Proof. We note that the proof of Theorem 9 could not be used here since if we remove (in the induction step) two paths p_i and q_j such that one of them is corrupted, we are not guaranteed that the k-active adversary becomes a $(k-1)$-active adversary (q_j may share a node with some other directed paths from A to B and that node could be corrupted).

First we describe the proof informally. The protocol is divided into two phases. In phase one of the protocol, A tries to transmit the secret message to B assuming at least one of the directed paths from B to A is not corrupted. This is done by running u concurrent sub-protocols in phase one, in each sub-protocol B uses one of the directed paths from B to A to send some feedback information to A. In the second phase of the protocol, A transmits shares of the secret message through the A to B paths excluding these paths which have intersection with B to A paths. B will use the information received in the second phase only if B detects that all directed paths from B to A are corrupted in phase one.

In phase one, A and B execute the following protocol on the path set $\{p_i : 1 \leq i \leq n\} \cup \{q\}$ for each directed path q from B to A. First A chooses $R_0 \in_R \mathbf{F}$ and sends shares of R_0 to B via the paths $p_1, \ldots, p_n$ using a $(k+1)$-out-of-n MDS secret sharing scheme. If B can correct the errors in the received shares (that is, there were at most $k-u$ errors), B recovers R_0. Otherwise B needs help from A and B sends the received shares back to A via the B to A path q. The problems are that: B may receive help even if B has never asked for. However B can detect this. Therefore B always works with A on such a protocol and recovers the correct R_0. Then A sends $R_1 = M^A - R_0$ using a $(k+1)$-out-of-n MDS secret sharing scheme. If B can correct the errors in the received shares of R_1, B has found the secret and can terminate the protocol. If B cannot correct these errors, B needs to continue the protocol. In this situation, B distinguishes the following two cases:

1. B has not asked for help in the transmission of R_0. B can ask for help now and B will then recover the secret M^A.
2. B has asked for help in the transmission of R_0. In this case B cannot ask for help (otherwise the adversary may learn both the values of R_0 and R_1 and thus may recover the secret). The sub-protocol needs to be restarted (that is, A constructs different R_0 and R_1 for M^A and sends them to B again). Each time when A and B restart this sub-protocol, A sends the shares of R_0 and R_1 only via these "non-corrupted" paths from A to B. The "non-corrupted" paths are computed from the feedbacks that A has received from the path q. If q is not corrupted, then the computation is reliable. However, if q is corrupted, then the computation is unreliable. If there is at least one non-corrupted path q_{i_0} from B to A, then B recovers the secret from the sub-protocol running on the path set $\{p_1, \ldots, p_n\} \cup \{q_{i_0}\}$. Otherwise B cannot recover the secret in phase one and we will go to phase two.

If B asks for help in the transmission of R_0, then both A and B "identify" the corrupted paths from A and B according to the information that B sends to A via the path q. If k' dishonest paths from A to B have been (correctly or incorrectly) identified at the restart of the sub-protocol, A uses a $(k+1)$-out-of-$(3k+1-u-k')$ MDS secret sharing scheme. This MDS secret sharing scheme will only be used for error detection (or message recovery in the case that no error occurs), thus it can be used to detect $3k+1-u-k'-k-1 = 2k-u-k' \geq k-k'$ errors. Due to the fact that this MDS secret sharing scheme cannot detect k errors we need to organize ourselves that B will never use incorrectly identified paths from A to B since otherwise B could compute the incorrect "secret". This is easy to be addressed by having B detect whether the path q from B to A is dishonest or not. This is done by having A reliably send to B what A received via the path q. Since a $(k+1)$-out-of-$(3k+1-u)$ MDS secret sharing scheme can detect k errors and simultaneously correct $k-u$ errors, both A and B identify at least $k' \geq k-u+1$ dishonest paths from A to B in the first run of the sub-protocol. During each following run of the sub-protocol, B will either recover the secret message (when no error occurs) or detect at least one corrupted path from A to B (A could also detect the corrupted path from A to B according to the information A received on the path q). Thus the sub-protocol will be restarted at most u times.

