Two-Round Perfectly Secure Message Transmission with Optimal Transmission Rate

Nicolas Resch[1] and Chen Yuan[2]

[1] Cryptology Group
CWI, Amsterdam, The Netherlands
nar@cwi.nl
[2] School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University, Shanghai, China
chen_yuan@sjtu.edu.cn

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 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 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 $2(1+o_{\ell\to\infty}(1))n\ell$ symbols. Under a reasonable assumption (which is satisfied by all known efficient two-round PSMT protocols), we complement this with a lower bound showing that $2n\ell$ symbols are necessary for Alice to privately and reliably communicate her secret. This provides strong evidence that our construction is optimal (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 to t channels in such a way that she can listen to and/or overwrite the message passing through these t corrupted channels. Here, we assume Eve 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*).[3] PSMT can be done in multiple communication rounds. During each round, one party acts as the sender and the other acts as the receiver. They are not permitted 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.

Prior Work. 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 a 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 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. Our protocol builds off of their ideas, as we discuss at the end of this introduction. Their work also answers 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.

Our Results. Our results are two-fold. Our first contribution is a two-round PSMT protocol communicating a length ℓ secret with transmission rate $2(1 +$

[3] One can also consider the model of secure message transmission where privacy and/or reliability is only guaranteed to hold with high probability [FW00]. However, in this work, we focus exclusively on the case of *perfect* privacy and reliability.

$o_{\ell\to\infty}(1))n$.[4] 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 two-round PSMT protocols. Specifically, under a reasonable assumption, we show that Alice and Bob have to transmit at least $2n\ell$ bits so as to securely communicate an ℓ-bit secret. Our assumption comes from the observation that all known efficient constructions such as [ACdH06,KS08,SZ16] allow the adversary to learn the whole transmission in the second round of communication. This means the adversary can recover the transmission of *all* n channels by only listening to t of them. The reason is that in the second round, Alice encodes the message via an error correcting code which ensures the correctness of the transmission but sacrifices privacy. Therefore, in the security analysis of their protocols, they assume that the adversary could learn the whole transmission in the second round. Under this assumption, our two-round PSMT protocol actually achieves the optimal transmission rate. In this sense, our lower bound argument reveals an inherent limit for optimizing two-round PSMT: to beat our protocol, one must design a two-round PSMT protocol bypassing this assumption.

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 $[n, t+1, n-t]_q$ Reed-Solomon code[5] 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 $\sum_{i=1}^{t+1} \lambda_i \mathbf{c}_i = \sum_{i=1}^{t+1} \lambda_i \tilde{\mathbf{c}}_i$. To simplify the following expressions, denote $\bar{\mathbf{c}} := \sum_{i=1}^{t+1} \lambda_i \mathbf{c}_i = \sum_{i=1}^{t+1} \lambda_i \tilde{\mathbf{c}}_i$.

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[6] $\lambda_1, \ldots, \lambda_{t+1}$ 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

[4] Here and throughout, $o_{\ell\to\infty}(1)$ denotes a quantity which tends to 0 as $\ell \to \infty$, holding n fixed.

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

[6] 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.

$\lambda_1, \ldots, \lambda_{t+1}$ 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 $\mathbf{s}_i = \mathbf{He}_i$ 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 $[n, t+1, n-t]$ 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 do the same thing as [SZ16] so as to reduce the communication complexity: we spot a corrupted codeword with error weight at least r by applying linear operations to the $\tilde{\mathbf{c}}_j$'s.[7] We take a different approach which simplifies the argument; for details, please see Algorithm 4. 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 the message data $\lambda_{ij}, j \in S$ 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

[7] 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 $\ell \log n = \omega(n \log n)$-bit secret. The interested reader can find the details in our proof.

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.

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 $2(1+o_{\ell\to\infty}(1))n$ if Alice communicates to Bob an $\ell\log n=\omega(n\log n)$-bit secret.

Lower Bound. Let us first formalize PSMT by defining Alice and Bob's moves. Assume that Alice wants to communicate an ℓ-bit 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 $\mathbf{a}$.

Next, we justify our assumption that Eve learn the whole transmission in the second round of communication. We design an adversary Eve 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. In the meanwhile, if the code of distance $n=2t+1$ used by Alice and Bob in the second round is known to Eve, Eve will learn $\mathbf{a}$. In fact, all known efficient constructions use the same code book in this situation. Their protocol only protect the correctness of the transmission in the second round not the privacy.[8] In the following argument, we assume that Eve knows $\mathbf{b}$ if there is no corruption in the first round. Therefore, 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 Eve can listen to t channels, this means any t shares should learn nothing of this secret. This

[8] It might be possible that Alice and Bob use different codes with same minimum distance $n=2t+1$ in the second round. In this case, Bob and Alice must share the code information which is kept secret from Eve. We are not aware of any construction with this property and can not be sure that such strategy will gain them any advantage.

