Two-Round Perfectly Secure Message Transmission with Optimal Transmission Rate*

Nicolas Resch† Chen Yuan‡

February 18, 2021

Abstract

In the model of *Perfectly Secure Message Transmission (PSMT)*, a sender Alice is connected to a receiver Bob via n parallel two-way channels, and Alice holds an ℓ symbol secret that she wishes to communicate to Bob. There is an unbounded adversary Eve that controls t of the channels, where $n = 2t + 1$. Eve is able to corrupt any symbol sent through the channels she controls, and furthermore may attempt to infer Alice's secret by observing the symbols sent through the channels she controls. The transmission is required to be (a) *reliable*, i.e., Bob must always be able to recover Alice's secret, regardless of Eve's corruptions; and (b) *private*, i.e., Eve may not learn anything about the Alice's secret. We focus on the two-round model, where Bob is permitted to first transmit to Alice, and then Alice responds to Bob.

In this work we provide tight upper and lower bounds for the PSMT model when the length of the communicated secret ℓ is asymptotically large. Specifically, we first construct a protocol that allows Alice to communicate an ℓ symbol secret to Bob by transmitting at most $2(1 + o(1))n\ell$ symbols. We complement this with a lower bound showing that $2n\ell$ symbols are necessary for Alice to privately and reliably communicate her secret. Thus, we completely determine the optimal transmission rate in this regime, even up to the leading constant.

1 Introduction

Background. Perfectly secure message transmission (PSMT) was first introduced by Dolev et al. in [DDWY93]. This problem involves two parties, the sender Alice and the receiver Bob. Alice wishes to communicate a secret to Bob over n parallel channels in the presence of a computationally unbounded adversary Eve. Eve is able to take control of up

*NR is partially supported by ERC H2020 grant No.74079 (ALGSTRONGCRYPTO). Part of this work was done while CY was at Centrum Wiskunde & Informatica.

†Cryptology Group, Centrum Wiskunde & Informatica. nar@cwi.nl

‡School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. chen_yuan@sjtu.edu.cn

to t channels in such a way that she can listen to and/or overwrite the message passing through these t corrupted channels. Eve considered here is *static*, i.e., she chooses up to t channels to corrupt before the protocol and will not change corrupted channels during the protocol. The goal of PSMT is to devise a procedure permitting Alice and Bob to communicate the secret reliably and privately. More precisely, it is guaranteed that Bob always completely recovers the secret (*reliability*) and Eve learns absolutely nothing about the secret (*privacy*). PSMT can be done in multiple communication rounds. During each round, one party acts as a sender and another party acts as a receiver. They are not allowed to change their roles in one round. It is clear that for $t > n/2$, PSMT is not possible, regardless of how many rounds the protocol uses. One can treat all the message transmitted over these n channels as a codeword of length n. Assume $\mathbf{c}_1$ represents the secret 1 and $\mathbf{c}_0$ represents the secret 0 that Alice wants to communicate to Bob. Since the distance of these two codewords is at most n and the number of errors t is more than the half the distance between $\mathbf{c}_1$ and $\mathbf{c}_0$, unique decoding is not possible.

The original paper in [DDWY93] showed that one-round PSMT is possible if $n \geq 3t+1$. The same paper also showed that PSMT is possible when $n \geq 2t+1$ if two or more rounds are performed. There have since been a number of efforts to devise improved PSMT protocols in various settings. The most challenging case is two-round PSMT with $n = 2t+1$ channels. To measure the performance of a PSMT protocol in this case, we use the metric of *transmission rate*, which is the total number of bits transmitted divided by the length (in bits) of the secret communicated.

In what follows, we focus on the case that $n = 2t + 1$. Sayeed and Abu-Amara [SA96] first presented a two-round PSMT achieving transmission rate $O(n^3)$. Agarwal et al. [ACdH06] further improved it to $O(n)$ which is asymptotically optimal as the lower bound of n was proved in [SNR04]. However, implementing this protocol requires an inefficient exponential-time algorithm. A breakthrough was achieved by Kurosawa and Suzuki [KS08] whose protocol achieves transmission rate $6n$, and can be run in the polynomial time. Inspired by this protocol, Spini and Zémor [SZ16] further reduced the transmission rate to $5n$, and moreover their protocol is arguably simpler than those that preceded it. They also answer in the affirmative an open problem posed in [KS08] of whether it is possible to achieve $O(n)$ transmission rate for a secret of size at most $O(n^2 \log n)$.

Hence, in reviewing the literature on PSMT, we note that the only known lower bound on the transmission rate for two-round PSMT is n, while the current state-of-the-art construction in [SZ16] achieves transmission rate $5n$. While both bounds are $\Theta(n)$, there is still a gap of $4n$ between the lower bound and the upper bound. In this work, we close this gap by showing that the optimal transmission rate is exactly $2n$.

Our Results. Our results are two-fold. Our first contribution is a two-round PSMT protocol with transmission rate $2(1 + o(1))n$.[1] This protocol improves over the state-of-the-art protocol in [SZ16] by $3n$. Furthermore, our protocol reaches this transmission rate when Alice and Bob merely communicate an $\omega(n \log n)$-bit secret, and moreover achieves transmission rate $O(n)$ when they communicate an $\Omega(n \log n)$-bit secret as in [SZ16].

Our second contribution is a lower bound on any protocol for two-round PSMT. Specifically, we show that Alice and Bob have to transmit at least $2n\ell$ bits so as to securely communicate an ℓ-bit secret. Note that this lower bound implies that our two-round PSMT protocol actually achieves the optimal transmission rate. In this sense, we settle the optimal transmission rate for two-round PSMT. As a byproduct, we also show that communicating a 1-bit secret via any two-round PSMT requires transmitting at least $n+n \log n$ bits, which also improves upon the previous lower bound of n.

Our Techniques. As mentioned above, we obtain tight upper and lower bounds for communicating an ℓ-bit secret in the model of two-round PSMT. We start by outlining the upper bound proof.

Upper Bound. For the upper bound, we construct a two-round PSMT protocol achieving transmission rate $\sim 2n$. Instead of presenting our optimal protocol immediately, we first present a simplified protocol which allows for communicating a $\log n$ bit secret securely, which we view as a symbol $m \in \mathbb{F}_q$ with $q \geq n$.

Bob first sends $t + 1$ codewords $\mathbf{c}_1, \ldots, \mathbf{c}_{t+1}$ which are picked independently and uniformly at random from a $[n, t + 1, n - t]$[2] Reed-Solomon code over $\mathbb{F}_q$. Alice receives the corrupted codewords $\tilde{\mathbf{c}}_i = \mathbf{c}_i + \mathbf{e}_i$. She uses the parity check matrix of this Reed-Solomon code to calculate the syndrome vectors $\mathbf{H}\tilde{\mathbf{c}}_i = \mathbf{s}_i$. Since Eve can corrupt at most t channels, there exist coefficients $\lambda_1, \ldots, \lambda_{t+1} \in \mathbb{F}_q$, not all zero, such that $\sum_{i=1}^{t+1} \lambda_i \mathbf{s}_i = \mathbf{0}$. From this one can show $\sum_{i=1}^{t+1} \lambda_i \mathbf{e}_i = \mathbf{0}$ and thus $\sum_{i=1}^{t+1} \lambda_i \mathbf{c}_i = \sum_{i=1}^{t+1} \lambda_i \tilde{\mathbf{c}}_i$; we denote this codeword by $\bar{\mathbf{c}}$.

Let $\mathbf{h} \in \mathbb{F}_q^n$ be a vector of weight n that is not orthogonal to the $[n, t + 1, n - t]$ Reed-Solomon code. Alice broadcasts[3] $\lambda_1, \ldots, \lambda_{t+1}$ together with $\langle \mathbf{h}, \bar{\mathbf{c}} \rangle + m$ to Bob where m is the secret; $\langle \mathbf{h}, \bar{\mathbf{c}} \rangle$ is a mask for the secret. Bob first uses $\lambda_1, \ldots, \lambda_{t+1}$ to recover $\bar{\mathbf{c}}$ and then obtains m by removing the mask $\langle \mathbf{h}, \bar{\mathbf{c}} \rangle$ from the last broadcasted message.

The privacy analysis is quite straightforward. First, Eve can calculate $\lambda_1, \ldots, \lambda_{t+1}$ by herself since each $\mathbf{s}_i = \mathbf{H}\mathbf{e}_i$ is available to her. This means we can reduce the privacy argument to the last message $\langle \mathbf{h}, \bar{\mathbf{c}} \rangle + m$ which is an immediate consequence of the $[n, t+1, n-t]$

[1] Here and throughout, $o(1)$ denotes a quantity which tends to 0 as the length of the secret increases, holding n fixed.

[2] A $[n, k, d]$ Reed-Solomon code has length n, dimension k and distance $d = n - k + 1$.

[3] To broadcast $\lambda \in \mathbb{F}_q$, Alice sends λ through every channel; note that Bob can easily recover λ by choosing the majority symbol.

Reed-Solomon code we use. This protocol allows Alice and Bob to securely communicate the secret $m \in \mathbb{F}_q$ at the cost of $n^2 \log n$ communication complexity (measured in bits).

Observe that if the syndrome space spanned by $\mathbf{s}_1, \ldots, \mathbf{s}_{t+1}$ has dimension r, Alice only needs to send $r+1$ coefficients instead of $t+1$ so as to share a common codeword with Bob. This observation leads to our most efficient two-round PSMT.

We now present the general protocol. Assume Alice and Bob want to communicate an $\ell \log n$-bit secret securely. We first split it into ℓ secrets $m_1, \ldots, m_\ell$, each of size $\log n$, which we think of as lying in $\mathbb{F}_q$ with $q \geq n$. Bob first sends $t+\ell$ codewords $\mathbf{c}_1, \ldots, \mathbf{c}_{t+\ell}$ which are picked independently and uniformly at random from a $[n, t+1, n-t]$ Reed-Solomon code over $\mathbb{F}_q$. Alice receives the corrupted codewords $\tilde{\mathbf{c}}_i = \mathbf{c}_i + \mathbf{e}_i$ for $i \in [t+\ell]$. She uses the parity-check matrix of this Reed-Solomon code to calculate the syndrome vectors $\mathbf{H}\tilde{\mathbf{c}}_i = \mathbf{s}_i$.

Assume that the space spanned by $\mathbf{s}_1, \ldots, \mathbf{s}_{t+\ell}$ has dimension r. Let $S \subset [t+\ell]$ be the index set of $\mathbf{s}_i$ that form the basis of this syndrome space. Without loss of generality, let us assume $S = \{t+\ell-r+1, t+\ell-r+2, \ldots, t+\ell\}$, the last r elements of $[t+\ell]$. For each $i \in [\ell]$, there exist not all zero coefficients λ_{ij} for $j \in S$ such that $\mathbf{s}_i = \sum_{j \in S} \lambda_{ij} \mathbf{s}_j$. In analogy to what we did in the simpler protocol, we let $\bar{\mathbf{c}}_i := \mathbf{c}_i - \sum_{j \in S} \lambda_{ij} \mathbf{c}_j = \tilde{\mathbf{c}}_i - \sum_{j \in S} \lambda_{ij} \tilde{\mathbf{c}}_j$.