In phase two of the protocol, A constructs $(k+1)$-out-of-$(3k+1-2u)$ MDS shares $(s_1, \ldots, s_{3k+1-2u})$ of the secret M^A and sends these shares to B via the $3k+1-2u$ paths which are node disjoint from the paths from the u paths from B

to A. Note that if B has determined that all these u paths from B to A have been corrupted in phase one, then B recovers the secret M^A from the received shares $(s_1^B, \ldots, s_{3k+1-2u}^B)$ in phase two since a $(k+1)$-out-of-$(3k+1-2u)$ MDS secret sharing scheme can be used to detect and correct $k-u$ errors simultaneously. Note that if at least one path from B to A is honest in phase one, then B has recovered the secret in phase and can just ignore this last message.

Now we present the entire protocol formally.

Step 1 B sets BA_BAD $= \emptyset$. For each directed path q from B to A, A and B run the sub-protocol between Step 2 and Step 11 (the sup-protocols for the u paths could be run parallely).

Step 2 A sets AB_CHANNEL$^A = \{p_1, \ldots, p_n\}$ and $j^A = 0$. B sets AB_CHANNEL$^B = \{p_1, \ldots, p_n\}$ and $j^B = 0$.

Step 3 Let $n_j = |$AB_CHANNEL$^A|$. A chooses $R_0 \in_R \mathbf{F}$, and constructs $(k+1)$-out-of-n_j MDS secret shares $\{s_i^A : p_i \in$ AB_CHANNEL$^A\}$ of R_0. For each $p_i \in$ AB_CHANNEL^A, A sends s_i^A to B via the path p_i.

Step 4 For each $p_i \in$ AB_CHANNEL^B, B receives s_i^B from A via the path p_i. B distinguishes the following two cases:

1. B can recover R_0. If $j = 0$ and there are at most $k-u$ errors, B recovers R_0 by correcting the errors (note that a $(k+1)$-out-of-n MDS scheme can be used to detect k errors and simultaneously correct $k-u$ errors). If $j > 0$, then B recovers R_0 only if there is no error in the received shares. B sends "ok" to A via the path q.
2. B cannot recover R_0. B sends $\{s_i^B : p_i \in$ AB_CHANNEL$^B\}$ to A via the path q.

Step 5 A distinguishes the following two cases:

1. A receives "ok" via the path q. A reliably sends "ok" to B.
2. A receives $\{\bar{s}_i^B : p_i \in$ AB_CHANNEL$^A\}$ (or sets default values if the received values are not in valid format). A sets BAD$^A = \{p_i : \bar{s}_i^B \neq s_i^A\}$ and reliably sends $\{\bar{s}_i^B : p_i \in$ AB_CHANNEL$^A\}$ and BAD^A to B. A sets AB_CHANNEL^A = AB_CHANNEL$^A \setminus$ BAD^A,

Step 6 B distinguishes the following two cases:

1. B reliably receives "ok" from A. If B sent "ok" to A in the Step 4, then goes to Step 7. Otherwise, B sets BA_BAD = BA_BAD$\cup\{q\}$ and goes to Step 11.
2. B reliably receives $\{\bar{s}_i^B : p_i \in$ AB_CHANNEL$^B\}$ and BAD^B from A. If $\bar{s}_i^B = s_i^B$ for all $p_i \in$ AB_CHANNEL^B, then B sets AB_CHANNEL^B = AB_CHANNEL$^B \setminus$ BAD^B, recovers R_0 from $\{s_i^B : p_i \in$ AB_CHANNEL$^B\}$, and goes to Step 7. Otherwise, B sets BA_BAD = BA_BAD $\cup \{q\}$ and goes to Step 11.

Step 7 Let $n_j = |$AB_CHANNEL$^A|$. A constructs $(k+1)$-out-of-n_j MDS secret shares $\{s_i^A : p_i \in$ AB_CHANNEL$^A\}$ of $R_1 = M^A - R_0$. For each $p_i \in$ AB_CHANNEL^A, A sends s_i^A to B via the path p_i.

Step 8 For each $p_i \in$ AB_CHANNEL^B, B receives s_i^B from A via the path p_i. B distinguishes the following two cases:

1. B can recover R_1. B recovers R_1 only if there is no error in the received shares. B sends "ok" to A via the path q.
2. B cannot recover R_1. For this situation we need to distinguish two cases:
   - 2.a) B sent "ok" to A in Step 4. That is, B has not asked for help to recover R_0. Then B can ask for help now. B sends $\{s_i^B : p_i \in \text{AB\_CHANNEL}^B\}$ to A via the path q.
   - 2.b) B sent the received shares to A in Step 4. That is, B has asked for help to recover R_0. Then B cannot ask for help now. B sends "continue to the next round" to A via the path q.

