Secure Communication in Minimal Connectivity Models*

Matthew Franklin[†]
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304

Rebecca N. Wright
AT&T Labs Research
180 Park Avenue
Florham Park, NJ 07932

Abstract

Problems of secure communication and computation have been studied extensively in network models. In this work, we ask what is possible in the information-theoretic setting when the adversary is very strong (Byzantine) and the network connectivity is very low (minimum needed for crash-tolerance). We concentrate on a new model called "multicast lines", and show a sizable gap between the connectivity required for *perfect* security and for *almost perfect* security. Our results also have implications to the commonly studied simple channel model and to general secure multiparty computation.

1 Introduction

If two parties are connected by a private and authenticated channel, then secure communication between them is guaranteed. However, in most networks, many parties are only indirectly connected, as elements of an incomplete network of private and authenticated channels. The interplay of network connectivity and secure communication has received a lot of attention in the literature [Dol82, BGW88, CCD88, Bea89, RB89, BCG93, DDWY93, SA96]. Not only is secure communication important in its own right, but it is also an essential primitive from which general secure computation can be achieved [BGW88, CCD88].

Much is known if the channels are *simple*, i.e., if each channel has a single sender and a single recipient. If there are t faulty processors, and the faults are passive *gossipers*, then $t+1$ disjoint paths of channels between sender and receiver are necessary and sufficient for secure communication. The same is true for a setting in which the only faults are crash failures. In contrast, if the t faulty processors are active *Byzantine* failures, under the control of a computationally unbounded adversary, then $2t+1$ disjoint paths between sender and receiver are necessary and sufficient [DDWY93]. Notice the gap in the connectivity required to tolerate a weak adversary and a strong one.

Less is known when a channel may have multiple recipients. The case of passive faults in multi-recipient networks has been studied previously [FY95]. The case of active faults in the *public broadcast* model (which can be thought of as the largest possible multi-recipient channels) has also been studied previously [GGL91]. In this paper, we begin the study of active faults for other kinds of multi-recipient networks.

*A preliminary version appeared in *Advances in Cryptology – Eurocrypt '98 Proceedings*, Lecture Notes in Computer Science 1403, Springer-Verlag, pp. 346–360.
[†]Work done while at AT&T Labs Research.

Appears in *J. Cryptology*, Vol. 13, No. 1, 2000, pp. 9–30.

It is not immediately obvious whether the change from simple channels to multi-recipient channels helps or hurts an active adversary. On one hand, the adversary may benefit from the loss in privacy of every channel. On the other hand, the adversary too suffers from a restriction, since an incorrect transmission from a faulty processor on a channel will always be received identically by all of that channel's receivers. In the setting that we consider, we will see that the change hurts the adversary more than it helps.

There are limits to what we can expect to achieve in the most general case. Even against a passive adversary, it is known to be co-NP-complete to decide the possibility of secure message transmission for an arbitrary multi-recipient network [FY95]. Note, however, that this does not imply a similar result for active adversaries. The decision problem remains co-NP-complete against a passive adversary when restricted to "neighbor networks" [FY95]. In a neighbor network, there is a multi-recipient channel from each processor to all its neighbors in some underlying graph. The main difficulty in working with neighbor networks is that disjoint paths in the underlying graph do not necessarily correspond to disjoint paths in the neighbor network. Paths in the neighbor network are truly disjoint, called "neighbor disjoint", only when the neighborhoods of the paths in the underlying graph are disjoint as well.

In this paper, we will be working with neighbor networks that have paths that are neighbor disjoint. We can then ignore those processors that may be passive observers but not active participants (since moving a fault from a passive observer to an active participant on the same path can only help the adversary). Following Dolev et al. [DDWY93], we abstract away the network and consider that sender and recipient are connected by some number of *wires* or *simple lines*. Each wire is a disjoint collection of processors arranged linearly, with communication links only between adjacent processors. We add the assumption that anything sent to a neighbor on any line is received identically by the other neighbor, whether or not the originator is faulty. In the literature, this is known as *reliable multicast* [PSL80, Ch85, PG89]. Hence, we call this property *multicast*, turning simple lines into *multicast lines*. We emphasize that our multicast lines model is more restrictive than the neighbor network model in general. One could of course execute our protocols in the more general setting by finding neighbor disjoint paths to act as the separate multicast lines.

Our model is related to that of Bracha and Toueg [BT85], who use *echo-broadcast* to refer to a primitive that restricts the communication behavior of a faulty processor so that contradictory messages are not received by different parties. We remark that the radio network model studied by Alon et al. [ABLP89] is somewhat different from what we consider here. Their work addresses issues of coordination and scheduling that arise in packet radio networks, and does not consider privacy. Note that it is implicit in our model that all nodes know the full network structure. In contrast, Burmester et al. show that the situation may be quite different if the network structure is not known by all parties [BDK97].

We briefly discuss some possibilities of physical realizations of multicast lines. Suppose that all processors are located on a flat physical plane, and equipped with equally powerful radio transmitter-receivers. Suppose that distances and radio strengths can be adjusted so that all one's immediate neighbors are in radio range (for both receiving and transmitting), while all other processors are out of radio range (for both receiving and transmitting). Suppose that the adversary can change the behavior of processors, but cannot tamper with the radios (e.g., cannot change their strengths or move their locations). In this setting, some number of disjoint multicast lines are realizable, e.g., $n = 2$ disjoint multicast lines between all pairs of processors equidistant around a circle, and $n = 3$ between most pairs of processors on the gridpoints of a hexagonal lattice. To get many disjoint multicast lines from radio broadcast seems to require additional physical assumptions, such as radios tuned to specific frequencies for transmission and reception (which the adversary cannot change), physical barriers to block transmission and reception for certain processors (e.g., rough terrain), or a third dimension for placing transmitter-receivers (e.g., in deep space).

There are other ways to achieve multicast lines without using radio broadcast. One approach is to use overlapping token rings or Ethernet buses: give an active tap to one processor for putting messages onto the ring, and give a passive tap to its immediate neighbors for listening only. This works under the assumption that the adversary can influence the behavior of the faulty processors, but cannot affect the behavior of the physical communication links. Another approach, effective against a polynomially-bounded adversary, is to broadcast encrypted messages using shared cryptographic keys. Yet another is to rely on a reliable multicast primitive [Ch85] supported by some modern distributed operating systems.

Our Results This paper has two main areas of contribution. First, we provide a complete characterization of when secure communication is possible over multicast lines and an almost complete characterization of when it is efficient. Second, we compare the power of multicast lines to the power of simple lines alone and to the power of simple lines with a broadcast channel. We show that all three models are of equivalent strength when the security is required to be perfect. In contrast, if a small probability of failure is allowed, then multicast lines are strictly more powerful than simple lines alone, but are equivalent to simple lines with broadcast.

More specifically, we consider two different measures of security: *perfect* (i.e., zero probability that the protocol fails to be secure); and *almost perfect* (i.e., an arbitrarily small probability that the protocol fails to be secure).

We begin by fully exploring the capabilities of multicast lines. Our results for multicast lines are summarized in Figure 1. Note that $(t+1)$-connectivity is sufficient to tolerate t arbitrarily malicious faults—closing the connectivity gap between tolerating a passive adversary and an active one that exists for simple lines—if we are willing to tolerate a small probability of error.

	privacy		
reliability	none	almost perfect	perfect
almost perfect	$n > t$ (§3.1)	$n > t$ (§4.1)	$n > \lceil 3t/2 \rceil$ (§4.2) $n > t$ (§4.3)
perfect	$n > 2t$ (§3.2)	$n > 2t$ (§4.4)	$n > 2t$ (§4.4)

Figure 1: Necessary and sufficient connectivity for secure message transmission over multicast lines

In §3, we first consider reliability alone, giving protocols that will be used as building blocks when we consider reliability with privacy. In §3.1, we give a protocol over any $n > t$ multicast lines for transmitting a message with almost perfect security. That is, there remain arbitrarily small probabilities δ and ϵ that the protocol fails to be reliable or private, respectively. The protocol is efficient, in the sense that the round complexity and bit complexity are (low-degree) polynomials of the size of the network, $\log \frac{1}{\delta}$ and $\log \frac{1}{\epsilon}$ (Theorem 4.3). The main building block for this protocol is an efficient subprotocol for message transmission over $n > t$ multicast lines with almost perfect reliability but with no privacy (Theorem 3.5). This protocol uses novel authentication techniques for guaranteeing that the correct message “outscores” the wrong ones, as well as techniques of privacy amplification that are related to the approach of Bennett, Brassard, and Robert [BBR88].

