# Secure Message Transmission with Small Public Discussion

Juan Garay* Clint Givens† Rafail Ostrovsky‡

October 26, 2009

**Abstract**

In the problem of Secure Message Transmission in the public discussion model (SMT-PD), a Sender wants to send a message to a Receiver privately and reliably. Sender and Receiver are connected by $n$ channels, up to $t < n$ of which may be maliciously controlled by a computationally unbounded adversary, as well as one public channel, which is reliable but not private.

The SMT-PD abstraction has been shown instrumental in achieving secure multi-party computation on sparse networks, where a subset of the nodes are able to realize a broadcast functionality, which plays the role of the public channel. However, the *implementation* of such public channel in point-to-point networks is highly costly and non-trivial, which makes minimizing the use of this resource an intrinsically compelling issue.

In this paper, we present the first SMT-PD protocol with *sublinear* (i.e., logarithmic in $m$, the message size) communication on the public channel. In addition, the protocol incurs a private communication complexity of $O(\frac{mn}{n-t})$, which, as we also show, is *optimal*. By contrast, the best known bounds in both public and private channels were linear. Furthermore, our protocol has an optimal round complexity of $(3, 2)$, meaning three rounds, two of which must invoke the public channel.

Finally, we ask the question whether some of the lower bounds on resource use for a single execution of SMT-PD can be beaten *on average* through amortization. In other words, if Sender and Receiver must send several messages back and forth (where later messages depend on earlier ones), can they do better than the naïve solution of repeating an SMT-PD protocol each time? We show that amortization can indeed drastically reduce the use of the public channel: it is possible to limit the total number of uses of the public channel to *two*, no matter how many messages are ultimately sent between two nodes. (Since two uses of the public channel are required to send any reliable communication whatsoever, this is best possible.)

**Key words:** Secure message transmission, information-theoretic security, almost-everywhere secure computation, randomness extractors.

## 1 Introduction

Dolev, Dwork, Waarts and Yung [DDWY93] introduced the model of *Secure Message Transmission* (SMT) in an effort to understand the connectivity requirements for secure communication in the information-theoretic setting. Generally speaking, an SMT protocol involves a sender, $\mathcal{S}$, who wishes to transmit a message $M$ to a receiver, $\mathcal{R}$, using a number $n$ of channels ("wires"), some of which are controlled by a malicious adversary $\mathcal{A}$. The goal is to send the message both *privately* and *reliably*. Since its introduction, SMT has been widely studied and optimized with respect to several different settings of parameters (for example—and non-exhaustively, see [SA96, SNP04, ACH06, FFGV07, KS08]).

*AT&T Labs – Research, 180 Park Ave., Florham Park, NJ 07932. Email: garay@research.att.com.

†Department of Mathematics, UCLA. Email: cgivens@math.ucla.edu. Work supported in part by NSF VIGRE Fellowship.

‡Department of Computer Science and Mathematics, UCLA. Email: rafail@cs.ucla.edu. Work supported in part by IBM Faculty Award, Xerox Innovation Group Award, OKAWA Research Award, NSF grants 0430254, 0716835, 0716389, 0830803, 0916574, BSF grant, and U.C. MICRO grant.

Garay and Ostrovsky [GO08] studied a model they called Secure Message Transmission *by Public Discussion* (SMT-PD) as an important building block for achieving secure multi-party computation [BGW88, CCD88] on sparse (i.e., not fully connected) networks. (An equivalent setup was studied earlier in a different context by Franklin and Wright [FW98].) In this model, in addition to the wires in the standard SMT formulation, called "common" or "private" wires from now on, $\mathcal{S}$ and $\mathcal{R}$ gain access to a *public* channel which the adversary can read but not alter. In this new setting, secure message transmission is achievable even if the adversary corrupts up to $t < n$ of the private wires—i.e., up to all but one.

The motivation for this abstraction comes from the feasibility in partially connected settings for a subset of the nodes in the network to realize a broadcast functionality despite the limited connectivity [DPPU86, Upf92, BG93][1], which plays the role of the public channel. (The private wires would be the multiple paths between them.) As such, the *implementation* of the public channel in point-to-point networks is costly and highly non-trivial in terms of rounds of computation and communication, as already the sending of a single message to a node that is not directly connected is simulated by sending the message over multiple paths, not just blowing up the communication but also incurring a slowdown factor proportional to the diameter of the network, and this is a process that must be repeated many times—linear in the number of corruptions for deterministic, error-free broadcast protocols (e.g., [GM98]), or expected (but high) constant for randomized protocols [FM97, KK06].

A main goal of this work is to minimize the use of this expensive resource, both in terms of communication as well as in the number of times it must be used when sender and receiver must send many messages back and forth, as it is the case in secure multi-party computation. We first present an SMT-PD protocol with a logarithmic (in $m$, the message size) communication complexity on the public channel; the best known bound, due to Shi, Jiang, Safavi-Naini, and Tuhin [SJST09], was linear (see related work below). In addition, our protocol incurs a private communication complexity of $O(\frac{mn}{n-t})$, which, as we also show, is *optimal*, thus providing an affirmative answer to the question posed in [SJST09] of whether the $O(n)$ private transmission rate could be improved. Furthermore, our protocol has an optimal round complexity of $(3, 2)$, meaning 3 rounds, 2 of which must invoke the public channel [SJST09].

Regarding the number of times the public channel must be used when considering SMT-PD as a subroutine in a larger protocol, we ask the question whether some of the lower bounds on resource use for a single execution of SMT-PD can be beaten *on average* through amortization. In other words, if a sender and receiver must send several messages back and forth (where later messages depend on earlier ones), can they do better than the naïve solution of repeating an SMT-PD protocol each time, incurring a cost of three rounds and two public channel transmissions per message? We show that amortization can in fact drastically reduce the use of the public channel: indeed, it is possible to limit the total number of uses of the public channel to *two*, no matter how many messages are ultimately sent between two nodes. (Since two uses of the public channel are required to send any reliable communication whatsoever, this is best possible.)

**Prior work.** The first variant of SMT considered in the literature is *perfectly secure message transmission* (PSMT), in which both privacy and reliability are perfect [DDWY93]. It is shown in the original paper that PSMT is possible if and only if $n \geq 2t + 1$. For such $n$, 2 rounds are necessary and sufficient for PSMT, while one-round PSMT is possible if and only if $n \geq 3t + 1$.

The communication complexity of PSMT depends on the number of rounds. For 1-round PSMT, Fitzi *et al.* [FFGV07] show that transmission rate $\geq \frac{n}{n-3t}$ is necessary and sufficient. (Recall that $n > 3t$ is required in this case.) For 2-round PSMT, Srinathan *et al.* [SNP04] show that a transmission rate $\geq \frac{n}{n-2t}$ is required[2]; this was extended in [SPR07], which showed that increasing the number of rounds does not help. Kurosawa and Suzuki [KS08] construct the first efficient (i.e., polynomial-time) 2-round PSMT protocol which matches this optimal transmission rate.

A number of relaxations of the perfectness requirements of PSMT are considered in the literature to

[1] Called "almost-everywhere" agreement, or broadcast, in this setting.

[2] The authors claim a matching upper bound as well, but this was shown to be flawed [ACH06].

achieve various tradeoffs (see for example [CPRS08] for a detailed discussion of variants of SMT). The most general version of SMT (or SMT-PD) is perhaps $(\epsilon, \delta)$-SMT. We call a protocol for SMT(-PD) an $(\epsilon, \delta)$-SMT(-PD) protocol provided that the adversary's advantage in distinguishing any two messages is at most $\epsilon$, and the receiver correctly outputs the message with probability $1 - \delta$. The lower bound $n \geq 2t + 1$ holds even in this general setting (at least for non-trivial protocols, such as those satisfying $\epsilon + \delta < 1/2$); hence the most interesting case for SMT-PD is the case when the public channel is required: $t < n \leq 2t$. As noted above, this requires round complexity (3,2) [SJST09]. Franklin and Wright [FW98] show that perfectly reliable ($\delta = 0$) SMT-PD protocols are impossible when $n \leq 2t$. On the other hand, perfect privacy ($\epsilon = 0$) is possible, and is achieved by previous SMT-PD constructions (see below).

The communication complexity lower bounds noted above all apply to PSMT; for more general SMT bounds, we are aware only of [KS07]. They consider the problem of *almost-secure message transmission*, which is only slightly less restrictive than PSMT. Namely, the problem requires perfect privacy, and that the Receiver *never* output an incorrect message, though he may output "failure" with probability $\delta$. The authors show that in this model, there is a communication complexity lower bound of $n(m + \log(1/\delta))$ (up to an additive constant).

A number of protocols for SMT-PD appear in previous work. The first such comes in [FW98] as a consequence of the equivalence shown there between networks with multicast and those with simple lines and broadcast (i.e., the public discussion model). Their solution has optimal round complexity $(3, 2)$[3]; however, when $t < n < \lceil \frac{3t}{2} \rceil$ (including the worst case $t = n + 1$), their protocol has (pick your poison) either positive privacy error $\epsilon > 0$, or *exponential* communication complexity. Garay and Ostrovsky [GO08] first describe a (4,3)-round $(0, \delta)$ protocol which was subsequently improved to (3,2) rounds. The protocol has linear transmission rate (in terms of message size) on the public and private channels. Shi *et al.* [SJST09] give the first protocol with constant transmission rate on the public channel (for messages of sufficient, modest size)[4], with linear transmission rate on the private channels as well; however, the *communication complexity* of their protocol is linear.

**Our contributions.** By contrast, we obtain the first round-optimal SMT-PD protocol with *sublinear* (logarithmic) communication complexity on the public channel. The protocol also enjoys a private communication complexity of $O\left(\frac{nm}{n-t}\right)$, which (just by itself) improves on previous constructions and, as we also show, is optimal. At a high level, the protocol has the same structure as previous 3-round SMT-PD protocols, with the following important differences: (1) our use of randomness extractors allows us to reduce the amount of transmitted randomness, which is reflected in the gain in private communication, and (2) typically in previous protocols the message is transmitted in the last round over the public channel, blinded by the private randomness thought not to have been tampered with; our improvement to public communication comes from the transmission of the (blinded) message on the *private* wires, provided that the sender *authenticates* the transmission making use of the public channel, which in turn requires smaller communication. Additionally, we achieve these improved communication bounds even for messages of smaller required size than Shi *et al.* [SJST09]. [5] Finally, the protocol achieves perfect privacy.

We arrive at this result through a series a transformations. First, we design a generic SMT-PD protocol with linear public communication and $O\left(\frac{nm}{n-t}\right)$ private communication (note that this already improves on existing results); second, we consider instantiations of the generic protocol's "black boxes" with different randomness extractors, each providing its own benefits (perfect privacy *vis-à-vis* smaller message size); and

[3]The round complexity is not apparent from the text, for two reasons: (1) The protocol is described in terms of the multicast model, not SMT-PD directly; and (2) the authors consider synchronous "rounds" not in the abstract SMT-PD model, but in the more concrete setting of nodes relaying messages in the underlying network.

[4]It is also claimed in [SJST09] that constant transmission rate on the public channel is optimal; as we show here, that is not the case.

[5]Specifically, [SJST09] require message size $m = \Omega(n^2(\log(1/\delta))^2)$, where we require only $m = \Omega(n(\log n + \log(1/\delta)) \log q)$, with $q \approx mn/(n-t)$.

last, we obtain the final protocol by essentially running two perfect-privacy instantiations of the generic protocol in parallel, one for the message itself and a "smaller" version for the authentication key. These results are presented in Section 3.

As noted above, we also show (Section 4) an $\Omega(\frac{nm}{n-t})$ lower bound on private communication. The lower bound holds for SMT without public discussion as well. The bound itself is weaker than previous, but it holds for a more general class of SMT protocols. In particular, it is the first communication complexity lower bound to consider non-perfect privacy, as well as the first to allow for the Receiver outputting an incorrect message.

Finally, we show in Section 5 how amortization can drastically reduce the use of the public channel, allowing sender and receiver to communicate *indefinitely* after using the public channel twice and a limited initial message. Our approach is to separate Sender and Receiver's interaction following the first execution of SMT-PD into two modes: a *Normal Mode* and a *Fault-Recovery Mode*. At a high level, in the Normal Mode, secure communication is successful provided the adversary does not interfere; this is implemented by a one-round protocol satisfying a relaxed version of the problem that we call *Weak* SMT-PD. Fault-Recovery Mode is entered if corruption is detected.[6]

Preliminaries and definitions are given in Section 2. For the purpose of readability, many of the proofs, as well as some complementary material, are presented in the appendix.

## 2 Model and Preliminaries

**Definition 2.1** *If $X$ and $Y$ are random variables over a discrete space $S$, the* statistical distance between $X$ and $Y$ *is defined to be*

$$\Delta(X,Y) \stackrel{def}{=} \frac{1}{2}\sum_{s\in S}\left|\Pr[X=s]-\Pr[Y=s]\right|.$$

*We say that $X$ and $Y$ are* $\epsilon$-close *if* $\Delta(X,Y)\leq\epsilon$.

**The public discussion model.** The *public discussion model* for secure message transmission [GO08] consists of a Sender $\mathcal{S}$ and Receiver $\mathcal{R}$ (PPTMs) connected by $n$ communication channels, or *wires*, and one *public channel*. $\mathcal{S}$ wishes to send a message $M_{\mathcal{S}}$ from message space $\mathcal{M}$ to $\mathcal{R}$, and to this end $\mathcal{S}$ and $\mathcal{R}$ communicate with each other in synchronous rounds in which one player sends information across the wires and/or public channel. Communication on the public channel is reliable but public; the common wires may be corrupted and so are not necessarily reliable or private.

$\mathcal{A}$ is a computationally unbounded adversary who seeks to disrupt the communication and/or gain information on the message. $\mathcal{A}$ may *adaptively* corrupt up to $t<n$ of the common wires (potentially all but one!). Corrupted wires are actively controlled by $\mathcal{A}$: he can eavesdrop, block communication, or place forged messages on them. Further, we assume $\mathcal{A}$ is *rushing*—in each round, he observes what is sent on the public channel and all corrupted wires before deciding what to place on corrupted wires, or whether to corrupt additional wires (which he then sees immediately).

An *execution* $E$ of an SMT-PD protocol is determined by the random coins of $\mathcal{S}$, $\mathcal{R}$, and $\mathcal{A}$ (which we denote $C_{\mathcal{S}}$, $C_{\mathcal{R}}$, $C_{\mathcal{A}}$ respectively), and the message $M_{\mathcal{S}}\in\mathcal{M}$. The *view of a player $\mathcal{P}\in\{\mathcal{S},\mathcal{R},\mathcal{A}\}$ in an execution $E$*, denoted $\mathrm{View}_{\mathcal{P}}$, is a random variable consisting of $\mathcal{P}$'s random coins and all messages received (or overheard) by $\mathcal{P}$. ($\mathcal{S}$'s view also includes $M_{\mathcal{S}}$). Additionally, let $\mathrm{View}_{\mathcal{P}}(M_0)$ denote the distribution on $\mathrm{View}_{\mathcal{P}}$ induced by fixing $M_{\mathcal{S}}=M_0$. In each execution, $\mathcal{R}$ outputs a received message $M_{\mathcal{R}}$, a function of $\mathrm{View}_{\mathcal{R}}$.

We can now define an $(\epsilon,\delta)$-SMT-PD protocol (cf. [FW98, GO08, SJST09]):

[6] Effectively, this is an instantiation in the SMT context of the "fast-track" approach (e.g., [Lam87, GRR98]), where if things are "smooth" then the algorithm or protocol performs very efficiently, reverting to a more punctilious mode otherwise.