Step 9 A distinguishes the following three cases:

1. A receives "ok" via the path q. A reliably sends "ok" to B.
2. A receives "continue to the next round" via the path q. A sets $j^A = j^A + 1$, reliably sends "continue to the next round" to B, and goes to Step 3.
3. A receives $\{\bar{s}_i^B : p_i \in \text{AB\_CHANNEL}^A\}$ (or sets default values if the received values are in invalid format). A sets BAD$^A = \{p_i : \bar{s}_i^B \neq s_i^A\}$, AB_CHANNEL^A = AB_CHANNEL$^A \setminus$ BAD^A, and reliably sends $\{\bar{s}_i^B : p_i \in \text{AB\_CHANNEL}^A\}$ and BAD^A to B.

Step 10 B distinguishes the following three cases:

1. B reliably receives "ok" from A. If B sent "ok" to A in the Step 8, then B has recovered the secret. B terminates the entire protocol. Otherwise, B sets BA_BAD = BA_BAD $\cup \{q\}$ and goes to Step 11.
2. B reliably receives "continues to the next round". If B sent "continues to the next round" to A in the Step 8, then B sets $j^B = j^B{+}1$ and goes to Step 3. Otherwise, B sets BA_BAD = BA_BAD$\cup$ $\{q\}$ and goes to Step 11.
3. B reliably receives $\{\bar{s}_i^B : p_i \in \text{AB\_CHANNEL}^B\}$ and BAD^B from A. If $\bar{s}_i^B = s_i^B$ for all $p_i \in$ AB_CHANNEL^B, then B sets AB_CHANNEL^B = AB_CHANNEL$^B \setminus$ BAD^B, recovers R_1 from $\{s_i^B : p_i \in \text{AB\_CHANNEL}^B\}$, recovers the secret M^B from both R_0 and R_1, and terminates the entire protocol. Otherwise, B sets BA_BAD = BA_BAD $\cup \{q\}$ and goes to Step 11.

Step 11 B waits until all u sub-protocols in phase one finish. If |BA_BAD| = u then B goes to Step 12. Otherwise, B has recovered the secret message, thus terminates the entire protocol.

Step 12 A constructs $(k{+}1)$-out-of-$(3k{+}1{-}2u)$ MDS shares $(s_1, \ldots, s_{3k+1-2u})$ of the secret M^A and sends these shares to B via the $3k + 1 - 2u$ paths which are node disjoint from the u B to A paths. Note that if |BA_BAD| = u, then B can recover the secret message M^B from the received shares $(s_1^B, \ldots, s_{3k+1-2u}^B)$ since a $(k+1)$-out-of-$(3k+1-2u)$

MDS secret sharing scheme can be used to detect and correct $k - u$ errors simultaneously.

It is straightforward to show that at the beginning of each run of the sub-protocol between Step 2 and Step 11, Both A and B have the same sets of AB_CHANNEL, that is, AB_CHANNEL^A = AB_CHANNEL^B at Step 2. From the analysis before the above protocol, it is straightforward that the above protocol is a $(0, 0)$-secure message transmission protocol against a k-active adversary. Q.E.D.

6 Secure message transmissions in hypergraphs

Applications of hypergraphs in secure communications have been studied by Franklin and Yung in [8]. A hypergraph H is a pair (V, E) where V is the node set and E is the hyperedge set. Each hyperedge $e \in E$ is a pair (A, A^*) where $A \in V$ and A^* is a subset of V. In a hypergraph, we assume that any message sent by a node A will be received identically by all nodes in A^*, whether or not A is faulty, and all parties outside of A^* learn nothing about the content of the message.

Let $A, B \in V$ be two nodes of the hypergraph $H(V, E)$. We say that there is a "*direct link*" from node A to node B if there exists a hyperedge (A, A^*) such that $B \in A^*$. We say that there is an "*undirected link*" from A to B if there is a directed link from A to B or a directed link from B to A. If there is a directed (undirected) link from A_i to A_{i+1} for every i, $0 \le i < k$, then we say that there is a "*directed path*" ("*undirected path*") from A_0 to A_k. A and B are "*strongly k-connected*" ("*weakly k-connected*") in the hypergraph $H(V, E)$ if for all $S \subset V - \{A, B\}$, $|S| < k$, there remains a directed (undirected) path from A to B after the removal of S and all hyperedges (X, X^*) such that $S \cap (X^* \cup \{X\}) \neq \emptyset$. Franklin and Yung [8] showed that reliable and private communication from A to B is possible against a *k-passive* adversary if and only if A and B are strongly 1-connected and weakly $(k + 1)$-connected. It should be noted that B and A are strongly k-connected does not necessarily mean that A and B are strongly k-connected.

