Perfectly Secure Message Transmission Revisited

(Extended Abstract)

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] Karthika Technologies Inc., ywang@karthika.com

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 [4] 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., [1–4, 10]). For example, Dolev [3] and Dolev et al. [4] 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

* Part of the work was done when this author was with Certicom Corp.

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 [4], 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 u to v is cheap and a channel from v to u is expensive but not impossible. Another example is that u has access to more resources than v does.

Goldreich, Goldwasser, and Linial [9], Franklin and Yung [7], Franklin and Wright [6], and Wang and Desmedt [15] 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 [7] 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 [6], and, Wang and Desmedt [15]. Note that Goldreich, Goldwasser, and Linial [9] 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. [9] 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. [6]), 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 5 is devoted to reliable message transmission over hypergraphs, and Section 6 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 $u, v \in V$, Throughout this paper, n denotes the number of vertex disjoint paths between 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 [6, 8, 13, 14]) and $\mathrm{auth}(M, a, b, c) := aM^2 + bM + c$ (following [15]). 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 the 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. 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: $\mathbf{F} \to \mathbf{F}^n$ with the property that for any $m \in \mathbf{F}$ and $(v_1, \ldots, v_n) = \mathrm{S}(m)$, 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 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 $n - k - 1$ [11]. We say that the $(k + 1)$-out-of-n secret sharing scheme can correct k' errors if from any $(v_1, \ldots, v_n) = \mathrm{S}(m)$ and any tuple $(u_1, \ldots, u_n)$ over 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 $(n - k - 1)/2$ [11]. 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 [11, p. 10]). Maximal Distance Separable (MDS) secret sharing schemes can be constructed from any MDS codes, for example, from Reed-Solomon code [12].

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 [6])*

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 [4] 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 $2(k - u) + 1$ directed node disjoint paths from A to B, where $1 \leq u \leq k$. We wonder how many 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 [6] showed that if there is no channel from B to A, then $2k+1$ channels from A to B is necessary for $(1-\delta)$-reliable (assuming that $\delta < \frac{1}{2}$) message transmission from A to B against a k-active adversary. In the following, we first show that this condition is sufficient also.

Theorem 1. *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. (Sketch) The necessity was proved in Franklin and Wright [6]. Let p_1, $\ldots, p_{2k+1}$ be the $2k+1$ directed node disjoint paths from A to B. Let $s^A \in \mathbf{F}$ be the secret that A wants to send to B. A constructs $(k+1)$-out-of-$(2k+1)$ secret shares $v = (s_1^A, \ldots, s_{2k+1}^A)$ of s^A. The protocol proceeds from round 1 through round $2k+1$. In round i, A chooses $\{(a_{i,j}^A, b_{i,j}^A) \in_R \mathbf{F}^2 : 1 \leq j \leq 2k+1\}$, sends $(s_i^A, \mathrm{auth}(s_i^A, a_{i,1}^A, b_{i,1}^A), \ldots, \mathrm{auth}(s_i^A, a_{i,2k+1}^A, b_{i,2k+1}^A))$ to B via path p_i, and sends $(a_{i,j}^A, b_{i,j}^A)$ to B via path p_j for each $1 \leq j \leq 2k+1$. In round i, B receives $(s_i^B, c_1^B, \ldots, c_{2k+1}^B)$ via path p_i, and receives $(a_{i,j}^B, b_{i,j}^B)$ via path j for each $1 \leq j \leq 2k+1$. B computes $t = |\{j : c_j^B = \mathrm{auth}(s_i^B, a_{i,j}^B, b_{i,j}^B)\}|$. If $t \geq k+1$, then B decides that s_j^B is a valid share. Otherwise B discards s_j^B. It is easy to check that after the round $2k+1$, with high probability, B will get at least $k+1$ valid shares to s^A. Thus, with high probability, B will recover the secret s^A. In the full version of this paper, we will show that this protocol is a $(0,\delta)$-secure message transmission protocol from A to B. Q.E.D.

Theorem 2. *Let $G(V,E)$ be a directed graph, $A, B \in V$, and $u \geq 1$. If there are $2(k-u)+1 \geq k+1$ directed node disjoint paths from A to B, then a necessary condition for private message transmission from A to B against a k-active adversary is that there are u directed node disjoint paths (these u paths are also node disjoint from the $2(k-u)+1$ paths from A to B) from B to A.*

Proof. First assume that there are less than u directed node disjoint paths from B to A. A strategy that will now be used by the adversary is that controlling $u-1$ nodes to disconnect all directed paths from B to A and controlling $k-u+1$ directed paths from A to B. Thus the adversary could make sure that no feedback message will be sent from B to A. This means that we are left with the same situation as Theorem 1 using $2(k-u)+1$ one-way channels. Since $k-u+1$ of these paths are controlled by the adversary, by Theorem 1, we need $2(k-u+1)+1 > 2(k-u)+1$ directed paths from A to B. This is a contradiction.