We also show (Theorem 3.6) that perfect reliability over multicast lines cannot be achieved if $n \leq 2t$, providing matching upper and lower bounds. We then turn to the case of perfect privacy. We modify the almost perfectly private protocol to achieve *perfect* privacy and almost perfect reliability when $n > \lceil 3t/2 \rceil$ (Corollary 4.5). Using quite different techniques, we can achieve message transmission with perfect privacy and almost perfect reliability over any $n > t$ multicast lines (Theorem 4.14). While the round complexity of this protocol is low, the bit complexity is exponential in n. All of our almost perfectly secure protocols have the desirable property that if no faults actually occur, then they will actually provide perfect security.

We then consider the models of simple lines only and simple lines with broadcast. In [DDWY93], they consider only perfect security. Here, we show that their $2t+1$-connectivity requirement holds even to achieve almost perfect security. Hence, this connectivity requirement can be considered a property of simple lines, rather than a property of perfect security. Further, we show that multicast lines are essentially equivalent to simple lines plus a broadcast channel. As shown by [DDWY93], $2t+1$ simple lines are required for message transmission with perfect privacy and perfect reliability. We show (Theorem 5.1) that this remains true when privacy is not required and there is a fairly large probability of failure of reliability.

The comparison between the three models is summarized in Figure 2. We remark that it is not immediately obvious that the lower bound techniques for simple lines do not generalize to multicast lines with almost perfect security, which makes our $t+1$ sufficiency results all the more surprising.

	almost perfect security	**perfect security**
simple lines only	$n > 2t$ (Thm 5.1)	$n > 2t$ [DDWY93]
with b/c channel	$n > t$ (Cor 5.2)	$n > 2t$ (Thm 5.4)
multicast lines	$n > t$ (Thm 4.3)	$n > 2t$ (Cor 4.15)

Figure 2: Necessary and sufficient connectivity: comparison of simple and multicast lines

Our results can also be used to strengthen the secure multiparty computation result of Rabin and Ben-Or [RB89]. In their setting, $n \geq 2t+1$ parties are connected by a complete graph of private authenticated single-receiver channels, and also have broadcast. We show that the channel connectivity can be reduced to $t+1$ in this case (Corollary 5.3).

2 The Model

We begin by precisely defining our model. Throughout the paper, n denotes the number of multicast lines and t denotes the number of faults under the control of the adversary.

Communication Model Party A (the message transmitter) and party B (the message recipient) are connected by n *lines*. The jth line is a sequence of $m+2$ nodes $X_{0,j}, X_{1,j}, \ldots, X_{m,j}, X_{m+1,j}$, where $X_{0,j} = A$ and $X_{m+1,j} = B$. It is assumed that $m \geq 1$. (Allowing the degenerate case $m = 0$ would change some of our results.) We may use the ordered pair (i,j) to denote the node $X_{i,j}$, and V to denote the set of all nodes $\{(i,j) : 0 \leq i \leq m+1, 1 \leq j \leq n\}$. Let $G = (V, E)$ be the undirected graph with edges $E = \{(X_{i,j}, X_{i+1,j}) : 0 \leq i \leq m, 1 \leq i \leq n\}$, i.e., neighbors on a line are neighbors in G. We may use the term *internal node* to denote $V - \{A, B\}$. It simplifies the exposition of our protocols to assume all lines are of the same length, but it is clear how to modify all our protocols and lower bound proofs to the case where lines are of different lengths.

We consider *multicast* as our only communication primitive. A message that is multicast by any node is received by all its neighbors (i.e., both neighbors of an internal node, or all n neighbors of A or B). Furthermore, a multicast value is received with privacy (i.e., non-neighbors learn nothing about what was sent) and authentication (i.e., neighbors are guaranteed to receive the value that was multicast and to know which neighbor multicast it).

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 probability distribution Pr. At the end of the protocol, the receiver B outputs a message $M^B \in \mathcal{M}$. We consider a synchronous system in which messages are sent via multicast in *rounds*. During each round of the protocol, each node first receives any messages that were multicast by its neighbors at the end of the previous round, then flips coins and performs local computations, and then possibly multicasts a message. For all of the protocols in this paper, $\mathcal{M}$ must be representable as a subset of a finite field $\mathbf{F}$.

Adversary Model We consider *active*, or *Byzantine*, attacks, in which t internal nodes are under the control of an adversary of unlimited computational power. The adversary is assumed to know the complete protocol specification, message space $\mathcal{M}$, size of network, and any inputs—other than M^A—held by any party (i.e., all relevant information except M^A and the coin flips used by V during the execution). At the start of the protocol, the adversary chooses the message distribution Pr and the t faulty nodes. It is a simplifying assumption that all faults are chosen before the start of the protocol, but the results in this paper are not affected if the adversary is given the additional power to choose faults during the execution of the protocol. The adversary can view all the behavior at the faulty nodes (coin flips, computations, messages received) as well as control the messages that they multicast. The adversary cannot violate the multicast constraint, i.e., whatever is received by one neighbor of a faulty node is received by both neighbors.

For any execution of the protocol, let adv be the adversary's view of the entire protocol, i.e., the behavior of the faulty nodes in every round, the initial state of the adversary, and the coin flips of the adversary in every round. We write $\text{adv}(m, r)$ to denote the adversary's view when $M^A = m$ and when the sequence of coin flips used by the adversary is r. Note that adv and $\text{adv}(m, r)$ are random variables, e.g., $\text{adv}(m, r)$ depends on the coin flips of the honest parties.

Privacy: A message transmission protocol is *ϵ-private* if, for every two messages $m_0, m_1 \in \mathcal{M}$ and every r, $\sum_c |\Pr[\text{adv}(m_0, r) = c] - \Pr[\text{adv}(m_1, r) = c]| \leq 2\epsilon$. The probabilities are taken over the coin flips of the honest parties, and the sum is over all possible values of the adversary's view.

Reliability: 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 choice of M^A and the coin flips of V and the adversary.

Security: A message transmission protocol is (ϵ, δ)*-secure* if it is ϵ-private and δ-reliable.

Efficiency: An (ϵ, δ)-secure message transmission protocol is *efficient* if its round complexity and bit complexity are polynomial in the size of the network, $\log \frac{1}{\epsilon}$ (if $\epsilon > 0$), and $\log \frac{1}{\delta}$ (if $\delta > 0$).

Note that if $t \geq n$, then it is possible to achieve neither reliable nor private message transmission, since an adversary can place one fault on each line and either block or monitor all communication between A and B. We therefore assume $t < n$ throughout the remainder of the paper.

Authentication Codes Our protocols make use of information-theoretically secure authentication over a finite field. For simplicity, we use the same authentication code throughout this paper:

Definition 1 *Let* $\mathbf{F}$ *be a finite field, and let* $a, b, M \in \mathbf{F}$. *We define* $\mathrm{auth}(M, a, b) = aM + b$.

This function has been used for similar purposes by many papers (cf. [RB89, Rab94]).

Throughout the paper, we write $|S|$ to denote the number of elements in the set S. We write $x \in_R S$ to indicate a choice with respect to the uniform distribution on S.

3 Reliable Communication Over Multicast Lines

In this section, we address the question of reliable communication, with no requirement of privacy. We consider almost perfect reliability first, in §3.1, and show that it is achievable whenever $n > t$. In §3.2, we consider perfect reliability, and show that it is possible only when $n > 2t$.

3.1 Almost Perfect Reliability

In this section, we show how to efficiently achieve δ-reliable communication for $\delta > 0$ when $n > t$. To achieve reliable communication, we use two sub-protocols. In the Basic Propagation Protocol, A tries to propagate a value s^A to B. To do this, the multicast lines are used essentially as simple lines. First, A sends s^A to its neighbors. In turn, each (non-faulty) node receives and propagates s^A "down" the simple line towards B.

Basic Propagation Protocol

- In round 1, party A multicasts s^A.
- In round ρ for $2 \leq \rho \leq m$, each $X_{\rho-1,j}$ $(1 \leq j \leq n)$ expects to receive a single element from $X_{\rho-2,j}$. Let $u_{\rho-1,j}$ be this value if a value was in fact received, or a publicly known default element otherwise. At the end of round ρ, party $X_{\rho-1,j}$ multicasts $u_{\rho-1,j}$.
- In round $m+2$, party B receives a single element from each $X_{m,j}$, or substitutes the default element. Let s^B_j be the value received or substituted on line j.