Following Franklin and Yung [8], and, Franklin and Wright [7], we consider multicast as our only communication primitive in this section. A message that is multicast by any node A in a hypergraph is received by all nodes A^* with privacy (that is, nodes not in A^* learn nothing about what was sent) and authentication (that is, nodes in A^* are guaranteed to receive the value that was multicast and to know which node multicast it). We assume that all nodes in the hypergraph know the complete protocol specification and the complete structure of the hypergraph.

Definition 2. *Let $H(V, E)$ be a hypergraph, $A, B \in V$ be distinct nodes of H, and $k \ge 0$. A, B are* k-separable *in H if there is a node set $W \subset V$ with at most k nodes such that any directed path from A to B goes through at least one node in W. We say that W* separates A, B.

Remark. Note that there is no straightforward relationship between strong connectivity and separability in hypergraphs.

Theorem 11. *Let $H(V,E)$ be a hypergraph, $A,B \in V$ be distinct nodes of H, and $k \geq 0$. The nodes A and B are not $2k$-separable if and only if there are $2k+1$ directed node disjoint paths from A to B in H.*

Proof. This follows directly from the maximum-flow minimum-cut theorem in classical graph theory. For details, see, e.g., [6]. Q.E.D.

Theorem 12. *Let $H(V,E)$ be a hypergraph, $A,B \in V$ be distinct nodes of H, and $k \geq 0$. A necessary and sufficient condition for reliable message transmission from A to B against a k-active adversary is that A and B are not $2k$-separable in H.*

Proof. First assume that A and B cannot be separated by a $2k$-node set. By Theorem 11, there are $2k+1$ directed node disjoint paths from A to B in H. Thus reliable message transmission from A to B is possible.

Next assume that A and B can be separated by a $2k$-node set W in H. We shall show that reliable message transmission is impossible. Suppose that π is a message transmission protocol from A to B and let $W = W_0 \cup W_1$ be a $2k$-node separation of A and B with W_0 and W_1 each having at most k nodes. Let m_0 be the message that A transmits. The adversary will attempt to maintain a simulation of the possible behavior of A by executing π for message $m_1 \neq m_0$. The strategy of the adversary is to flip a coin and then, depending on the outcome, decide which set of W_0 or W_1 to control. Let W_b be the chosen set. In each execution step of the transmission protocol, the adversary causes each node in W_b to follow the protocol π as if the protocol were transmitting the message m_1. This simulation succeeds with nonzero probability. Since B does not know whether $b=0$ or $b=1$, at the end of the protocol B cannot decide whether A has transmitted m_0 or m_1 if the adversary succeeds. Thus with nonzero probability, the reliability is not achieved. Q.E.D.

Theorem 12 gives a sufficient and necessary condition for achieving reliable message transmission against a k-active adversary over hypergraphs. In the following example, we show that this condition is not sufficient for achieving privacy against a k-active adversary (indeed, even not for a k-passive adversary).

Example 1 *Let $H(V,E_h)$ be the hypergraph in Figure 1 where $V = \{A, B, v_1, v_2, v, u_1, u_2\}$ and $E_h = \{(A,\{v_1,v_2\}), (v_1,\{v,B\}), (v_2,\{v,B\}), (A,\{u_1,u_2\}), (u_1,\{v,B\}), (u_2,\{v,B\})\}$. Then the nodes A and B are not 2-separable in H. Theorem 12 shows that reliable message transmission from A to B is possible against a 1-active adversary. However, the hypergraph H is not weakly 2-connected (the removal of the node v and the removal of the corresponding hyperedges will disconnect A and B). Thus, the result by Franklin and Yung [8] shows that private message transmission from A to B is not possible against a 1-passive adversary.*