**Definition 2.2** *A protocol $\Pi$ in the model above, in which $\mathcal{S}$ attempts to send a message $M_{\mathcal{S}}$ to $\mathcal{R}$, is* $(\epsilon, \delta)$-secure *(or simply, is an* $(\epsilon, \delta)$-SMT-PD protocol*) if it satisfies:*

PRIVACY*: For any two messages $M_0, M_1 \in \mathcal{M}$,* $\mathrm{View}_{\mathcal{A}}(M_0)$ *and* $\mathrm{View}_{\mathcal{A}}(M_1)$ *are $\epsilon$-close.*

RELIABILITY*: For all $M_{\mathcal{S}} \in \mathcal{M}$ and all adversaries $\mathcal{A}$, $\mathcal{R}$ should correctly receive the message with probability at least $1 - \delta$; i.e.,* $\Pr[M_{\mathcal{R}} = M_{\mathcal{S}}] \geq 1 - \delta$*. (The probability is taken over all players' random coins.)*

**Error-correcting codes and consistency checks for codewords.** For our purposes, the following definition of error-correcting codes is sufficient:

**Definition 2.3** *Given a finite alphabet $\Sigma$, an* error-correcting code $\mathcal{E}$ of minimum distance $d$ *is a pair of mappings $Enc : \Sigma^K \to \Sigma^N$, where $K < N$ and $Dec : \Sigma^N \to \Sigma^K$, such that (1) any two distinct elements $x, y$ in the image of $Enc$ (the* codewords*) have $dist(x, y) \geq d$ in the Hamming metric; (2) $Dec(Enc(x)) = x$ for all $x \in \mathbb{F}_q^K$.*[7] *We say $\mathcal{E}$ has* rate $K/N$ *and* relative minimum distance $d/N$.

We require a family of codes of increasing input length which is *asymptotically good*, that is, $\mathcal{E}$ should have *constant* rate and *constant* relative minimum distance $D$. See, e.g., [MS83] for a standard reference.

Of particular interest for us are the well-known Reed-Solomon codes over $F_q$, obtained by oversampling polynomials in $\mathbb{F}_q[X]$. Given an input in $\mathbb{F}_q^K$, we interpret it as a polynomial $f$ of degree $\leq K - 1$; to obtain a codeword from $f$, we simply evaluate it at $N$ distinct points in $\mathbb{F}_q$, for any $N > K$. Indeed, any two such polynomials agree on at most $K - 1$ points, therefore the Reed-Solomon code has minimum distance $N - K + 1$.

Our protocols make use of a simple method to probabilistically detect when codewords sent on the private wires are altered by $\mathcal{A}$. Simply put, the sender of the codeword reveals a small subset of the codeword symbols. Formally, suppose $\mathcal{S}$ sends a codeword $\mathcal{C} \in \Sigma^N$ to $\mathcal{R}$ over one of the private wires, and $\mathcal{R}$ receives the (possibly altered) codeword $\widetilde{\mathcal{C}}$. (If $\mathcal{R}$ receives a non-codeword, he immediately rejects it.) Then to perform the consistency check, $\mathcal{S}$ chooses a random set $J = \{j_1, j_2, \ldots, j_\ell\} \subset [N]$ and sends $(J, \mathcal{C}|_J)$ to $\mathcal{R}$, where $\mathcal{C}|_J$ represents the codeword $\mathcal{C}$ restricted to the indices in $J$. If the revealed symbols match, then the consistency check succeeds; otherwise the check fails and $\mathcal{R}$ rejects $\widetilde{\mathcal{C}}$ as tampered.

Suppose $\mathcal{A}$ alters $\mathcal{C}$ to a different codeword, $\widetilde{\mathcal{C}} \neq \mathcal{C}$. symbols. Therefore, the probability that they agree on a randomly chosen index is $\leq 2/3$, and so

$$\Pr[\mathcal{R} \text{ accepts } \widetilde{\mathcal{C}}] = \Pr[\mathcal{C}|_J = \widetilde{\mathcal{C}}|_J] \leq (2/3)^\ell.$$

Thus, with probability $\geq 1 - (2/3)^\ell$, $\mathcal{R}$ will reject a tampered codeword. Of course, the validity of the check depends upon $\mathcal{A}$ not knowing $J$ at the time of potential corruption of $\mathcal{C}$.

**Average min-entropy and average-case randomness extractors.** Recall that the *min-entropy* of a distribution $X = (X_1, \ldots, X_N)$ over $\{0,1\}^N$ is defined as

$$H_\infty(X) = \min_x \left(-\log\left(\Pr[X = x]\right)\right),$$

and gives a measure of the amount of randomness "contained" in a weakly random source. We say a distribution $X$ is a *$k_{min}$-source* if $H_\infty(X) \geq k_{min}$.

A *(seeded)* $(N, M, k_{min}, \epsilon)$*-strong extractor* is a (deterministic) function

$$\mathrm{Ext} : \{0,1\}^N \times \{0,1\}^D \to \{0,1\}^M$$

[7] Note in particular that this allows us to test for membership in the image $Enc(\Sigma^K)$ by first decoding and then re-encoding.

such that for *any* $k_{min}$-source $X$, the distribution $U_D \circ \mathrm{Ext}(X, U_D)$ is $\epsilon$-close to $U_D \circ U_M$ (where $U_k$ represents the uniform distribution on $\{0,1\}^k$). The input to the extractor is the $N$-bit $k_{min}$-source, $X$, together with a truly random seed $s$, which is uniformly distributed over $\{0,1\}^D$. Its output is an $M$-bit string which is statistically close to uniform, *even conditioned on the seed $s$ used to generate it.*

This notion of min-entropy, and of a general randomness extractor, may be an awkward fit when considering an adversary with side information $Y$ as above. In these cases, a more appropriate measure may be found in the *average min-entropy* of $X$ given $Y$, defined in [DORS08] by

$$\widetilde{H}_\infty(X|Y) = -\log\left(\mathbb{E}_{y \leftarrow Y}\left[\max_x \Pr\left[X = x | Y = y\right]\right]\right).$$

Note that this definition is based on the *worst-case* probability for $X$, conditioned on the *average distribution* (as opposed to worst-case probability) of $Y$. The rationale is that $Y$ is assumed to be outside of the adversary's control; however, once $Y$ is known, the adversary then predicts the *most likely* $X$, given that particular $Y$.

[DORS08] use average min-entropy to define an object closely related to extractors: A *(seeded) average-case* $(N, M, k_{min}, \epsilon)$*-strong extractor* is a (deterministic) function

$$\mathrm{Ext} : \{0,1\}^N \times \{0,1\}^D \to \{0,1\}^M$$

such that the distribution of $(U_D \circ \mathrm{Ext}(X, U_D), I)$ is $\epsilon$-close to $(U_D \circ U_M, I)$, whenever $(X, I)$ is jointly distributed pair satisfying $\widetilde{H}_\infty(X|I) \geq k_{min}$. The similarity to an ordinary extractor is clear. [DORS08] prove the following fact about average min-entropy:

**Fact 2.4** *If $Y$ has at most $2^\ell$ possible values, then $\widetilde{H}_\infty(X|(Y,Z)) \geq \widetilde{H}_\infty(X|Z) - \ell$.*

**Extracting randomness from $\mathbb{F}_q$.** We will make use of a special-purpose *deterministic* (seedless) extractor $\mathrm{Ext}_q$ which operates at the level of field elements in $\mathbb{F}_q$ as opposed to bits.

$\mathrm{Ext}_q$ works not on general min-entropy sources, but on the restricted class of *symbol-fixing sources*, which are strings in $\mathbb{F}_q^N$ such that some subset of $K$ symbols is distributed independently and uniformly over $\mathbb{F}_q$, while the remaining $N - K$ symbols are fixed. Given a sample from any such source, $\mathrm{Ext}_q$ outputs $K$ field elements which are uniformly distributed over $\mathbb{F}_q^K$.

$\mathrm{Ext}_q$ works as follows: Given $\alpha \in \mathbb{F}_q^N$, construct $f \in \mathbb{F}_q[X]$ of degree $\leq N-1$, such that $f(i) = \alpha_i$ for $i = 0, \ldots, N-1$. Then $\mathrm{Ext}_q(\alpha) = (f(N), f(N+1), \ldots, f(N+K-1))$. (Of course we require $N+K \leq q$.) This extractor has proven useful in previous SMT protocols as well (see, e.g., [ACH06, KS08]).

# 3 SMT-PD with Small Public Discussion

In this section we present our main positive results. First, we construct a basic $(\epsilon, \delta)$-SMT-PD protocol, $\Pi_{\mathrm{Gen}}$ (for "generic"), with optimal private communication and linear public communication. We then consider possible instantiations of $\Pi_{\mathrm{Gen}}$; using, in particular, Reed-Solomon codes and the extractor $\mathrm{Ext}_q$, improves it to a 0-private protocol. Finally, we use $\Pi_{\mathrm{Gen}}$ (instantiated with Reed-Solomon codes) as a building block to construct our main protocol $\Pi_{\mathrm{SPD}}$, which achieves *logarithmic* public communication while maintaining optimal private communication (and other desirable properties).

## 3.1 A generic protocol with optimal private communication

Protocol $\Pi_{\mathrm{Gen}}$ achieves essentially optimal communication complexity on the private wires of $O(\frac{mn}{n-t})$, where $m$ is the length of the message, while maintaining linear communication complexity on the public channel. (See Section 4 for a precise statement of the lower bound.) This is the first SMT-PD protocol to

achieve sublinear transmission rate on the private wires, and as such provides an affirmative answer to the question posed in [SJST09] of whether $O(n)$ private-wire transmission rate can be improved.

$\Pi_{\mathrm{Gen}}$ relies on two primitives as black boxes: an error-correcting code $\mathcal{E}$ and an average-case strong extractor, $\mathrm{Ext}_A$. The efficiency of the protocol depends on the interaction between the basic parameters of the protocol—$\epsilon$, $\delta$, $m$, $n$, and $t$—and the parameters of $\mathcal{E}$ and $\mathrm{Ext}_A$. After presenting the protocol and proving its security, we will examine its complexity in terms of these parameters.

At a high level, the protocol has the same structure as previous 3-round SMT-PD protocols: (1) in the first round, one of the parties (in our case $\mathcal{R}$) sends lots of randomness on each private wire; (2) using the public channel, $\mathcal{R}$ then sends checks to verify the randomness sent in (1) was not tampered with; (3) $\mathcal{S}$ discards any tampered wires, combines each remaining wire's randomness to get a one-time pad $R$, and sends $C = M \oplus R$ on the public channel. However, our use of extractors allows us to reduce the amount of transmitted randomness, which is reflected in the gain in private communication.

We remark that one may modify $\Pi_{\mathrm{Gen}}$ to have interaction order $\mathcal{S}$-$\mathcal{R}$-$\mathcal{S}$, instead of $\mathcal{R}$-$\mathcal{R}$-$\mathcal{S}$ as we present it. One advantage of $\mathcal{R}$-$\mathcal{R}$-$\mathcal{S}$ is that when instantiated with deterministic extractors (see below), it does not require any random coins for $\mathcal{S}$ (in contrast to $\mathcal{S}$-$\mathcal{R}$-$\mathcal{S}$, where both parties use randomness crucially).

Now we turn to the details of protocol $\Pi_{\mathrm{Gen}}$. Let error-correcting code $\mathcal{E}$ have encoding and decoding functions $Enc : \{0,1\}^K \to \{0,1\}^N$ and $Dec : \{0,1\}^N \to \{0,1\}^K$, respectively, and relative minimum distance $D$. (We will specify $K$ below.) While $N > K$ may be arbitrarily large for the purpose of correctness, we will want $K/N$ and $D$ both to be constant for our complexity analysis—such codes are called *asymptotically good* .

Second, let $\mathrm{Ext}_A$ be an average-case $(nK, m, k_{min}, \epsilon/2)$-strong extractor. Here $K$ is, as above, the source length of the error-correcting code $\mathcal{E}$, and $m$ and $\epsilon$ are the message-length and privacy parameters of $\Pi_{\mathrm{Gen}}$. $k_{min}$ is the min-entropy threshold. Now clearly $m \le k_{min} \le nK$. On the other hand, we require $k_{min} = O(m)$ for our complexity claim to hold—that is, $\mathrm{Ext}_A$ should extract a constant fraction of the min-entropy. Further, the extractor's seed length $s$ should be $O(n+m)$.

Finally, let $b = \frac{1}{1-D}$, and then set $\ell = \log_b(t/\delta)$. Now with foresight, we set $K = \lceil k_{min}/(n-t) \rceil + \ell$.[8] Note that if $D$ and $k_{min}$ are constant and $k_{min} = O(m)$, then $K = O(m)/(n-t) + \ell$. The protocol, $\Pi_{\mathrm{Gen}}$, is presented in Fig. 1.

**Theorem 3.1** *Let $t < n$. Protocol $\Pi_{\mathrm{Gen}}$ is a $(3,2)$-round $(\epsilon,\delta)$-SMT-PD protocol with communication complexity $O(\frac{mn}{n-t})$ on the private wires provided that $m/(n-t) = \Omega(\ell)$, and $\max(O(\ell(n+\log m)), O(m+n))$ on the public channel, provided only that $m = \Omega(\ell)$.*

*Proof.* **Privacy.** We first claim that if we omit $C$, then $\mathcal{A}$ has essentially no information (up to $\epsilon$) on $\mathcal{S}$'s output of the average-case extractor, $\widetilde{R} = \mathrm{Ext}_A(\widetilde{\alpha}, seed)$. Formally:

**Claim 3.1** *The distribution $(U_s, \widetilde{R}, \mathrm{View}_{\mathcal{A}} \setminus C)$ is $\epsilon/2$-close to $(U_s, U_m, \mathrm{View}_{\mathcal{A}} \setminus C)$.*

(Proof in the appendix.) The remainder of the proof of $\epsilon$-privacy is by contradiction: We show that, if there exists an adversary $\mathcal{A}$ and messages $M_0, M_1$ such that $\Delta(\mathrm{View}_{\mathcal{A}}(M_0), \mathrm{View}_{\mathcal{A}}(M_1)) > \epsilon$, then there exists a distinguisher $\mathcal{D}$ which can distinguish $(U_s, \widetilde{R}, \mathrm{View}_{\mathcal{A}} \setminus C)$ from $(U_s, U_m, \mathrm{View}_{\mathcal{A}} \setminus C)$, in contradiction to the above claim.

So suppose such an $\mathcal{A}$, $M_0$, $M_1$ exist. Then there exists a distinguisher $\mathcal{D}_0$ which satisfies

$$|\Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_0)) = 1] - \Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_1)) = 1]| > \epsilon$$

In particular it follows that either

$$(1) \qquad \left|\Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_0)) = 1] - \Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_\$)) = 1]\right| > \epsilon/2$$

[8] As a sanity check, observe that $k_{min} \le nK = n(k_{min}/(n-t) + \ell)$, so the extractor we define can exist.

**Protocol** $\Pi_{\mathrm{Gen}}(\epsilon, \delta, m, n, t, \mathcal{E}, \mathrm{Ext}_A)$

1. ($\mathcal{R} \overset{PRI}{\rightarrow} \mathcal{S}$). For each wire $i$, $\mathcal{R}$ chooses a random $r_i \in \{0,1\}^K$ and sends the codeword $\mathcal{C}_i = Enc(r_i)$ along wire $i$. Let $\widetilde{\mathcal{C}}_i$ be the codeword received by $\mathcal{S}$, and $\widetilde{r}_i = Dec(\widetilde{\mathcal{C}}_i)$.