It is clear that if there are no faults on a given line, then the value received on that line by B in round $m+1$ is A's starting value:

Fact 1 *If there are no faults on the* j*th line, then* $s^B_j = s^A$.

In the Full Distribution Protocol, each internal node $X_{i,j}$ tries to transmit an element $s_{i,j}$ to A and B. As in the Basic Propagation Protocol, the lines are used essentially as simple lines. In order to help recipients keep track of which messages should be propagated and which messages should be ignored, the "intended" recipient or recipients of a message are included. Specifically, we say X_i is an intended recipient of m if X_i receives (m, X_i) or (m, X) where $X_i \in X$.

Full Distribution Protocol

- In round 1, each $X_{i,j}$ multicasts $(s_{i,j}, \{X_{i-1,j}, X_{i+1,j}\})$.
- In round ρ for $2 \leq \rho \leq m+1$:
  - For $1 \leq j \leq n$ and $\rho \leq i \leq m$, party $X_{i,j}$ expects to be the intended recipient of an element from $X_{i-1,j}$ (initiated by $X_{i-\rho+1,j}$). Let $u_{i,j}$ be the received value or a default value if none is received.
  - For $1 \leq j \leq n$ and $1 \leq i \leq m-\rho+1$, party $X_{i,j}$ expects to be the intended recipient of an element from $X_{i+1,j}$ (initiated by $X_{i+\rho-1,j}$). Let $v_{i,j}$ be the received or default value.
  - For $1 \leq j \leq n$, party A expects to be the intended recipient on the jth line of a single element (initiated by $X_{\rho-1,j}$). Let $s^A_{\rho-1,j}$ be the received or default value.
  - For $1 \leq j \leq n$, party B expects to be the intended recipient on the jth line of a single element from $X_{m+\rho,j}$. Let $s^B_{m+\rho,j}$ be the received or default value.
  - $X_{i,j}$ multicasts $(u_{i,j}, X_{i+1,j})$ if $\rho \leq i \leq m$, and $(v_{i,j}, X_{i-1,j})$ if $1 \leq i \leq m-\rho+1$.

As with the Basic Propagation Protocol, it is clear in the Full Distribution Protocol that messages originating on non-faulty lines are correctly received at their destinations:

Fact 2 *If there are no faults on the jth line, then $s^A_{i,j} = s^B_{i,j} = s_{i,j}$ for all $1 \leq i \leq m$.*

In addition, since a message sent by a faulty node is multicast identically to its neighbors (and if no message is sent, the neighbors both substitute the same publicly known default value), it follows that even on lines with one fault, A and B agree on the value originated at the faulty node. Note that this captures *precisely* the advantage that multicast lines give to the parties over simple lines.

Fact 3 *If $X_{i,j}$ is the only fault on the jth line, then $s^A_{i,j} = s^B_{i,j}$.*

To achieve reliable message transmission, each internal node chooses a random authentication key. A's message M^A is authenticated with respect to each of these mn random authentication keys. The adversary can only reliably forge an authentication if it has seen the key, i.e., for keys initiated on a line with at least one fault. By contrast, A and B agree on at least one authentication key from each fault-free and single-fault line. If all received messages are ranked by B according to the number of lines from which corroborating authentication keys originated, then the real message will almost always get the highest rank.

Reliable Transmission Protocol

- In rounds 1 through $m+2$, the nodes of V execute an instance of the Full Distribution Protocol. The element that $X_{i,j}$ initiates is $(a_{i,j}, b_{i,j}) \in_R \mathbf{F}^2$. Let $(a^A_{i,j}, b^A_{i,j})$ and $(a^B_{i,j}, b^B_{i,j})$ be the values that A and B receive or substitute as the element initiated by $X_{i,j}$.
- In rounds $m+3$ through $2m+4$, the nodes of V execute an instance of the Basic Propagation Protocol from A to B. The element that A initiates is $(M^A, \{(i, j, \mathrm{auth}(M^A, a^A_{i,j}, b^A_{i,j})) : 1 \le i \le m, 1 \le j \le n\})$. In round $2m+4$, node B receives or substitutes $(M^B_k, \{(i, j, u^B_{i,j,k}) : 1 \le i \le m, 1 \le j \le n\})$ on the kth line, $1 \le k \le n$.
- Let $r(k) = |\{j : \exists i. u^B_{i,j,k} = \mathrm{auth}(M^B_k, a^B_{i,j}, b^B_{i,j})\}|$. Node B outputs M^B_k for the k that maximizes $r(k)$.

Let w_0 denote the number of lines with no faults, w_1 denote the number with exactly one fault, and w_+ denote the number with two or more faults. Recall that n is the number of multicast lines, t is the number of faults under the control of the adversary, and n is assumed to be larger than t.

Lemma 3.1 *$w_0 > w_+$.*

Proof: $w_0 + w_1 = n - w_+ > t - w_+ \ge (w_1 + 2w_+) - w_+ = w_1 + w_+$, so $w_0 > w_+$. ■

Lemma 3.2 *There exists k such that $r(k) \ge w_0 + w_1$ and $M^B_k = M^A$.*

Proof: Since $n > t$, there exists at least one line k with no faults. By Fact 2, the value received by B on this line in round $2m+4$ is the same as the value multicast by A in round $m+3$. That is, $M^B_k = M^A$ and $u^B_{i,j,k} = \mathrm{auth}(M^A, a^A_{i,j}, b^A_{i,j})$ for $1 \le i \le m$ and $1 \le j \le n$. Since $M^B_k = M^A$, the authentication test $u^B_{i,j,k} \stackrel{?}{=} \mathrm{auth}(M^B_k, a^B_{i,j}, b^B_{i,j})$ that B performs succeeds whenever $(a^A_{i,j}, b^A_{i,j}) = (a^B_{i,j}, b^B_{i,j})$. By Facts 2 and 3, this happens for every i when the jth line has no faults (w_0 times), and for at least one i when the jth line has one fault (w_1 times). Thus, $r(k) \ge w_0 + w_1$. ■

Lemma 3.3 *Let $a, b \in_R \mathbf{F}$, let $M \in \mathbf{F}$, and let $v = \mathrm{auth}(M, a, b)$. Suppose $\mathcal{P}$ is any procedure (possibly randomized, not necessarily polynomial time) that, on input M, v, outputs $M^*, v^* \ne M, v$. Then the probability that $v^* = \mathrm{auth}(M^*, a, b)$ is at most $\frac{1}{|\mathbf{F}|}$, where the probability is taken over the coin flips of the procedure and the uniform choices of a and b.*

Proof: Since $v = aM + b$, it follows that $v^* = aM^* + b$ if and only if $a = (v^* - v)(M^* - M)^{-1}$. Thus $\mathcal{P}$ is essentially guessing a value for a from input M, v. But all values of $a \in \mathbf{F}$ are equally likely given M, v, since there exists a unique b that is consistent with every possible M, v, a. Thus $\mathcal{P}$ cannot guess the value for a with success greater than $\frac{1}{|\mathbf{F}|}$. ■

Lemma 3.4 *The probability that there exists k such that $M^B_k \ne M^A$ and $r(k) > w_1 + w_+$ is less than $\frac{mn^2}{|\mathbf{F}|}$.*

Proof: Suppose that $r(k) > w_1 + w_+$ and $M_k^B \neq M^A$. Let $N = \{1, \ldots, n\}$, and let $W_0 \subseteq \{1, \ldots, n\}$ be the lines with no faults. Then we must have $\mathrm{auth}(M_k^B, a_{i,j}^B, b_{i,j}^B) = u_{i,j,k}^B$ for at least one i, j, k such that $j \in W_0$ and $k \in N - W_0$. By Lemma 3.3, this can be achieved with probability at most $\frac{1}{|\mathbf{F}|}$ for any given i, j, k. Thus it is achieved over all candidate i, j, and k with probability at most $m \cdot |W_0| \cdot |(N - W_0)|/|\mathbf{F}| = mw_0(n - w_0)/|\mathbf{F}| < mn^2/|\mathbf{F}|$. ■

Theorem 3.5 *If $\delta > 0$ and $n > t$, the Reliable Transmission Protocol is an efficient δ-reliable message transmission protocol when $|\mathbf{F}| \geq \frac{mn^2}{\delta}$.*

Proof: Suppose $\delta > 0$ and $\mathbf{F} \supseteq \mathcal{M}$ such that $|\mathbf{F}| \geq \frac{mn^2}{\delta}$. The Reliable Transmission Protocol takes $2m + 4$ rounds, and the bit complexity is a low degree polynomial in m, n, and $\log \frac{1}{\delta}$, so it is efficient. To see that it is reliable, consider a run of the protocol in which A starts with the message M^A and B outputs M^B. By Lemma 3.2, there is some k such that $r(k) \geq w_0 + w_1$ and $M_k^B = M^A$. By Lemma 3.4, the probability that there exists k' such that $M_{k'}^B \neq M^A$ and $r(k') > w_1 + w_+$ is less than $\frac{mn^2}{|\mathbf{F}|} \leq \delta$. By Lemma 3.1, $w_1 + w_+ < w_0 + w_1$, so it follows that $\Pr[M^B = M^A] \geq 1 - \delta$. Hence the Reliable Transmission Protocol is δ-reliable. ■

Since reliable communication is not possible when $t > n$, this protocol provides matching upper and lower bounds for almost perfect reliability without privacy.