Secondly we assume that there are u directed node disjoint paths q_i from B to A, $2(k-u)+1$ paths p_i from A to B, and that p_1 is not node disjoint from q_1. A strategy that will now be used by the adversary is that using one node to control both paths p_1 and q_1, using other $u-1$ nodes to disconnect all directed paths from B to A, controlling $k-u$ other directed paths from A to B. A similar argument as above will show a contradiction. Q.E.D.

In the following we prove a simple sufficient condition.

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_1 and p_2) 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. (Sketch) Let $s^A \in \mathbf{F}$ be the secret message that A wants to send to B. In the following we describe the protocol briefly without proof. The details and a generalization will be given in the full version of this paper.

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 $s_1^A = s^A - s_0^A$. 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 $c_i^B = \mathrm{auth}(s_i^B, a_{1-i}^B, b_{1-i}^B)$. 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 terminate 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 recover B's authentication key (a^A, b^A). A sends $(s^A, \mathrm{auth}(s^A, a^A, b^A))$ to B via the uncorrupted path from A to B.

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

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 reliable message transmission in directed graphs. We first start with necessary conditions.

Theorem 4. *Replacing in Theorem 2: $2(k-u)+1$ by $3(k-u)+1$, provides necessary conditions for $(0,0)$-secure message transmission from A to B.*

Proof. Use an argument as in the proof of Theorem 2, but use the $3k+1$ bound from [4] instead of the $2k+1$ one. Q.E.D.

We will show that if there are $3k+1-u$ paths from A to B and u paths from B to A, then $(0,0)$-secure message transmission from A to B is possible. We first show the simple case for $u = 1$.

Theorem 5. *Let $G(V,E)$ be a directed graph, $A,B \in V$. If there are $3k \geq 2k+1$ directed node disjoint paths from A to B and one directed path from B to A (the* *directed path from B to A is node disjoint from the paths from A to B) then there is a $(0,0)$-secure message transmission protocol from A to B against a k-active adversary.*

Proof. Let $p_1, \ldots, p_{3k}$ be the directed paths from A to B and q be the directed path from B to A. The protocol π proceeds as follows:

Step 1 B sets A_STOP $= 0$ and B_STOP $= 0$.

Step 2 A chooses a $key^A \in_R \mathbf{F}$ and constructs $(k+1)$-out-of-$3k$ MDS secret shares $v = (s_1^A, ..., s_{3k}^A)$ of key^A. For each $1 \leq i \leq 3k$, A sends s_i to B via the path p_i.

Step 3 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 key^B from the shares, sends "stop" to A via the path q, and sets B_STOP $= 1$. 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 4 A distinguishes the following two cases:
1. A receives $v^A = (s_1^A, ..., s_{3k}^A)$ from the path q. A reliably sends $\mathcal{P} = \{i : s_i^A \neq s_i\}$ to B.
2. A received "stop" or anything else via q. A reliably sends "stop" to B.

Step 5 B distinguishes the following two cases:
1. B reliably receives "stop" from A. B sets A_STOP $= 1$.
2. B reliably receives $\mathcal{P}$ from A. If B_STOP $= 0$ then B recovers key^B from the shares $\{s_i^B : i \notin \mathcal{P}\}$ (note that $|\{s_i^B : i \notin \mathcal{P}\}| = 2k$).

Step 6 A reliably transmits $key^A + m^A$ to B, where m^A is the message to be transmitted.

Step 7 B reliably receives the ciphertext c^B and decrypts the message $m^B = c^B - key^B$.

Note that if B sends v^B to A in Step 3 then k paths from A to B are corrupted and the path q is honest. Thus the adversary will not learn v^B and key. 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.

Before proving our main theorem, we describe a variant π' of the protocol π in the proof of Theorem 5. We call B_STOP during the i-th execution of π B_STOP(i) and similar for A_STOP(i). The new protocol π' proceeds as follows:

Step 1 Instead of sending the secret key^A, A first sends $R_1 \in_R \mathbf{F}$ using π.

Step 2 A, B execute Steps 1 and 2 of π for the message R_2 where $R_1 + R_2 = key^A$.

Step 3 If B_STOP(2) = 1 (B_STOP(1) = 1 or 0), then B computes the secret key^B.