2. ($\mathcal{R} \overset{PUB}{\rightarrow} \mathcal{S}$). $\mathcal{R}$ chooses a random subset $J = \{j_1, j_2, \ldots, j_\ell\} \subset [N]$ of codeword indices, $|J| = \ell$. Let

$$\mathcal{C}_i|_J = (\mathcal{C}_{i,j_1}, \mathcal{C}_{i,j_2}, \ldots, \mathcal{C}_{i,j_\ell}) \in \{0,1\}^\ell$$

be the codeword $\mathcal{C}_i$ restricted to the indices of $J$. $\mathcal{R}$ sends $(J, \{\mathcal{C}_i|_J\}_{i\in[n]})$ to $\mathcal{S}$ over the public channel.

3. ($\mathcal{S} \overset{PUB}{\rightarrow} \mathcal{R}$). $\mathcal{S}$ rejects any wire $i$ which is syntactically incorrect (including the case that $\widetilde{\mathcal{C}}_i$ is not a valid codeword), or for which $\mathcal{C}_i|_J$ conflicts with $\widetilde{\mathcal{C}}_i$. Call the set of remaining, accepted wires $\mathsf{ACC}$, and let $B \in \{0,1\}^n$, where $b_i = 1 \iff i \in \mathsf{ACC}$.

   Let $\widetilde{\alpha}$ denote the concatenation of $\widetilde{r}_i$ for all $i \in \mathsf{ACC}$, padded with zeroes so that $|\widetilde{\alpha}| = nK$. $\mathcal{S}$ chooses $seed \in \{0,1\}^s$ uniformly at random. He applies $\mathrm{Ext}_A : \{0,1\}^{nK} \times \{0,1\}^s \rightarrow \{0,1\}^m$ to obtain $\widetilde{R} = \mathrm{Ext}_A(\widetilde{\alpha}, seed)$, where $|\widetilde{R}| = m$. $\mathcal{S}$ puts $C = M_{\mathcal{S}} \oplus \widetilde{R}$, and sends $(B, C, seed)$ on the public channel.

   **Receiver:** $\mathcal{R}$ uses $B$ to reconstruct $\mathsf{ACC}$. He forms $\alpha$ by concatenating $r_i$ for each $i \in \mathsf{ACC}$, and padding with zeroes to size $nK$. He applies $\mathrm{Ext}_A : \{0,1\}^{nK} \rightarrow \{0,1\}^m$, obtaining $R = \mathrm{Ext}_A(\alpha, seed)$. He then recovers $M_{\mathcal{R}} = C \oplus R$.

Figure 1: *A generic SMT-PD protocol with optimal communication complexity on the private wires and linear communication complexity on the public channel.*

or

$$(2) \qquad \left| \Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_\$)) = 1] - \Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_1)) = 1] \right| > \epsilon/2.$$

Here $\mathrm{View}_{\mathcal{A}}(M_\$)$ denotes the random variable obtained by first sampling $M_\$$ uniformly from $\{0,1\}^m$, and then sampling from $\mathrm{View}_{\mathcal{A}}$ conditioned on $M_{\mathcal{S}} = M_\$$. (If the probability distribution on $\mathcal{M}$ is uniform, then the distribution of $\mathrm{View}_{\mathcal{A}}(M_\$)$ is identically that of $\mathrm{View}_{\mathcal{A}}$, but we do not assume this here.)

Without loss of generality, we assume case (1) above holds. Now we describe $\mathcal{D}$, which uses $\mathcal{D}_0$ as a black box in order to distinguish $(U_s, \widetilde{R}, \mathrm{View}_{\mathcal{A}} \setminus C)$ and $(U_s, U_m, \mathrm{View}_{\mathcal{A}} \setminus C)$. First, the challenger flips a coin. On heads, he samples $u \leftarrow (U_s, \widetilde{R}, \mathrm{View}_{\mathcal{A}} \setminus C)$, and on tails, $u \leftarrow (U_s, U_m, \mathrm{View}_{\mathcal{A}} \setminus C)$. In either case he obtains $u = (u_s, u_{test}, u_{view})$ which he passes on to $\mathcal{D}$. $\mathcal{D}$ forms $C_{\mathcal{D}} = M_0 \oplus u_{test}$, which plays the role of $C$ in the protocol. He passes $u_{view} \cup C_{\mathcal{D}}$ to $\mathcal{D}_0$, which returns a bit $b$ representing its guess that $u_{view} \cup C_{\mathcal{D}}$ was sampled from $\mathrm{View}_{\mathcal{A}}(M_b)$. If $b = 0$, then $\mathcal{D}$ outputs a guess of "heads" (i.e., guesses $u_{test}$ was sampled from $\widetilde{R}$), otherwise $\mathcal{D}$ guesses "tails" ($u_{test}$ was sampled from $U_m$).

Now consider the success probability of $\mathcal{D}$ when the challenger flips heads, so that $u_{test} \sim \widetilde{R}$. In this case, $C_{\mathcal{D}} = M_0 \oplus \widetilde{R}$ is obtained exactly as in $\Pi_{\mathrm{Gen}}$, and therefore $u_{view} \cup C_{\mathcal{D}}$ is distributed identically with $\mathrm{View}_{\mathcal{A}}(M_0)$. Thus $\Pr[\mathcal{D}(u) = 1|\text{heads}] = \Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_0)) = 1]$. Alternatively, suppose the challenger flips tails, and $u_{test}$ is uniform. Then $C_{\mathcal{D}} = M_0 \oplus u_{test}$ is uniform, which is also the distribution of $C$ if we choose $M = M_{\mathcal{S}}$ uniformly at random. Thus $\Pr[\mathcal{D}(u) = 1|\text{tails}] = \Pr[D_0(\mathrm{View}_{\mathcal{A}}(M_\$)) = 1]$. Putting these together, we discover

$$\begin{aligned} &\left| \Pr[\mathcal{D}(U_s, \widetilde{R}, \mathrm{View}_{\mathcal{A}} \setminus C) = 1] - \Pr[\mathcal{D}(U_s, U_m, \mathrm{View}_{\mathcal{A}} \setminus C) = 1] \right| \\ &\qquad = \left| \Pr[\mathcal{D}_0(\mathrm{View}_{\mathcal{A}}(M_0)) = 1] - \Pr[\mathcal{D}_0(\mathrm{View}_{\mathcal{A}}(M_\$)) = 1] \right| > \epsilon/2, \end{aligned}$$

which contradicts the above claim. This completes the verification of $\epsilon$-privacy.

**Reliability.** Observe that $M_{\mathcal{R}} = C \oplus R$ and $M_{\mathcal{S}} = C \oplus \widetilde{R}$. Therefore,

$$\begin{aligned}\mathcal{R} \text{ fails to decode correctly } (M_{\mathcal{R}} \neq M_{\mathcal{S}}) &\iff \mathrm{Ext}(\alpha, seed) = R \quad \neq \quad \widetilde{R} = \mathrm{Ext}(\widetilde{\alpha}, seed) \\ &\implies \alpha \neq \widetilde{\alpha} \\ &\implies \exists i \in \mathsf{ACC} \text{ s.t. } r_i \neq \widetilde{r}_i \\ &\implies \exists i \in \mathsf{ACC} \text{ s.t. } \mathcal{C}_i \neq \widetilde{\mathcal{C}}_i.\end{aligned}$$

The latter event only happens if $\mathcal{A}$ succeeds in altering $\mathcal{C}_i$ without $\mathcal{S}$ detecting it. By construction, our consistency check (Section 2) guarantees that this happens with probability at most $(1-D)^\ell = \delta/t$ for a single wire, hence (taking a union bound over corrupt wires) probability at most $\delta$ overall. Consequently, $\Pr[M_{\mathcal{R}} = M_{\mathcal{S}}] \geq 1 - \delta$.

**Complexity.** The private wires are used only in round 1, to send $Enc(r_i)$ on each wire. The total complexity is therefore $nN = O(nK)$ (for $\mathcal{E}$ of constant rate). As noted above, our assumptions on $\mathcal{E}$ and $\mathrm{Ext}_A$ imply that $K = O(m/(n-t) + \ell)$, and therefore the total private wire complexity is $O(mn/(n-t) + n\ell)$, which is $O(mn/(n-t))$ provided $m/(n-t) = \Omega(\ell)$.

The public channel is used in Rounds 2 and 3. In Round 2, $\mathcal{R}$ transmits $J \subset [N]$ of size $\ell$, and the restricted codewords $\mathcal{C}_i|_J$, at total cost $\ell n + \ell \log N = \ell n + \ell(\log K + O(1)) = \ell n + O(\ell(\log(m/(n-t)+\ell)))$. Provided that $m = \Omega(\ell)$, this is $O(\ell(n + \log m))$.

In Round 3, $\mathcal{S}$ uses the public channel to send $(B, C, seed)$ where $B$ indicates accepted wires, $C$ hides the message $M_{\mathcal{S}}$, and $seed$ is a seed for $\mathrm{Ext}_A$. Thus the Round 3 public communication is $n + m + s$, which is $O(n + m)$ for any extractor with reasonable seed length. □

## 3.2 Instantiating the generic protocol

Here we consider possible instantiations of $\Pi_{\mathrm{Gen}}$. Since our main interest is in 0-private protocols, the most important instantiation will be that with Reed-Solomon codes and the extractor $\mathrm{Ext}_q$ of Section 2. Nevertheless, other choices of (explicit) extractor are possible, and we examine one such in particular.

**Kamp and Zuckerman's symbol-fixing extractor.** The first extractor we suggest is the deterministic symbol-fixing extractor of Kamp and Zuckerman [KZ06, Theorem 1.3]. This extractor, like $\mathrm{Ext}_q$, works for the class of symbol-fixing sources; here the symbols come from an alphabet of constant size $d \geq 3$. It is an efficiently computable $(n, m, O(m + 1/\epsilon), \epsilon)$-extractor (the $O$ hides constants depending on $d$); it extracts a constant fraction of min-entropy and has output exponentially close to uniform. Additionally, it works for sources of *any* min-entropy rate—that is, $k_{min}$ has no dependence on $n$. It also has the advantage of being deterministic, thus obviating the need for $\mathcal{S}$ to choose and send a seed in Round 3. In this case, $\mathcal{S}$ *does not require any random coins at all*.

To convert the extractor of Kamp and Zuckerman into an average-case extractor as in Section 2, we may invoke the following fact, proven in [DORS08]:

**Fact 3.2** *For any $\gamma > 0$, if $\mathrm{Ext} : \{0,1\}^N \times \{0,1\}^D \to \{0,1\}^M$ is an $(N, M, k_{min}, \epsilon)$-strong extractor, then $\mathrm{Ext}$ is an $(N, M, k_{min} + \log(1/\gamma), \epsilon + \gamma)$-strong average-case extractor.*

Taking $\gamma = \epsilon$, we obtain a $2\epsilon$-extractor, while the additional additive error of $\log(1/\epsilon)$ is absorbed into the $O(\log(1/\epsilon))$-term already appearing in Kamp and Zuckerman's extractor. Provided that $\epsilon = \Omega(2^{-cm})$, we have $\log(1/\epsilon) = O(m)$, and so the new min-entropy satisfies $k'_{min} = O(m)$, as required in the complexity analysis.

We note that since the extractor of [KZ06] works on alphabets of fixed constant size, it is able to achieve optimal communication complexity for messages of size $\Omega(n\ell)$, as in $\Pi_{\mathrm{Gen}}$. In contrast, the instantiation with Reed-Solomon has a dependence on the field size $\log q \approx \log(mn/(n-t))$ as well.

**Reed-Solomon codes and the extractor $\mathrm{Ext}_q$.** Statistical error is a feature of all general-purpose randomness extractors. To get around it, we can exploit the fact that the sources arising from $\Pi_{\mathrm{Gen}}$ are not general min-entropy sources. Rather, conditioning on the adversary's view, each good wire carries independent, uniform randomness, and the corrupt wires carry fixed values. Thus the source we are interested in actually carries quite a great deal of structure. In particular, we may view it as a symbol-fixing source as described in Section 2, since we may group bits into symbols, and the adversary has no information on the symbols carried by good wires.

Consider an instantiation of $\Pi_{\mathrm{Gen}}$ using the extractor $\mathrm{Ext}_q : \mathbb{F}_q^{kN} \to \mathbb{F}_q^r$ of Section 2, which is indeed errorless. (Here $r = m/\log q$ is the size of $M_{\mathcal{S}}$ in field elements.) $\mathrm{Ext}_q$ is, according to our notation, a $(kN, r, r, 0)$ extractor for sources over $\mathbb{F}_q$: It extracts 100% of the randomness from its input with no statistical error. (It is also deterministic, hence trivially strong.) Since $\mathrm{Ext}_q$ operates at the level of field elements, Reed-Solomon codes are a natural choice for the error-correcting code $\mathcal{E}$ of $\Pi_{\mathrm{Gen}}$. We choose $\mathcal{E}$ to be $\mathrm{Ext}_q : \mathbb{F}_q^K \to \mathbb{F}_q^{2K}$, which has relative minimum distance $1/2$.

We now describe two requirements imposed by this instantiation. First, the description of $\Pi_{\mathrm{Gen}}$ assumes an extractor which operates on bits rather than field elements. This presents no real problem, as all statements can be recast in a straightforward way to this new setting. However, as mentioned above, the move from $\{0,1\}$ to $\mathbb{F}_q$ does have the effect of adding a $\log q$ term to the message size required for optimal communication complexity (see statement of and complexity analysis for Theorem 3.2 ).

Second, we must specify the appropriate field size $q$ in terms of the basic parameters $m, n, t, \delta$. Recall $\ell = \log(t/\delta)$. We require (with foresight):

$$q \log q = \Omega(mn/(n-t)) \qquad \text{and} \qquad (q - 2\ell)\log q > \frac{2m}{n-t}.$$

Thus $M_{\mathcal{S}} \in \mathbb{F}_q^r$, where $r = m/\log q$.

For the proof of privacy, we require $\mathrm{Ext}_q : \mathbb{F}_q^{nK} \to \mathbb{F}_q^r$ is in fact a perfect randomness extractor—so we need $q \geq nK + r$. Since $K = r/(n-t) + \ell$, we have (using $m = \Omega(n\ell)$):

$$\begin{aligned} nK + r &= n \cdot (\frac{r}{n-t} + \ell) + r = r(\frac{n}{n-t} + 1) + n\ell \\ &= \frac{m}{(\log q)} \cdot \frac{n}{n-t} + O(m) = O(\frac{m}{(\log q)} \cdot \frac{n}{n-t}). \end{aligned}$$

Thus, for $q \geq nK + r$ it suffices that $q \log q = \Omega(mn/(n-t))$, which is our first assumption on $q$.

Now observe that in order for our codeword authentication to be valid, we need $q \geq 2K = 2r/(n-t) + 2\ell$. Thus we require:

$$\begin{aligned} q \geq 2r/(n-t) + 2\ell &\iff q \geq \frac{2m}{(\log q)(n-t)} + 2\ell \\ &\iff q \log q \geq \frac{2m}{n-t} + 2\ell \log q \\ &\iff (q - 2\ell)\log q \geq \frac{2m}{n-t}, \end{aligned}$$

which gives our second condition on $q$.

### 3.3 A protocol with logarithmic public communication

In this section we present a protocol for SMT-PD which is the first to achieve logarithmic communication complexity (in $m$) *on the public channel*. The protocol is perfectly private, achieves the optimal communication complexity of $O(\frac{mn}{n-t})$ on the private wires, and has optimal round complexity of $(3, 2)$.