3.2 Perfect Reliability

In this section, we show that perfect reliability is unachievable over n multicast lines when $n \leq 2t$. The proof follows Dolev et al. [DDWY93].

Theorem 3.6 *0-reliable message transmission over n multicast lines is impossible when $n \leq 2t$.*

Proof: Note that it is sufficient to show that 0-reliable message transmission is impossible when $n = 2t$, since an adversary can always choose to use fewer than t of its allowed faults. Consider a graph of $n = 2t$ multicast lines, each of length $m \geq 1$, and suppose that Π is a message transmission protocol. The adversary behaves as follows. All faults will be placed on the first processor of some line (i.e. $X_{1,j}$, for some j). The adversary flips a coin to decide whether to disrupt $W_0 = \{1, \ldots, t\}$ (first half of the lines) or $W_1 = \{t+1, \ldots, 2t\}$ (second half of the lines). Let W_b denote the faulty subset, and let W_{1-b} denote the honest subset. The adversary will attempt to maintain a simulation of a possible behavior of A executing Π for some other message.

Let $s_\rho^{i,j}$ be the message multicast by processor $X_{i,j}$ in round ρ of the execution. Let s_ρ^A (respectively s_ρ^B) be the message multicast by A (respectively B) in round ρ of the execution. Let $\widehat{s}_\rho^A$ be the message, chosen by the adversary, that A supposedly multicast in round ρ of the simulation.

In each round ρ, the adversary causes each $X_{1,j}$ in W^b to follow the protocol Π as if the messages that it received from A were $\widehat{s}_1^A, \ldots, \widehat{s}_{\rho-1}^A$. That is, the message $s_\rho^{1,j}$ that the adversary will cause to be multicast by $X_{1,j}$ in round ρ is a function of these simulated messages from A, the real messages $s_1^{2,j}, \ldots, s_{\rho-1}^{2,j}$ from X_{2j}, and local coin flips for $X_{1,j}$ chosen at random by the adversary.

With nonzero probability, all of the adversary's choices for $\widehat{s}_1^A, \ldots, \widehat{s}_\rho^A$ are consistent with a possible behavior of A executing Π for some other message, so B cannot halt at the end of round ρ and output M^B with certainty. ■

Note that the nonzero probability of this adversary attack succeeding is very small, and depends on the number of random bits used by the processors. Further, the proof does not exclude the possibility of a δ-reliable protocol whose complexity is a function of $\frac{1}{\delta}$. Note also that, unlike the simple lines setting of Dolev et al., the sender A learns which nodes are faulty during the execution of Π. The proof shows that this extra information does not help A and B.

4 Secure Communication Over Multicast Lines

In this section, we consider reliable *and* private communication. By Theorem 3.6, we can not hope to achieve perfect reliability unless $n > 2t$. Hence, we first consider the case of almost perfect privacy with almost perfect reliability. We show in §4.1 that almost perfect security is achievable whenever $n > t$. In §4.2, we show that it is possible to efficiently achieve perfect privacy with almost perfect reliability when $n > \lceil 3t/2 \rceil$. We do not know whether it is possible to efficiently achieve perfect privacy when $t < n \leq \lceil 3t/2 \rceil$, but we are able to give an inefficient solution in §4.3. In §4.4, we point out that the protocol of [DDWY93], combined with our protocols, can be modified to work for perfect privacy with perfect reliability over multicast lines if $n > 2t$.

4.1 Almost Perfect Security

In this section, we show it is possible to efficiently achieve (ϵ, δ)-secure message transmission over multicast lines.

In the Private Propagation Protocol, A tries to propagate a different $s_j^A \in \mathbf{F}$ to B on each line j, $1 \leq j \leq n$. This protocol demonstrates that it does not matter whether the multicast property is extended to sender and receiver in our model, since they can use it to communicate a different value to each of their neighbors.

Private Propagation Protocol

- In round 1, each $X_{1,j}$ multicasts $r_j \in_R \mathbf{F}$.
- In round 2, A multicasts $(u_1, \ldots, u_n)$, where each $u_j = s_j^A + r_j$, $1 \leq j \leq n$.
- In rounds 3 through $m+4$, each $X_{1,j}$ now proceeds as in the Basic Propagation Protocol with the value $s_j = u_j - r_j$. Let s_j^B be the element ultimately received by B on the jth line.

Fact 4 *If there are no faults on the jth line, then $s_j^B = s_j^A$ and* $\Pr[s_j^A = s|\mathrm{adv}] = \Pr[s_j^A = s]$.

Using the Private Propagation Protocol, we can achieve private message transmission. Intuitively, the protocol works as follows. A privately propagates a different random one-time pad on each line to B. Using the Reliable Transmission Protocol from the preceding section and a randomized authentication procedure, A and B determine which pads have been received identically at both ends. A then encrypts the message using the sum of the pads that pass the test, and transmits this encryption reliably (and non-privately) to B. A similar protocol appears in [BF97]. Formally, we have the following:

Private Transmission Protocol

- In rounds 1 through $m+4$, the nodes of V execute an instance of the Private Propagation Protocol. A propagates to B the values $c_j^A, d_j^A \in_R \mathbf{F}^2$ on each line j. Let c_j^B, d_j^B be the values received by B on the j line.

- For $1 \leq j \leq n$, B chooses $r_j^B \in_R \mathbf{F}$, and computes $s_j^B = \mathrm{auth}(r_j^B, c_j^B, d_j^B)$. In rounds $m+5$ through $3m+9$, the nodes of V execute an instance of the Reliable Transmission Protocol. B then $\min(\epsilon, \frac{\delta}{3})$-reliably transmits to A the values r_j^B, s_j^B. Let r_j^A, s_j^A for $1 \leq j \leq n$ be the values received by A as the output of the Reliable Transmission Protocol.

- A computes $W^A = \{j : s_j^A = \mathrm{auth}(r_j^A, c_j^A, d_j^A)\}$ and $z^A = M^A + \sum_{j \in W^A} c_j^A$. In rounds $3m+10$ through $5m+13$, the nodes of V execute another instance of the Reliable Transmission Protocol. A $\frac{\delta}{3}$-reliably transmits to B the values W^A and z^A. Let W^B, z^B be the values received by B as the output of the Reliable Transmission Protocol.

- B computes $M^B = z^B - \sum_{j \in W^B} c_j^B$.

Lemma 4.1 *The Private Transmission Protocol is ϵ-private.*

Proof: Since $n > t$, there exists a non-faulty line j^*. Since j^* is a non-faulty line, $c_{j^*}^A = c_{j^*}^B$ and $d_{j^*}^A = d_{j^*}^B$. Suppose the reliable transmission from B to A succeeds; let RT denote this event. Then, for all $1 \leq j \leq n$, $r_j^B = r_j^A$ and $s_j^B = s_j^A$. In particular, this implies that $s_{j^*}^A = s_{j^*}^B = c_{j^*}^B r_{j^*}^B + d_{j^*}^B = c_{j^*}^A r_{j^*}^A + d_{j^*}^A$, and so $j^* \in W^A$. Every $c_{j^*}^A$ is equally likely given r_j^A, s_j^A. Since $z^A = M^A + c_{j^*}^A + \sum_{j \in W^A, j \neq j^*} c_j^A$, we have that every M^A is equally likely given r_j^A, s_j^A, z^A. Since this is the only relevant information about M^A in adv, other than Pr, we have that $\Pr[\mathrm{adv}(m_0, r) = c | RT] = \Pr[\mathrm{adv}(m_1, r) = c | RT]$ for every pair of messages m_0, m_1, adversary coin flips r, and possible view c. We know that $\Pr[RT] \geq 1 - \epsilon$, and thus $\Pr[RT | M^A = M, r] \geq 1 - \epsilon$ for all M and all adversary coin flips r (since the adversary can choose Pr and r).