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, Eve may inject t errors to change $\mathbf{a}_1$ to $\mathbf{a}_2$. Then, Alice can not detect any corruption and take the move as if no corruption happens. However, This will lead to the situation that Alice and Bob 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. Although we do not pin down the actual value of optimal two-round PSMT, our lower bound shows that any two-round PSMT beating our lower bound must bypass this assumption. We leave this as a future direction.

Comparison to Previous Version. Our previous version does not include this assumption and prove the same lower bound. However, one of the conference referees points out that Eve may not learn the whole transmission in the second round if the code used by Alice and Bob are not fixed in this situation. We thank for his valuable comment which helps us to fix this bug. We also emphasize that in all known efficient PSMT protocols, Eve can predict the code used by Alice and Bob. This means our new assumption holds for these constructions. To beat our construction, one has to design a PSMT protocol bypassing this assumption.

Technical Comparison to Previous Works. Our protocol achieving transmission rate $2n$ utilizes ideas from prior works, and we would like to take a moment here to properly acknowledge them. The idea of leveraging the syndrome space and pseudobasis to correct errors was first introduced by Kurosawa and Suzuki in [KS08]. They also proposed the idea of generalized broadcast to decrease the communication cost of the second round. Spini and Zémor [SZ16] further developed this idea by showing how to spot a codeword with large error. They also abandon the dependency on the codeword communicated in the first round in [KS08] which greatly simplified the technique. These ideas also appear in our protocol; in particular, the first round of our protocol matches that of [SZ16].

To obtain a more efficient PSMT protocol, we observe that the protocol in [SZ16] divided the size of the global support of the errors into two cases: the small and the big one. In the second round, Alice transmits information for both of the potential cases. Thus, in some sense, half of her communication is wasted. Dealing with both cases simultaneously required a more careful analysis of the syndrome space to generate the required masks: we exploit linear dependencies amongst the syndromes, unlike [SZ16] that used a decoding algorithm, which itself was already a key improvement over the protocol in [KS08]. Furthermore, the approach in [SZ16] sends back syndrome vectors whose lengths are always $t+1$. In our protocol, we exploit the codewords in the pseudobasis S to correct the error, allowing us to only send back $|S|$ symbols to identify the vector. The bigger $|S|$ is, the more errors can be detected, permitting the use of more efficient generalized broadcast.

On the other hand, the lower bound argument is new, except that the need for broadcast in the second round is also mentioned in the $O(n)$ lower bound argument [SNR04].

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. By $o_{\ell \to \infty}(1)$ we refer to a quantity which tends to 0 as $\ell \to \infty$ (fixing all other parameters, including n), and we write $f(\ell) \sim g(\ell)$ if $\lim_{\ell \to \infty} \frac{f(\ell)}{g(\ell)} = 1$ (again, fixing all other parameters).

Remark 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 $d(\mathbf{x}, \mathbf{y}) = |\mathrm{supp}(\mathbf{x} - \mathbf{y})|$. For a vector $\mathbf{x} \in \mathbb{F}_q^n$ and a subset $S \subseteq [n]$, $\mathbf{x}|_S := (x_i)_{i \in S}$ denotes the length $|S|$ vector obtained by projecting on the coordinates indexed by S. 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 *maximum 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,[9] 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 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 1 (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 uniformly random codeword $\mathbf{c} \in \mathcal{C}$, the scalar product $\langle \mathbf{h}, \mathbf{c} \rangle$ is a uniformly random element of $\mathbb{F}_q$, even when conditioned on any t symbols of $\mathbf{c}$.*

Formally, for any $1 \leq i_1 < i_2 < \cdots < i_t \leq n$ and $\alpha_1, \alpha_2, \ldots, \alpha_t, \beta \in \mathbb{F}_q$, we have

$$\Pr[\langle \mathbf{h}, \mathbf{c} \rangle = \beta | \mathbf{c}_{i_1} = \alpha_1, \mathbf{c}_{i_2} = \alpha_2, \ldots, \mathbf{c}_{i_t} = \alpha_t] = \frac{1}{q} \ ,$$

where the randomness is over the uniformly random $\mathbf{c} \in \mathcal{C}$.