In its Round 3 communication, $\Pi_{\mathrm{Gen}}$ incurs a cost of size $m$ on the public channel, which we wish to reduce to $O(\log m)$. Our improvement comes from the insight that $\mathcal{S}$ can send the third-round message ($C$,

in the notation of $\Pi_{\text{Gen}}$) on the *common* wires, provided that $\mathcal{S}$ *authenticates* the transmission (making use of the public channel).

$\mathcal{S}$ could simply send $C$ on every common wire and authenticate $C$ publicly. The downside of this approach is that the private wire complexity would be $\Omega(mn)$ rather than $O(\frac{mn}{n-t})$—no longer optimal. Our solution is to take $C$ and encode it *once again* using Reed-Solomon into shares $C_1, \ldots, C_n$, each of size $\approx \frac{m}{n-t}$, such that any $n-t$ correct $C_i$'s will reconstruct $C$. $\mathcal{S}$ then sends $C_i$ on wire $i$, and authenticates each $C_i$ publicly.

This authentication uses a short secret key, $\widetilde{s}$, of size $\ell(n+\log(\frac{cm}{n-t}))$ (which is the cost of authenticating $n$ messages of size $cm/(n-t)$, using the consistency check of Section 2; $c$ is an absolute constant defined below). Thus, $\mathcal{S}$ and $\mathcal{R}$ will run two processes in parallel: a "small" strand, in which $\mathcal{S}$ privately sends the short key to $\mathcal{R}$; and a "big" strand, in which $\mathcal{S}$ sends $M_{\mathcal{S}}$ to $\mathcal{R}$, making use of the shared key in the third round. The small protocol sends the short key using any reasonably efficient SMT-PD protocol; for ease of exposition, we use $\Pi_{\text{Gen}}$, instantiated with Reed-Solomon codes. We also use $\Pi_{\text{Gen}}$ with Reed-Solomon codes for the big strand of the protocol in order to achieve perfect privacy and optimal private wire complexity.

We now describe the protocol in detail. Many of the parameters are the same as in (the Reed-Solomon instantiation of) $\Pi_{\text{Gen}}$: We set $\ell = \log(t/\delta)$, and fix a prime $q$ such that

$$q \log q = \Omega(mn/(n-t)) \qquad \text{and} \qquad (q-2\ell)\log q \geq \frac{2m}{n-t}.$$

The message space is $\mathcal{M} = \mathbb{F}_q^r$, that is, an $m$-bit message is considered as a sequence of $r = \lceil m/\log q \rceil$ field elements in $\mathbb{F}_q$. (However, we also assume, for the purpose of the Round 3 authentication, that the field elements are actually *represented* as bit-strings of length $r \log q$.) Set $K = \lceil r/(n-t) \rceil + \ell$ and $N = 2K$.

In addition to the above parameters, we will also define their small-strand counterparts, which we notate using variables with hats. Set $\hat{m} = \ell(n+\log(cK \log q))$—as noted above, this is the size of the shared secret which will be used to authenticate the $C_i$'s. Here the constant $c > 1$ is the expansion factor of an efficiently computable, constant-rate error-correcting code $\mathcal{E}'$ of relative minimum distance (say) $1/3$. (We caution that $\mathcal{E}'$ plays a different role in $\Pi_{\text{SPD}}$ than $\mathcal{E}$ did in $\Pi_{\text{Gen}}$, hence the different name.) We will use $Enc$ and $Dec$ to denote the encoding and decoding functions of $\mathcal{E}'$; we use $Enc_{RS}$ and $Dec_{RS}$ for the encoding and decoding functions of the Reed-Solomon code which functions as $\mathcal{E}$ for $\Pi_{\text{SPD}}$.

Fix $\hat{q}$ to be a prime such that

$$\hat{q} \log \hat{q} = \Omega(\frac{\hat{m}n}{n-t}) \qquad \text{and} \qquad (\hat{q}-2\ell)\log \hat{q} > \frac{2\hat{m}}{n-t},$$

Set $\hat{r} = \hat{m}/\log \hat{q}$, $\hat{K} = \hat{r}/(n-t) + \ell$, and $\hat{N} = 2\hat{K}$. Finally, set $\ell_{3/2} = \log_{3/2}(t/\delta)$.

The protocol, $\Pi_{\text{SPD}}$ (for "small public discussion"), is shown in Figure 2. Keep in mind the high-level understanding of the protocol: The first two rounds are simply parallel versions of Rounds 1 and 2 of $\Pi_{\text{Gen}}$, run with different (big and small) parameters. In Round 3, we complete the small instance of $\Pi_{\text{Gen}}$ as usual, and use the resulting shared secret to blind the (public-channel) authentication of the $C_i$'s which encode $C$. The latter have been sent on the unreliable private wires, unlike in $\Pi_{\text{Gen}}$, where no authentication was required in Round 3 since $C$ itself was sent on the public channel.

**Theorem 3.2** *Protocol $\Pi_{\text{SPD}}$ (Fig. 2) is a valid $(3,2)$-round $(0, 3\delta)$-SMT-PD protocol. It has communication complexity $O(\frac{mn}{n-t})$ on the private wires and $O(n\ell \log m)$ on the public channel, provided $m = \Omega(n\ell \log q)$.*

# 4 Private Communication Lower Bound

In this section we prove a lower bound of $\Omega(\frac{nm}{n-t})$ for the expected communication complexity on the private wires, for *any* $(\epsilon, \delta)$-SMT-PD protocol (where $\epsilon$ and $\delta$ are considered constants). Since protocol $\Pi_{\text{Gen}}$ of

**Protocol** $\Pi_{\mathrm{SPD}}$

1. ($\mathcal{R} \overset{PRI}{\rightarrow} \mathcal{S}$). **(small)** For each wire $i$, $\mathcal{R}$ chooses a random $\hat{f}_i \in \mathbb{F}_{\hat{q}}[X]$ such that $\deg(\hat{f}_i) \leq \hat{K}$. $\mathcal{R}$ sends the Reed-Solomon (RS) codeword $\hat{\mathcal{C}}_i = \big(\hat{f}_i(1), \hat{f}_i(2), \ldots, \hat{f}_i(\hat{N})\big)$ along wire $i$. Let $\widetilde{\hat{\mathcal{C}}}_i$ be the codeword received by $\mathcal{S}$, and $\widetilde{\hat{f}}_i = Dec_{RS}(\widetilde{\hat{\mathcal{C}}}_i)$.

   **(big)** For each wire $i$, $\mathcal{R}$ chooses a random $f_i \in \mathbb{F}_q[X]$ such that $\deg(f_i) \leq K$. $\mathcal{R}$ sends the RS codeword $\mathcal{C}_i = \big(f_i(1), f_i(2), \ldots, f_i(N)\big)$ along wire $i$. Let $\widetilde{\mathcal{C}}_i$ be the codeword received by $\mathcal{S}$, and $\widetilde{f}_i = Dec_{RS}(\widetilde{C}_i)$.

2. ($\mathcal{R} \overset{PUB}{\rightarrow} \mathcal{S}$). **(small)** $\mathcal{R}$ chooses a random subset $\hat{J} = \{\hat{j}_1, \ldots, \hat{j}_\ell\} \subset [\hat{N}]$ of codeword indices, $|\hat{J}| = \ell$. $\mathcal{R}$ performs codeword verification as in Section 2 by sending $\hat{J}$, as well as $\{\hat{\mathcal{C}}_i|_{\hat{J}}\}$ for each wire $i$, over the public channel.

   **(big)** $\mathcal{R}$ chooses a random subset $J = \{j_1, \ldots, j_\ell\} \subset [N]$ of codeword indices, $|J| = \ell$. $\mathcal{R}$ performs codeword verification as in Section 2 by sending $J$, as well as $\{\mathcal{C}_i|_J\}$ for each wire $i$, over the public channel.

3. ($\mathcal{S} \overset{PUB+PRI}{\longrightarrow} \mathcal{R}$). $\mathcal{S}$ rejects any wire $i$ which is syntactically incorrect or which fails one of the consistency checks in Round 2. Call the set of remaining, accepted wires $\mathsf{ACC}$.

   **(small)** Let $\widetilde{\hat{\alpha}}$ denote the concatenation of $\widetilde{\hat{f}}_i$ for each $i \in \mathsf{ACC}$, padded with $0 \in \mathbb{F}_q$ so its length is $\hat{K}n$. Applying $\mathrm{Ext}_{\hat{q}} : \mathbb{F}_{\hat{q}}^{\hat{K}n} \rightarrow \mathbb{F}_{\hat{q}}^{\hat{r}}$ of Section 2, $\mathcal{S}$ obtains $\widetilde{s} = \mathrm{Ext}_{\hat{q}}(\widetilde{\hat{\alpha}})$.

   **(big)** Let $\widetilde{\alpha}$ denote the concatenation of $\widetilde{f}_i$ for each $i \in \mathsf{ACC}$, padded with $0 \in \mathbb{F}_q$ so its length is $Kn$. Applying the randomness extractor $\mathrm{Ext}_q : \mathbb{F}_q^{Kn} \rightarrow \mathbb{F}_q^r$, $\mathcal{S}$ obtains $\widetilde{R} = \mathrm{Ext}_q(\widetilde{\alpha})$.

   Now $M_{\mathcal{S}}$ and $\widetilde{R}$ are both vectors in $\mathbb{F}_q^r$; $\mathcal{S}$ puts $C = \widetilde{R} + M_{\mathcal{S}}$. Now $\mathcal{S}$ applies the Reed Solomon code $\mathbb{F}_q^r \rightarrow \mathbb{F}_q^{Kn}$ to $C$, obtaining a codeword $D \in \mathbb{F}_q^{Kn}$. Let $D = (D_1, \ldots, D_n)$ where each $D_i \in \mathbb{F}_q^K$. View $D_i$ as a bit-string of length $K \log q$, and let $E_i = Enc(D_i)$, so that $|E_i| = cK \log q$ (in bits). $\mathcal{S}$ sends $E_i$ on wire $i \in \mathsf{ACC}$; let $\widetilde{E}_i$ denote the message received by $\mathcal{R}$ on wire $i$.

   To authenticate each $E_i$, $\mathcal{S}$ chooses a random subset $J' \subseteq [cK \log q]$, $|J'| = \ell_{3/2}$. Put $auth_{\mathcal{S}} = (J', \{E_i|_{J'}\}_{i \in \mathsf{ACC}})$; we have $|auth_{\mathcal{S}}| \leq \hat{m}$ (with equality if every wire is in $\mathsf{ACC}$). Padding as necessary, view $auth_{\mathcal{S}}$ as an element of $\mathbb{F}_{\hat{q}}^{\hat{r}}$. $\mathcal{S}$ sets $V = \widetilde{s} + auth_{\mathcal{S}}$ and sends $(V, B)$ over the public channel, where $B$ is an $n$-bit string representing the set $\mathsf{ACC}$.

   **Receiver:** $\mathcal{R}$ learns $\mathsf{ACC}$ from $B$. For $i \in \mathsf{ACC}$, he forms $\alpha$, the concatenation of $f_i$ for each $i \in \mathsf{ACC}$ (padded with $0 \in \mathbb{F}_q$ to length $Kn$). He applies $\mathrm{Ext}_q$ to obtain $R = \mathrm{Ext}_q(\alpha) \in \mathbb{F}_q^r$.

   Similarly, for $i \in \mathsf{ACC}$, he forms $\hat{\alpha}$, the concatenation of $\hat{f}_i$ for each $i \in \mathsf{ACC}$ (padded with $0 \in \mathbb{F}_q$ to length $\hat{K}n$). He applies $\mathrm{Ext}_{\hat{q}}$ to obtain $s = \mathrm{Ext}_{\hat{q}}(\hat{\alpha}) \in \mathbb{F}_{\hat{q}}^{\hat{r}}$.

   Next $\mathcal{R}$ forms $V - s$, which he parses as $auth_{\mathcal{R}} = (\widetilde{J}', \{check_i\}_{i \in \mathsf{ACC}})$. For each (correctly formed) $\widetilde{E}_i$, $\mathcal{R}$ verifies its authenticity by checking that $\widetilde{E}_i|_{\widetilde{J}'} = check_i$. For those which pass, he recovers $\widetilde{D}_i = Dec(\widetilde{E}_i)$, $\widetilde{D}_i \in \mathbb{F}_q^K$. Once $\mathcal{R}$ has recovered at least $n - t$ valid $\widetilde{D}_i$'s, he has $K(n-t) = r$ symbols in $\mathbb{F}_q$, which he uses to decode the RS code used by $\mathcal{S}$ to encode $C$. (This is simply interpolation.) Call the result $\widetilde{C} \in \mathbb{F}_q^r$. Finally, $\mathcal{R}$ obtains $M_{\mathcal{R}} = \widetilde{C} - R$.

   (On failure to authenticate at least $n - t$ $\widetilde{E}_i$'s, or to parse $auth_{\mathcal{R}}$ correctly, $\mathcal{R}$ outputs $\perp$.)

Figure 2: *SMT-PD protocol with small (logarithmic) public communication and optimal private communication.*

the previous section meets this bound, we provide a complete answer to the question raised in [SJST09] of determining the optimal transmission rate on private wires for an $(\epsilon, \delta)$-SMT-PD protocol.

Our communication lower bound holds even for a weakened adversary who is *passive* and *non-adaptive*—that is, $\mathcal{A}$ chooses which wires to corrupt at the start of the protocol and only eavesdrops thereafter. It also holds even if we modify $\delta$-reliability so that the probability that $M_{\mathcal{R}} = M_{\mathcal{S}}$ is taken over the the choice of $M_{\mathcal{S}}$ as well (and not just the players' coins). Further, as noted in the Introduction, it also holds in the case of SMT with no public channel, *mutatis mutandis*.

For the lower bound, we assume that $M_{\mathcal{S}}$ is chosen uniformly at random from $\mathcal{M}$; in this case $H(M_{\mathcal{S}}) = \log|\mathcal{M}|$. (Refer to Appendix A for entropy definitions and formulas.) In the following lemmas (proofs in Appendix B) we assume $\Pi$ is a valid $(\epsilon, \delta)$-SMT-PD protocol, and probabilities are over all players' coins as well as the random selection of $M_{\mathcal{S}} \in \mathcal{M}$.

The first two lemmas are complementary, establishing entropy versions of $\epsilon$-privacy and $\delta$-reliability, respectively. Namely, in Lemma 4.1, we show that in any $\epsilon$-private protocol, the entropy of $M_{\mathcal{S}}$ remains high given the adversary's view. Then in Lemma 4.2, we show that for any $\delta$-reliable protocol (with passive adversary), the entropy of $M_{\mathcal{S}}$ given the entire transcript of communications is low. Though these statements are quite intuitive, their proofs are relatively delicate.

**Lemma 4.1** *For all adversaries $\mathcal{A}$ and all $\epsilon$-private protocols, $H\left(M_{\mathcal{S}} \mid \mathrm{View}_{\mathcal{A}}\right) \geq -\log(1/|\mathcal{M}| + 2\epsilon)$.*[9]

The *transcript* $T$ of an $(\epsilon, \delta)$-SMT-PD protocol execution is the random variable consisting of the list of messages the players send on public and private channels over the course of the protocol. Thus in the case of a passive adversary, $T$ is completely determined by $M_{\mathcal{S}}$ and the coins of $\mathcal{S}$ and $\mathcal{R}$. For a given set of wires $S$, we will let $T_S$ denote the transcript restricted to communications on the wires in $S$. In the sequel we use PUB, PRIV, CORR, and SEC to denote respectively the public channel, private wires, corrupted wires, and secure (uncorrupted and private) wires.

We use $H_2(\cdot)$ to denote the binary entropy function, $H_2(p) = -p\log p - (1-p)\log(1-p)$.