Let C_i be the set of adversary views where $M^A = m_i$ and RT succeeded; let $\bar{C}_i$ be the set of adversary views where $M^A = m_i$ and RT failed. By the analysis of the preceding paragraph, $\sum_{c \in C_i} |\Pr[\mathrm{adv}(m_0, r) = c] - \Pr[\mathrm{adv}(m_1, r) = c]| = 0$ and $\sum_{c \in \bar{C}_i} |\Pr[\mathrm{adv}(m_0, r) = c] - \Pr[\mathrm{adv}(m_1, r) = c]| \leq \epsilon$. Thus $\sum_c |\Pr[\mathrm{adv}(m_0, r) = c] - \Pr[\mathrm{adv}(m_1, r) = c]| \leq 2\epsilon$. ■

Lemma 4.2 *If both reliable transmissions succeed, then* $\Pr[M^B \neq M^A] \leq \frac{n}{|\mathbf{F}|}$.

Proof: Suppose both reliable transmissions succeed. Then $r_j^A = r_j^B$ and $s_j^A = s_j^B$ for all $1 \leq j \leq n$, and $W^A = W^B$ and $z^A = z^B$. Therefore, if $j \in W^A$, then $c_j^A r_j^A + d_j^A = s_j^A = s_j^B = c_j^B r_j^B + d_j^B = c_j^B r_j^A + d_j^B$ which implies that $r_j^A = (d_j^B - d_j^A)(c_j^A - c_j^B)^{-1}$. Since c_j^A, d_j^A, c_j^B, and d_j^B are fixed before the random choice of r_j^A, it follows that for any fixed $j \in W^A$, $\Pr[c_j^A \neq c_j^B] \leq 1/|\mathbf{F}|$.

If both reliable transmissions succeed and $M^B \neq M^A$, then $c_j^A \neq c_j^B$ for at least one $j \in W^A$. By the above, this occurs with probability at most $|W^A|/|\mathbf{F}| < n/|\mathbf{F}|$. ■

Theorem 4.3 *If $\epsilon > 0$, $\delta > 0$, and $n > t$, the Private Transmission Protocol is an efficient (ϵ, δ)-secure message transmission protocol when $|\mathbf{F}| \geq \frac{3n}{\delta}$.*

Proof: Take $\mathbf{F} \supseteq \mathcal{M}$ such that $|\mathbf{F}| \geq \frac{3n}{\delta}$. By Lemma 4.1, the Private Transmission Protocol is ϵ-private. It is efficient since the Reliable Transmission Protocol is efficient. By Theorem 3.5, the probability that either reliable message transmission fails is no more than $\frac{2\delta}{3}$. Hence, by Lemma 4.2, $\Pr[M^A \neq M^B] \leq \frac{2\delta}{3} + \frac{n}{|\mathbf{F}|} \leq \delta$. ■

Since secure communication is not possible when $t > n$, this protocol provides matching upper and lower bounds for almost perfect privacy with almost perfect reliability.

4.2 Perfect Privacy when $n > \lceil 3t/2 \rceil$

Note that the requirement that $\epsilon > 0$ is necessary since the second step of the protocol requires a $\min(\epsilon, \frac{\delta}{3})$-reliable transmission. In fact, there is an adversary attack against the protocol that succeeds in compromising privacy with nonzero (at most ϵ) probability. First, the adversary listens on t lines in the private propagation phase. The adversary then partially disrupts the first reliable transmission from B to A, affecting on each of the t faulty lines all of the values associated with the fault-free lines. If the adversary successfully guesses the appropriate unseen authentication in the Reliable Transmission Protocol (as in Lemma 3.4), the disruption succeeds, and no authentication check by A passes for any fault-free line. In this case, W^A contains only faulty lines, allowing the adversary to determine M^A from z^A.

It is possible to foil this attack if the first reliable transmission from B to A is done using $\frac{\delta}{3}$-reliable message transmission such that A can detect when the correct message is not received. Then A could send nothing when this reliable transmission fails. In fact, the proof of Lemma 4.1 is easily adapted to show that the adversary never learns *any* information about the message. Fortunately, the Reliable Transmission Protocol of §3.1 can easily be modified to provide this property when $n > \lceil 3t/2 \rceil$.

Definition 2 *A message transmission protocol is* perfectly detecting *if B either terminates with $M^B = M^A$ or terminates and outputs nothing.*

Corollary 4.4 (to Theorem 3.5) *If $\delta > 0$ and $n > \lceil 3t/2 \rceil$, then there exists an efficient perfectly-detecting δ-reliable message transmission protocol.*

Proof: We change the output rule for B in the final step of the Reliable Transmission Protocol to the following: If there is a unique k such that $r(k) > t$, B outputs M_k^B. Otherwise, B terminates without output.

When $n > \lceil 3t/2 \rceil$, we have that $w_0 + w_1 > t \geq w_1 + w_+$. By Lemma 3.2, there will always be some k such that $M_k^B = M^A$ and $r(k) > t$. Thus B will always either output the correct message or will output nothing, and so the modified protocol is perfectly detecting. By Lemma 3.4, the probability that the protocol outputs nothing is at most δ when $|\mathbf{F}| \geq \frac{mn^2}{\delta}$. ■

Corollary 4.5 *If $\delta > 0$ and $n > \lceil 3t/2 \rceil$, then there exists an efficient $(0, \delta)$-secure message transmission protocol.*

4.3 Perfect Privacy when $t < n < \lceil 3t/2 \rceil$

In §4.2, we showed how to efficiently achieve perfect privacy and almost perfect reliability when $n > \lceil 3t/2 \rceil$. In this section, we continue our investigation of perfect privacy, and show that perfect privacy and almost perfect reliability can be achieved at minimum connectivity of $n > t$, although the bit complexity is exponential in the number of lines. Subsequently, Wang and Desmedt have shown an efficient $(0, \delta)$-secure message transmission protocol that works for all $n < t$ [WD99].

Intuitively, our protocol proceeds as follows. The receiver attempts to transmit to the sender many random, uniquely labeled, one-time pads. The sender is able to find one pad that was transmitted with perfect privacy and almost perfect reliability. The sender then transmits to the receiver—with almost perfect reliability *and without privacy*—the encryption of the message using the one-time pad, together with the label of the pad. The receiver can look up the one-time pad from the label, and decrypt the message.

Formally, define a *probe set* S to be a subset of nodes such that no two nodes are in the same line: If $(i, j) \in S$ and $i' \neq i$ then $(i', j) \notin S$. Let $\mathcal{L}$ denote the set of all probe sets. Let $\psi : \mathcal{L} \rightarrow \mathbf{F}$ be an injective mapping from probe sets to elements of $\mathbf{F}$. Given a function $f(x) = (y_1, y_2, y_3)$, we write $f_i(x)$ to denote y_i. We define a *double masking* procedure for authentication with secrecy:

$$\mathrm{DoubleMask}(M, a, b, c) = (aM + b, M + c)$$

Without knowledge of the "secret key" a, b, c, no information about the "encrypted" value M can be inferred, and any tampering is almost always detected. We define the corresponding *unmask* procedure:

$$\mathrm{Unmask}((u, v), a, b, c) = \begin{cases} v - c & \text{if } a(v - c) = (u - b) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Then the protocol is as follows:

Perfectly Private Transmission Protocol

- In rounds 1 through $m+1$, the nodes of V execute an instance of the Full Distribution Protocol. The element that $X_{i,j}$ initiates is $f_{i,j} : \mathcal{L} \rightarrow \mathbf{F}^3$, chosen uniformly at random from the set of all complete functions from $\mathcal{L}$ to $\mathbf{F}^3$. Let $f^A_{i,j}$ and $f^B_{i,j}$ be the elements received by A and B, respectively, corresponding to the element initiated by $X_{i,j}$. Let $\alpha(S) = \sum_{(i,j) \in S} f^A_{i,j}(S)$, and $\beta(S) = \sum_{(i,j) \in S} f^B_{i,j}(S)$, for every $S \in \mathcal{L}$. (The summations are component-wise addition over the finite field.)