Before entering into the second round, we need to do the same thing as in [SZ16] so as to reduce communication complexity. That is, we spot a corrupted codeword with error weight at least r[4] by applying linear operations to the $\tilde{\mathbf{c}}_j$'s. We take a different approach which simplifies the argument; for details, please see Algorithm 2. Assume that Alice has managed to spot a corrupted codeword $\tilde{\mathbf{c}} = \sum_{j \in S} \lambda_j \tilde{\mathbf{c}}_j$ with error weight at least r. Alice first broadcasts the index set S together with λ_j for $j \in S$ and $\tilde{\mathbf{c}}$ to Bob. Then, Alice uses an $[n, r+1, n-r]$ Reed-Solomon code to encode message $\lambda_{ij}, j \in S$ and $\langle \mathbf{h}, \bar{\mathbf{c}}_i \rangle + m_i$ for $i \in [\ell]$.

Once Bob receives the messages, he can correctly recover the index set S and λ_j for $j \in S$ and $\tilde{\mathbf{c}}$ as these messages are broadcasted. By applying the same linear operation on the codewords in S, Bob will obtain $\mathbf{c} = \sum_{j \in S} \lambda_j \mathbf{c}_j$ which is at least distance r away from $\tilde{\mathbf{c}}$. Bob then ignores the r channels that cause the inconsistency between $\mathbf{c}$ and $\tilde{\mathbf{c}}$. Bob can decode the rest of Alice's messages correctly which were encoded by the $[n, r+1, n-r]$ Reed-Solomon code since Eve can only cause r erasures and $t-r$ errors now. The recovery procedure is exactly the same as in the first protocol. The privacy argument is also quite straightforward. First of all, the coefficients λ_{ij} can be computed by Eve on her own. Then, the privacy of the secret m_i can be reduced to the privacy of $\mathbf{c}_i$ for $i \in [r]$ which is guaranteed by the $[n, t+1, n-t]$ Reed-Solomon code.

[4]Note that Eve has to corrupt at least r channels so as to make the syndrome space have dimension r. To simplify our discussion here, we assume $r \leq \frac{t}{3}$; otherwise the protocol will be little more complicated. Specifically, Alice first broadcasts a corrupted codeword with error weight $\frac{t}{3}$ and then sends all corrupted codewords in S to Bob via a $[n, \frac{t}{3}, n-\frac{t}{3}+1]$ Reed-Solomon code. This extra cost will not affect transmission rate as we can amortize it out by communicating $\ell \log n = \Omega(n \log n)$-bits secret. The interested reader can find the details in our proof.

It remains to bound the communication complexity. The first-round communication complexity is $(\ell+t)n\log n$. The second-round communication complexity is $nr\log(t+\ell)+(r+n)n\log n+\frac{n}{r+1}(r+1)\ell\log n$. Thus, the transmission rate is $2n+O(\frac{n^2}{\ell})$ which becomes $2(1+o(1))n$ if Alice communicates to Bob an $\ell\log n=\omega(n\log n)$-bit secret.

Lower Bound. Assume that Alice wants to communicate an ℓ-bits secret s securely to Bob via a two-round PSMT. In the first round, Bob sends a vector $\mathbf{a}=(a_1,\ldots,a_n)$ to Alice, and Alice receives a corrupted vector $\tilde{\mathbf{a}}$. Based on $\tilde{\mathbf{a}}$ and the secret $s\in[2^\ell]$, Alice sends back a vector $\mathbf{b}=(b_1,\ldots,b_n)$ to Bob. On receiving the corrupted vector $\tilde{\mathbf{b}}$, Bob tries to decode the correct secret s with the help of $\mathbf{a}$. Since PSMT is reliable, we can assume that the algorithm used by Alice and the algorithm used by Bob to decode is deterministic. That means $\mathbf{b}$ is uniquely determined by $\tilde{\mathbf{a}}$ and s.

Next, we design an adversary to force Alice and Bob to transmit at least $2\ell n$ bits so as to securely send the ℓ-bit secret. In the first round, Eve does nothing. That means Alice will receive a correct vector $\mathbf{a}$. Moreover, she has no idea which channels are corrupted. She must therefore assume that any subset of t channels are *equally likely* to be corrupted in the second round. Given $\mathbf{a}$, Alice has to use a code of distance $n=2t+1$ to encode the secret $s\in[2^\ell]$ so as to achieve reliability. This gives a lower bound ℓn on the second round communication complexity.

Meanwhile, from this argument, we can see that Eve knows exactly $\mathbf{b}$ if she does nothing in the first round. To achieve perfect security, Alice and Bob must share a private key of size ℓ in the first round. We also notice that the message sent by Bob in the first round is *independent of* Eve's strategy, which means that the lower bound on the communication complexity of the first round can be applied to the case Eve does nothing in the first round.

We construct a secret sharing scheme by treating $\mathbf{a}=(a_1,\ldots,a_n)$ as n shares and this private key as a secret. Since the adversary can listen to t channels, this means any t shares should learn nothing of this secret. This implies that such a secret sharing scheme has t-privacy. We next show that such secret sharing scheme must have $t+1$-reconstruction.

Let $\mathbf{a}_1$ be any share vector of secret s_1 and $\mathbf{a}_2$ be any share vector of secret s_2. If $\mathbf{a}_1$ and $\mathbf{a}_2$ are within distance t, the adversary may inject t errors to change $\mathbf{a}_1$ to $\mathbf{a}_2$. Then, Alice and Bob will share a wrong key and thus Alice fails to recover the correct secret. This implies the share vectors associated with different secrets must have distance $t+1$ and thus any $n-(t+1)+1=t+1$ shares can reconstruct the secret. As we have t-privacy and $t+1$-reconstruction, our secret sharing scheme is threshold, which implies that the number of bits communicated in the first round is also at least ℓn. Putting it all together, we obtain the desired $2\ell n$ lower bound on the communication of the two-round PSMT.

2 Preliminaries

Notations. For an integer $n \geq 1$, we denote $[n] := \{1, 2, \ldots, n\}$. By default, log denotes the base-2 logarithm.

Throughout, $\mathbb{F}_q$ denotes the finite field with q elements, for q a prime power. We let n denote the number of channels through which Alice and Bob may communicate and t the number of channels Eve may corrupt; we focus exclusively on the $n = 2t + 1$ case. The complexity measure of a protocol that concerns us is its *transmission rate*, defined as the total number of symbols communicated divided by the number of symbols of the transmitted secret. The length of the transmitted secret is denoted by ℓ.

Remark 2.1. As usual, a *bit* refers to an element of $\{0, 1\}$, while in this work, a *symbol* refers to an element from the field $\mathbb{F}_q$, and we will need $q \geq n$. While it is most natural to measure the total communication in bits, as our protocols will involve transmitting elements of $\mathbb{F}_q$ it is more convenient for us to talk about the number of symbols transmitted. Note that when we compute the transmission rate and we assume the length of the secret is a growing parameter, whether we measure the communication in bits or symbols does not matter. However, when we present our lower bound proof in Section 4 it will be most convenient for us to talk about bits.

Codes. As in previous works, our protocols rely crucially on linear codes with desirable properties. For two vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{F}_q^n$, the *(Hamming) distance* between them is $d(\mathbf{x}, \mathbf{y}) := |\{i \in [n] : x_i \neq y_i\}|$. Given a vector $\mathbf{x}$ and a subset $\mathcal{Y} \subseteq \mathbb{F}_q^n$ we denote $d(\mathbf{x}, \mathcal{Y}) := \min\{d(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in \mathcal{Y}\}$. The *(Hamming) weight* of a vector is $\mathrm{wt}(\mathbf{x}) := d(\mathbf{x}, \mathbf{0})$. The *support* of $\mathbf{x}$ is $\mathrm{supp}(\mathbf{x}) := \{i \in [n] : x_i \neq 0\}$. Note that $\mathrm{wt}(\mathbf{x}) = |\mathrm{supp}(\mathbf{x})|$ and $d(\mathbf{x}, \mathbf{y}) = |\mathrm{supp}(\mathbf{x} - \mathbf{y})|$. By a *(linear) code*, we refer to a linear subspace $\mathcal{C} \leq \mathbb{F}_q^n$; n is the *block-length*, $k = \dim(\mathcal{C})$ is the *dimension* and $d = \min\{\mathrm{wt}(\mathbf{c}) : \mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\}\}$ is the *(minimum) distance*. We refer to such a code as an $[n, k, d]_q$ code.

A code is called *maximimum distance separable (MDS)* if $d = n-k+1$. Such codes exist whenever $q \geq n$ and are furnished by the well-known Reed-Solomon (RS) codes defined via the evaluations of degree $\leq k - 1$ polynomials. However, in this work, we will not directly use the specific structure of RS codes,[5] so we will state our results for arbitrary linear MDS codes.

Any linear code $\mathcal{C}$ may be described as the kernel of a matrix, i.e., $\mathcal{C} = \{\mathbf{x} \in \mathbb{F}_q^n : \mathbf{H}\mathbf{x} = \mathbf{0}\}$. Such a matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ is called a *parity-check matrix*.

Given two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^n$ we define their *inner product* via $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^n x_i y_i$. We will need the following lemma from [SZ16]. It states that there exists an MDS code $\mathcal{C} \leq \mathbb{F}_q^n$ of dimension t for $n = 2t + 1$ for which one can find a vector $\mathbf{h} \in \mathbb{F}_q^n$ such that, even

[5] Although in order to implement the protocol efficiently we will use the existence of efficient encoding and decoding algorithms for RS codes.

once t coordinates are revealed from a codeword $\mathbf{c} \in \mathcal{C}$, the inner-product $\langle \mathbf{h}, \mathbf{c} \rangle \in \mathbb{F}_q$ is completely unconstrained.

Lemma 2.2 (Lemma 1 from [SZ16])**.** *For any n and any $t < n$ there exists a linear MDS code $\mathcal{C}$ of parameters $[n, t+1, n-t]$ and a vector $\mathbf{h} \in \mathbb{F}_q^n$ is such that given a random codeword $\mathbf{c} \in \mathcal{C}$, the scalar product $\langle \mathbf{h}, \mathbf{c} \rangle$ is completely undetermined even when any t symbols of $\mathbf{c}$ are known.*

Remark 2.3. We note that any such vector $\mathbf{h}$ must not lie in the dual of $\mathcal{C}$, and moreover that it must have weight at least $t+1$.