**Lemma 4.2** *For all $\delta$-reliable protocols, $H(M_{\mathcal{S}}|T) \leq H_2(\sqrt{\delta}) + 2\sqrt{\delta}H(M_{\mathcal{S}})$.*

Given Lemmas 4.1 (a proof of "high" entropy) and 4.2 (a proof of "low" entropy), we take the difference of the two inequalities (leaving still a "high" amount of entropy), and show that this bounds from below $H(T_{\mathsf{SEC}}|\mathsf{SEC})$. This is intuitive: the adversary knows which wires are secure, and yet it is only from these wires that $\mathcal{S}$ and $\mathcal{R}$ can leverage any privacy at all. Therefore the entropy of the messages on them should be high.

**Lemma 4.3** $-\log(1/|\mathcal{M}| + 2\epsilon) - H_2(\sqrt{\delta}) - 2\sqrt{\delta}\log|\mathcal{M}| \leq H\left(T_{\mathsf{SEC}} \mid \mathsf{SEC}\right)$.

Our main lower bound theorem follows. The idea is straightforward. Since the set of secure wires is unknown to $\mathcal{S}$ and $\mathcal{R}$ (for a passive adversary, say), it must be that, in an average sense, *every* set of $n-t$ private wires carries the requisite entropy. Then we use Han's inequality (see proof) to "average" the entropy over all subsets of $n-t$ wires and obtain an estimate for the total entropy on private wires, completing the proof.

**Theorem 4.1** *Let $\Pi$ be any $(\epsilon, \delta)$-SMT-PD protocol with $n \leq 2t$, in the presence of a passive, non-adaptive adversary $\mathcal{A}$. Let $C$ denote the expected communication (in bits) over the private wires (the expectation is taken over all players' coins and the choice of $M_S \in \mathcal{M}$). Then*

$$C \geq \frac{n}{n-t} \cdot \left(-\log(1/|\mathcal{M}| + 2\epsilon) - H_2(\sqrt{\delta}) - 2\sqrt{\delta}\log|\mathcal{M}|\right)$$

*In particular, if $\epsilon = O(1/|\mathcal{M}|)$ and $\delta = O(1)$, then $C = \Omega(mn/(n-t))$.*

[9] This entropy lemma is not directly equivalent to a seemingly related probability version (as in [SJST09], Lemma 2).

**Corollary 4.4** *Provided that* $\epsilon = O(1/|\mathcal{M}|)$*, and* $\delta = O(1)$*, protocols* $\Pi_{\mathrm{Gen}}$ *and* $\Pi_{\mathrm{SPD}}$ *have optimal private communication complexity* $O\left(\frac{nm}{n-t}\right)$ *for messages of size* $m = \Omega(n\ell)$ *and* $m = \Omega(n\ell \log q)$*, respectively.*

# 5 Amortized Use of the Public Channel

A natural question when considering SMT-PD as a subroutine in a larger protocol is whether some of the lower bounds on resource use for a single execution of SMT-PD can be beaten *on average* through amortization. For instance, an almost-everywhere secure computation protocol may invoke an SMT-PD subroutine every time any two nodes in the underlying network need to communicate. Must they use the public channel twice every single time, or can the nodes involved, say, save some state information which allows them to reduce their use of the public channel in later invocations?

Our next result shows that amortization can in fact drastically reduce the use of the public channel: indeed, it is possible to limit the total number of uses of the public channel to *two*, no matter how many messages are ultimately sent between two nodes. (Since two uses of the public channel are required to send any reliable communication whatsoever, this is best possible.)

Of course, $\mathcal{S}$ and $\mathcal{R}$ may use the first execution of SMT-PD to establish a shared secret key, which can be used for message encryption and authentication on the common wires. The Sender computes a ciphertext and sends it (with authentication) on every common wire. With overwhelming probability, no forged message is accepted as authentic, and the Receiver accepts the unique, authentic message which arrives on any good wire. However, since we are considering the information-theoretic setting, each use of the shared key reduces its entropy with respect to the adversary's view. If the parties know in advance an upper bound on the total communication they will require, and can afford to send a proportionally large shared key in the first execution of SMT-PD, then this approach is tenable by itself.

In some situations, however, the players may not know a strict upper bound on the number of messages they will send. And even when they do, it may happen that the protocol terminates early with some probability, so that an initial message with large entropy is mostly wasted. With these considerations in mind, we now explore strategies which allow $\mathcal{S}$ and $\mathcal{R}$ to communicate *indefinitely* after using only two broadcast rounds and a limited initial message. Our approach is to separate Sender and Receiver's interaction following the first execution of SMT-PD into two modes: a *Normal Mode* and a *Fault-Recovery Mode*.

In the Normal Mode, $\mathcal{S}$ and $\mathcal{R}$ communicate over the common wires without making use of their shared key; they are successful provided the adversary does not actively interfere. If the adversary does interfere, one of the players (say $\mathcal{R}$) will detect this and enter Fault-Recovery Mode, in which he uses the shared key to broadcast information about the messages he received on each common wire, allowing $\mathcal{S}$ to determine at least one corrupted wire (which he then informs $\mathcal{R}$ about, authentically).

In this way, $\mathcal{S}$ and $\mathcal{R}$ communicate reliably and privately so long as the adversary is passive; and any time he is active, they are able to eliminate at least one corrupted wire[10]. (Of course, once they have eliminated all $t$ corrupt wires, communication becomes *very* efficient.) In the sequel we describe implementations of Normal Mode and Fault-Recovery Mode, as well as how the two modes interact with each other.

**Normal Mode.** Let us first define a weaker version of SMT by public discussion in which reliability is only guaranteed for a passive adversary. Let $\Pi$ be a protocol which attempts to send a message from $\mathcal{S}$ to $\mathcal{R}$ using *only the common wires* (and not relying on any shared secret key). Then we say $\Pi$ is a *Weak* $(\epsilon, \delta)$ *SMT-PD* protocol if it satisfies Definition 2.2 where we (1) add to the adversary's view a bit indicating whether $\mathcal{R}$ accepted a message or not (see next point), and (2) replace RELIABILITY with:

WEAK RELIABILITY:

[10]This is akin to the "slow" PSMT original protocol in [DDWY93]. .

- *(Correctness with passive adversary)* If the adversary only eavesdrops, then $\mathcal{R}$ receives the message correctly.
- *(Detection of active adversary)* If the adversary actively corrupts any wire, then with probability $\geq 1-\delta$, *either* $\mathcal{R}$ receives the message correctly ($M_{\mathcal{R}} = M_{\mathcal{S}}$), *or* $\mathcal{R}$ outputs "Corruption detected."

The first change above affects $\epsilon$-privacy since it alters the definition of $\mathrm{View}_{\mathcal{A}}$; this is necessary because in the compiled, amortized protocol using Weak SMT-PD as a subroutine, the adversary will learn whether $\mathcal{R}$ accepted a message based on whether $\mathcal{R}$ does or does not enter Fault-Recovery Mode.

We remark in passing that Weak SMT-PD is similar in spirit to *almost* SMT from the standard (non-public discussion) model [KS07], in that both are relaxations which allow one-round transmission (for Weak SMT-PD, only with a passive adversary). The difference is that in the ordinary model, definitions for almost SMT require that the message be correctly received with overwhelming probability regardless of the adversary's actions; in the public discussion model, when the adversary controls a majority of wires, this is impossible, so we only require that corruptions be detected. Indeed, we cannot guarantee reliability in a single round even when the adversary simply *blocks* transmission on corrupted wires (otherwise a minority of wires would carry enough information to recover the message, thus violating privacy).

If we do not require the Weak SMT-PD protocol to finish in *one round*, then there is a simple solution: use the common wires to *simulate* the public channel wire in an ordinary SMT-PD protocol. Any time a party would use the public channel, they instead send the public-channel message over *every* common wire. Two possibilities arise: (1) The adversary never tampers with any such "virtual" public channel invocation. In this case, the virtual public channel functions like an actual public channel, and the protocol succeeds with the same probability as the underlying SMT-PD protocol. (2) The adversary at some point tampers with a virtual public channel invocation. If he does, then the receiving party in that round will detect tampering, and can notify the other player by sending a flag on every channel (or, if the receiving player is $\mathcal{R}$ and it is the final round, he just outputs "Corruption Detected").[11]

The above Weak SMT-PD protocol is conceptually simple (given a pre-existing SMT-PD protocol!), but we might hope to do Weak SMT-PD in a single round, as opposed to the three rounds required for ordinary SMT-PD. The following simple scheme shows one way this can be done.

Assume the Sender wants to send a single field element $M_{\mathcal{S}} = \alpha \in \mathbb{F}_q$. The one-round protocol, $\Pi_{\mathrm{W-SMT-PD}}$, is shown in Figure 3. Essentially, the sender performs a $3t+2$-out-of-$3n$ Shamir secret sharing of the message; however, rather than sending externally specified shares on each wire $i$ (such as $f(1), f(2), f(3)$ on wire 1), he chooses a set of *random* points on which to evaluate $f$.

**Protocol** $\Pi_{\mathrm{W-SMT-PD}}$

1. ($\mathcal{S} \overset{PRI}{\rightarrow} \mathcal{R}$). $\mathcal{S}$ chooses a random polynomial $f \in \mathbb{F}_q[x]$ with $\deg(f) \leq 3t+1$ and $f(0) = \alpha$, and a random sequence $x_1^{(1)}, x_1^{(2)}, x_1^{(3)}, x_2^{(1)}, x_2^{(2)}, x_2^{(3)}, \ldots, x_n^{(1)}, x_n^{(2)}, x_n^{(3)}$ of $3n$ distinct elements of $\mathbb{F}_q \setminus \{0\}$. On wire $i$ $\mathcal{S}$ sends to $\mathcal{R}$ the three pairs $(x_i^{(1)}, f(x_i^{(1)})), (x_i^{(2)}, f(x_i^{(2)})), (x_i^{(3)}, f(x_i^{(3)}))$. (The indexing is unimportant; what matters is that on each wire he sends three evaluation points, and the resulting shares.)

   **Receiver:** $\mathcal{R}$ receives $(\widetilde{x}_i^{(j)}, \widetilde{y}_i^{(j)})$ for $j = 1, 2, 3$, on each wire $i$. He verifies that all $3n$ $\widetilde{x}_i$'s are distinct, and that the $3n$ points $(\widetilde{x}_i, \widetilde{y}_i)$ lie on a polynomial $\widetilde{f}$ of degree $\leq 3t+1$. If so, he outputs $M_{\mathcal{R}} = \widetilde{f}(0)$; otherwise (or in case some wire is syntactically incorrect) he outputs "Corruption detected."

Figure 3: *A one-round Weak SMT-PD protocol.*

[11] We do not consider here whether such a protocol preserves ($\epsilon$-)privacy when the adversary knows whether $\mathcal{R}$ detects corruption; obviously this depends on the details of the protocol. Therefore this is not quite a black-box reduction.

**Lemma 5.1** *The protocol of Figure 3 is a Weak $(\delta, \delta)$-SMT-PD protocol for $q$ sufficiently large ($\Omega(t/\delta)$).*

We are now ready to describe Normal Mode for $\mathcal{S}$ and $\mathcal{R}$: it is simply the repeated execution of the Weak SMT-PD protocol, with the two players alternating the role of Sender and Receiver, until one of them as Receiver outputs "Corruption detected." At that time, that player's next message to the other party will alert them to enter Fault-Recovery Mode.

**Fault-Recovery Mode.** Specifically, suppose $\mathcal{R}$ detects corruption in a message sent by $\mathcal{S}$. He will then use the shared secret established in the initial execution of (ordinary) SMT-PD to secretly and authentically send the following on all wires: (1) a flag signalling Fault-Recovery Mode; (2) a list of specific wires known to be corrupted (if any); (3) the received transmission on all wires not known to be corrupt.

Since at least one of the wires is not corrupted, $\mathcal{S}$ will receive this communication on it and (verifying its authenticity) enter Fault-Recovery Mode also. $\mathcal{S}$ recovers the set of received transmissions and determines which ones were tampered with. He then sends the following to $\mathcal{R}$, again using the shared secret for privacy and authentication: (1) the message $M_{\mathcal{S}}$ on which $\mathcal{R}$ detected corruption; (2) an updated list of specific wires known to be corrupted. At this time, $\mathcal{R}$ has received the intended message and Normal Mode resumes with $\mathcal{R}$ now playing the role of Sender.

Each time Fault-Recovery Mode occurs, $\mathcal{S}$ and $\mathcal{R}$ are able to detect at least one previously unknown corrupt wire. If at any point $\mathcal{S}$ and $R$ have jointly detected $t$ wires as corrupt, they will simply send all future transmissions on the remaining, good wires, guaranteeing perfect privacy and reliability.

**Theorem 5.1** *Given an initial shared secret consisting of $O(n^2)$ field elements, $\mathcal{S}$ and $\mathcal{R}$ can communicate* indefinitely *using only the private wires. The probability that one of them will* ever *accept an incorrect message is $\leq t\delta$. Moreover, with probability $\geq 1 - t\delta$, $\mathcal{A}$ gains at most $\delta$ information on each of $t$ different messages, and no information on any other message.*

# References


[ACH06] S. Agarwal, R. Cramer, and R.de Haan. Asymptotically optimal two-round perfectly secure message transmission. In *Advances in Cryptology–CRYPTO'06*. Springer-Verlag, 2006.

[BG93] P. Berman and J. Garay. Fast consensus in networks of bounded degree. *Distributed Computing*, 2(7):62–73, 1993. Preliminary version in *WDAG'90*.

[BGW88] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. 20th Annual ACM Symposium of the Theory of Computation*, pages 1–10, May 1988.

[CCD88] D. Chaum, C. Crepeau, and I. Damgard. Multiparty unconditionally secure protocols. In *Proceedings 20th Annual Symposium on Theory of Computing, STOC*. Association for Computing Machinery, May 1988.

[CPRS08] A. Choudhary, A. Patra, C. P. Rangan, and K. Srinathan. Unconditionally reliable and secure message transmission in undirected synchronous networks: Possibility, feasibility and optimality. Cryptology ePrint Archive, Report 2008/141, 2008. http://eprint.iacr.org/.

[CT91] T. Cover and J. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.

[DDWY93] D. Dolev, C. Dwork, O. Waarts, and M. Young. Perfectly secure message transmission. *Journal of ACM*, 1(40):17–47, 1993.

[DORS08] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[DPPU86] C. Dwork, D. Peleg, N. Pippinger, and E. Upfal. Fault tolerance in networks of bounded degree. In *Proc. 18th Annual Symposium on the Theory of Computing*, pages 370–379, 1986.

[FFGV07] M. Fitzi, M. Franklin, J. Garay, and S. Harsha Vardhan. Towards optimal and efficient perfectly secure message transmission. In *Proc. 4th Theory of Cryptography Conference (TCC '07)*, Lecture Notes in Computer Science, February 2007.

[FM97] P. Feldman and S. Micali. An optimal probabilistic protocol for synchronous Byzantine agreement. *SIAM J. Comput.*, 26(4):873–933, 1997.

[FW98] M. Franklin and R. Wright. Secure communications in minimal connectivity models. In *Advances in Cryptology–EUROCRYPT'98*, Lecture Notes in Computer Science, pages 346–360. Springer Verlag, June 1998.

[GM98] J. Garay and Y. Moses. Fully polynomial Byzantine agreement for $n > 3t$ processors in $t + 1$ rounds. *SIAM J. Comput.*, 27(1):247–290, 1998. Preliminary version in *STOC '92*.

[GO08] J. Garay and R. Ostrovsky. Almost-everywhere secure computation. In *Advances in Cryptology–Eurocrypt'08*, Lecture Notes in Computer Science 4965, pages 307–323. Springer, April 2008.