- B computes $g^B : \mathcal{L} \rightarrow \mathbf{F}^2$, where $g^B(S) = \mathrm{DoubleMask}(r^B_S, \beta_1(S), \beta_2(S), \beta_3(S))$ and $r^B_S \in_R \mathbf{F}$ for every $S \in \mathcal{L}$. In rounds $m + 2$ through $2m + 3$, party B propagates g^B to A using the Basic Propagation Protocol. Let g^A_j be the element that A receives on the jth line in round $2m + 3$.

- Next, A computes the tuple $z^A = (\psi(S'), M^A + r^A_{S',j})$, where $r^A_{S',j} = \mathrm{Unmask}(g^A_j(S'), \alpha_1(S'), \alpha_2(S'), \alpha_3(S'))$ and no larger probe set leads to a successful unmasking for any j. In rounds $2m + 4$ through $4m + 7$, A sends z^A to B using the (almost perfect) Reliable Transmission Protocol. Let $z^B = (x^B, y^B)$ be the element that B accepts as the outcome of the Reliable Transmission Protocol.

- B outputs $M^B = y^B - r^B_{\psi^{-1}(x^B)}$.

The remainder of this section will prove that this protocol achieves perfect privacy and almost perfect reliability. Note, however, that the protocol is not efficient, since the message sent in the first step is the description of a function on the set of probe sets, which is of size $(m+1)^n$.

We first prove the protocol is almost perfectly reliable.

Lemma 4.6 *Let $(u,v) = \mathrm{DoubleMask}(M,a,b,c)$, where $a,b,c \in_R \mathbf{F}$, and where $M \in \mathbf{F}$. Let $\mathcal{P}$ be any procedure (possibly randomized, not necessarily polynomial time) that, given input (u,v,M) outputs $(u',v') \neq (u,v)$. Then the probability that $\mathrm{Unmask}((u',v'),a,b,c)$ is defined is at most $\frac{1}{|\mathbf{F}|}$ for every M. Here the probability is over the coin flips of the procedure and the uniform choices of a, b, and c.*

Proof: First note that $a(v-c) = (u-b)$. Second, note that if $\mathrm{Unmask}((u',v'),a,b,c)$ is defined then $a(v'-c) = (u'-b)$. Thus a successful unmask with $(u',v') \neq (u,v)$ must have $v' \neq v$. Now let $\mathcal{P}$ be as in the condition of the lemma. If $\mathcal{P}$ is successful, then it has essentially guessed the values of a and b: $a = (u-u')(v-v')^{-1}$ and $b = u - M(u-u')(v-v')^{-1}$. But a,b cannot be guessed from (u,v,M) with success greater than $\frac{1}{|\mathbf{F}|}$, since there is a unique b,c that is consistent with every possible M,u,v,a. ■

Lemma 4.7 *Let S' be the probe set found by A in round $2m+4$. Then $|S'| \geq w_0 + w_1$.*

Proof: By construction, there is no probe set larger than S' that leads to a successful unmasking for any j. Thus it suffices to show that there exists a probe set of size $w_0 + w_1$ that leads to a successful unmasking for some j. By Facts 2 and 3, there exists $S^* \in \mathcal{L}$ such that $|S^*| = w_0 + w_1$ and $\alpha(S^*) = \beta(S^*)$. By Fact 1, there exists j^* such that $g^A_{j^*} = g^B$. Since $g^B(S^*)$ is a double masking with respect to $\beta_1(S^*), \beta_2(S^*), \beta_3(S^*)$, it follows that S^* leads to a successful unmasking for j^*. ■

Lemma 4.8 *Let $r^A_{S',j}$ be the unmasked value found by A in round $2m+4$. Then $\Pr[r^A_{S',j} \neq r^B_{S'}] \leq \frac{n|\mathcal{L}|}{|\mathbf{F}|}$*

Proof: As in the proof of Lemma 4.7, there exists a line j^* and a probe set S^*, $|S^*| = w_0 + w_1$, such that $r^A_{S^*,j^*} = r^B_{S^*}$. We next show that, for any probe set S of equal size or larger, and for any j, the probability that the unmasking is successful and $r^A_{S,j} \neq r^B_S$ is at most $\frac{1}{|\mathbf{F}|}$. Fix $S \in \mathcal{L}$ such that $|S| \geq w_0 + w_1$ and fix $j \in \{1,\ldots,n\}$. By Lemma 3.1, $|S| > w_1 + w_+$. Thus, at the end of round $m+1$, the adversary has no information about $\alpha(S)$ or $\beta(S)$. If $\alpha(S) = \beta(S)$, then it follows from Lemma 4.6 that $\Pr[r^A_{S,j} \neq r^B_S] \leq \frac{1}{|\mathbf{F}|}$. If $\alpha(S) \neq \beta(S)$, then the probability that $r^A_{S,j} \neq r^B_S$ is at most the probability that the authentication by B succeeds, which is at most $\frac{1}{|\mathbf{F}|}$ by Lemma 3.3. Since in either case $\Pr[r^A_{S,j} \neq r^B_S] \leq \frac{1}{|\mathbf{F}|}$, it follows that $\Pr[r^A_{S',j} \neq r^B_{S'}] \leq \frac{t}{|\mathbf{F}|}|\{S \in \mathcal{L} : |S| \geq w_0 + w_1\}| < \frac{n|\mathcal{L}|}{|\mathbf{F}|}$. ■

Lemma 4.9 *Let $\delta > 0$ and $n > t$. Then the Perfectly Private Transmission Protocol is δ-reliable when $|\mathbf{F}| \geq \frac{1}{\delta}(n(m+1)^n + \frac{1}{2}mn^2)$.*

Proof: Choose $\mathbf{F} \supseteq \mathcal{M}$ such that $|\mathbf{F}| > \frac{1}{\delta}(n(m+1)^n + \frac{1}{2}mn^2)$. If $r^A_{S',j} = r^B_{S'}$ and the Reliable Transmission Protocol is successful, then $M^A = M^B$. By Lemma 4.8, the first requirement fails with probability at most $\frac{n|\mathcal{L}|}{|\mathbf{F}|}$. Since the message to be transmitted by the Reliable Transmission Protocol in rounds $2m+4$ through $4m+7$ is from $\mathbf{F}^2$, it follows from Theorem 3.5 that the second requirement fails with probability at most $\frac{mn^2}{2|\mathbf{F}|}$. Hence, the probability that either requirement fails is at most $\frac{n|\mathcal{L}|}{|\mathbf{F}|} + \frac{mn^2}{2|\mathbf{F}|} = \frac{1}{|\mathbf{F}|}(n(m+1)^n + \frac{1}{2}mn^2) \le \delta$. ■

Next, we turn to proving perfect privacy. We write adv_{2m+3} to denote the random variable corresponding to the adversary's view through the end of round $2m+3$.

Lemma 4.10 *If* $a, b, c \in_R \mathbf{F}$ *then* $\Pr[R = r | \mathrm{DoubleMask}(R, a, b, c)] = \Pr[R = r]$

Proof: For every $u, v, r \in \mathbf{F}$, the equation $(u, v) = \mathrm{DoubleMask}(r, (a, b, c))$ has exactly $|\mathbf{F}|^2$ solutions, from which the lemma follows. ■

Lemma 4.11 $\Pr[r^B_{S'} = r | \mathrm{adv}_{2m+3}] = \frac{1}{|\mathbf{F}|}$ *for all* r *and for all* S' *such that* $|S'| \ge w_0 + w_1$.

Proof: Let $L \subset \{1, \ldots, n\}$ denote the lines that have no faults. Let $G = \{(i,j) : (i,j) \in S', j \in L\}$. Let $\gamma^B(S') = \sum_{(i,j)\in G} f^B_{i,j}(S')$. By definition, $\Pr[r^B_{S'} = r|\mathrm{adv}_{2m+3}] = \Pr[r^B_{S'} = r|g^B(S'), \{f^B_{i,j}(S')\}_{(i,j)\in S'-G}] = \Pr[r^B_{S'} = r|\mathrm{DoubleMask}(r^B_{S'}, \beta(S')), \beta(S') - \gamma^B(S')]$. Since $|S'| \ge w_0 + w_1 > w_1 + w_+ = n - |L|$, we know that G is nonempty. For each $(i,j) \in G$ we have that $f^B_{i,j}(S') = f_{i,j}(S') \in_R \mathbf{F}^3$. This implies that all possible values of $\gamma^B(S')$ are equally likely given adv_{2m+3}. The result then follows from Lemma 4.10. ■

Lemma 4.12 $\Pr[r^A_{S'} = r | \mathrm{adv}_{2m+3}] = \frac{1}{|\mathbf{F}|}$ *for all* r *and for all* S' *such that* $|S'| \ge w_0 + w_1$.