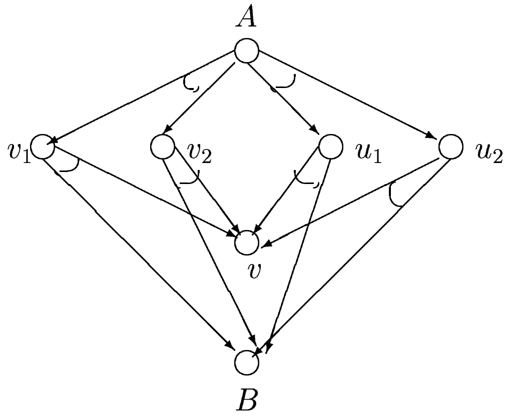

Fig. 1. The hypergraph $H(V, E_h)$ in Example 1

Theorem 13. *Let $\delta > 0$ and A and B be two nodes in a hypergraph $H(V, E)$ satisfying the following conditions:*

1. *A and B are not $2k$-separable in H.*
2. *B and A are not $2k$-separable in H.*
3. *A and B are strongly k-connected in H.*

Then there is a $(0, \delta)$-secure message transmission protocol from A to B against a k-active adversary.

Proof. Assume that the conditions of the theorem is satisfied. For each k-node subset set S of $V \setminus \{A, B\}$, let p_S be a directed path from A to B which witnesses that A and B are strongly k-connected by removing the nodes in S and corresponding hyperedges in H. Let $\mathcal{S} = \{S : S \subset V \setminus \{A, B\}, |S| = k\}$ and $\mathcal{P} = \{p_S : S \in \mathcal{S}\}$. Then A transmits the message M^A to B using the following protocol.

Step 1 For each $S \in \mathcal{S}$, A chooses a random pair $(a_S, b_S) \in_R \mathbf{F}^2$, and transmits this pair to B via the path p_S.

Step 2 For each $S \in \mathcal{S}$, B receives a pair (a_S^B, b_S^B) from A via the path p_S.

Step 3 For each $S \in \mathcal{S}$, B chooses a random $r_S \in_R \mathbf{F}$ and computes $s_S = \text{auth}(r_S; a_S^B, b_S^B)$.

Step 4 B reliably transmits $s = \langle\langle r_S, s_S\rangle : S \in \mathcal{S}\rangle$ to A.

Step 5 A reliably receives the value $s = \langle\langle r_S, s_S\rangle : S \in \mathcal{S}\rangle$ from B.

Step 6 A computes the key index set $K_{\text{index}} = \{i_S : s_S = \text{auth}(r_S; a_S^A, b_S^A)\}$ and the shared secret $K^A = \sum_{i_S \in K_{\text{index}}} a_S^A$.

Step 7 A reliably transmits $\langle K_{\text{index}}, M^A + K^A\rangle$ to B, where M^A is the secret message.

Step 8 B reliably receives the value $\langle K_{\text{index}}, c\rangle$ from A. B computes the shared secret $K^B = \sum_{i_S \in K_{\text{index}}} a_S^B$, and decrypts the message $M^B = c - K^B$.

It is possible that $a_S^A \neq a_S^B$ but $\text{auth}(r_S; a_S^A, b_S^A) = \text{auth}(r_S; a_S^B, b_S^B)$ for some $S \in \mathcal{S}$. However this probability is negligible. Thus the above protocol is reliable with high probability. Since A and B are strongly k-connected in H, there is a pair (a_S, b_S) such that (a_S, b_S) reliably reaches B and the adversary cannot infer any information of a_S from its view. Thus the above protocol is $(0, \delta)$-secure against a k-active adversary if one chooses sufficiently large $\mathbf{F}$. Q.E.D.

The results in Sections 3 and 4 show that the condition in Theorem 13 is not necessary.

7 Secure message transmission over neighbor networks

7.1 Definitions

A special case of the hypergraph is the *neighbor networks.* A neighbor network is a graph $G(V, E)$. In a neighbor network, a node $A \in V$ is called a neighbor of another node $B \in V$ if there is an edge $(A, B) \in E$. In a neighbor network, we assume that any message sent by a node A will be received identically by all its neighbors, whether or not A is faulty, and all parties outside of A's neighbor learn nothing about the content of the message.

For a neighbor network $G(V, E)$ and two nodes A, B in it, Franklin and Wright [7], and, Wang and Desmedt [17] showed that if there are n multicast lines (that is, n paths with disjoint neighborhoods) between A and B and there are at most k malicious (Byzantine style) processors, then the condition $n > k$ is necessary and sufficient for achieving efficient probabilistically reliable and perfect private communication.