[GRR98] R. Gennaro, M. Rabin, and T. Rabin. Simplified vss and fast-track multiparty computations with applications to threshold cryptography. In *Proc. 17th Annual ACM Symp. on Principles of Distributed Computing, PODC*, pages 101–111. ACM, 1998.

[KK06] J. Katz and C. Koo. On expected constant-round protocols for Byzantine agreement. In *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 445–462. Springer, 2006.

[KS07] K. Kurosawa and K. Suzuki. Almost secure (1-round, n-channel) message transmission scheme. Cryptology ePrint Archive, Report 2007/076, 2007. http://eprint.iacr.org/.

[KS08] K. Kurosawa and K. Suzuki. Truly efficient 2-round perfectly secure message transmission scheme. Lecture Notes in Computer Science 4965, pages 324–340. Springer, April 2008.

[KZ06] J. Kamp and D. Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM J. Comput.*, 36(5):1231–1247, 2006.

[Lam87] L. Lamport. A fast mutual exclusion algorithm. *ACM Transactions on Computer Systems*, 5(1):1–11, 1987.

[MS83] F. Macwilliams and N. Sloane. *The Theory of Error-Correcting Codes*. North Holland, January 1983.

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

[SJST09] H. Shi, S. Jiang, R. Safavi-Naini, and M. Tuhin. Optimal secure message transmission by public discussion. In arXiv.org e-Print archive, arXiv:0901.2192v1, January 2009.

[SNP04] K. Srinathan, A. Narayanan, and C. Pandu Rangan. Optimal perfectly secure message transmission. In *Advances in Cryptology–CRYPTO'04*, volume 3152 of *Lecture Notes in Computer Science*, pages 545–561. Springer-Verlag, 2004.

[SPR07] K. Srinathan, N. R. Prasad, and C. P. Rangan. On the optimal communication complexity of multiphase protocols for perfect communication. *Security and Privacy, IEEE Symposium on*, 0:311–320, 2007.

[Upf92] E. Upfal. Tolerating linear number of faults in networks of bounded degree. In *Proc. 11th ACM Symposiyum on Principles of Distributed Computing*, pages 83–89, 1992.

# A Some Entropy Formulas

Here we briefly review some facts about entropy. For a good reference, see, e.g., [CT91]. The *entropy* of a discrete random variable $X$ (defined over a space $\mathcal{X}$) is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} \Pr[X = x] \log \Pr[X = x].$$

Entropy provides a measure of the expected information content of $X$; to be precise, it measures the expected number of bits in a representation of $X$. The *conditional entropy of $X$ given $Y$*, denoted $H(X \mid Y)$, expresses the expected additional information given by $X$ if we already know $Y$. It satisfies the following formula, known as the chain rule for conditional entropy:

$$H(X \mid Y) = H(X, Y) - H(Y). \tag{1}$$

Here $H(X, Y)$ is the entropy of the random variable $(X, Y)$ in the product space $\mathcal{X} \times \mathcal{Y}$; it measures the expected information on learning the values of $X$ and $Y$ (which may be correlated). Some additional formulas which we will find useful include:

(E1) $H(X) \leq \log |\mathcal{X}|$, with equality iff $X$ is uniformly distributed over $\mathcal{X}$.

(E2) $H(X \mid Y) \leq H(X \mid f(Y))$.

(E3) $H(X \mid Z) \leq H(X \mid Y) + H(Y \mid Z)$.

(E4) $H(X \mid Y) \leq H(X))$.

# B Proofs

We repeat the statements of the claims here for convenience.

**Claim 3.1** *The distribution $(U_s, \widetilde{R}, \mathrm{View}_{\mathcal{A}} \setminus C)$ is $\epsilon/2$-close to $(U_s, U_m, \mathrm{View}_{\mathcal{A}} \setminus C)$.*

*Proof.* (Claim.) We first show that $\widetilde{H}_\infty(\widetilde{\alpha} | \mathrm{View}_{\mathcal{A}} \setminus C) \geq k_{min}$. Recall that $\widetilde{\alpha}$ is, by definition, the concatenation of $\widetilde{r}_i$ for $i \in \mathsf{ACC}$ (padded to length $nK$). Let $\mathsf{SEC}$ denote the set of secure (private, uncorrupted) wires. Then for $i \in \mathsf{SEC}$, we have that $\widetilde{r}_i = r_i$, and therefore

$$\begin{aligned}\widetilde{H}_\infty(\widetilde{r}_i | \mathrm{View}_{\mathcal{A}} \setminus C) &= \widetilde{H}_\infty(r_i | \mathrm{View}_{\mathcal{A}} \setminus C) \\ &= \widetilde{H}_\infty(r_i | (J, \mathcal{C}_i|_J)),\end{aligned}$$

where the latter equality follows since $r_i$ is independent of everything in $\mathcal{A}$'s view except $(J, \mathcal{C}_i|_J)$ (and $C$, which we exclude). Now we apply Fact 2.4: since $\mathcal{C}_i|_J$ has at most $2^\ell$ possible values, then

$$\begin{aligned}\widetilde{H}_\infty(r_i | (J, \mathcal{C}_i|_J)) &\geq \widetilde{H}_\infty(r_i | J) - \ell \\ &= \widetilde{H}_\infty(r_i) - \ell && \text{(independence of } r_i, J) \\ &= K - \ell = \lceil k_{min}/(n-t) \rceil + \ell - \ell = \lceil k_{min}/(n-t) \rceil.\end{aligned}$$

Since the transmissions on the private wires, and associated codeword verifications, are all mutually independent, we then have

$$\begin{aligned}\widetilde{H}_\infty(\widetilde{\alpha} | \mathrm{View}_{\mathcal{A}} \setminus C) &= \widetilde{H}_\infty(\{\widetilde{r}_i\}_{i \in \mathsf{ACC}} | \mathrm{View}_{\mathcal{A}} \setminus C) \\ &\geq \widetilde{H}_\infty(\{r_i\}_{i \in \mathsf{SEC}} | \mathrm{View}_{\mathcal{A}} \setminus C) \\ &= \sum_{i \in \mathsf{SEC}} \widetilde{H}_\infty(r_i | (J, \mathcal{C}_i|_J)) \\ &\geq (n-t) \lceil k_{min}/(n-t) \rceil \geq k_{min},\end{aligned}$$

which shows as we claimed that $\widetilde{H}_\infty(\widetilde{\alpha}|\mathrm{View}_{\mathcal{A}} \setminus C) \geq k_{min}$. From here the overall claim follows immediately because $\mathrm{Ext}_A$ is an $(nK, m, k_{min}, \epsilon/2)$-strong average-case extractor, and $\widetilde{R} = \mathrm{Ext}_A(\widetilde{\alpha}, seed)$ with $seed$ uniformly random. □

**Theorem 3.2** *Protocol $\Pi_{\mathrm{SPD}}$ (Fig. 2) is a valid $(3,2)$-round $(0, 3\delta)$-SMT-PD protocol. It has communication complexity $O(\frac{mn}{n-t})$ on the private wires and $O(n\ell \log m)$ on the public channel, provided $m = \Omega(n\ell \log q)$.*

*Proof.* **Reliability.** We first claim that, with high probability, $\mathcal{R}$ receives $auth_{\mathcal{S}} = (J', \{E_i|_{J'}\}_{i\in\mathsf{ACC}})$ correctly in Round 3—that is, $(\widetilde{J}', \{check_i\}_{i\in\mathsf{ACC}}) = (J', \{E_i|_{J'}\}_{i\in\mathsf{ACC}})$). Since $(\widetilde{J}', \{check_i\}_{i\in\mathsf{ACC}}) = V - s$ and $(J', \{E_i|_{J'}\}_{i\in\mathsf{ACC}}) = V - \widetilde{s}$, it is enough to show that $s = \widetilde{s}$. According to the protocol,

$$s = \mathrm{Ext}_{\hat{q}}(\hat{\alpha}) \qquad \text{and} \qquad \widetilde{s} = \mathrm{Ext}_{\hat{q}}(\widetilde{\hat{\alpha}}),$$

so again it is enough to show that $\hat{\alpha} = \widetilde{\hat{\alpha}}$. These variables are defined as the concatenations (over $i \in \mathsf{ACC}$) of $\hat{f}_i$ and $\widetilde{\hat{f}}_i$ respectively (appropriately padded). Thus it suffices to show that for every $i \in \mathsf{ACC}$, $\hat{f}_i = \widetilde{\hat{f}}_i$. This in turn will only *fail* to be the case provided that $\mathcal{A}$ tampers with one of the codewords $\hat{\mathcal{C}}_i$ transmitted in Round 1, and is not caught by the consistency check in Round 2. By construction, the probability that $\mathcal{A}$ beats the consistency check on a given wire is at most $2^{-\ell} = \delta/t$, and so the probability that he beats any of the consistency checks at all, is bounded by $t \cdot 2^{-\ell} = \delta$. This proves

**Claim B.1** *Let $G_1$ be the event in which $\mathcal{R}$ receives $(J', \{E_i|_{J'}\}_{i\in\mathsf{ACC}})$ correctly. The probability of $G_1$ is $\geq 1 - \delta$.*

Note that if $G_1$ holds, then the Round 3 consistency check $\mathcal{R}$ performs on the $\widetilde{E}_i$'s is done using the correct authentication, $auth_{\mathcal{R}} = auth_{\mathcal{S}}$. Nevertheless, $\mathcal{R}$ may still incorrectly accept some $\widetilde{E}_i \neq E_i$ if $\mathcal{A}$ tampers with $E_i$ and even the correct authentication vector fails to detect the tamper. We claim that the probability that this happens is also $\leq \delta$ (provided $G_1$ holds). For in order for $\mathcal{R}$ to accept a tampered $\widetilde{E_i}$, it must fail the Round 3 consistency check. Now $\mathrm{View}_{\mathcal{A}}$ is statistically independent of $auth_{\mathcal{S}} = (J', \{E_i|_{J'}\}_{i\in\mathsf{ACC}})$, since the latter is sent using a 0-private instantiation of $\Pi_{\mathrm{Gen}}$.

Therefore the Round 3 consistency check is valid, and (since we use a code of relative minimum distance 1/3) the probability $\mathcal{A}$ successfully tampers any given corrupt wire is again $\leq (2/3)^{\ell_{3/2}} = \delta/t$, and the chance of any successful tampering is $\leq \delta$. We have thus established

**Claim B.2** *Suppose $G_1$ holds (so that $\mathcal{R}$ uses the correct authentication vector). Let $G_2$ be the event in which the first $n - t$ $\widetilde{E}_i$'s accepted by $\mathcal{R}$ are in fact valid ($\widetilde{E}_i = E_i$). Then the probability of $G_2$, given $G_1$, is $\geq 1 - \delta$.*

(Note that if $G_1$ does not hold, then there is no guarantee that $\mathcal{R}$ will accept any $\widetilde{E}_i$'s, let alone $n - t$ of them. On the other hand, if it does hold, then $\mathcal{R}$ will correctly accept $\widetilde{E}_i = E_i$ on every good wire $i$, and the only question is whether he accepts a tampered $\widetilde{E}_i$ *first* (since the protocol has $\mathcal{R}$ use the first $n - t$ accepted $\widetilde{E}_i$'s to reconstruct $C$. The claim above shows that this bad outcome happens with negligible probability.)

Now if $G_1$ and $G_2$ both hold, then it follows that for every accepted $\widetilde{E}_i$, $\widetilde{D_i} = Dec(\widetilde{E}_i) = Dec(E_i) = D_i$, and hence $\mathcal{R}$ decodes correctly using these $\widetilde{D_i}$'s, and we thus have $\widetilde{C} = C$. In this case, since $M_{\mathcal{S}} = C - \widetilde{R}$ and $M_{\mathcal{R}} = \widetilde{C} - R$, reliability will hold provided that $\widetilde{R} = R$. This last equality is satisfied provided $\mathcal{A}$ does not successfully tamper with any of the $\mathcal{C}_i$'s sent in Round 1—therefore it is satisfied with probability $\geq 1 - \delta$. Therefore we have:

**Claim B.3** *Suppose $G_1$ and $G_2$ hold. Then $\Pr[M_{\mathcal{R}} = M_{\mathcal{S}}] \geq 1 - \delta$.*

Taking the three previous claims all together, we find:

$$\begin{aligned}\Pr[M_{\mathcal{R}} = M_{\mathcal{S}}] &\geq \Pr[M_{\mathcal{R}} = M_{\mathcal{S}} | G_1, G_2] \cdot \Pr[G_1, G_2] \\ &\geq (1-\delta) \Pr[G_2|G_1] \Pr[G_1] \\ &\geq (1-\delta)^3 \geq 1 - 3\delta,\end{aligned}$$

which completes the proof of $3\delta$-reliability.

**(Perfect) Privacy.** The proof is essentially the same as that in the privacy analysis for protocol $\Pi_{\mathrm{Gen}}$. In fact it is simpler, since here we specialize to $\epsilon = 0$.

In a nutshell, suppose we exclude from the adversary's view $C = M_{\mathcal{S}} + \widetilde{R}$ (which is to say, we exclude the codewords $E_i$ which he could use to reconstruct $C$). Then according to this restricted view, $\widetilde{R} = \mathrm{Ext}_q(\widetilde{\alpha})$ is distributed uniformly at random over $\mathbb{F}_q^r$, because $\mathrm{Ext}_q$ is an error-less extractor. Thus replacing $C$ in his view still gives $\mathcal{A}$ no information on $M_{\mathcal{S}}$.

**Complexity.** By its definition, $\hat{m} = \ell(n + \log(cK \log q)) = \ell n + O(\ell \log \frac{m}{n-t})$. This is $O(m)$ provided that $m = \Omega(n\ell \log q)$, since $\log q = \Omega(\log(n/(n-t)))$ by the second condition on its size. Since $\hat{m} = O(m)$, and since Rounds 1 and 2 have the same structure for the big and small strands, it follows that the communication complexity in these two rounds is dominated by the big strand, so we consider only those messages.

Note also that our assumption $m/(n-t) = \Omega(\ell \log q)$ is equivalent to $r/(n-t) = \Omega(\ell)$ (since $m = r \log q$). As a result, we freely replace $K = r/(n-t) + \ell$ with $r/(n-t)$ in the estimates below.

Regarding communication on the private wires, note that they are used first in Round 1 to send $2K$ field elements per wire, at a total cost of

$$2Kn \log q = O\left(\frac{r}{n-t} \cdot n \cdot \log q\right).$$

which yields a communication complexity of $O(\frac{mn}{n-t})$ since $m = r \log q$.

The private wires are used again in Round 3 to send the $E_i$'s, each of which has size $cK \log q = \frac{cm}{n-t}$. Thus, since we send at most $n$, we again get communication complexity $O(mn/(n-t))$ on the private wires.

The public channel is used in Rounds 2 and 3; in Round 2 $\mathcal{R}$ specifies $\ell$ positions out of $[N]$, and sends a total of $n\ell$ field elements. Hence the Round 2 public communication is

$$\begin{aligned}\ell \log(N) + n\ell \log q &= O(\ell \log(K) + n\ell \log q) \\ &= O(\ell \log r + n\ell \log q).\end{aligned}$$

Now $r = m/\log q < q$ (by assumption on $q$). Hence $\log r < \log q$, so the above is $O(n\ell \log q)$. By our choice of $q$, we have $\log q + \log\log q \geq \log m + \log(n/(n-t)) + O(1)$.[12] Therefore the Round 2 public communication is $O(n\ell(\log m + \log(n/(n-t))))$; in particular it is $O(n\ell \log m)$ provided that $m = \Omega(n/(n-t))$, per our assumption.