Step 4 If B_STOP(1) = 1 and B_STOP(2) = 0, then B and A continue with the rest of π for R_2, and B will be able to compute the secret key^B.

Step 5 If B_STOP(1) = 0 and B_STOP(2) = 0 then A_STOP(2) = 0. In this case, k corrupted paths should have already been identified by both A and B in the second run of π (though A does not know whether it has correctly identified the corrupted paths). A "restarts" the protocol by sending key^A using a $(k+1)$-out-of-$2k$ secret sharing scheme along the $2k$ non-corrupted paths. B excludes the known k bad paths and computes the secret from the secret sharing scheme.

Note that due to the malicious information A received, A may restart the protocol even though B may have already computed the correct secret. In this case, B can just ignore these messages.

Theorem 6. *Let $G(V, E)$ be a directed graph, $A, B \in V$. If there are $3k+1-u \geq 2k+1$ (which implies $k \geq u$) directed node disjoint paths from A to B and u directed paths from B to A (the directed paths from B to A are node disjoint from the paths from A to B) then there is a $(0,0)$-secure message transmission protocol from A to B against a k-active adversary.*

Proof. Let $p_1, \ldots, p_{3k+1-u}$ be the directed paths from A to B, and $q_1, \ldots, q_u$ be the directed paths from B to A. The protocol will be based on the variant protocol π' of Theorem 5. Before we begin, we note that a $(k+1)$-out-of-$(3k+1-u)$ MDS secret sharing scheme can detect k errors and simultaneously correct $k-u$ errors. In the following, we informally describe the protocol. The full protocol will be presented in the full version of this paper.

Step 1 A chooses $R_0 \in_R \mathbf{F}$ and sends R_0 to B via the $3k+1-u$ paths using a $(k+1)$-out-of-$(3k+1-u)$ MDS secret sharing scheme.

Step 2 If B can correct the errors (i.e. there were at most $k-u$ errors in the received shares), B finds R_0. Otherwise B needs help from A (that is, B will send the received shares back to A via all B to A paths). The problems are that:
- B may receive help even if B has never asked. However B can detect this. Therefore B will always work with A on such a protocol.
- A may receive u different versions of "asking for help".

For each of the u paths from B to A, B and A will keep track of the "dishonest" paths from A to B according to the information A received on this path.

Step 3 A now sends R_1 using a $(k+1)$-out-of-$(3k+1-u)$ MDS secret sharing scheme where $key = R_0 + R_1$.

Step 4 If B can correct the errors, B has found the secret. However, B may need to play with A prolonging the protocol due to incorrect paths from B to A. B distinguishes the following two cases:

1. B has not asked help in Step 2. B can ask help now and B will then recover the secret key.
2. B has asked help in Step 2. In this case B cannot ask for help again (otherwise the enemy may learn the secret). The protocol needs to be restarted from Step 1 on. We know that in this case there is at least one honest path from B to A. (Indeed, if B asked for help in Step 2, then the number of dishonest paths from A to B is at least $k' \geq k - u + 1$. Assume that all paths from B to A were dishonest then the total number of dishonest parties is $k' + u \geq k + 1$, which is a contradiction.) Since A and B identified (correctly or incorrectly) dishonest parties on the paths from A to B (the version corresponding to the honest B to A path should have correctly identified the dishonest paths), they will only use these paths that were not identified as dishonest. If k' dishonest paths from A to B have been (correctly or incorrectly) identified, a $(k{+}1)$-out-of-$(3k{+}1{-}u{-}k')$ MDS secret sharing scheme will be used. 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 a path from B to A is dishonest or not. This is done by having A reliably sends to B what A received via the path q_i from B to A. During each run of the protocol, B will either recover the secret message (when no error occurs) or detect 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 honest B to A path—though A may not know which path from B to A is honest). Thus the protocol will be restarted at most u times.

After the initial run, B will first use the path q_1 to send the "asking for help" message. Then it will use the path q_2, and then q_3, etc. These steps can be run in parallel. Q.E.D.

Theorem 6 can be strengthened as follows.

Theorem 7. *Let $G(V, E)$ be a directed graph, $A, B \in V$. Assume that there are $3k+1-u \geq 2k+1$ (which implies $k \geq u$) directed node disjoint paths from A to B and u directed paths 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 a $(0,0)$-secure message transmission protocol from A to B against a k-active adversary.*

Proof. The protocol proceeds in the same way as the protocol in the proof of Theorem 6. In addition, at the end of the protocol, A constructs a $(k+1)$-out-of-$(3k+1-2u)$ MDS shares $(s_1, \ldots, s_{3k+1-2u})$ of the secret key^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. If B has determined that all these u paths from B to A have been corrupted, then B will recover the secret key^A 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. Note that if at least one path from B to A is honest, then B has recovered the secret already and can just ignore this last message. Q.E.D.