Remark 2 *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. (A similar discussion of the utility of pseudobases is available in Section 3.2 of [SZ16].) 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

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

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 1 (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* $\mathrm{span}\{\mathbf{H}\mathbf{y}_1, \ldots, \mathbf{H}\mathbf{y}_s\}$.

In other words, one computes a basis for the space spanned by $\mathbf{H}\mathbf{y}_1, \ldots, \mathbf{H}\mathbf{y}_s \in \mathbb{F}_q^{n-k}$, 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 3 *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*

$$\mathbf{H}\mathbf{y}_i = \mathbf{H}(\mathbf{c}_i + \mathbf{e}_i) = \mathbf{H}\mathbf{c}_i + \mathbf{H}\mathbf{e}_i = \mathbf{H}\mathbf{e}_i ,$$

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 a crucial idea, which is a method for Alice and Bob to agree on a random codeword that is not completely revealed to Eve. As we elaborate upon further in Remark 5, this means of extracting this secret codewords represents our core improvement over [SZ16].

Next, in Section 3.2, we show how to generalize the protocol to the case of $\ell \geq 1$, and achieve communication rate $(4 + o_{\ell\to\infty}(1))n$. 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.[10] 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 $(2 + o_{\ell\to\infty}(1))n$, we note that we only need to do something different if Eve invests many corruptions in the first round.[11] In order to handle this, we ask Alice to send a bit more information to Bob to indicate a 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.

Notations for this section. 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 1. Also, $\mathbf{H} \in \mathbb{F}_q^{t\times n}$ denotes a parity-check matrix for $\mathcal{C}$. The datum $(\mathcal{C}, \mathbf{h}, \mathbf{H})$ is *public*, fixed prior to the execution of the protocol and available to Alice, Bob and Eve throughout the execution. 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 providing a simple protocol which allows Alice to transmit one secret symbol $m \in \mathbb{F}_q$ to Bob. While this does not achieve our main goal, we find that it clarifies our means of extracting a codeword known to both Alice and Bob but secret from Eve, which we call $\bar{\mathbf{c}}$ and $\mathbf{c}'$. As we discuss further in Remark 5, this idea is the core of what allows us to go beyond the protocol of [SZ16] and eventually compress Alice's communication to just $\sim n\ell$ symbols. The details of the protocol are provided in Algorithm 1.

We now sketch why the protocol indeed yields a PSMT.

Reliability. First, we argue that Lines 8 and 9 from Algorithm 1 are justified, i.e., that Alice can indeed find $p \in [t+1]$ and $\lambda_j \in \mathbb{F}_q$ for $j \in [t+1] \setminus \{p\}$ such that $\mathbf{s}_p = \sum_{j\neq p} \mathbf{s}_j$. As $\mathbf{s}_1, \ldots, \mathbf{s}_{t+1} \in \mathbb{F}_q^t$ are $t+1$ vectors in a t-dimensional space, they must satisfy a nontrivial linear dependence $\sum_{j=1}^{t+1} \lambda'_j \mathbf{s}_j = \mathbf{0}$. Alice can thus pick any $p \in [t+1]$ for which $\lambda'_p \neq 0$, and then set $\lambda_j = -\lambda'_j/\lambda'_p$ for $j \in [t+1] \setminus \{p\}$.

[10] 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.
[11] More precisely, if the dimension of the syndrome space exceeds $t/3$.

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

procedure Round 1: Bob Transmits
  Bob samples $\mathbf{c}_1, \ldots, \mathbf{c}_{t+1} \in \mathcal{C}$ independently and uniformly at random.
  For $j = 1, \ldots, t+1$, Bob transmits the i-th coordinate of $\mathbf{c}_j$ through the i-th channel.
end procedure
procedure Round 2: Alice Transmits
  For $j = 1, \ldots, t+1$, Alice receives the vectors $\tilde{\mathbf{c}}_j$ where $d(\mathbf{c}_j, \tilde{\mathbf{c}}_j) \leq t$.
  For $j = 1, \ldots, t+1$, Alice computes $\mathbf{s}_j = \mathbf{H}\tilde{\mathbf{c}}_j \in \mathbb{F}_q^t$.
  Alice finds a coordinate $p \in [t+1]$ such that $\mathbf{s}_p \in \text{span}\{\mathbf{s}_j : j \neq p\}$.
  Alice finds $\lambda_j \in \mathbb{F}_q$ for $j \in [t+1] \setminus \{p\}$ such that $\mathbf{s}_p = \sum_{j \neq p} \lambda_j \mathbf{s}_j$.
  $\bar{\mathbf{c}} \leftarrow \tilde{\mathbf{c}}_p - \sum_{j \neq p} \lambda_j \tilde{\mathbf{c}}_j$
  Alice broadcasts p, $(\lambda_j : j \neq p)$ and the symbol $m' \leftarrow m + \langle \mathbf{h}, \bar{\mathbf{c}} \rangle$.
end procedure
procedure Output Phase
  Bob receives p, $(\lambda_j : j \neq p)$ and the symbol m'.
  $\mathbf{c}' \leftarrow \mathbf{c}_p - \sum_{j \neq p} \lambda_j \mathbf{c}_j$
  return $m' - \langle \mathbf{h}, \mathbf{c}' \rangle$.
end procedure