Proof: The argument is similar to the proof of Lemma 4.11. The unmasking of $r^A_{S'}$ was successful, so $\mathrm{DoubleMask}(r^A_{S'}, \alpha(S')) = g^A(S')$. It follows that $\Pr[r^A_{S'} = r|\mathrm{adv}_{2m+3}] = \Pr[r^A_{S'} = r|g^A(S'), \{f^A_{i,j}(S')\}_{(i,j)\in S'-G}] = \Pr[r^A_{S'} = r|\mathrm{DoubleMask}(r^A_{S'}, \alpha(S')), \alpha(S') - \gamma^A(S')]$, where $\gamma^A(S') = \sum_{(i,j)\in G} f^A_{i,j}(S') \in_R \mathbf{F}^3$. By Lemma 4.10, $\Pr[r^A_{S'} = r|\mathrm{adv}_{2m+3}] = \Pr[r^A_{S'} = r] = \frac{1}{|\mathbf{F}|}$. ■

Lemma 4.13 *The Perfectly Private Transmission is* 0*-private for every* $n > t$.

Proof: The only information about M^A that the adversary sees, other than Pr, is $M^A + r^A_{S',j}$ for one probe set S'. By Lemma 4.7 and Lemma 4.12, we have that all values of M^A are equally likely given what the adversary sees, other than Pr. Thus for all m_0, m_1, r, c, we have that $\Pr[\mathrm{adv}(m_0, r) = c] = \Pr[\mathrm{adv}(m_1, r) = c]$, from which the lemma follows. ■

The security of the protocol follows immediately from the Lemmas 4.9 and 4.13.

Theorem 4.14 *Let* $\delta > 0$ *and* $n > t$. *Then the Perfectly Private Transmission Protocol is* $(0, \delta)$*-secure when* $|\mathbf{F}| \ge \frac{1}{\delta}(n(m+1)^n + \frac{1}{2}mn^2)$.

4.4 Perfect Security

To complete our treatment of secure communications over multicast lines, we note that it is easy to achieve perfectly secure message transmission over $n > 2t$ multicast lines. The idea is to use the Private Propagation Protocol from §4.1 to simulate the protocol of [DDWY93] for $n > 2t$ simple lines.

Corollary 4.15 *If $n > 2t$, then there exists an efficient $(0, 0)$-secure message transmission protocol.*

Note that this protocol can also be used for almost perfect privacy with perfect reliability, so we have now addressed all combinations of reliability and privacy.

5 Secure Communication Without Multicast

In this section, we compare the multicast model to simple lines with and without broadcast. We say that there are n simple lines connecting sender and receiver if they are connected by n disjoint paths of single-receiver channels. In this model, it does not help the adversary to have more than one fault on any line. Each line is then either faulty or honest, and anything transmitted to one party on an honest line is guaranteed to have come from the other party and to be hidden from the adversary. (See [DDWY93] for a more detailed description of this model.) In addition, we say that there is broadcast if any party can send an authenticated message that will be received by all parties.

5.1 Simple Lines

Dolev et al. [DDWY93] showed that $2t + 1$ simple lines are necessary and sufficient for perfectly secure message transmission. We showed in §3.2 that, similarly, $2t+1$ multicast lines are necessary and sufficient for perfectly secure message transmission. However, as shown in §4.1, only $t + 1$ multicast lines are needed for almost perfectly secure message transmission. In contrast, we show in this section that the $2t+1$ bound in the simple lines model holds even for almost perfect security. Thus, multicast lines are strictly more powerful than simple lines alone when a small probability of failure is allowed, but are equivalent to simple lines if no failure is allowed.

Specifically, we show that $2t+1$ simple lines are required for reliable message transmission even if we allow a substantial probability of failure. It is easy to achieve 1/2-reliability when $n = 2t$: send M^A on all lines, and have B take a majority vote, where B uses a coin flip to break a t-to-t tie. The following theorem shows that it is not possible to do substantially better.

Theorem 5.1 *If $n \leq 2t$ and $\delta < \frac{1}{2}(1 - \frac{1}{|\mathcal{M}|})$, then δ-reliable message transmission over n simple lines is impossible.*

Proof: Let $n = 2t$, and let Π be a message transmission protocol from A to B. The set of all possible transcripts of B for Π is drawn from a probability distribution that depends on Pr, the coin flips C^A of A, the coin flips C^B of B, the choice of faulty lines by the adversary, and the other random choices of the adversary. Without loss of generality, we can assume that the protocol proceeds in phases, where A is silent during even phases and B is silent during odd phases (see [DDWY93]).

Suppose that the adversary behaves as follows. First, it chooses Pr to be the uniform distribution on $\mathcal{M}$. Then it chooses to disrupt either the first element on each of the first t lines or of the last t lines, according to the uniform distribution on two elements. Next, the adversary chooses a message $\widehat{M}^A \in \mathbf{F}$ according to the same probability distribution from which the actual message M^A was drawn, and also chooses a sequence of coin flips $\widehat{C}^A$ sufficiently long to simulate the behavior of A over the course of the protocol. During the first phase, the adversary simulates A for input message $\widehat{M}^A$ and coin flip sequence $\widehat{C}^A$, and puts the corresponding values on the faulty lines. During the second phase, the adversary prevents any transmission from B to A on the faulty lines. The transcript E_2^A of A after two phases includes its coin flips and the messages from B in the second phase on the honest lines. The transcript $\widehat{E}_2^A$ of the simulated A after two phases includes its simulated coin flips and the messages from B in the second phase on the faulty lines. In general, during phase $2i+1$, the adversary simulates the behavior of A with input message $\widehat{M}^A$, coin flip sequence $\widehat{C}^A$, and transcript $\widehat{E}_{2i}^A$. The appropriate messages are inserted on the faulty lines during phase $2i+1$. During phase $2i+2$, the adversary prevents any transmission from B to A on the faulty lines.

Given such an adversary, an execution is completely determined by $M^A, \widehat{M}^A$, the adversary's coin flip to choose the fault set, the prefix of C^A actually used by A, the prefix of C^B actually used by B, and the prefix of $\widehat{C}^A$ actually used by the adversary in its simulation. For some executions, B will halt and output a guess for M^B, based on its transcript and coin flips. Let $\mathcal{E}$ be the executions such that B halts and outputs $M^B = M^A$, where $M^A \neq \widehat{M}^A$, and let $E \in \mathcal{E}$. Suppose that E makes use of the first r^A bits of C^A, the first r^B bits of C^B and the first $\widehat{r}^A$ bits of $\widehat{C}^A$. Then let $\mathrm{sw}(E)$ be the execution where the values of M^A and $\widehat{M}^A$ are swapped, the adversary's choice of faulty lines is switched, and the prefixes of C^A and $\widehat{C}^A$ are swapped. Then, for any C^B, the transcript of B is identical for E and $\mathrm{sw}(E)$. Furthermore, the probability p_E that E occurs is the same as the probability $p_{\mathrm{sw}(E)}$ that $\mathrm{sw}(E)$ occurs: $p_E = p_{\mathrm{sw}(E)} = \Pr[M^A]\Pr[\widehat{M}^A]2^{-r^A-r^B-\widehat{r}^A-1}$. Thus $\Pr[B$ halts and $M^B = M^A | M^A \neq \widehat{M}^A] = \sum_{E\in\mathcal{E}} p_E = \sum_{E\in\mathcal{E}} p_{\mathrm{sw}(E)} \leq \Pr[B$ halts with $M^B = \widehat{M}^A | M^A \neq \widehat{M}^A] \leq \Pr[\text{protocol fails } | M^A \neq \widehat{M}^A]$. This implies that the probability that the protocol fails is at least $\frac{1}{2}\Pr[M^A \neq \widehat{M}^A] = \frac{1}{2}(1 - \frac{1}{|\mathcal{M}|})$. ■

5.2 Simple Lines with Broadcast

In this section, we show that simple lines with broadcast are equivalent to multicast lines in their connectivity requirements for secure communication.