Broadcast. Next, observe that since Eve controls at most $t < n/2$ of the channels, if Alice transmits the same symbol through all n channels, then Bob can always recover Alice's intended symbol by choosing the majority symbol. Of course, such a procedure does not guarantee any privacy, i.e., Eve will always learn the symbol Alice transmits to Bob.

2.1 Pseudobases

An important technical tool in our protocols are *pseudobases*, as introduced in the work of Kurosawa and Suzuki [KS08]. Before providing the definition, we explain their utility. Consider the scenario where Bob has sent a codeword $\mathbf{c} \in \mathcal{C}$ to Alice by sending the i-th coordinate c_i through the $i-$th channel. In order to guarantee privacy, as Eve can observe t of the channels, it must be that $\dim \mathcal{C} \geq t+1$. However, by the Singleton bound, that forces the distance of $\mathcal{C}$ to be at most $n-(t+1)+1 = n-t = t+1$, which means that Bob can uniquely decode Alice's transmission only if Eve introduces $\leq t/2$ errors. However, as Eve can introduce up to t errors, it appears that we do not have an effective means of enforcing reliability.

However, consider the following scenario: instead of sending a single codeword through the channel in this way, Bob sends many codewords $\mathbf{c}_1, \ldots, \mathbf{c}_r$. Privacy is preserved so long as the transmissions are not correlated in any way (say, each one is sampled independently and uniformly at random). However, Alice now has an advantage in decoding: all of the corruptions introduced by Eve are confined to the same set of t coordinates. The idea is to exploit this fact to allow Alice and Bob to agree on some codeword $\bar{\mathbf{c}}$ of which Eve knows at most t coordinates (which in turn means that $\langle \mathbf{h}, \bar{\mathbf{c}} \rangle$ can effectively mask the secret m). Using the concept of pseudobases, it turns out that this is possible (so long as the distance of $\mathcal{C}$ is at least $t+1$, as is the case when $\mathcal{C}$ is MDS).

We now provide the formal definition of a pseudobasis.

Definition 2.4 (Pseudobasis [KS08])**.** Let $\mathbf{y}_1, \ldots, \mathbf{y}_s \in \mathbb{F}_q^n$ be vectors. A *pseudobasis* for $\mathbf{y}_1, \ldots, \mathbf{y}_s$ is a subcollection $\mathbf{y}_{i_1}, \ldots, \mathbf{y}_{i_r}$ with $1 \leq i_1 < \cdots < i_r \leq s$ such that $\mathbf{H}\mathbf{y}_{i_1}, \ldots, \mathbf{H}\mathbf{y}_{i_r} \in \mathbb{F}_q^{n-k}$ is a basis for the linear space $\operatorname{span}\{\mathbf{H}\mathbf{y}_1, \ldots, \mathbf{H}\mathbf{y}_s\}$.

In other words, one computes a basis for the space spanned by $\mathbf{Hy}_1, \ldots, \mathbf{Hy}_s \in \mathbb{F}_q^{n-k}$, and then the preimage of the basis vectors in $\mathbb{F}_q^n$ provides a pseudobasis. Observe that, given access to $\mathbf{H}$, such a pseudobasis can be found in time polynomial in n, and furthermore that it consists of at most $n-k$ vectors.

Remark 2.5. Note that if we have a code $\mathcal{C} \leq \mathbb{F}_q^n$ with parity-check matrix $\mathbf{H}$ and we write $\mathbf{y}_i = \mathbf{c}_i + \mathbf{e}_i$ for each $i \in [s]$ with $\mathbf{c}_i \in \mathcal{C}$, then as

$$\mathbf{Hy}_i = \mathbf{H}(\mathbf{c}_i + \mathbf{e}_i) = \mathbf{Hc}_i + \mathbf{He}_i = \mathbf{He}_i \ ,$$

we conclude that $\mathbf{y}_{i_1}, \ldots, \mathbf{y}_{i_r}$ forms a pseudobasis for $\mathbf{y}_1, \ldots, \mathbf{y}_s$ if and only if $\mathbf{e}_{i_1}, \ldots, \mathbf{e}_{i_r}$ forms a pseudobasis for $\mathbf{e}_1, \ldots, \mathbf{e}_s$.

This observation will be crucial for us in our privacy analysis. We will be in the scenario that Alice has received potentially corrupted codewords from Bob, which we write as $\tilde{\mathbf{c}}_i = \mathbf{c}_i + \mathbf{e}_i$, where $\mathbf{e}_i$ denotes the errors introduced by Eve. Alice will then broadcast some information about a pseudobasis for her received vectors to Bob. This does not leak any information to Eve, as she could have computed the same pseudobasis from the error vectors $\mathbf{e}_i$ that she knows.

3 The Protocol

In this section, we present our protocol which allows Alice to privately and reliably transmit an ℓ symbol secret $(m_1, \ldots, m_\ell) \in \mathbb{F}_q^\ell$ to Bob. In order to ease readability, we present two simplifications of our full protocol first before presenting the full construction. The first construction, presented in Section 3.1, allows Alice to transmit a one symbol secret $m \in \mathbb{F}_q$. Despite being fairly simple, it already introduces the most crucial idea for our protocol, which is a method for Alice and Bob to agree on a random codeword that is not completely revealed to Eve.

Next, in Section 3.2, we show how to generalize the protocol to the case of $\ell \geq 1$, and achieve communication rate $(4 + o(1))n$. Intuitively, this requires Alice and Bob to agree on ℓ random codewords that are not completely known to Eve. In order to guarantee small transmission rate, we need a few more tricks. As in [SZ16], one useful technique we employ is a method for Alice to find a vector which indicates many of the channels that Eve is corrupting, allowing Bob to safely ignore those channels.[6] Informally, this transforms symbol corruptions into erasures, and erasures are easier to recover from. In particular, Alice can encode her data with a code of higher rate and Bob will still be able to uniquely-decode. To get our final protocol achieving transmission rate $(2 + o(1))n$, we note that we only need to do something different if Eve invests many corruptions in the first round.[7] In order to handle this, we ask Alice to send a bit more information to Bob to indicate a

[6] There is a procedure with the same guarantee in [SZ16]; however, we believe our procedure is simpler, and moreover does not use the specific structure of RS codes.

[7] More precisely, if the dimension of the syndrome space exceeds $t/3$.

larger number of corrupted channels, which transforms more of the symbol corruptions into erasures in the subsequent transmissions, and hence allows Alice to use an error-correcting code of higher rate. We describe the necessary modifications in Section 3.3.

Throughout, $\mathcal{C} \leq \mathbb{F}_q^n$ denotes an MDS code of dimension $t+1$ and $\mathbf{h} \in \mathbb{F}_q^n$ a vector satisfying the conclusion of Lemma 2.2. Also, $\mathbf{H} \in \mathbb{F}_q^{t \times n}$ denotes a parity-check matrix for $\mathcal{C}$. Lastly, we denote by $E \subseteq [n]$ the set of t channels that Eve controls. Of course, this set is unknown to Alice and Bob; we introduce this notation exclusively for the analysis.

3.1 A Simple Protocol for $\ell = 1$

We begin by describing a simple protocol which allows Alice to transmit one secret symbol $m \in \mathbb{F}_q$ to Bob.

Algorithm 1 A first protocol for transmitting a one symbol secret $m \in \mathbb{F}_q$.

1: **procedure** ROUND 1: BOB TRANSMITS
2: Bob samples $\mathbf{c}_1, \ldots, \mathbf{c}_{t+1} \in \mathcal{C}$ independently and uniformly at random.
3: For $j = 1, \ldots, t+1$, Bob transmits the i-th coordinate of $\mathbf{c}_j$ through the i-th channel.
4: **end procedure**
5: **procedure** ROUND 2: ALICE TRANSMITS
6: For $j = 1, \ldots, t+1$, Alice receives the vectors $\tilde{\mathbf{c}}_j$ where $d(\mathbf{c}_j, \tilde{\mathbf{c}}_j) \leq t$.
7: For $j = 1, \ldots, t+1$, Alice computes $\mathbf{s}_j = \mathbf{H}\tilde{\mathbf{c}}_j \in \mathbb{F}_q^t$.
8: Alice finds a coordinate $p \in [t+1]$ such that $\mathbf{s}_p \in \operatorname{span}\{\mathbf{s}_j : j \neq p\}$.
9: Alice finds $\lambda_j \in \mathbb{F}_q$ for $j \in [t+1] \setminus \{p\}$ such that $\mathbf{s}_p = \sum_{j \neq p} \lambda_j \mathbf{s}_j$.
10: $\bar{\mathbf{c}} \leftarrow \tilde{\mathbf{c}}_p - \sum_{j \neq p} \lambda_j \tilde{\mathbf{c}}_j$
11: Alice broadcasts p, $(\lambda_j : j \neq p)$ and the symbol $m + \langle \mathbf{h}, \bar{\mathbf{c}} \rangle$.
12: **end procedure**
13: **procedure** OUTPUT PHASE
14: Bob receives p, $(\lambda_j : j \neq p)$ and the symbol m'.
15: $\mathbf{c}' \leftarrow \mathbf{c}_p - \sum_{j \neq p} \lambda_j \mathbf{c}_j$
16: **return** $m' - \langle \mathbf{h}, \mathbf{c}' \rangle$.
17: **end procedure**

We now sketch why the protocol indeed yields a PSMT.

Reliability. First, observe that we may choose $\lambda_j \in \mathbb{F}_q$ for $j \in [t+1] \setminus \{p\}$, as $\mathbf{s}_1, \ldots, \mathbf{s}_{t+1} \in \mathbb{F}_q^t$ are $t+1$ vectors in a t-dimensional space, and therefore satisfy a nontrivial linear dependence. Hence, Line 9 from the algorithm is justified.

The important observation is that since the code $\mathcal{C}$ has distance $t+1$, we have $\mathbf{c}' = \bar{\mathbf{c}}$.

Indeed, first note that $\bar{\mathbf{c}} \in \mathcal{C}$, as

$$\mathbf{H}\bar{\mathbf{c}} = \mathbf{H}\left(\tilde{\mathbf{c}}_p - \sum_{j \neq p} \lambda_j \tilde{\mathbf{c}}_j\right) = \mathbf{H}\tilde{\mathbf{c}}_p - \sum_{j \neq p} \lambda_j \mathbf{H}\tilde{\mathbf{c}}_j = \mathbf{s}_p - \sum_{j \neq p} \lambda_j \mathbf{s}_j = \mathbf{0} \ .$$

Now, if $E \subseteq [n]$ denotes the channels that the adversary controls, then the coordinates on which each $\mathbf{c}_j$ can disagree with $\tilde{\mathbf{c}}_j$ are confined to the set E. Thus, the support of $\left(\mathbf{c}_p - \sum_{j \neq p} \lambda_j \mathbf{c}_j\right) - \left(\tilde{\mathbf{c}}_p - \sum_{j \neq p} \lambda_j \tilde{\mathbf{c}}_j\right)$ is also contained in the set E. As $|E| \leq t$, we conclude that the codewords $\mathbf{c}' = \mathbf{c}_p - \sum_{j \neq p} \lambda_j \mathbf{c}_j$ and $\tilde{\mathbf{c}} = \tilde{\mathbf{c}}_p - \sum_{j \neq p} \lambda_j \tilde{\mathbf{c}}_j$ are distance at most t from one another; as $\mathcal{C}$ has distance $t+1$, they must be the same vector.