We close our discussion on secure message transmission in directed graphs with an application of Theorem 7. Up to now, we have concentrated on the situation that there are more paths from A to B than paths from B to A. The following theorem address the situation that there are more paths from B to A.

Theorem 8. *Let $G(V,E)$ be a directed graph, $A, B \in V$. Assume that there are $k+1$ directed node disjoint paths from A to B and $2k+1$ directed node disjoint paths from B to A. If $k+1$ paths among these $2k+1$ paths from B to A are node disjoint from the $k+1$ paths from A to B, then there is a $(0,\delta)$-secure message transmission protocol from A to B against a k-active adversary.*

Proof. See the full version of this paper. Q.E.D

5 Secure message transmissions in hypergraphs

Hypergraphs have been studied by Franklin and Yung in [7]. 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 (v, v^*) where $v \in V$ and v^* is a subset of V. In a hypergraph, we assume that any message sent by a node v will be received identically by all nodes in v^*, whether or not v is faulty, and all parties outside of v^* learn nothing about the content of the message.

Let $v, u \in V$ be two nodes of the hypergraph $H(V,E)$. We say that there is a "*direct link*" from node v to node u if there exists a hyperedge (v, v^*) such that $u \in v^*$. We say that there is an "*undirected link*" from v to u if there is a directed link from v to u or a directed link from u to v. If there is a directed (undirected) link from v_i to v_{i+1} for every i, $0 \leq i < k$, then we say that there is a "*directed path*" ("*undirected path*") from v_0 to v_k. v and u are "*strongly k-connected*" ("*weakly k-connected*") in the hypergraph $H(V,E)$ if for all $S \subset V - \{v,u\}$, $|S| < k$, there remains a directed (undirected) path from v to u after the removal of S and all hyperedges (x, x^*) such that $S \cap (x^* \cup \{x\}) \neq \emptyset$. Franklin and Yung [7] showed that reliable and private communication from v to u is possible against a k-*passive* adversary if and only if v and u are strongly 1-connected and weakly $k+1$-connected. It should be noted that u and v are strongly k-connected does not necessarily mean that v and u are strongly k-connected.

Following Franklin and Yung [7], and, Franklin and Wright [6], we consider multicast as our only communication primitive in this section. A message that is multicast by any node v in a hypergraph is received by all nodes v^* with privacy (that is, nodes not in v^* learn nothing about what was sent) and authentication (that is, nodes in v^* 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 $k \geq 0$. 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 9. *The nodes A, B of a hypergraph H is 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., [5]. Q.E.D.

Theorem 10. *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 9, 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, 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 will 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 10 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 10 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 [7] shows that private message transmission from A to B is not possible against a 1-passive adversary.*

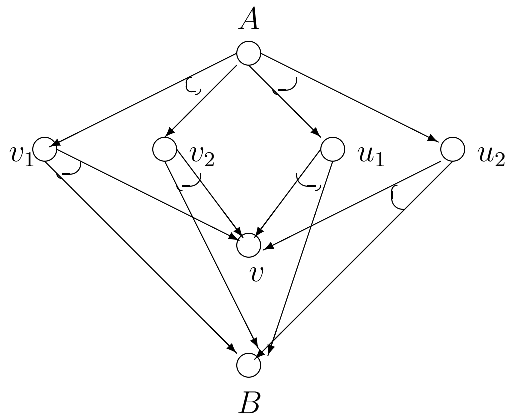

Fig. 1. The hypergraph $H(V, E_h)$ in Example 1

Theorem 11. *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. See the full version of this paper. Q.E.D.

The results in Sections 3 and 4 show that the condition in Theorem 11 is not necessary.

6 Secure message transmission over neighbor networks

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

For a neighbor network $G(V, E)$ and two nodes v, u in it, Franklin and Wright [6], and, Wang and Desmedt [15] showed that if there are n multicast lines (that is, n paths with disjoint neighborhoods) between v and u 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 function. $H_G(V, E_h)$ is defined by letting E_h be the set of hyperedges (v, v^*) where $v \in V$ and v^* is the set of neighbors of v.