In Round 3, $\mathcal{S}$ sends $(V, B)$ publicly. $B$ just has size $n$, and $V$ has size $\hat{m} = \ell(n + \log(\frac{cm}{n-t}))$, which are dominated by the Round 2 public communication. □

**Lemma 4.1** *For all adversaries $\mathcal{A}$ and all $\epsilon$-private protocols, $H\left(M_{\mathcal{S}} \,\middle|\, \mathrm{View}_{\mathcal{A}}\right) \geq -\log(1/|\mathcal{M}| + 2\epsilon)$.*

*Proof.* Let $\mathcal{V}$ denote the support of $\mathrm{View}_A$, and for brevity let $V = \mathrm{View}_{\mathcal{A}}$. By the $\epsilon$-privacy condition,

[12] It is easy to check that the second condition on $q$'s size does not affect our conclusions.

we have that for any two messages $m_0, m_1 \in \mathcal{M}$,

$$\sum_{v\in\mathcal{V}} \Big| \Pr[V = v|M_{\mathcal{S}} = m_0] - \Pr[V = v|M_{\mathcal{S}} = m_1] \Big| \leq 2\epsilon$$
$$\sum_{v\in\mathcal{V}} \Big| \frac{\Pr[V = v, M_{\mathcal{S}} = m_0]}{\Pr[M_{\mathcal{S}} = m_0]} - \frac{\Pr[V = v, M_{\mathcal{S}} = m_1]}{\Pr[M_{\mathcal{S}} = m_1]} \Big| \leq 2\epsilon$$
$$\sum_{v\in\mathcal{V}} \Big| \Pr[M_{\mathcal{S}} = m_0, V = v] - \Pr[M_{\mathcal{S}} = m_1, V = v] \Big| \leq \frac{2\epsilon}{|\mathcal{M}|}.$$

Summing over messages $m_0$, we obtain that for all $m_1 \in \mathcal{M}$:

$$\sum_{m_0\in\mathcal{M}} \sum_{v\in\mathcal{V}} \Big| \Pr[M_{\mathcal{S}} = m_0, V = v] - \Pr[M_{\mathcal{S}} = m_1, V = v] \Big| \leq 2\epsilon.$$

For any fixed $v \in \mathcal{V}$, let $m_v$ denote a maximally probable value of $M_{\mathcal{S}}$, given that $\text{View}_A = v$. That is, for all $m_1 \in \mathcal{M}$, $\Pr[M_{\mathcal{S}} = m_v|V = v] \geq \Pr[M_{\mathcal{S}} = m_1|V = v]$, or equivalently $\Pr[M_{\mathcal{S}} = m_v, V = v] \geq \Pr[M_{\mathcal{S}} = m_1, V = v]$. We then remove all summands in the previous inequality except those with $m_0 = m_v$, which results in the valid inequality (for all $m_1 \in \mathcal{M}$):

$$\sum_{v\in\mathcal{V}} \Big( \Pr[M_{\mathcal{S}} = m_v, V = v] - \Pr[M_{\mathcal{S}} = m_1, V = v] \Big) \leq 2\epsilon$$
$$\sum_{v\in\mathcal{V}} \Pr[M_{\mathcal{S}} = m_v, V = v] \leq \sum_{v\in\mathcal{V}} \Big( \Pr[M_{\mathcal{S}} = m_1, V = v] \Big) + 2\epsilon$$
$$= \Pr[M_{\mathcal{S}} = m_1] + 2\epsilon = 1/|\mathcal{M}| + 2\epsilon. \tag{2}$$

Now consider $H(M_{\mathcal{S}}|\text{View}_A)$. Since $H(M_{\mathcal{S}}|V = v) \geq H_\infty(M_{\mathcal{S}}|V = v)$, it follows that

$$\begin{aligned} H(M_{\mathcal{S}}|V) &= \mathbb{E}_{v\leftarrow V}[H(M_{\mathcal{S}}|V = v)] \geq \mathbb{E}_{v\leftarrow V}[H_\infty(M_{\mathcal{S}}|V = v)] \\ &= \sum_{v\in\mathcal{V}} \Pr[V = v] \left( -\log \left( \max_{m\in\mathcal{M}} \Pr[M_{\mathcal{S}} = m|V = v] \right) \right) \\ &= \sum_{v\in\mathcal{V}} \Pr[V = v] \left( \log \left( \frac{\Pr[V = v]}{\max_{m\in\mathcal{M}} \Pr[M_{\mathcal{S}} = m, V = v]} \right) \right) \\ &\geq \log \left( \frac{1}{\sum_{v\in\mathcal{V}} \max_{m\in\mathcal{M}} \Pr[M_{\mathcal{S}} = m, V = v]} \right) \\ &= -\log \left( \sum_{v\in\mathcal{V}} \Pr[M_{\mathcal{S}} = m_v, V = v] \right), \end{aligned} \tag{3}$$

where the next-to-last line is an application of the log sum inequality.[13] Finally, we can substitute (2) into (3) to obtain

$$H(M_{\mathcal{S}}|V) \geq -\log(1/|\mathcal{M}| + 2\epsilon).$$

□

**Lemma 4.2** *For all $\delta$-reliable protocols, $H(M_{\mathcal{S}}|T) \leq H_2(\sqrt{\delta}) + 2\sqrt{\delta}H(M_{\mathcal{S}})$.*

*Proof.* The proof of the lemma follows from a series of several claims.

Call a given transcript $\tau$ *good* if $\Pr[M_{\mathcal{S}} = M_{\mathcal{R}}|T = \tau] \geq 1 - \sqrt{\delta}$. Otherwise $\tau$ is *bad*.

[13] For $a_i, b_i \geq 0$ with $\sum a_i = a$ and $\sum b_i = b$, the log sum inequality states $\sum_i \left( a_i \log \frac{a_i}{b_i} \right) \geq a \log \frac{a}{b}$.

**Claim B.4** *With probability $\geq 1 - \sqrt{\delta}$, $T$ is a good transcript.*

*Proof.* Suppose not, then we will show $\Pr[M_{\mathcal{R}} = M_{\mathcal{S}}] < 1 - \delta$, contradicting $\delta$-reliability. Indeed:

$$\begin{aligned}
\Pr[M_{\mathcal{R}} = M_{\mathcal{S}}] &= \sum_{\tau \in T, m \in \mathcal{M}} \Pr[M_{\mathcal{R}} = m \wedge M_{\mathcal{S}} = m \wedge T = \tau] \\
&= \sum_{\tau \text{ good}} \Pr[M_{\mathcal{R}} = M_{\mathcal{S}} | T = \tau] \Pr[T = \tau] + \sum_{\tau \text{ bad}} \Pr[M_{\mathcal{R}} = M_{\mathcal{S}} | T = \tau] \Pr[T = \tau] \\
&< \sum_{\tau \text{ good}} \Pr[T = \tau] + \sum_{\tau \text{ bad}} (1 - \sqrt{\delta}) \Pr[T = \tau] \\
&= \Pr[T \text{ good}] + (1 - \sqrt{\delta}) \Pr[T \text{ bad}] \\
&= 1 - \Pr[T \text{ bad}] + (1 - \sqrt{\delta}) \Pr[T \text{ bad}] \\
&= 1 - \sqrt{\delta} \Pr[T \text{ bad}] < 1 - \delta.
\end{aligned}$$

□

Our next claim relates the probability that $M_{\mathcal{S}} = M_0$, given fixed transcript $\tau$ and coins for $\mathcal{R}$, to the probability of $M_{\mathcal{S}} = M_0$ given only the transcript $\tau$.

**Claim B.5** *For all messages $M_0$, coins $c_r$ for $\mathcal{R}$, and transcripts $\tau$:*

$$\Pr[M_{\mathcal{S}} = M_0 | C_{\mathcal{R}} = c_r \wedge T = \tau] \leq \frac{\Pr[C_{\mathcal{R}} = c_r]}{\Pr[C_{\mathcal{R}} = c_r | T = \tau]} \Pr[M_{\mathcal{S}} = M_0 | T = \tau].$$

*Proof.* The LHS is equal to $\Pr[M_S = M_0 \wedge C_{\mathcal{R}} = c_r \wedge T = \tau] / \Pr[C_{\mathcal{R}} = c_r \wedge T = \tau]$. Considering only the numerator of this expression to begin with, we have:

$$\begin{aligned}
&\Pr[M_S = M_0 \wedge C_{\mathcal{R}} = c_r \wedge T = \tau] \\
&= \sum_{c'_s} \Pr[M_S = M_0 \wedge C_{\mathcal{R}} = c_r \wedge T = \tau \wedge C_{\mathcal{S}} = c'_s] \\
&= \sum_{c'_s} \Pr[M_{\mathcal{S}} = M_0 \wedge C_{\mathcal{S}} = c'_s] \Pr[C_{\mathcal{R}} = c_r] \{\tau = T(M_0, c_r, c'_s)\}.
\end{aligned}$$

The previous line uses our independence assumptions on $\mathcal{R}$'s coins. It also uses a bracket notation to simplify summation notation: here $\{P\} = 1$ if the predicate $P$ is true and 0 otherwise. Continuing, the above is

$$\begin{aligned}
&= \Pr[C_{\mathcal{R}} = c_r] \sum_{c'_s} \Pr[M_{\mathcal{S}} = M_0 \wedge C_{\mathcal{S}} = c'_s] \{\tau = T(M_0, c_r, c'_s)\} \\
&\leq \Pr[C_{\mathcal{R}} = c_r] \sum_{c'_s} \Pr[M_{\mathcal{S}} = M_0 \wedge C_{\mathcal{S}} = c'_s] \sum_{c'_r} \{\tau = T(M_0, c'_r, c'_s)\} \\
&= \Pr[C_{\mathcal{R}} = c_r] \sum_{c'_s} \sum_{c'_r} \Pr[M_{\mathcal{S}} = M_0 \wedge C_{\mathcal{S}} = c'_s] \{\tau = T(M_0, c'_r, c'_s)\} \\
&= \Pr[C_{\mathcal{R}} = c_r] \sum_{c'_s} \Pr[M_{\mathcal{S}} = M_0 \wedge C_{\mathcal{S}} = c'_s \wedge T = \tau]
\end{aligned}$$

The last equality may be easier to understand if you work backwards from the last line to the previous one. It says that we can get the probability of a fixed message $M_0$ and coins of $\mathcal{S}$ $c'_s$ and transcript $\tau$, by summing

over all coins for $\mathcal{R}$, the probability that the message and coins appear, conditioned on the fact that the transcript generated by the message and coins in that summand *is* in fact $\tau$.

$$= \Pr[C_{\mathcal{R}} = c_r] \Pr[M_{\mathcal{S}} = M_0 \wedge T = \tau] = \Pr[C_{\mathcal{R}} = c_r] \Pr[T = \tau] \Pr[M_{\mathcal{S}} = M_0 | T = \tau].$$

Dividing back through by the denominator $\Pr[C_{\mathcal{R}} = c_r \wedge T = \tau]$ yields the statement of the claim. □

For a given transcript $\tau$, let $M_{max}(\tau)$ denote the maximally probable $M_{\mathcal{S}}$ given that $T = \tau$. (If two are equiprobable, break the tie arbitrarily.) The next claim asserts that for a good transcript, the sender's message equals $M_{max}(\tau)$ with high probability.

**Claim B.6** *For any good transcript $\tau$, $\Pr[M_{\mathcal{S}} = M_{max}(\tau) | T = \tau] \geq 1 - \sqrt{\delta}$.*

*Proof.* By Claim B.4, we have $\Pr[M_{\mathcal{R}} = M_{\mathcal{S}} | T = \tau] \geq 1 - \sqrt{\delta}$. On the other hand,

$$\begin{aligned}
\Pr[M_{\mathcal{R}} = M_{\mathcal{S}} | T = \tau] &= \sum_{c_r} \Pr[M_{\mathcal{R}} = M_{\mathcal{S}} | C_{\mathcal{R}} = c_r \wedge T = \tau] \Pr[C_{\mathcal{R}} = c_r | T = \tau] \\
&= \sum_{c_r} \Pr[M_{\mathcal{S}} = M_{\mathcal{R}}(\tau, c_r) | C_{\mathcal{R}} = c_r \wedge T = \tau] \Pr[C_{\mathcal{R}} = c_r | T = \tau].
\end{aligned}$$

Now apply Claim B.5 with $M_0 = M_{\mathcal{R}}(\tau, c_r)$:

$$\begin{aligned}
&\leq \sum_{c_r} \Pr[M_{\mathcal{S}} = M_{\mathcal{R}}(\tau, c_r) | T = \tau] \Pr[C_{\mathcal{R}} = c_r] \\
&\leq \sum_{c_r} \Pr[M_{\mathcal{S}} = M_{max}(\tau) | T = \tau] \Pr[C_{\mathcal{R}} = c_r] \\
&= \Pr[M_{\mathcal{S}} = M_{max}(\tau) | T = \tau],
\end{aligned}$$

as desired. □

Now (finally) we consider $H(M_{\mathcal{S}}|T)$. By definition,

$$\begin{aligned}
H(M_{\mathcal{S}}|T) &= \sum_{\tau} \Pr[T = \tau] H(M_{\mathcal{S}} | T = \tau) \\
&= \sum_{\tau} \Pr[T = \tau] \sum_{M} \Pr[M_{\mathcal{S}} = M | T = \tau] \log \Pr[M_{\mathcal{S}} = M | T = \tau].
\end{aligned}$$

We estimate the sum by splitting its domain depending on whether $\tau$ is good or bad. Considering first the sum over bad $\tau$:

$$\begin{aligned}
\sum_{\tau \text{ bad}} \Pr[T = \tau] H(M_{\mathcal{S}} | T = \tau) &\leq \sum_{\tau \text{ bad}} \Pr[T = \tau] H(M_{\mathcal{S}}) \qquad \text{(E4)} \\
&= \Pr[T \text{ bad}] H(M_{\mathcal{S}}) \leq \sqrt{\delta} H(M_{\mathcal{S}}).
\end{aligned}$$

Moving to the sum over good $\tau$, we re-interpret it as an entropy $H((M_{\mathcal{S}})_{good}|T_{good})$, where the random variables are those induced by restricting the space of executions to ones which produce good transcripts. Then we apply Fano's inequality [CT91], which states that for random variables $X, Y$ over a discrete space $S$, if $\hat{p} = \Pr[X \neq Y]$, then

$$H\left(X \,\middle|\, Y\right) \leq H_2(\hat{p}) + \hat{p} \log(|S| - 1).$$

In our case this becomes

$$\begin{aligned}
H((M_{\mathcal{S}})_{good} | T_{good}) &\leq H((M_{\mathcal{S}})_{good} | M_{max}(T_{good})) \qquad \text{(E2)} \\
&\leq H_2(\sqrt{\delta}) + \sqrt{\delta} \log(|\mathcal{M}| - 1) \\
&\leq H_2(\sqrt{\delta}) \sqrt{\delta} H(M_{\mathcal{S}}),
\end{aligned}$$

where the second line applies Fano's inequality to the fact that (for good transcripts) $\Pr[M_{\mathcal{S}} \neq M_{max}(\tau)] \leq \sqrt{\delta}$ (Claim B.6).

Combining the two estimates, we see that $H(M_{\mathcal{S}}|T) \leq H_2(\sqrt{\delta}) + 2\sqrt{\delta}H(M_{\mathcal{S}})$, as required. □

**Lemma 4.3** $-\log(1/|\mathcal{M}| + 2\epsilon) - H_2(\sqrt{\delta}) - 2\sqrt{\delta}\log|\mathcal{M}| \leq H\left(T_{\mathsf{SEC}} \;\middle|\; \mathsf{SEC}\right)$.