Thus, in particular, $\langle \mathbf{h}, \mathbf{c}' \rangle = \langle \mathbf{h}, \bar{\mathbf{c}} \rangle$, so $m' - \langle \mathbf{h}, \mathbf{c}' \rangle = m + \langle \mathbf{h}, \bar{\mathbf{c}} \rangle - \langle \mathbf{h}, \mathbf{c}' \rangle = m$, i.e., Bob returns Alice's intended secret m.

Privacy. As Eve can only see t symbols from each transmitted codeword and the code $\mathcal{C}$ has dimension $t+1$ and is MDS, Eve learns only t symbols from $\mathbf{c}_1, \ldots, \mathbf{c}_{t+1}$. Now, after seeing $(p, \lambda_j : j \neq p)$, Eve knows that

$$\mathbf{0} = \mathbf{s}_p - \sum_{j \neq p} \lambda_j \mathbf{s}_j = \mathbf{H}\tilde{\mathbf{c}}_p - \sum_{j \neq p} \lambda_j \mathbf{H}\tilde{\mathbf{c}}_j = \mathbf{e}_p - \sum_{j \neq p} \lambda_j \mathbf{H}\mathbf{e}_j \ .$$

However, as she already knows $\mathbf{e}_1, \ldots, \mathbf{e}_{t+1}$ and $\mathbf{H}$, she does not learn anything from this transmission. In particular, Eve still only knows t symbols of $\mathbf{c}' = \bar{\mathbf{c}} = \tilde{\mathbf{c}}_p - \sum_{j \neq p} \lambda_j \tilde{\mathbf{c}}_j$ which is a codeword distributed uniformly at random in C, and so Lemma 2.2 guarantees that Eve has no information on $\langle \mathbf{h}, \bar{\mathbf{c}} \rangle$. Thus, even after observing $m + \langle \mathbf{h}, \bar{\mathbf{c}} \rangle$, she has no information on m, as desired.

Communication Cost. In the first round, Bob transmits $(t+1)n \sim n^2/2$ symbols. In the second round, Alice transmits $\log_q(t+1) + tn + n \sim n^2/2$ symbols. Hence, to communicate a single symbol, the total communication requirement of Algorithm 1 is $\sim n^2$. In terms of bits, as we require $q \geq n$, we conclude that Alice and Bob must transmit $\sim n^2 \log n$ bits.

3.2 A Protocol with $(4 + o(1))n$ Transmission Rate

In this subsection, we provide a protocol that will allow Alice to transmit an ℓ symbol secret to Bob requiring only $\sim 4n\ell$ symbols to be communicated. We begin by outlining some of the new ingredients we need.

Generalized Broadcast. One technique that we will use in our protocol is *generalized broadcast*, as introduced in previous works. The situation that motivates the idea of generalized broadcast is the following: imagine that in some way, Bob has become aware that Eve is controlling some set $R \subseteq [n]$ of the channels. Then, when decoding a transmission

from Alice, he can replace the symbols he receives through the channels in R by an erasure symbol. Thus, instead of decoding from t symbol corruptions, he only has to perform the easier task of decoding from $t - r$ symbol corruptions and r erasures, where $r = |R|$.

In particular, to uniquely decode from t errors where $n = 2t + 1$, if Alice wants to guarantee that the codeword she transmits can be uniquely-decoded by Bob, then she must use a code with distance $2t + 1 = n$: by the Singleton bound, she must use an MDS code of dimension 1, i.e., she can only send a single symbol. A natural example of a dimension 1 MDS code is the repitition code: this precisely recovers broadcast as introduced earlier.

However, if Bob knows a subset R as above, then he can uniquely decode so long as the code has distance at least $2(t - r) + r + 1 = n - r$. Thus, if Alice uses an MDS code of dimension $r + 1$, Bob can recover her intended transmission. We refer to this as r-generalized broadcast, which we now formally define.

Definition 3.1 (Generalized Broadcast)**.** For an integer $r \geq 0$, *r-generalized broadcast* refers to the procedure where Alice uses an $[n, r + 1, n - r]_q$ code $\mathcal{C}_r$ to transmit $r + 1$ symbols $(x_1, \ldots, x_{r+1}) \in \mathbb{F}_q^{r+1}$ by encoding the message $(x_1, \ldots, x_{r+1})$ into a codeword $\mathbf{c} \in \mathcal{C}_r$, and sending the i-th symbol of $\mathbf{c}$ through the i-th channel for each $i \in [n]$.

For succinctness, we write Alice *r-broadcasts* $(x_1, \ldots, x_{r+1})$ to indicate that Alice uses the r-generalized broadcast to transmit the data $(x_1, \ldots, x_{r+1})$ to Bob.

Remark 3.2. Assuming Alice and Bob communicate with a dimension $r+1$ Reed-Solomon code, then both encoding the message and decoding from r erasures and $t - r$ symbol corruptions can be done in polynomial time [WB86].

Thus, r-generalized broadcast allows Alice to reliably transmit $r+1$ times more information to Bob than standard (i.e., 0-)broadcast, which can greatly improve the transmission rate of the protocol if r is sufficiently large.

Finding a Set of Corrupted Channels. In light of the above discussion, we would like to allow Bob to find a large set of corrupted channels. For general ℓ, we will have Bob transmit $t + \ell$ uniformly random codewords in the first round, and Alice receives the corrupted codewords $\tilde{\mathbf{c}}_j = \mathbf{c}_j + \mathbf{e}_j$, where the support of each $\mathbf{e}_j$ is contained in the t channels Eve controls, E.

Now, if Alice were aware that $\mathbf{e}_j$ has large weight for some j, then she could just broadcast $\tilde{\mathbf{c}}_j$ and the index j to Bob. Bob could then compute the set $\mathrm{supp}(\tilde{\mathbf{c}}_j - \mathbf{c}_j)$ and subsequently ignore the transmissions sent through those channels. However, one problem is that there might not be $\mathbf{e}_j$ that has sufficiently large weight. More concerningly, Alice does not actually know $\mathbf{e}_1, \ldots, \mathbf{e}_{t+\ell}$!

Dealing with the first issue, note that it actually suffices to find multipliers λ_j such that $\sum_j \lambda_j \mathbf{e}_j$ has large weight: then Alice can broadcast the λ_j's and $\mathbf{y} := \sum_j \lambda_j \tilde{\mathbf{c}}_j$, and

then Bob can compute $\operatorname{supp}\left(\mathbf{y} - \sum_j \lambda_j \mathbf{c}_j\right)$ and ignore the subsequent transmissions sent through those channels.

Actually, in order to ensure a good transmission rate it will be important that the linear dependency is chosen to be relatively short; in particular, it should be independent of ℓ. It will turn out that we can find such a vector $\mathbf{y}$ which is a linear combination of a pseudobasis for the vectors $\tilde{\mathbf{c}}_1, \ldots, \tilde{\mathbf{c}}_{t+\ell}$. Recalling that the dimension of the syndrome space is at most t, this guarantees that we don't need to transmit too many multipliers λ_j.

However, we still haven't addressed the issue that Alice does not have direct access to the $\mathbf{e}_j$'s. But it turns out that this is not an problem: given a set of vectors with linearly independent syndromes, we will be able to find a linear combination $\sum_j \lambda_j \tilde{\mathbf{c}}_j$ that is far from *every* codeword. So, in particular, it will be far from $\sum_j \lambda_j \mathbf{c}_j$, as required.

Specifically, if $r \leq t/3$ and $\mathbf{y}_1, \ldots, \mathbf{y}_r \in \mathbb{F}_q^r$ are vectors such that the syndromes $\mathbf{H}\mathbf{y}_1, \ldots, \mathbf{H}\mathbf{y}_r \in \mathbb{F}_q^t$ are linearly independent, then Algorithm 2 finds a vector $\mathbf{y}$ in the span of $\mathbf{y}_1, \ldots, \mathbf{y}_r$ that satisfies $d(\mathbf{y}, \mathcal{C}) \geq r$. There is a procedure in [SZ16] with the same guarantee; however, we believe our algorithm is a bit simpler, so we have chosen to present it. In particular, we do not need to apply a unique-decoding algorithm as is required by the procedure in [SZ16]; we just use simple linear-algebraic operations.

Algorithm 2 A procedure for Alice to find a vector whose distance from $\mathcal{C}$ is at least r for $r \leq \frac{t}{3}$.

procedure MANY-ERRORS($\mathbf{y}_1, \ldots, \mathbf{y}_r$)
  For $i = 1, \ldots, r$, let $\mathbf{x}_i \in \mathcal{C}$ denote the codeword agreeing with $\mathbf{y}_i$ on the last $t+1$ coordinates. ▷ This is possible, as every subset of $t+1$ coordinates forms an information set for $\mathcal{C}$.
  For $i = 1, \ldots, r$, $\mathbf{e}_i \leftarrow \mathbf{y}_i - \mathbf{x}_i$.
  Let M denote the matrix in $\mathbb{F}_q^{r \times n}$ whose rows are $\mathbf{e}_1, \ldots, \mathbf{e}_r$.
  Using Gaussian elimination, put M in reduced row echelon form; let $\mathbf{e}_1^*, \ldots, \mathbf{e}_r^*$ denote the rows.
  if $\exists i \in [r]$ s.t. $\operatorname{wt}(\mathbf{e}_i^*) \geq r$ **then** $\mathbf{e} \leftarrow \mathbf{e}_i^*$
  else
    for $j = 2, 3, \ldots, r$ **do**
      if $\operatorname{wt}\left(\sum_{i=1}^j \mathbf{e}_i^*\right) \geq r$ **then** $\mathbf{e} \leftarrow \sum_{i=1}^j \mathbf{e}_i^*$
      end if
    end for
  end if
  Choose $\lambda_1, \ldots, \lambda_r \in \mathbb{F}_q$ such that $\mathbf{e} = \sum_{i=1}^r \lambda_i \mathbf{e}_i$.
  $\mathbf{y} \leftarrow \sum_{i=1}^r \lambda_i \mathbf{y}_i$
  return y
end procedure

Lemma 3.3. *Let $\mathbf{y}_1, \ldots, \mathbf{y}_r$ have linearly independent syndromes and assume $r \leq \frac{t}{3}$. Then the vector $\mathbf{y}$ returned by Algorithm 2 has distance at least r from $\mathcal{C}$.*

Proof. By assumption, we have that the syndromes $\mathbf{s}_i = \mathbf{H}\mathbf{y}_i \in \mathbb{F}_q^t$ for $i = 1, \ldots, r$ are linearly independent. We claim that the vectors $\mathbf{e}_1, \ldots, \mathbf{e}_r \in \mathbb{F}_q^n$ are linearly independent. Suppose $\lambda_1, \ldots, \lambda_r \in \mathbb{F}_q$ are such that $\sum_{i=1}^r \lambda_i \mathbf{e}_i = \mathbf{0}$. Then

$$\mathbf{0} = \sum_{i=1}^r \lambda_i \mathbf{H}\mathbf{e}_i = \sum_{i=1}^r \lambda_i \mathbf{H}(\mathbf{y}_i - \mathbf{x}_i) = \sum_{i=1}^r \lambda_i \mathbf{s}_i \ .$$

As $\mathbf{s}_1, \ldots, \mathbf{s}_r$ are linearly independent, this implies $\lambda_1 = \cdots = \lambda_r = 0$, as desired.