Now, 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, recalling that $E \subseteq [n]$ denotes the channels that the adversary controls, 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. In the first round of the protocol, Eve can only see $|E| \leq t$ symbols from each transmitted codeword. As the code $\mathcal{C}$ has dimension $t+1$ and is MDS, Eve learns only learns these $|E|$ symbols from $\mathbf{c}_1, \ldots, \mathbf{c}_{t+1}$.

In the second round, Eve sees $(p, \lambda_j : j \neq p)$. However, she already knows $\mathbf{e}_1, \ldots, \mathbf{e}_{t+1}$ and $\mathbf{H}$ and, using the fact that $\mathbf{s}_j = \mathbf{H}\tilde{\mathbf{c}}_j = \mathbf{H}\mathbf{e}_j$ for $j \in [t+1]$, $(p, \lambda_j : j \neq p)$ can be computed from $\mathbf{e}_1, \ldots, \mathbf{e}_{t+1}$ and $\mathbf{H}$. Thus, she does not learn anything from the second transmission.

We conclude that after the protocol, Eve has only learned the symbols indexed by the corrupted channels E from $\mathbf{c}_1, \ldots, \mathbf{c}_{t+1}$. In particular, Eve 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 $\mathcal{C}$, and so Lemma 1 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 $(4 + o_{\ell\to\infty}(1))n$ 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 [KS08,SZ16]. 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 repetition 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 2 (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 4 *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 an $\mathbf{e}_j$ 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 $\mathrm{supp}\left(\mathbf{y} - \sum_j \lambda_j \mathbf{c}_j\right)$ 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 4 finds a vector $\mathbf{y}$ in the span of $\mathbf{y}_1, \ldots, \mathbf{y}_r$ that satisfies $d(\mathbf{y}, \mathcal{C}) \geq r$. This procedure and its analysis are presented in Appendix A.

Remark 5 *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.*

A more significant difference between our protocols concerns the communication of the masked secrets. For each of the message symbols $m_1, \ldots, m_\ell$, the most efficient protocol of [SZ16] requires Alice to broadcast two *symbols $z_1^{(i)}, z_2^{(i)} \in \mathbb{F}_q$ which each mask the message symbol m_i in a different way. The symbol $z_1^{(i)}$ uses the mask $\langle \mathbf{h}, \mathbf{y}_{p_i} \rangle$; $z_2^{(i)}$ uses the mask $\langle \mathbf{h}, \tilde{\mathbf{c}}_{p_i} \rangle$ where $\tilde{\mathbf{c}}_{p_i}$ is the decoding of $\mathbf{y}_{p_i}$, or $z_2^{(i)}$ is just set to 0 if the decoding failed. Bob then chooses which mask to open, depending on the size of the pseudobasis. The authors comment they could* *use generalized broadcast for these symbols (as we do) to somewhat decrease the communication cost; however, even this change would not bring the second round communication down to $\sim n\ell$. Thus, a key difference between our protocols can be observed: by more carefully exploiting the structure of the pseudobasis, our extraction of the codewords $\bar{\mathbf{c}}_{p_i} = \mathbf{c}'_{p_i}$ to yield the masks $\langle \mathbf{h}, \bar{\mathbf{c}}_i \rangle$ prevents us from needing to use two different masks to guarantee that Bob can reliably recover the message symbols.*

The Protocol. We are now in position to give our PSMT for transmitting an ℓ symbol secret: the details are in Algorithm 2.

Algorithm 2 A protocol for transmitting an ℓ-symbol secret $(m_1, \ldots, m_\ell) \in \mathbb{F}_q^\ell$, which achieves transmission rate $(4 + o_{\ell\to\infty}(1))n$.