*Proof.* Since a secure $(\epsilon, \delta)$-SMT-PD protocol must work for any adversary $\mathcal{A}$, it is enough to show that the inequality holds when $\mathcal{A}$ is passive and only uses randomness to decide which wires to corrupt. In this case $\text{View}_{\mathcal{A}} = (T_{\mathsf{PUB}\cup\mathsf{CORR}}, \mathsf{SEC})$.

By Lemmas 4.1 and 4.2 and the properties of entropy, we have (assuming $\mathcal{A}$ is passive):

$$\begin{aligned}
-\log(1/|\mathcal{M}| + 2\epsilon) - H_2(\sqrt{\delta}) - 2\sqrt{\delta}\log|\mathcal{M}| &\leq H\left(M_{\mathcal{S}} \;\middle|\; \text{View}_A\right) - H\left(M_{\mathcal{S}} \;\middle|\; T\right) \quad \text{(Lemmas 4.1, 4.2)} \\
&\leq H\left(T \;\middle|\; \text{View}_{\mathcal{A}}\right) \quad \text{(E3)} \\
&= H\left(T_{\mathsf{PUB}\cup\mathsf{CORR}}, T_{\mathsf{SEC}} \;\middle|\; T_{\mathsf{PUB}\cup\mathsf{CORR}}, \mathsf{SEC}\right) \\
&= H\left(T_{\mathsf{SEC}} \;\middle|\; T_{\mathsf{PUB}\cup\mathsf{CORR}}, \mathsf{SEC}\right) \\
&\leq H\left(T_{\mathsf{SEC}} \;\middle|\; \mathsf{SEC}\right). \quad \text{(E2)}
\end{aligned}$$

□

**Theorem 4.1** *Let $\Pi$ be any $(\epsilon, \delta)$-SMT-PD protocol with $n \leq 2t$, in the presence of a passive, non-adaptive adversary $\mathcal{A}$. Let $C$ denote the expected communication (in bits) over the private wires (the expectation is taken over all players' coins and the choice of $M_S \in \mathcal{M}$). Then*

$$C \geq \frac{n}{n-t} \cdot (-\log(1/|\mathcal{M}| + 2\epsilon) - H_2(\sqrt{\delta}) - 2\sqrt{\delta}\log|\mathcal{M}|)$$

*In particular, if $\epsilon = O(1/|\mathcal{M}|)$ and $\delta = O(1)$, then $C = \Omega(mn/(n-t))$.*

*Proof.* Let $\mathcal{A}$ be a passive adversary who chooses $t$ wires uniformly at random to corrupt. Let $\mathcal{T}$ refer to the space of possible transcripts, and likewise for a wire $i$ or set of wires $S$, let $\mathcal{T}_i$ and $\mathcal{T}_S$ denote the space of possible restricted transcripts $T_i$ and $T_S$. By the definition of entropy, $H(T_{\mathsf{SEC}}|\mathsf{SEC}) = H(T_{\mathsf{SEC}}, \mathsf{SEC}) - H(\mathsf{SEC})$ is equal to:

$$-\sum_{\substack{I\subseteq[n]\\|I|=n-t}} \sum_{\tau\in\mathcal{T}_I} \Pr[T_I = \tau \wedge \mathsf{SEC} = I] \cdot \log\left(\Pr[T_I = \tau \wedge \mathsf{SEC} = I]\right) \quad - \quad H(\mathsf{SEC}).$$

But since $\mathcal{A}$ is passive, $\mathcal{S}$ and $\mathcal{R}$ have no information on which wires are in $\mathsf{CORR}$ or $\mathsf{SEC}$. It follows that the transcript $T_I$ is independent of $\mathsf{SEC}$; hence

$$\Pr[T_I = \tau \wedge \mathsf{SEC} = I] \quad = \quad \Pr[T_I = \tau]\Pr[\mathsf{SEC} = I] \quad = \quad \Pr[T_I = \tau]\frac{1}{\binom{n}{n-t}},$$

since every set of size $n-t$ is equally likely to be $\mathsf{SEC}$. The previous sum is then equal to

$$\begin{aligned}
&-\sum_{\substack{I\subseteq[n]\\|I|=n-t}} \sum_{\tau\in\mathcal{T}_I} \Pr[T_I = \tau]\frac{1}{\binom{n}{n-t}} \cdot \left(\log(\Pr[T_I = \tau]) + \log\frac{1}{\binom{n}{n-t}}\right) \quad - \quad H(\mathsf{SEC}) \\
&= \frac{-1}{\binom{n}{n-t}} \sum_{\substack{I\subseteq[n]\\|I|=n-t}} \sum_{\tau\in\mathcal{T}_I} \Pr[T_I = \tau] \cdot \log\left(\Pr[T_I = \tau]\right) \\
&\qquad\qquad - \quad \frac{1}{\binom{n}{n-t}} \log\frac{1}{\binom{n}{n-t}} \sum_{\substack{I\subseteq[n]\\|I|=n-t}} \sum_{\tau\in\mathcal{T}_I} \Pr[T_I = \tau] \quad - \quad H(\mathsf{SEC}).
\end{aligned}$$

Now by the definition of entropy we have that $-\sum_{\tau\in\mathcal{T}_I}\Pr[T_I=\tau]\cdot\log\left(\Pr[T_I=\tau]\right)=H(T_I)$. Additionally, $\log\frac{1}{\binom{n}{n-t}}=-\log\binom{n}{n-t}$, and $\sum_{\tau\in\mathcal{T}_I}\Pr[T_I=\tau]=1$. Making these substitutions, the previous expression is equal to

$$\begin{aligned}
&\frac{1}{\binom{n}{n-t}}\sum_{\substack{I\subseteq[n]\\|I|=n-t}}H(T_I) \quad + \quad \frac{1}{\binom{n}{n-t}}\log\binom{n}{n-t}\left(\sum_{\substack{I\subseteq[n]\\|I|=n-t}}1\right) \quad - \quad H(\mathsf{SEC})\\
&=\frac{1}{\binom{n}{n-t}}\sum_{\substack{I\subseteq[n]\\|I|=n-t}}H(T_I) \quad + \quad \frac{1}{\binom{n}{n-t}}\log\binom{n}{n-t}\binom{n}{n-t} \quad - \quad H(\mathsf{SEC})\\
&=\frac{1}{\binom{n}{n-t}}\sum_{\substack{I\subseteq[n]\\|I|=n-t}}H(T_I) \quad + \quad \log\binom{n}{n-t} \quad - \quad H(\mathsf{SEC})\\
&=\frac{1}{\binom{n}{n-t}}\sum_{\substack{I\subseteq[n]\\|I|=n-t}}H(T_I),
\end{aligned}$$

since, by (E1), $H(\mathsf{SEC})=\log\binom{n}{n-t}$.

So far we have shown the inequality

$$\left(-\log(1/|\mathcal{M}|+2\epsilon)-H_2(\sqrt{\delta})-2\sqrt{\delta}\log|\mathcal{M}|\right)\le\frac{1}{\binom{n}{n-t}}\sum_{\substack{I\subseteq[n]\\|I|=n-t}}H(T_I),$$

and rearranging gives

$$\binom{n}{n-t}\left(-\log(1/|\mathcal{M}|+2\epsilon)-H_2(\sqrt{\delta})-2\sqrt{\delta}\log|\mathcal{M}|\right)\le\sum_{\substack{I\subseteq[n]\\|I|=n-t}}H(T_I).$$

Now we apply Han's inequality [CT91]: For a random variable $X=(X_1,\ldots,X_n)$ and $1\le q\le n$, we define

$$H_q(X)=\frac{1}{\binom{n-1}{q-1}}\sum_{\substack{I\subseteq[n]\\|I|=q}}H(X|_I)$$

(where $X|_I$ is the random variable $(X_i)_{i\in I}$).

Then Han's inequality states that

$$H_1(X)\ge H_2(X)\ge\cdots\ge H_n(X)=H(X).$$

For our purposes, $q=n-t$ and $X=T_{\mathsf{PRIV}}=(T_1,\ldots,T_n)$ is the transcript restricted to the private wires.

Dividing the previous inequality on both sides by $\binom{n-1}{n-t-1}$ yields

$$\begin{aligned}
\frac{n}{n-t}\Big( -\log(1/|\mathcal{M}|+2\epsilon) - H_2(\sqrt{\delta}) - 2\sqrt{\delta}\log|\mathcal{M}|\Big) &\le \frac{1}{\binom{n-1}{n-t-1}} \sum_{\substack{I\subseteq[n]\\ |I|=n-t}} H(T_I) \\
&= H_{n-t}(T_{\mathsf{PRIV}}) \\
&\le H_1(T_{\mathsf{PRIV}}) \\
&= \sum_{i\in[n]} H(T_i) \\
&\le \sum_{i\in[n]} \mathbb{E}[\text{\# bits transmitted on wire } i] \\
&= \mathbb{E}[\text{total \# bits transmitted on private wires}].
\end{aligned}$$

The last line is exactly $C$, which shows as desired that

$$C \ge \frac{n}{n-t}\Big( -\log(1/|\mathcal{M}|+2\epsilon) - H_2(\sqrt{\delta}) - 2\sqrt{\delta}\log|\mathcal{M}|\Big).$$

□

**Lemma 5.1** *The protocol of Figure 3 is a Weak $(\delta,\delta)$-SMT-PD protocol for $q$ sufficiently large ($\Omega(t/\delta)$).*

*Proof.* If the adversary is passive, it is clear that $\mathcal{R}$ will receive the correct message.

Consider an active adversary. We will look at the worst case, when $n = t+1$ (i.e., only one good wire). Also, we assume the adversary "materially alters" the shares ($(x_i, y_i)$ pairs) on the wires he controls—which is to say, alters them in some way besides simply permuting the $3t$ shares sent by the Sender. (Clearly, simply permuting the shares leads to $\mathcal{R}$ accepting the correct message.) We claim that with overwhelming probability, if the adversary materially alters the shares, then $\mathcal{R}$ will detect corruption.

Let $\widetilde{f}$ denote the polynomial obtained from interpolating the $3t+2$ shares consisting of all shares on corrupted wires, and 2 of the 3 shares on the good wire. Then $\widetilde{f}$ is a polynomial of degree $\le 3t+1$. Note that $\mathcal{R}$ will accept some message precisely if the remaining share on the good wire, which was not used to define $\widetilde{f}$, also lies on $\widetilde{f}$; otherwise he will output "Corruption detected." Now if the adversary materially alters the shares, then he does at least one of the following: (1) alter the value of $f$ at one or more of the evaluation points chosen by $\mathcal{S}$; or (2) alter the set of evaluation points chosen by $\mathcal{S}$. In case (1) it is certainly the case that $\widetilde{f} \ne f$, since they disagree on a point. In case (2), let $\widetilde{x}_i$ be an altered evaluation point which does not appear in the Sender's original list. Observe that since the adversary sees only $3t$ points in a perfect $3t+2$-out-of-$3n$ secret sharing scheme, then regardless of his knowledge of $\alpha = f(0)$, the value of $f$ at any other point remains uniformly distributed over $\mathbb{F}_q$ according to his view. Therefore the probability that $\widetilde{f} = f$ in this case is $\le 1/q$, the probability that $\widetilde{f}(\widetilde{x}_i) = f(\widetilde{x}_i)$, which proves our claim.

Now we prove weak reliability. The situation we want to avoid is when the remaining share lies on $\widetilde{f}$ *and* $\widetilde{f}(0) \ne f(0) = \alpha$, because this is exactly the situation in which $\mathcal{R}$ accepts a false message. To this end, we'll bound the probability that the remaining share lies on $\widetilde{f}$, given that $\widetilde{f} \ne f$.

Next we claim that, in the case that $\widetilde{f} \ne f$, $\mathcal{R}$ detects corruption with overwhelming probability. So assume $\widetilde{f} \ne f$. Two distinct polynomials of degree $\le 3t+1$ can agree on at most $3t+1$ points. Consider the share on the good wire which was not used to interpolate $\widetilde{f}$—let us assume it was $(x_1^{(3)}, f(x_1^{(3)}))$. Conditioned on the set of all other evaluation points chosen by the sender, $x_1^{(3)}$ is distributed randomly among the remaining $q-1-(3t+2) = q-3t-3$ nonzero points in $\mathbb{F}_q$. Thus, since $\widetilde{f}$ and $f$ agree on at most $3t+1$ points,

$$\Pr[\widetilde{f}(x_1^{(3)}) = f(x_1^{(3)}) | \widetilde{f} \ne f] \le \frac{3t+1}{q-3t-3}.$$

The above analysis then shows that the overall probability with which $\mathcal{R}$ detects corruption, in the case of material alteration, is $\geq 1 - 1/q - (3t+1)/(q-3t-3) = 1 - \Theta(t/q)$. Taking $q$ sufficiently large this is $\geq 1 - \delta$, which proves weak reliability.

This also shows $\delta$-privacy. Note that of course if we consider the adversary's view as consisting only of the shares on corrupt wires, then in fact perfect privacy holds, since the secret sharing scheme is perfect. If we also include knowledge of whether $\mathcal{R}$ output a message or "Corruption detected," this only increases the adversary's advantage in distinguishing any two messages $m_0, m_1$ by a negligible amount ($\leq \delta$), since if the adversary does *not* materially alter the shares, then he knows already that $\mathcal{R}$ will output a message; and if he does, then he knows that $\mathcal{R}$ will output corruption detected except with probability $\delta$. □

**Theorem 5.1** *Given an initial shared secret consisting of $O(n^2)$ field elements, $\mathcal{S}$ and $\mathcal{R}$ can communicate* indefinitely *using only the private wires. The probability that one of them will* ever *accept an incorrect message is $\leq t\delta$. Moreover, with probability $\geq 1 - t\delta$, $\mathcal{A}$ gains at most $\delta$ information on each of $t$ different messages, and no information on any other message.*

*Proof.* Let us first argue that an initial secret consisting of $O(n^2)$ field elements is sufficient to realize the scheme described in Section 5 (using our one-round Weak SMT-PD protocol from Figure 3). The shared secret is only used during Fault-Recovery Mode. In $\mathcal{R}$'s first communication after entering Fault-Recovery Mode, $\mathcal{R}$ must encrypt and authenticate a message consisting of a flag, a list of wires, and the entire contents of $\mathcal{S}$'s previous transmission. The size of this message is $1+n+(\log q)3n = O(n \log q)$. In $\mathcal{S}$'s response, he encrypts and authenticates a message consisting of his original message $M_{\mathcal{S}}$ and a list of wires, of total size $\log q + n$, which of course is also $O(n \log q)$. This entire process occurs at most $t$ times, for a total cost of $O(tn \log q) = O(n^2 \log q)$. Therefore if $\mathcal{S}$ and $\mathcal{R}$ share an initial secret consisting of $O(n^2)$ field elements of $\mathbb{F}_q$ using ordinary SMT-PD, they can communicate at least a polynomial number of field elements thereafter, without using the public channel again, before the privacy and reliability errors blow up beyond control.

In fact, as the statement of the theorem indicates, the situation is much better than even that. $\mathcal{S}$ and $\mathcal{R}$ can actually communicate *indefinitely*, not just a polynomial number of times. The reason is that, although the privacy and reliability errors may accumulate, these errors *only occur when the adversary is (materially) active*. And since he is caught with overwhelming probability ($\geq 1 - \delta$) each such time, then with similarly overwhelming probability ($\geq 1 - t\delta$) he is caught the first $t$ times he does it, after which $\mathcal{S}$ and $\mathcal{R}$ are in the clear. □