Now, we note that if $\mathbf{e} = \sum_{i=1}^r \lambda_i \mathbf{e}_i$ is found such that $d(\mathbf{e}, \mathcal{C}) \geq r$, then it also follows that $\mathbf{y} = \sum_{i=1}^r \lambda_i \mathbf{y}_i$ satisfies $d(\mathbf{y}, \mathcal{C}) \geq r$. Indeed,

$$d(\mathbf{y}, \mathcal{C}) = d\left(\mathbf{e} + \sum_{i=1}^r \lambda_i \mathbf{x}_i, \mathcal{C}\right) = d\left(\mathbf{e}, \mathcal{C} + \sum_{i=1}^r \lambda_i \mathbf{x}_i\right) = d(\mathbf{e}, \mathcal{C}) \geq r$$

as $\sum_{i=1}^r \lambda_i \mathbf{x}_i \in \mathcal{C}$.

Now, for $\mathbf{e} \in \operatorname{span}\{\mathbf{e}_1, \ldots, \mathbf{e}_r\}$, to ensure $d(\mathbf{e}, \mathcal{C}) \geq r$, note that it is sufficient to show that $r \leq \operatorname{wt}(\mathbf{e}) \leq t - r + 1$. Indeed, as we have $d(\mathbf{0}, \mathbf{e}) = \operatorname{wt}(\mathbf{e}) \geq r$, it suffices to verify that for all nonzero codewords $\mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\}$ we have $d(\mathbf{e}, \mathbf{c}) \geq r$. And indeed, this follows as

$$t + 1 \leq d(\mathbf{0}, \mathbf{c}) \leq d(\mathbf{0}, \mathbf{e}) + d(\mathbf{e}, \mathbf{c}) \leq t - r + 1 + d(\mathbf{e}, \mathbf{c}) \ ,$$

and so $d(\mathbf{e}, \mathbf{c}) \geq r$.

Hence, we now show how the algorithm finds a vector $\mathbf{e} \in \operatorname{span}\{\mathbf{e}_1, \ldots, \mathbf{e}_r\}$ which satisfies $r \leq \operatorname{wt}(\mathbf{e}) \leq t - r + 1$. Consider the matrix

$$M = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \vdots \\ \mathbf{e}_r \end{bmatrix} \in \mathbb{F}_q^{r \times n}$$

whose rows are given by vectors $\mathbf{e}_1, \ldots, \mathbf{e}_r$.

Consider putting the matrix M into reduced row echelon form; denote the resulting rows $\mathbf{e}_1^*, \ldots, \mathbf{e}_r^*$. By the definition of row operations, $\operatorname{span}\{\mathbf{e}_1, \ldots, \mathbf{e}_r\} = \operatorname{span}\{\mathbf{e}_1^*, \ldots, \mathbf{e}_r^*\}$, so it suffices to find a vector $\mathbf{e}^* \in \operatorname{span}\{\mathbf{e}_1^*, \ldots, \mathbf{e}_r^*\}$ satisfying $r \leq \operatorname{wt}(\mathbf{e}^*) \leq t - r + 1$.

As the vectors $\mathbf{e}_1, \ldots, \mathbf{e}_r$ are linearly independent, there is a set $R \subseteq [n]$ of r pivot points: that is, we have indices $1 \leq j_1 < j_2 < \cdots < j_r \leq n$ such that for each $i, p \in [r]$:

$$(\mathbf{e}_i)_{j_p} = \begin{cases} 1 & \text{if } i = p \\ 0 & \text{otherwise} \end{cases} \ .$$

Therefore, for each $i \in [r]$ we have $\mathrm{supp}(\mathbf{e}_i^*) \subseteq ([t] \setminus R) \cup \{j_i\}$, so $\mathrm{wt}(\mathbf{e}_i^*) \leq t - r + 1$. Thus, if we are in the case that for some $i \in [r]$ we have $r \leq \mathrm{wt}(\mathbf{e}_i^*)$, we can just return the vector $\mathbf{e}_i^*$.

Assume now that for each i we have $\mathrm{wt}(\mathbf{e}_i^*) < r$. Consider the sequence of vectors $\sum_{i=1}^{j} \mathbf{e}_i^*$ for $j = 2, \ldots, r$. Note that $\mathrm{supp}\left(\sum_{i=1}^{r} \mathbf{e}_i^*\right) \supseteq R$, so $\mathrm{wt}\left(\sum_{i=1}^{r} \mathbf{e}_i^*\right) \geq |R| = r$. Hence, there exists $2 \leq j \leq r$ such that:

- $\mathrm{wt}\left(\sum_{i=1}^{j} \mathbf{e}_i^*\right) \geq r$;
- for all $1 \leq j' \leq j$, $\mathrm{wt}\left(\sum_{i=1}^{j'} \mathbf{e}_i^*\right) < r$.

We claim that $\mathbf{e}^* := \sum_{i=1}^{j} \mathbf{e}_i^*$ satisfies $r \leq \mathrm{wt}(\mathbf{e}^*) \leq t + 1 - r$. The lower bound is obvious by the definition of j. For the upper bound, we note that

$$\mathrm{wt}\left(\sum_{i=1}^{j} \mathbf{e}_i^*\right) \leq \mathrm{wt}\left(\sum_{i=1}^{j-1} \mathbf{e}_i^*\right) + \mathrm{wt}(\mathbf{e}_j^*) < r + r \leq t + 1 - r \ ,$$

where the upper bound on the weight of $\sum_{i=1}^{j-1} \mathbf{e}_i^*$ is again by the definition of j and the upper bound on $\mathrm{wt}(\mathbf{e}_j^*)$ follows from our earlier assumption. That $2r \leq t + 1 - r$ follows from $r \leq t/3$. □

The Protocol. We are now in position to give our PSMT for transmitting an ℓ symbol secret.

Theorem 3.4. *Algorithm 3 is a PSMT with transmission rate* $(4 + o(1))n$.

Proof. We first verify that the protocol is reliable. After, we show that it is private. Lastly, we compute its transmission rate.

Reliability. We first make a few observations to justify the algorithm. First, we note that the definition of T on Line 12 is valid: indeed, $r = |S| \leq t$ since a pseudobasis has size at most t, so there are at least ℓ elements in $[t + \ell] \setminus S$. Also, we note that $\mathbf{z} = \sum_{y \in S} \lambda_j \mathbf{c}_j \in \mathcal{C}$, so since $\mathbf{y}$ is at distance at least r' from $\mathcal{C}$, we have $|\mathrm{supp}(\mathbf{z} - \mathbf{y})| = d(\mathbf{z}, \mathbf{y}) \geq r'$, as stated in Line 19. Furthermore, as $\mathbf{y} = \sum_{j \in S} \lambda_j \tilde{\mathbf{c}}_j$, if $E \subseteq [n]$ denotes the set of channels that Eve controls, then $\mathrm{supp}(\mathbf{y} - \mathbf{z}) \subseteq E$. Hence, for each $i \in [\ell]$, the transmission from Alice to Bob of $(\lambda_{ij} : j \in S)$ and $\langle \mathbf{h}, \bar{\mathbf{c}}_{p_i} \rangle + m_i$ via r'-generalized broadcast is reliable.

As in the analysis in Section 3.1, the reliability of Algorithm 3 follows from the fact that for $i = 1, \ldots, \ell$, we have $\bar{\mathbf{c}}_{p_i} = \mathbf{c}'_{p_i}$. And once again, the argument proceeds by demonstrating that both $\bar{\mathbf{c}}_{p_i}$ and $\mathbf{c}'_{p_i}$ are elements of $\mathcal{C}$. This is clear for $\mathbf{c}'_{p_i}$; for $\bar{\mathbf{c}}_{p_i}$, we use the parity-check matrix $\mathbf{H}$:

$$\mathbf{H}\bar{\mathbf{c}}_{p_i} = \mathbf{H}\left(\tilde{\mathbf{c}}_{p_i} - \sum_{j \in S} \lambda_{ij} \tilde{\mathbf{c}}_j\right) = \mathbf{s}_{p_i} - \sum_{j \in S} \lambda_{ij} \mathbf{s}_j = \mathbf{0} \ .$$

Algorithm 3 A protocol for transmitting an ℓ-symbol secret $(m_1, \ldots, m_\ell) \in \mathbb{F}_q^\ell$, which achieves transmission rate $(4 + o(1))n$.