For each neighbor network $G(V, E)$, there is a hypergraph $H_G(V, E_h)$ which is equivalent to $G(V, E)$ in functionality. $H_G(V, E_h)$ is defined by letting E_h be the set of hyperedges (A, A^*) where $A \in V$ and A^* is the set of neighbors of A.

Let A and B be two nodes in a neighbor network $G(V, E)$. We have the following definitions:

1. A and B are *k-connected* in $G(V, E)$ if there are k node disjoint paths between A and B in $G(V, E)$.
2. A and B are *weakly k-hyper-connected* in $G(V, E)$ if A and B are weakly k-connected in $H_G(V, E_h)$.
3. A and B are *k-neighbor-connected* in $G(V, E)$ if for any set $V_1 \subseteq V \setminus \{A, B\}$ with $|V_1| < k$, the removal of $neighbor(V_1)$ and all incident edges from $G(V, E)$ does not disconnect A and B, where

   $$neighbor(V_1) = V_1 \cup \{A \in V : \text{thereexists} B \in V_1 (B, A) \text{ such that } \in E\} \setminus \{A, B\}.$$

4. A and B are *weakly (n, k)-connected* if there are n node disjoint paths $p_1, \ldots,$ p_n between A and B and, for any node set $T \subseteq (V \setminus \{A, B\})$ with $|T| \leq k$, there exists $1 \leq i \leq n$ such that all nodes on p_i have no neighbor in T.

It is easy to check that the following relationships hold.

weak $(n, k-1)$-connectivity $(n \geq k) \Rightarrow k$-neighbor-connectivity $\Rightarrow$ weak k-hyper-connectivity $\Rightarrow k$-connectivity

In the following examples, we show that these implications are strict.

Example 2 *Let $G(V,E)$ be the graph in Figure 2 where $V = \{A, B, C, D\}$ and $E = \{(A,C), (C,B), (A,D), (D,B), (C,D)\}$. Then it is straightforward to check that $G(V,E)$ is 2-connected but not weakly 2-hyper-connected.*

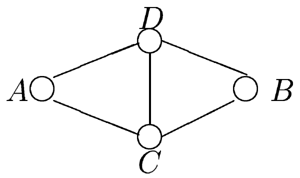

Fig. 2. The graph $G(V,E)$ in Example 2

Example 3 *Let $G(V,E)$ be the graph in Figure 3 where $V = \{A, B, C, D, F\}$ and $E = \{(A,C),\ (A,D),\ (C,B),\ (D,B),\ (C,F),\ (F,D)\}$. Then it is straightforward to check that A and B are weakly* 2*-hyper-connected but not* 2*-neighbor-connected.*

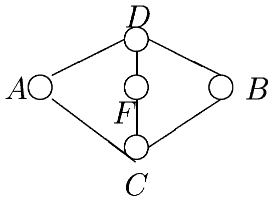

Fig. 3. The graph $G(V,E)$ in Example 3

Example 4 *Let $G(V,E)$ be the graph in Figure 4 where $V = \{A,\ B,\ C,\ D,\ E,\ F,\ G,\ H\}$ and $E = \{(A,C),\ (C,D),\ (D,E)\ (E,B),\ (A,F),\ (F,G),\ (G,H)\ (H,B),\ (C,H),\ (E,F)\}$. Then it is straightforward to check that A and B are* 2*-neighbor-connected but not weakly* $(2,1)$*-connected.*

Example 2 shows that k-connectivity does not necessarily imply weak k-hyper-connectivity. Example 3 shows that weak k-hyper-connectivity does not necessarily imply k-neighbor-connectivity. Example 4 shows that k-neighbor connectivity does not necessarily imply weak $(n, k-1)$-connectivity for some $n \geq k$.

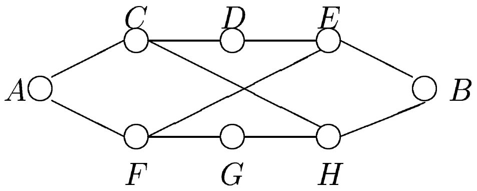

Fig. 4. The graph $G(V, E)$ in Example 4

7.2 $(0, \delta)$-Secure message transmission over neighbor networks

Wang and Desmedt [17] have given a sufficient condition for achieving $(0, \delta)$-security message transmission against a k-active adversary over neighbor networks. In this section, we show that their condition is not necessary.