5.2.1 Almost Perfect Security

There is a certain relationship between simple lines with broadcast and multicast lines. Specifically, anything done over simple lines can be simulated over multicast lines using the Private Propagation Protocol (§4.1), and vice versa. Similarly, broadcast and the Reliable Transmission Protocol (§3.1) have the same result. This allows translation of certain protocols from one setting to the other.

Corollary 5.2 *$(0, \delta)$-secure communication is possible over $n > t$ simple lines with broadcast.*

Proof: The Private Transmission Protocol can be directly translated into this setting, as follows. The Private Propagation step is done using the simple lines, and broadcast is used in place of the Reliable Transmissions. ■

Note that simple lines, as used for private propagation, have the same security properties as the Private Propagation Protocol, while the broadcast acts as a perfectly reliable transmission. This has the somewhat unintuitive effect that the translated Private Transmission Protocol achieves perfect *privacy* but is still only almost perfectly *reliable*, since an adversary can still disrupt the private propagation and cause the receiver to output the wrong message with nonzero probability.

Implications for Secure Multiparty Computation Corollary 5.2 can be used to strengthen the secure multiparty computation result of Rabin and Ben-Or [RB89]. In their setting, $n > 2t+1$ parties are connected by a complete graph of private, authenticated, single-receiver channels, and also any player can broadcast a message that will be received authentically by all players. The channel connectivity can be reduced to $t+1$, since the $(0,\delta)$-protocol from Corollary 5.2 can simulate the missing channels. The small probability δ that each simulation fails is not significant, since the protocol of Rabin and Ben-Or already has a negligible probability of failure. Indeed, this error is necessary, since error-free multiparty computation requires $3t+1$ connectivity [BGW88, CCD88, RB89].

Corollary 5.3 *Secure multiparty computation, with an arbitrarily small probability of error, is efficient over a* $(t+1)$*-connected network of at least* $2t+1$ *nodes in the private authenticated channels with broadcast model.*

5.2.2 Perfect Security

One might hope that the broadcast channel would allow us to break the $n > 2t$ connectivity requirement for perfect security. We show here that this is not the case. Together with Corollary 5.2, this shows that multicast lines and simple lines with broadcast are equivalent for secure communication.

Theorem 5.4 $(0,0)$*-secure message transmission over* n *simple lines with a broadcast channel is impossible if* $n \leq 2t$.

Proof: The proof follows [DDWY93]. Let $n = 2t$. We show that any protocol Π that achieves perfect reliability in the presence of active faults cannot achieve perfect privacy on its fault-free executions. A fault-free execution of Π is completely determined by the initial message and the coin flips used by A and B.

Let Z be the public channel together with any subset of t lines. It suffices to show that, for perfect reliability, the information on Z for every fault-free execution of the protocol must be consistent with at most one message in the support of Pr. Towards a contradiction, assume there exists $M_1 \neq M_2$ in the support of Pr, and coin flips $C_1^A, C_1^B, C_2^A, C_2^B$, such that the traffic on Z for fault-free executions $\Pi[M_1, C_1^A, C_1^B]$ and $\Pi[M_2, C_2^A, C_2^B]$ are identical.

Let the initial message be $M^A = M_1$, and let the coin flips of A and B be C_1^A and C_2^B respectively. Consider the following faulty execution of Π. The adversary controls all t lines not in Z. During the phases when A sends to B, the faulty lines send traffic consistent with the fault-free execution $\Pi[M_2, C_2^A, C_2^B]$. During the phases when B sends to A, the faulty lines send traffic consistent with the fault-free execution $\Pi[M_1, C_1^A, C_1^B]$. Then the view of A under these circumstances will be identical to the view of A for the fault-free execution $\Pi[M_1, C_1^A, C_1^B]$. Moreover, the view of B under these circumstances will be identical to the view of B for the fault-free execution $\Pi[M_2, C_2^A, C_2^B]$. Thus, for this faulty execution, A will halt while B will be unable to output $M^B = M^A$ with certainty. ■

6 Conclusions

We have considered the problem of secure communication over multicast lines. We have given a complete characterization of when it is possible to give a solution, and an almost complete characterization of when it is possible to give an efficient solution.

In addition, we compared multicast lines to the simple lines alone or with broadcast. We showed that all three models are of equivalent strength when the security is required to be perfect, while multicast lines and simple lines with broadcast are more powerful than simple lines alone when security need not be perfect. In particular, our results yield improved protocols for secure multiparty computation in a network of private authenticated channels with broadcast, reducing the necessary connectivity to $t+1$.

In all of the multicast protocols described in this paper, the multicast property is only *needed* to multicast values drawn from a uniform distribution. With simple modifications, the protocols would retain their security properties in a communication setting that had multicast lines for the first round and simple lines thereafter. This suggests that there may be a more fundamental communication "atom" than multicast for establishing secure communication with low connectivity.

A more general setting is a multicast graph, with a channel from each node to its neighborhood. If a graph has n disjoint paths whose neighborhoods are also disjoint, then our multicast lines protocols can be simulated on the multicast graph. However, if these n disjoint paths do not have disjoint neighborhoods, then an adversary may be able to foil our protocols with $t < n$ faults by using one fault to eavesdrop on two disjoint lines. An obvious direction of further research is to fully characterize secure communication in this more general setting.

References

[ABLP89] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg, "On the complexity of radio communication," *ACM STOC*, 274–285, (1989).

[Bea89] D. Beaver, "Multiparty protocols tolerating half faulty processors," *Proc. Crypto '89*, 560–572.

[BF97] A. Beimel and M. Franklin, "Reliable communication over partially authenticated networks," *Proc. Workshop on Distributed Algorithms (WDAG)*, Springer-Verlag LNCS 1320, 245–259, (1997).

[BCG93] M. Ben-Or, R. Canetti, and O. Goldreich, "Asynchronous secure computation," *ACM STOC*, 52–61, (1993).

[BGW88] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computing," *ACM STOC*, 1–10, (1988).

[BBR88] C. Bennett, G. Brassard, M. Robert, "Privacy amplification by public discussion," *SIAM J. Comput.* 17(2):210–229, (1988).

[BT85] G. Bracha and S. Toueg, "Asynchronous consensus and broadcast protocols", *JACM* 32(4): 824–840, (1985).

[BDK97] M. Burmester, Y. Desmedt, and G. Kabatianski, "Trust and Security: A New Look at the Byzantine Generals Problem", *Network Threats*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, Vol. 38, 75–83, (1997).

[CCD88] D. Chaum, C. Crepeau, and I. Damgard, "Multiparty unconditional secure protocols," *ACM STOC*, 11–19, (1988).

[Ch85] D. Cheriton and W. Zwaenepoel, "Distributed process group in the V kernel," *ACM Trans. on Computer Systems* 3, 77–107, (1985).

[Dol82] D. Dolev, "The Byzantine Generals strike again," *J. Algorithms* 3:14–30, (1982).

[DDWY93] D. Dolev, C. Dwork, O. Waarts, and M. Yung, "Perfectly secure message transmission," *JACM* 40(1):17–47, (1993).

[FY95] M. Franklin and M. Yung, "Secure hypergraphs: privacy from partial broadcast," *ACM STOC*, 36–44, (1995).

[GGL91] O. Goldreich, S. Goldwasser, and N. Linial, "Fault-tolerant computation in the full information model," *IEEE FOCS*, 447–457, (1991).

[PSL80] M. Pease, R. Shostak, and L. Lamport, "Reaching Agreement in the Presence of Faults", *JACM* 27(2): 228–234, (1980).

[PG89] F. M. Pittelli and H. Garcia-Molina, "Reliable Scheduling in a TMR Database System", *ACM Transactions on Computer Systems* 7(1): 25–60, (1989).

[Rab94] T. Rabin, "Robust sharing of secrets when the dealer is honest or faulty," *JACM* 41(6):1089–1109, (1994).

[RB89] T. Rabin and M. Ben-Or, "Verifiable secret sharing and multiparty protocols with honest majority," *ACM STOC*, 73–85, (1989).

[SA96] H. Sayeed and H. Abu-Amara, "Efficient perfectly secure message transmission in synchronous networks," *Information and Computation* 126(1):53–61, (1996).

[WD99] Y. Wang and Y. Desmedt, "Secure communication in broadcast channels: The answer to Franklin and Wright's question", *Proc. Eurocrypt '99*, 446–458.