procedure ROUND 1: BOB TRANSMITS
  Bob samples $\mathbf{c}_1, \ldots, \mathbf{c}_{t+\ell} \in \mathcal{C}$ independently and uniformly at random.
  For $j = 1, \ldots, t + \ell$, Bob transmits the i-th symbol of $\mathbf{c}_j$ through the i-th channel.
end procedure
procedure ROUND 2: ALICE TRANSMITS
  For $j = 1, \ldots, t + \ell$, Alice receives the vectors $\tilde{\mathbf{c}}_j$ where $d(\mathbf{c}_j, \tilde{\mathbf{c}}_j) \leq t$.
  For $j = 1, \ldots, t + \ell$, Alice computes $\mathbf{s}_j = \mathbf{H}\tilde{\mathbf{c}}_j \in \mathbb{F}_q^t$.
  Alice computes a pseudobasis for $\tilde{\mathbf{c}}_1, \ldots, \tilde{\mathbf{c}}_{t+\ell}$. Let $S \subseteq [t + \ell]$ index the elements of the pseudobasis.
  $r \leftarrow |S|$ and $r' = \min\{r, \lfloor t/3 \rfloor\}$.
  Let $S' \subseteq S$ denote a subset of size r'.
  Let $\mathbf{y} \leftarrow$ MANY-ERRORS$(\tilde{\mathbf{c}}_j : j \in S')$; write $\mathbf{y} = \sum_{j \in S} \lambda_j \tilde{\mathbf{c}}_j$. ▷ Of course, for $j \in S' \setminus S$, we may put $\lambda_j = 0$.
  Let $T = \{p_1, \ldots, p_\ell\}$ denote the ℓ smallest elements of $[t + \ell] \setminus S$.
  For $i \in [\ell]$, choose coefficients $\lambda_{ij} \in \mathbb{F}_q$ such that $\mathbf{s}_{p_i} = \sum_{j \in S} \lambda_{ij} \mathbf{s}_j$, and define $\bar{\mathbf{c}}_{p_i} \leftarrow \tilde{\mathbf{c}}_{p_i} - \sum_{j \in S} \lambda_{ij} \tilde{\mathbf{c}}_j$.
  Alice broadcasts the information $(S, (\lambda_j : j \in S), \mathbf{y})$.
  For each $i \in [\ell]$, Alice r'-broadcasts the data $(\lambda_{ij} : j \in S)$ and $\langle \mathbf{h}, \bar{\mathbf{c}}_{p_i} \rangle + m_i$.
end procedure
procedure OUTPUT PHASE
  Bob recovers $(S, (\lambda_j : j \in S), \mathbf{y})$ and defines $\mathbf{z} \leftarrow \sum_{j \in S} \lambda_j \mathbf{c}_j$. He also lets $T = \{p_1, \ldots, p_\ell\}$ denote the ℓ smallest elements of $[t + \ell] \setminus S$.
  Bob ignores the channels in the set $\mathrm{supp}(\mathbf{y} - \mathbf{z})$, a set of cardinality at least r'.
  For each $i \in [\ell]$, Bob recovers the information $(\lambda_{ij} : j \in S)$ and m_i', defines $\mathbf{c}'_{p_i} \leftarrow \mathbf{c}_{p_i} - \sum_{j \in S} \lambda_{ij} \mathbf{c}_j$, and then defines $m_i \leftarrow m_i' - \langle \mathbf{h}, \mathbf{c}'_{p_i} \rangle$.
  return $(m_1, \ldots, m_\ell)$.
end procedure

Now, since $\mathrm{supp}(\mathbf{c}_j - \tilde{\mathbf{c}}_j) \subseteq E$ for each $j \in [t+\ell]$, we also have $\mathrm{supp}(\mathbf{c}'_{p_i} - \bar{\mathbf{c}}_{p_i}) = \mathrm{supp}\left(\left(\mathbf{c}_{p_i} - \sum_{j\in S}\lambda_{ij}\mathbf{c}_j\right) - \left(\tilde{\mathbf{c}}_{p_i} - \sum_{j\in S}\lambda_{ij}\tilde{\mathbf{c}}_j\right)\right) \subseteq E$, which implies $d(\mathbf{c}'_{p_i}, \bar{\mathbf{c}}_{p_i}) \leq |E| \leq t$. As $\mathcal{C}$ has distance $t+1$, it follows that $\mathbf{c}'_{p_i} = \bar{\mathbf{c}}_{p_i}$. In particular, we have $\langle \mathbf{h}, \mathbf{c}'_{p_i}\rangle = \langle \mathbf{h}, \bar{\mathbf{c}}_{p_i}\rangle$.

Hence, for each $i \in [\ell]$, $m'_i - \langle \mathbf{h}, \mathbf{c}'_{p_i}\rangle = m_i + \langle \mathbf{h}, \bar{\mathbf{c}}_{p_i}\rangle = m_i$, demonstrating reliability.

Privacy. Let $E \subseteq [n]$ denote the set of t channels that Eve is observing. In the first round, she observes $(\mathbf{c}_1)|_E, \ldots, (\mathbf{c}_{t+\ell})|_E$. In the second round, she first observes $(S, (\lambda_j : j \in S), \mathbf{y})$. Also, for each $i \in [\ell]$, she then observes $(\lambda_{ij} : j \in S)$ and $m'_i = \langle \mathbf{h}, \bar{\mathbf{c}}_{p_i}\rangle + m_i$.

We wish to establish that Eve learns nothing about the symbols m_i for each $i \in [\ell]$. To establish this, it suffices to show that Eve has no information on $\langle \mathbf{h}, \bar{\mathbf{c}}_{p_i}\rangle$. And to do this, according to Lemma 2.2, it suffices to show that from Eve's perspective, $\bar{\mathbf{c}}_{p_i}$ appears to be a uniformly random codeword from which Eve has observed only t coordinates.

First, note that the data $(S, (\lambda_j : j \in S))$ informs Eve that $\{\tilde{\mathbf{c}}_j : j \in S\}$ forms a pseudobasis for $\tilde{\mathbf{c}}_1, \ldots, \tilde{\mathbf{c}}_{t+\ell}$. However, that is equivalent to the assertion that $\{\mathbf{e}_j : j \in S\}$ forms a pseudobasis for $\mathbf{e}_1, \ldots, \mathbf{e}_{t+\ell}$: as Eve knows $\mathbf{e}_1, \ldots, \mathbf{e}_{t+\ell}$, this does not tell her anything she does not already know.

Now, the vector $\mathbf{y}$ is defined in terms of the set $\{\tilde{\mathbf{c}}_j : j \in S\}$, and so it may leak some information on these vectors. However, note that for each $i \in [\ell]$, it does not depend on the vector $\tilde{\mathbf{c}}_{p_i}$. Also, the coefficients $(\lambda_{ij} : j \in S)$ are chosen so that $\mathbf{s}_{p_i} = \sum_{j\in S}\lambda_j \mathbf{s}_j$ and Eve knows $\mathbf{s}_j = \mathbf{H}\mathbf{e}_j$ for all $j \in [t+\ell]$, so the data $(\lambda_{ij} : j \in S)$ tell her nothing she did not already know. Hence, even after observing $(S, (\lambda_j : j \in S), \mathbf{y}, (\lambda_{ij} : j \in S))$, Eve still has not learned anything about the vector $\tilde{\mathbf{c}}_{p_i}$ beyond the t coordinates of the set E that she learned in the first round.

Hence, using that the codeword $\bar{\mathbf{c}}_{p_i} = \tilde{\mathbf{c}}_{p_i} - \sum_{j\in S}\lambda_{ij}\tilde{\mathbf{c}}_j = \mathbf{c}_{p_i} - \sum_{j\in S}\lambda_{ij}\mathbf{c}_j$, regardless of how much information Eve has learned about $\sum_{j\in S}\lambda_{ij}\mathbf{c}_j$, the vector $\bar{\mathbf{c}}_{p_i}$ still looks like a uniformly random codeword in $\mathcal{C}$ that she has only observed t coordinates of. This establishes the privacy of the transmission, as desired.

Transmission Rate. In the first round, Bob sends $(t+\ell)n$ symbols. In the second round, Alice broadcasts $\frac{r\log(t+\ell)}{\log q} + r + n$ symbols and then r'-broadcasts $\ell(r+1)$ symbols: this requires her to send

$$\frac{nr\log(t+\ell)}{\log q} + (r+n)n + (r+1)\ell\frac{n}{r'+1}$$

elements from $\mathbb{F}_q$. Thus, if N is the total number of symbols transmitted,

$$\frac{N}{\ell} = \frac{tn}{\ell} + n + \frac{nr\log(t+\ell)}{\ell\log q} + \frac{rn}{\ell} + \frac{n^2}{\ell} + \frac{(r+1)n}{r'+1} \leq 4n + O\left(\frac{n^2}{\ell} + \frac{n^2\log\ell}{\ell\log n}\right), \quad (1)$$

where we used $q \geq n$ and $\frac{r+1}{r'+1} \leq 3$. Hence, as $r' = \min\{r, t/3\}$ and $r \leq t$, assuming $\ell = \omega(n)$ we have $\frac{N}{\ell} \sim 4n$, as promised. □

Remark 3.5. Note that if we had been in the case that $r = r'$, i.e., $r \leq \frac{t}{3}$, then the transmission rate of Algorithm 3 would have been $\sim 2n$. Hence, in order to get our desired transmission rate of $2n$, we will only have to amend the protocol in the case that $r > \frac{t}{3}$. This is what we do in the following subsection.

3.3 Protocol with $(2 + o(1))n$ Transmission Rate

In order to decrease the transmission rate to $\sim 2n$, we look more carefully at the transmission rate as computed in (1). We have a factor of $\sim n$ from the first round when Bob communicates to Alice, and then a factor of $\sim 3n$ when Alice replies to Bob in the second round. In our lower bound argument, we will show that both parties will have to communicate $n\ell$ symbols in each round; hence, our only hope of getting a $\sim 2n$ transmission rate will be to decrease the communication of Alice in the second round.

Now, we note that the dominant term in Alice's communication is the $\frac{(r+1)n}{r'+1}\ell$ term which comes from the ℓ r'-generalized broadcasts from Line 15; as $r' \leq \frac{t}{3}$ and r can be as large as t, this term could be as large as $3n\ell$. If Alice used r-generalized broadcast for each of these transmissions, then this communication would cost only $\sim n\ell$ symbols, and we would get the $\sim 2n$ transmission rate we desire. However, as $\mathbf{y}$ only informs Bob of r' corrupted channels, if $r > r' = \min\{r, \lfloor t/3 \rfloor\}$ then Alice will have to communicate some more information for Bob to learn of r corrupted channels, which will guarantee the reliability of the transmission.

The solution for this is rather simple. We assume from now on that $r > r'$, which is the same as saying $r > \frac{t}{3}$. First, Alice broadcasts $(\mathbf{y}, S, \lambda_j : j \in S)$ as before (see Line 14); thus, $t/3$-generalized broadcast is now reliable. Next, we have Alice $t/3$-generalized broadcast the entire pseudobasis to Bob, i.e., all the vectors $\tilde{\mathbf{c}}_j$ for $j \in S$. We claim that this implies that r-generalized broadcast will now be reliable. Indeed, this follows from the following simple lemma.

Lemma 3.6. *Let $\tilde{\mathbf{c}}_j = \mathbf{c}_j + \mathbf{e}_j$ for $j \in S$ with $\mathbf{c}_j \in \mathcal{C}$ and put $\mathbf{s}_j = \mathbf{H}\tilde{\mathbf{c}}_j = \mathbf{H}\mathbf{e}_j$. Assume that $\dim(\mathrm{span}\{\mathbf{s}_j : j \in S\}) = r$. Then $\left|\bigcup_{j\in S} \mathrm{supp}(\mathbf{e}_j)\right| \geq r$.*

Proof. Let $\mathbf{d}_i \in \mathbb{F}_q^n$ denote the vector whose i-th coordinate is 1 and the remaining coordinates are 0. Let $R = \bigcup_{j\in S} \mathrm{supp}(\mathbf{e}_j)$; then clearly $\mathrm{span}\{\mathbf{d}_i : i \in R\} \supseteq \mathrm{span}\{\mathbf{e}_j : j \in S\}$, so also

$$\mathrm{span}\{\mathbf{H}\mathbf{d}_i : i \in R\} \supseteq \mathrm{span}\{\mathbf{H}\mathbf{e}_j : j \in S\} = \mathrm{span}\{\mathbf{s}_j : j \in S\}.$$

As $\dim(\mathrm{span}\{\mathbf{H}\mathbf{d}_i : i \in R\}) \leq |R|$, we conclude $|R| \geq \dim(\mathrm{span}\{\mathbf{s}_j : j \in S\}) = r$, as desired. □

Thus, suppose Alice reliably transmits to Bob the vectors $\tilde{\mathbf{c}}_j$ for $j \in S$. From this, Bob can compute the set $\bigcup_{j\in S} \mathrm{supp}(\mathbf{c}_j - \tilde{\mathbf{c}}_j) = \bigcup_{j\in S} \mathrm{supp}(\mathbf{e}_j)$; this set has cardinality at least r, and moreover it is contained in E (where, as usual, E denotes the set of channels

Eve controls). Hence, there are now r channels that Bob can safely ignore, so Alice may reliably r-broadcast the ℓ transmissions $(\lambda_{ij} : j \in S)$ and $\langle \mathbf{h}, \bar{\mathbf{c}}_{p_i} \rangle + m_i$, as in Line 15.