Theorem 14. *(Wang and Desmedt [17]) If A and B are weakly (n, k)-connected for some $k < n$, then there is an efficient $(0, \delta)$-secure message transmission between A and B.*

The condition in Theorem 14 is not necessary. For example, the neighbor network G in Example 3 is not 2-neighbor-connected, thus not weakly $(2, 1)$-connected. In the following we present a $(0, \delta)$-secure message transmission protocol against a 1-active adversary from A to B for the neighbor network of Example 3 .

Message transmission protocol for neighbor network G in Example 3.

Step 1 A chooses two random pairs $(r_1^A, r_2^A) \in_R \mathbf{F}^2$ and $(r_3^A, r_4^A) \in_R \mathbf{F}^2$. A sends (r_1^A, r_2^A) to C and (r_3^A, r_4^A) to D.

Step 2 B chooses two random pairs $(r_1^B, r_2^B) \in_R \mathbf{F}^2$ and $(r_3^B, r_4^B) \in_R \mathbf{F}^2$. B sends (r_1^B, r_2^B) to C and (r_3^B, r_4^B) to D.

Step 3 C chooses a random pair $(a_1, b_1) \in_R \mathbf{F}^2$. C sends $(a_1 + r_1^A, b_1 + r_2^A)$ to A and $(a_1 + r_1^B, b_1 + r_2^B)$ to B.

Step 4 D chooses a random pair $(a_2, b_2) \in_R \mathbf{F}^2$. D sends $(a_2 + r_3^A, b_2 + r_4^A)$ to A and $(a_2 + r_3^B, b_2 + r_4^B)$ to B.

Step 5 From the messages received from C and D, A computes (a_1^A, b_1^A) and (a_2^A, b_2^A).

Step 6 From the messages received from C and D, B computes (a_1^B, b_1^B) and (a_2^B, b_2^B).

Step 7 B chooses a random $r \in_R \mathbf{F}$, computes $s_1 = \text{auth}(r; a_1^B, b_1^B)$ and $s_2 = \text{auth}(r; a_2^B, b_2^B)$. Using the probabilistically reliable message transmission protocol of Franklin and Wright [7], B transmits $\langle r, s_1, s_2 \rangle$ to A.

Step 8 Let $\langle r^A, s_1^A, s_2^A \rangle$ be the message received by A in the last step, A computes the key index set $K_{\text{index}} = \{i : s_i^A = \text{auth}(r^A; a_i^A, b_i^A)\}$. A also computes the shared secret $K^A = \sum_{i \in K_{\text{index}}} a_i^A$.

Step 9 Using the probabilistically reliable message transmission protocol of Franklin and Wright [7], A transmits $\langle K_{\text{index}}, M^A + K^A \rangle$ to B, where M^A is the secret message.

Step 10 Let $\langle K^B_{\text{index}}, c^B \rangle$ be the message that B received in the last step. B computes the shared secret $K^B = \sum_{i \in K^B_{\text{index}}} a_i^B$, and decrypts the message $M^B = c^B - K^B$.

It is straightforward to check that the above protocol is an efficient $(0, \delta)$-secure message transmission protocol from A to B against a 1-active adversary.

Example 1 shows that for a general hypergraph, the existence of a reliable message transmission protocol does not imply the existence of a private message transmission protocol. We show that this is true for probabilistic reliability and perfect privacy in neighbor networks also.

Example 5 *Let $G(V, E)$ be the neighbor network in Figure 5 where $V = \{A, B, C, D, E, F, G\}$ and $E = \{(A, C), (C, D), (D, B), (A, E), (E, F), (F, B), (G, C), (G, D), (G, E), (G, F)\}$. Then there is a probabilistic reliable message transmission protocol from A to B against a 1-active adversary in G. But there is no private message transmission from A to B against a 1-passive (or 1-active) adversary in G.*

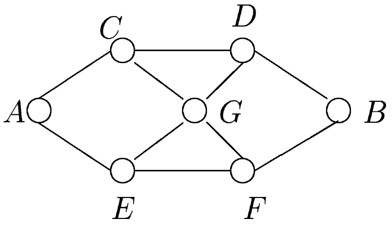

Fig. 5. The graph $G(V, E)$ in Example 5

Proof. It is straightforward to check that $G(V, E)$ is not weakly 2-hyper-connected. Indeed, in the hypergraph $H_G(V, E_h)$ of $G(V, E)$, the removal of node G and the removal of the corresponding hyperedges will disconnect A and B completely. Thus Franklin and Yung's result in [8] shows that there is no private message transmission protocol against a 1-passive (or 1-active) adversary from A to B. It is also straightforward to check that Franklin and Wright's [7] reliable message transmission protocol against a 1-active adversary works for the two paths (A, C, D, B) and (A, E, F, B). Q.E.D.