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

1. v and u are *k-connected* in $G(V, E)$ if there are k node disjoint paths between v and u in $G(V, E)$.
2. v and u are *weakly k-hyper-connected* in $G(V, E)$ if v and u are weakly k-connected in $H_G(V, E_h)$.
3. v and u are *k-neighbor-connected* in $G(V, E)$ if for any set $V_1 \subseteq V \setminus \{v, u\}$ with $|V_1| < k$, the removal of *neighbor*(V_1) and all incident edges from $G(V, E)$ does not disconnect v and u, where *neighbor*$(V_1) = V_1 \cup \{v \in V : \exists u \in V_1 (u, v)$ such that $\in E\} \setminus \{v, u\}$.
4. v and u are *weakly (n, k)-connected* if there are n node disjoint paths $p_1, \ldots,$ p_n between v and u and, for any node set $T \subseteq (V \setminus \{v, u\})$ with $|T| \leq k$, there exists an i $(1 \leq i \leq n)$ such that all nodes on p_i have no neighbor in T.

It is easy to check that the following relations hold.

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

In the following examples, we will 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.*

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.*

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.*

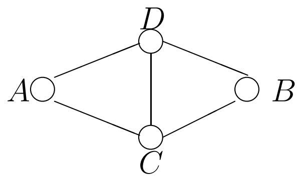

Fig. 2. The graph $G(V, E)$ in Example 2

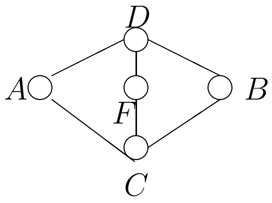

Fig. 3. The graph $G(V, E)$ in Example 3

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 $n \geq k$.

6.2 $(0, \delta)$-Secure message transmission over neighbor networks

Wang and Desmedt [15] have given a sufficient condition for achieving $(0, \delta)$-security message transmission against a k-active adversary over neighbor networks. In this section, we first show that their condition is not necessary.

Theorem 12. *(Wang and Desmedt [15]) 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 12 is not necessary. For example, the neighbor network G in Example 3 is not 2-neighbor-connected, thus not weakly $(2, 1)$-connected. In

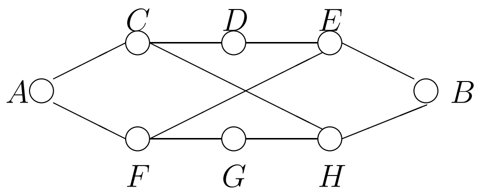

Fig. 4. The graph $G(V, E)$ in Example 4

the full version of this paper, we will present a $(0, \delta)$-secure message transmission protocol against a 1-active adversary from A to B.

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.*

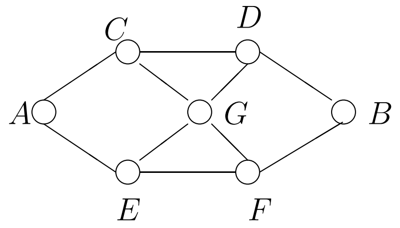

Fig. 5. The graph $G(V, E)$ in Example 5

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 [7] 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 [6] 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.

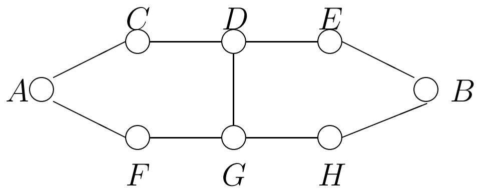

Fig. 6. The graph $G(V, E)$

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. D. Dolev. The Byzantine generals strike again. *J. of Algorithms*, **3**:14–30, 1982.
4. D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *J. of the ACM*, **40**(1):17–47, 1993.
5. L.R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, 1962.
6. M. Franklin and R. Wright. Secure communication in minimal connectivity models. *Journal of Cryptology*, **13**(1):9–30, 2000.
7. M. Franklin and M. Yung. Secure hypergraphs: privacy from partial broadcast. In: *Proc. ACM STOC '95*, pages 36–44, ACM Press, 1995.
8. E. Gilbert, F. MacWilliams, and N. Sloane. Codes which detect deception. *The BELL System Technical Journal*, **53**(3):405–424, 1974.
9. O. Goldreich, S. Goldwasser, and N. Linial. Fault-tolerant computation in the full information model. *SIAM J. Comput.* **27**(2):506–544, 1998.
10. V. Hadzilacos. *Issues of Fault Tolerance in Concurrent Computations*. PhD thesis, Harvard University, Cambridge, MA, 1984.
11. F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*. North-Holland Publishing Company, 1978.
12. R. J. McEliece and D. V. Sarwate. On sharing secrets and Reed-Solomon codes. *Comm. ACM*, **24**(9):583–584, September 1981.
13. T. Rabin. Robust sharing of secrets when the dealer is honest or faulty. *J. of the ACM*, **41**(6):1089–1109, 1994.
14. 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.
15. 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.