It is reasonable now to wonder if this will negatively impact the privacy of the protocol, as more information is revealed to Eve. However, by observing the proof of Theorem 3.4, one can see that even if Eve learns of $\tilde{\mathbf{c}}_j$ for $j \in S$, the inner-product $\langle \mathbf{h}, \bar{\mathbf{c}}_{p_i} \rangle$ is still wholly unknown to her, implying that they yield an effective mask for the secrets m_i.

Instead of completely rewriting the protocol, we just indicate in Algorithm 4 the changes that need to be made to Algorithm 3 to obtain the $\sim 2n$ transmission rate.

Algorithm 4 Our final protocol for transmitting an ℓ-symbol secret $(m_1, \ldots, m_\ell) \in \mathbb{F}_q^\ell$, which achieves transmission rate $(2 + o(1))n$. We just indicate what needs to be changed from Algorithm 3 when $r > r' = \min\{r, \lfloor t/3 \rfloor\}$.

1: **procedure** ROUND 1: BOB TRANSMITS
2: Bob performs lines 2-3 from Algorithm 3.
3: **end procedure**
4: **procedure** ROUND 2: ALICE TRANSMITS
5: Alice performs lines 6-14 from Algorithm 3.
6: **if** $r = r'$ **then**
7: Alice performs Line 15 from Algorithm 3.
8: **else**
9: Alice r'-broadcasts $\tilde{\mathbf{c}}_j$ for each $j \in S$.
10: For each $i \in [\ell]$, Alice r-broadcasts the data $(\lambda_{ij} : j \in S)$ and $\langle \mathbf{h}, \bar{\mathbf{c}}_{p_i} \rangle + m_i$.
11: **end if**
12: **end procedure**
13: **procedure** OUTPUT PHASE
14: Bob performs lines 18-19 from Algorithm 3.
15: Let $r \leftarrow |S|$.
16: **if** $r \leq t/3$ **then** Bob performs line 20
17: **else**
18: Bob recovers $\tilde{\mathbf{c}}_j$ for each $j \in S$.
19: Bob ignores the channels in the set $\bigcup_{j \in S} \operatorname{supp}(\tilde{\mathbf{c}}_j - \mathbf{c}_j)$, which has cardinality at least r.
20: For each $i \in [\ell]$, Bob recovers the information $(\lambda_{ij} : j \in S)$ and m'_i, defines $\mathbf{c}'_{p_i} \leftarrow \mathbf{c}_{p_i} - \sum_{j \in S} \lambda_{ij} \mathbf{c}_j$, and then defines $m_i \leftarrow m'_i - \langle \mathbf{h}, \mathbf{c}'_{p_i} \rangle$.
21: **end if**
22: **return** $(m_1, \ldots, m_\ell)$.
23: **end procedure**

Theorem 3.7. *Algorithm 4 is a PSMT with transmission rate* $(2 + o(1))n$.

Proof. As usual, we first establish reliability, then privacy, and lastly compute the trans-

mission rate. We just indicate the changes required to the proof of Theorem 3.4 to obtain Theorem 3.7, as most of the ideas are the same.

Reliability. In light of the reliability of Algorithm 3, in order to verify the reliability of Algorithm 4 it suffices to check that Bob can recover the information $(\lambda_{ij} : j \in S)$ and m'_i for each $i \in [\ell]$. That is, we need to ensure that r-generalized broadcast is reliable, even if $r > t/3$. This is certainaly the case if Bob knows at least r channels that Eve controls. But this is exactly what is guaranteed by Lemma 3.6: $\bigcup_{j \in S} \operatorname{supp}(\tilde{\mathbf{c}}_j - \mathbf{c}_j)$ is the set of r channels controlled by Eve that Bob knows.

Privacy. When $|S| > t/3$, Eve learns the vectors $\tilde{\mathbf{c}}_j$ for $j \in S$. However, as argued in the proof of Theorem 3.4, it is still the case that the vectors $\bar{\mathbf{c}}_{p_i} = \tilde{\mathbf{c}}_{p_i} - \sum_{j \in S} \lambda_{ij} \tilde{\mathbf{c}}_j = \mathbf{c}_{p_i} - \sum_{j \in S} \lambda_{ij} \mathbf{c}_j$ look like uniformly random codewords from which Eve has only observed t coordinates. So Lemma 2.2 still guarantees that the masks $\langle \mathbf{h}, \bar{\mathbf{c}}_{p_i} \rangle$ look like uniformly random elements of $\mathbb{F}_q$ to Eve, ensuring privacy of the transmission.

Transmission Rate. As the first round is unchanged from Algorithm 3, we simply need to establish that in the second round, Alice sends at most $n\ell + O(n^2 + n \log \ell / \log n)$ symbols. As noted in Remark 3.5, if $r = r'$ then this is the case. Hence, we now assume $r > r'$. In this case, Alice first r'-broadcasts the $r = |S|$ vectors $\tilde{\mathbf{c}}_j$ for $j \in S$ in Line 9; this requires $\frac{rn^2}{r'+1} \leq 3n^2$ symbols. Lastly, in Line 10, she uses ℓ invocations of r-generalized broadcast to transmit $r + 1$ symbols: this requires $\frac{(r+1)n\ell}{r+1} = n\ell$ symbols. Thus, Alice always communicates at most $n\ell + O(n^2 + n \log \ell / \log n)$ symbols in the second round, as desired. □

Remark 3.8. In light of our final algorithm (Algorithm 4) which occasionally requires Alice to send the entire pseudobasis to Bob, one could imagine the following very simple protocol: Alice always just broadcasts the entire pseudobasis to Bob, and then proceeds to r-broadcast the data $(\lambda_{ij} : j \in S)$ and $m_i + \langle \mathbf{h}, \bar{\mathbf{c}}_{p_i} \rangle$ for $i \in [\ell]$, as in Line 10. Such a protocol would indeed yield a PSMT with transmission rate $2n(1 + o(1))$. However, broadcasting the entire pseudobasis costs $\Theta(n^3)$ symbols (assuming $|S| = r = \Theta(n)$), so the total communication will include a $\Theta(n^3)$ term, rather than the $O(n^2 + n \log \ell / \log n)$ term as we have now. Hence, we would require $\ell = \omega(n^2)$ in order to ensure a communication cost of $\sim 2n\ell$; in order to match prior works (which just required $\ell = \omega(n)$), we have presented the more complicated Algorithm 4.

4 Lower Bound

In this section, we will show that our two-round PSMT in Algorithm 4 is actually optimal by proving a tight lower bound of $2n$ on the transmission rate.

Theorem 4.1. *Any two-round perfectly secure message transmission of an ℓ-bit secret requires communicating $2n\ell$ bits.*

An important step in our lower bound argument involves extracting a t-threshold secret-sharing scheme from a PSMT protocol. In order to make this precise, we provide in the following section the definition of a t-threshold secret-sharing scheme. The reader that is familiar with these notions may safely proceed to Section 4.2.

4.1 Secret-Sharing Schemes

For more details, we refer the reader to the treatment of secret-sharing schemes provided in [CDN15, Section 11.9]. Informally, a *t-threshold secret-sharing scheme* is a method for a secret to be distributed amongst n parties so as to guarantee (a) t-privacy, which guarantees that any set of t parties can learn nothing about the secret; and (b) $(t+1)$-reconstruction, which guarantees that any set of $(t+1)$ parties can fully recover the secret.

Given a vector of random variables $\mathbf{X} = (X_0, X_1, \ldots, X_n)$ and a subset $B \subseteq [n]$, we denote by $\mathbf{X}_B$ the vector of random variables indexed by the set B. We now provide the definition of a t-threshold secret-sharing scheme.

Definition 4.2 (t-Threshold Secret-Sharing Scheme)**.** A t-threshold secret-sharing scheme is a vector of random variables $\mathbf{X} = (X_0, X_1, \ldots, X_n)$ with each $X_i \in \mathcal{X}_i$ such that the following holds:

- The random variable X_0 is uniform over $\mathcal{X}_0$.
- t-privacy: Given any subset $B \subseteq [n]$ with $|B| \leq t$, any $x_0 \in \mathcal{X}_0$ and any $\mathbf{x}_B \in \prod_{i \in B} \mathcal{X}_i$ with $\Pr[\mathbf{X}_B = \mathbf{x}_B | X_0 = x_0] > 0$, $\Pr[X_0 = x_0 | \mathbf{X}_B = \mathbf{x}_B] = 1/|\mathcal{X}_0|$. That is, the shares in the set B provide no information on the secret.
- $(t+1)$-reconstruction: Given any subset $B \subseteq [n]$ with $|B| \geq t+1$ and any $\mathbf{x}_B \in \prod_{i \in B} \mathcal{X}_i$ with $\Pr[\mathbf{X}_B = \mathbf{x}_B | X_0 = x_0] > 0$, there is a unique $x_0 \in \mathcal{X}_0$ such that $\Pr[X_0 = x_0 | \mathbf{X}_B = \mathbf{x}_B] = 1$. That is, the shares in the set B uniquely determine the secret.

Finally, we will require the following observation, which (to the best of our knowledge) is folklore. It states that the size of each share must exceed the size of the secret.

Observation 4.3. If $\mathbf{X} = (X_0, X_1, \ldots, X_n)$ with each $X_i \in \mathcal{X}_i$ is a t-threshold secret-sharing scheme, then for all $i \in [n]$ we have $|\mathcal{X}_i| \geq |\mathcal{X}_0|$.

The justification for this is as follows. Suppose that for some $i \in [n]$, $|\mathcal{X}_i| < |\mathcal{X}_0|$, and let $B \subseteq [n] \setminus \{i\}$ be any set of size t. Then if the shares in B are known, as the i-th share must determine the secret, there can be only $|\mathcal{X}_i|$ choices remaining for the secret. But all $|\mathcal{X}_0|$ elements of $\mathcal{X}_0$ should be equally likely, a contradiction.