Though weak k-hyper-connectivity is a necessary condition for achieving probabilistically reliable and perfectly private message transmission against a $(k-1)$-active adversary, we do not know whether this condition is sufficient. We conjecture that there is no probabilistically reliable and perfectly private message transmission protocol against a 1-active adversary for the weakly 2-hyper-connected neighbor network $G(V, E)$ in Figure 6, where $V = \{A, B, C, D, E, F, G, H\}$ and $E = \{(A,C), (C,D), (D,E), (E,B), (A,F), (F,G), (G,H), (H,B), (D,G)\}$. Note that in order to prove or refute our conjecture, it is sufficient to show whether there is a probabilistically reliable message transmission protocol against a 1-active adversary for the neighbor network. For this specific neighbor network, the trick in our previous protocol could be used to convert any probabilistically reliable message transmission protocol to a probabilistically reliable and perfectly private message transmission protocol against a 1-active adversary.

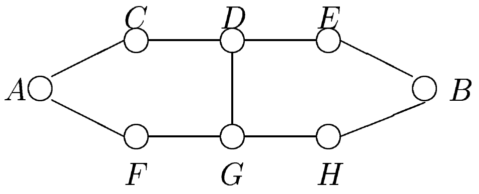

Fig. 6. The graph $G(V, E)$

References

1. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computing. In: *Proc. ACM STOC, '88*, pages 1–10, ACM Press, 1988.
2. D. Chaum, C. Crepeau, and I. Damgard. Multiparty unconditional secure protocols. In: *Proc. ACM STOC '88*, pages 11–19, ACM Press, 1988.
3. Y. Desmedt and Y. Wang. Perfectly secure message transmission revisited. In: *Proc. Eurocrypt '02*, pages 502–517, Lecture Notes in Computer Science 2332, Springer-Verlag, 2002.
4. D. Dolev. The Byzantine generals strike again. *J. of Algorithms*, **3**:14–30, 1982.
5. D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *J. of the ACM*, **40**(1):17–47, 1993.
6. L.R. Ford and D. R. Fulkerson. *Flows in Networks.* Princeton University Press, Princeton, NJ, 1962.
7. M. Franklin and R. Wright. Secure communication in minimal connectivity models. *Journal of Cryptology*, **13**(1):9–30, 2000.
8. M. Franklin and M. Yung. Secure hypergraphs: privacy from partial broadcast. In: *Proc. ACM STOC '95*, pages 36–44, ACM Press, 1995.
9. E. Gilbert, F. MacWilliams, and N. Sloane. Codes which detect deception. *The BELL System Technical Journal*, **53**(3):405–424, 1974.
10. O. Goldreich, S. Goldwasser, and N. Linial. Fault-tolerant computation in the full information model. *SIAM J. Comput.* **27**(2):506–544, 1998.

11. V. Hadzilacos. *Issues of Fault Tolerance in Concurrent Computations.* PhD thesis, Harvard University, Cambridge, MA, 1984.
12. F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes.* North-Holland Publishing Company, 1978.
13. R. J. McEliece and D. V. Sarwate. On sharing secrets and Reed-Solomon codes. *Comm. ACM*, **24**(9):583–584, September 1981.
14. T. Rabin. Robust sharing of secrets when the dealer is honest or faulty. *J. of the ACM*, **41**(6):1089–1109, 1994.
15. T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In: *Proc. ACM STOC '89*, pages 73–85, ACM Press, 1989.
16. A. Shamir. How to share a secret. *Commun. ACM*, **22**:612–613, November 1979.
17. Y. Wang and Y. Desmedt. Secure communication in multicast channels: the answer to Franklin and Wright's question. *J. of Cryptology*, **14**(2):121–135, 2001.