4.2 The Proof

We now provide the proof of Theorem 4.1.

Proof. First of all, let us characterize the behaviours of the sender Alice and the receiver Bob in a two-round PSMT. We can assume that in the first round Bob sends a message to Alice and in the second round Alice sends back a message containing the information of the secret to Bob. Otherwise, the second round communication does not help the receiver Bob decode the secret and one can then reduce this protocol to a one-round PSMT. Under this assumption, the following holds:

1. In the first round, Bob runs a randomized algorithm $A(\ell)$ to generate a message $\mathbf{a} = (a_1, \ldots, a_n) \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ where the randomness is only available to Bob. Bob sends $\mathbf{a}$ to Alice such that a_i is sent through the i-th channel.

2. Alice receives the corrupted vector $\tilde{\mathbf{a}}$ and runs the algorithm $B(\tilde{\mathbf{a}}, s)$ to generate the message $\mathbf{b} = (b_1, \ldots, b_n) \in \mathcal{B}_1 \times \cdots \times \mathcal{B}_n$ where $s \in [2^\ell]$ is the secret. Then Alice sends $\mathbf{b}$ to Bob such that b_i is sent through the i-th channel.

3. Bob receives the corrupted vector $\tilde{\mathbf{b}}$ and runs the algorithm $C(\tilde{\mathbf{b}}, \mathbf{a})$ to recover the secret. The protocol succeeds if C outputs s and Eve learns nothing about the secret.

Next, we will characterize the capabilities of the adversary Eve in this protocol. Eve is static, which means she has to choose up to t channels to corrupt before the beginning of this protocol. During the protocol, she can listen to the messages and change the messages transmitted through the corrupted channels. Eve succeeds if she learns anything about the secret or Bob fails to recover the secret. The total communication complexity is $\sum_{i=1}^{n}(\log |\mathcal{A}_i| + \log |\mathcal{B}_i|)$.

Since this protocol is reliable, we can assume that the algorithms B and C are deterministic. Otherwise, they must be correct for any random bits used by the algorithms B and C, and we can just analyze these two algorithms after setting all random bits to zero.

We first analyze the communication complexity of the second round in the scenario that Eve does nothing in the first round. In this scenario, Alice will receive the correct vector $\mathbf{a}$ and learns nothing about Eve. That means, from Alice and Bob's perspective, Eve can corrupt any t channels in the second round. We now demonstrate that one can extract from Bob's transmission a code with distance $2t + 1 = n$.

Claim 4.4. *Let* $\mathbf{b}_s = B(\mathbf{a}, s)$ *for* $s \in [2^\ell]$. *The set of codewords* $\{\mathbf{b}_s : s \in [2^\ell]\}$ *forms a code with minimum distance* $2t + 1$.

Note that this claim implies $\min_i \log |\mathcal{B}_i| \geq \ell$ and thus the communication complexity of the second round $\sum_{i=1}^{n} \log |\mathcal{B}_i| \geq \ell n$.

Proof of Claim 4.4. We note that, for $s_1 \neq s_2$, $\mathbf{b}_{s_1}$ and $\mathbf{b}_{s_2}$ must not agree on any index. Otherwise, Eve can inject t errors to cause Bob to receive the same vector $\tilde{\mathbf{b}}$ if $\mathbf{b}_{s_1}$ or $\mathbf{b}_{s_2}$ was sent. In one of the two cases, the $C(\tilde{\mathbf{b}}, \mathbf{a})$ does not output the correct secret, contradicting reliability. □

Now, we turn to lower-bounding the necessary communication in the first round. From Claim 4.4, Eve learns the vector transmitted in the second round as the codewords have distinct values in each coordinate. This implies that Bob and Alice must share a private key of length ℓ in the first round to achieve perfect security. More precisely, in the first round, Bob sends a vector of length n to Alice containing the information of this private key while Eve who observes any t symbols of this vector will learn nothing about this private key. One can deduce t-privacy from this if we treat such a vector as n shares and the private key as a secret. We formalize this as follows.

Observe that in the first round, the message sent by Bob is a function of ℓ and some randomness only available to Bob which is independent of Eve's strategy. This implies that the communication complexity of the first round is the same, regardless of Eve's strategy.

For any $s \in [2^\ell]$, consider the set $C_s = \{A(\ell, r) : B(A(\ell, r), s) = \mathbf{b}_1, r \in \mathcal{R}\}$, where r is the random string used by A and $\mathcal{R}$ denotes the set of random strings it could sample. This is the set of possible transmissions by Bob in the first round, given that Alice responds with the transmission $\mathbf{b}_1$ in the second round. The following claim states that one may extract a t-threshold secret sharing scheme from these sets.

Claim 4.5. *Define the random vector $\mathbf{X} = (X_0, X_1, \ldots, X_n) \in [2^\ell] \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ as follows.*

- *Sample a uniformly random secret $s \in [2^\ell]$ and put $X_0 = s$.*
- *Sample a random string r subject to the condition that $A(\ell, r) = \mathbf{a} = (a_1, \ldots, a_n) \in C_s$. For all $i \in [n]$, define $X_i = a_i \in \mathcal{A}_i$.*

Then the random vector $\mathbf{X}$ provides a t-threshold secret-sharing scheme.

Proof of Claim 4.5. We verify privacy and reconstruction.

- t-Privacy. Let $B \subseteq [n]$ denote any set of at most t coordinates. By the privacy of the PSMT protocol, it must be that Eve learns nothing about the secret s, even after seeing any t coordinates from the transmissions of Alice and Bob; in particular, this applies to the coordinates in the set B. Also, from the above discussion, we know that the transmission from Alice to Bob is completely revealed to Eve. Thus, if she sees that the second transmission is $\mathbf{b}_1$, even after seeing $(a_i : i \in E)$, she cannot know anything about the secret. Formally, for any fixed $s_0 \in [2^\ell]$ and $a_{0i} \in S_i$ for each $i \in B$, if $\Pr[\forall i \in B, a_i = a_{0i} | s = s_0] > 0$, we have

$$\frac{1}{2^\ell} = \Pr[s = s_0 | a_i = a_{0i}, \forall i \in B] = \Pr[X_0 = s_0 | \mathbf{X}_i = a_{0i}, \forall i \in B] \ ,$$

demonstrating privacy.

- $(t+1)$-Reconstruction. We notice that if for $s_1 \neq s_2$ there are two codewords $\mathbf{a}_{s_1} \in C_{s_1}$ and $\mathbf{a}_{s_2} \in C_{s_2}$ within distance t, Eve may corrupt t channels in the first round so as to change $\mathbf{a}_{s_1}$ to $\mathbf{a}_{s_2}$. Then, if Alice wants to send the secret message s_2 to Bob, she will transmit $B(\mathbf{a}_{s_2}, s_2) = \mathbf{b}_1$ to Bob. However, when Bob receives $\mathbf{b}_1$, the algorithm $C(\mathbf{a}_{s_2}, \mathbf{b}_1)$ will output s_2 instead of s_1 according to our definition of C_s. Thus, we conclude that this protocol fails. From this, we conclude that the codewords in different sets have distance at least $t+1$. In other words, given any set $B \subseteq [n]$ of $n-(t+1)+1 = t+1$ shares $(a_i : i \in B) = (X_i : i \in B)$, one can recover the secret $X_0 = s$.

□

As stated in Observation 4.3, in any t-threshold secret sharing scheme, the share size must be at least the secret size. We thus conclude $\sum_{i=1}^{n} \log |\mathcal{A}_i| \geq n\ell$, i.e., we obtain another $n\ell$ communication complexity in the first round. We emphasize that the message sent by Bob in the first round is *independent of* Eve's strategy. That means, the lower bound on the communication complexity of the first round can be applied to the case Eve does nothing in the first round. Therefore, we obtain the lower bound $2n\ell$ on the communication complexity of two-round PSMT, as desired. □

From above proof, we notice that if Alice and Bob want to communicate 1-bit secret via two-round PSMT, the threshold secret sharing scheme forces the communication complexity in the first round to be $n \log n$ [CCX13]. Thus, we have the following theorem.

Theorem 4.6. *Any two-round perfectly secure message transmission of* 1*-bit secret requires communicating at least* $n + n \log n$ *bits.*

5 Conclusion

We have precisely pinned down the optimal transmission rate in the model of perfectly secure message transmission with two rounds of communication when a (slim) majority of the channels are uncorrupted, i.e., when $n = 2t+1$. Namely, $\sim 2n\ell$ bits are necessary and sufficient to transmit an ℓ-bit message, and this is tight as soon as $\ell \log n = \omega(n)$.

The natural open question which remains from our work is to determine the amount of communication necessary when ℓ is smaller. Of particular interest is the case when $\ell = 1$, i.e., Alice's secret consists of a single bit. Our Theorem 4.6 informs us that $n + n \log n$ bits of communication are necessary, which improves upon the previous best lower bound. Our protocol from Section 3.1 tells us that $\sim n^2 \log n$ bits are necessary which, other than a slight improvement in the leading coefficient, does not improve over prior results [SZ16]. Thus, pinning down the communication complexity between the $\Omega(n \log n)$ lower bound

and the $O(n^2 \log n)$ upper bound for transmitting a one bit secret remains a tantalizing open problem.

6 Acknowledgement

CY would like to thank Serge Fehr for introducing him to this problem.

References

[ACdH06] Saurabh Agarwal, Ronald Cramer, and Robbert de Haan. Asymptotically optimal two-round perfectly secure message transmission. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 394–408. Springer, 2006.

[CCX13] Ignacio Cascudo, Ronald Cramer, and Chaoping Xing. Bounds on the threshold gap in secret sharing and its applications. *IEEE Trans. Inf. Theory*, 59(9):5600–5612, 2013.

[CDN15] Ronald Cramer, Ivan Bjerre Damgård, and Jesper Buus Nielsen. *Secure multiparty computation and secret sharing*. Cambridge University Press, 2015.

[DDWY93] Danny Dolev, Cynthia Dwork, Orli Waarts, and Moti Yung. Perfectly secure message transmission. *J. ACM*, 40(1):17–47, 1993.

[KS08] Kaoru Kurosawa and Kazuhiro Suzuki. Truly efficient 2-round perfectly secure message transmission scheme. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 324–340. Springer, 2008.

[SA96] Hasan Md. Sayeed and Hosame Abu-Amara. Efficient perfectly secure message transmission in synchronous networks. *Inf. Comput.*, 126(1):53–61, 1996.

[SNR04] K. Srinathan, Arvind Narayanan, and C. Pandu Rangan. Optimal perfectly secure message transmission. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 545–561. Springer, 2004.

[SZ16] Gabriele Spini and Gilles Zémor. Perfectly secure message transmission in two rounds. In *Theory of Cryptography Conference*, pages 286–304. Springer, 2016.

[WB86] Lloyd R Welch and Elwyn R Berlekamp. Error correction for algebraic block codes, December 30 1986. US Patent 4,633,470.