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 $d(\mathbf{c}_j, \tilde{\mathbf{c}}_j) \le t$.
  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 $r' \leftarrow \min\{r, \lfloor t/3 \rfloor\}$.
  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 $T \leftarrow \{p_1, \ldots, p_\ell\}$ 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 $m'_i \leftarrow m_i + \langle \mathbf{h}, \bar{\mathbf{c}}_{p_i} \rangle$.
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

Theorem 1 *Algorithm 2 is a PSMT with transmission rate $(4 + o_{\ell\to\infty}(1))n$.*

Proof. We first verify that the protocol is reliable. After, we show that it is private. Lastly, we compute its transmission rate. Throughout the proof, we let $E \subseteq [n]$ denote the set of t channels that Eve is corrupting.

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 $\mathbf{z} = \sum_{j \in S} \lambda_j \mathbf{c}_j \in \mathcal{C}$, 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 2 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 $\mathbf{H}$:

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

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

Privacy. First, we describe Eve's view of the protocol. In the first round, she observes $(\mathbf{c}_1)|_E, \ldots, (\mathbf{c}_{t+\ell})|_E$. In the second round, she first observes $(S, (\lambda_j : j \in S), \mathbf{y})$. Then, for each $i \in [\ell]$, she 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, conditioned on Eve's view, $\langle \mathbf{h}, \bar{\mathbf{c}}_{p_i} \rangle$ is a uniformly random element of $\mathbb{F}_q$. And to do this, according to Lemma 1, it suffices to show that from Eve's perspective, $\bar{\mathbf{c}}_{p_i}$ is a uniformly random codeword from which Eve has observed only t coordinates.

First of all, as $\mathbf{c}_1, \ldots, \mathbf{c}_{t+\ell}$ are sampled independently and uniformly from $\mathcal{C}$ and $\mathcal{C}$ has dimension $t+1$ and is MDS, after the first round Eve only learns $(\mathbf{c}_j)|_E$ for each $j \in [t+\ell]$.

Next, we consider the second round. We begin by noting that Eve can compute S from $\mathbf{H}$ and $\mathbf{e}_1, \ldots, \mathbf{e}_{t+\ell}$, which she knows. Indeed, as $\mathbf{s}_j = \mathbf{H}\tilde{\mathbf{c}}_j = \mathbf{H}\mathbf{e}_j$, Eve can also compute the pseudobasis S. So she learns nothing from this transmission. Once she has computed S Eve can then compute the set T and subsequently $(\lambda_{ij} : j \in S)$ for each $i \in [\ell]$, as the λ_{ij}'s are a function of the sets S and T and the syndromes $\mathbf{s}_1, \ldots, \mathbf{s}_{t+\ell}$, to which she has access.

Next, consider revealing to Eve the codewords $(\mathbf{c}_j : j \in S)$. Then, she can compute the corrupted codeword $\tilde{\mathbf{c}}_j = \mathbf{c}_j + \mathbf{e}_j$ for $j \in S$, so she can then compute the vector $\mathbf{y}$ and the multipliers $(\lambda_j : j \in S)$. Hence, what Eve sees in the second round is at most as informative as $(\mathbf{c}_j : j \in S)$.

Hence, at the termination of the protocol, what Eve can infer from her view about the masks $\langle \mathbf{h}, \bar{\mathbf{c}}_{p_i} \rangle$ for $i \in [\ell]$ is no more than what she can infer about them from the following data:

- The codewords $(\mathbf{c}_j : j \in S)$;
- The coordinates of all the codewords indexed by E, i.e., $(\mathbf{c}_j)|_E$ for $j \in [t+\ell]$.

Recall that, for each $i \in [\ell]$, $\bar{\mathbf{c}}_{p_i} = \mathbf{c}'_{p_i} = \mathbf{c}_{p_i} - \sum_{j \in S} \lambda_{ij} \mathbf{c}_j$. On the one hand, from the two pieces of data above, we have shown that Eve can compute exactly $\sum_{j \in S} \lambda_{ij} \mathbf{c}_j$. On the other hand, as the $\mathbf{c}_j$'s are sampled independently, the above data reveals nothing about $\mathbf{c}_{p_i}$ other than the coordinates indexed by E. Thus, from Eve's perspective, $\bar{\mathbf{c}}_{p_i} = \mathbf{c}_{p_i} - \sum_{j \in S} \lambda_{ij} \mathbf{c}_j$ is a uniformly random codeword from which she has only observed the coordinates indexed by E. Therefore the messages $m'_i = m_i + \langle \mathbf{h}, \bar{\mathbf{c}}_{p_i} \rangle$ reveal nothing about the secret vector $(m_1, \ldots, m_\ell)$. This concludes the proof of the assertion that the protocol is private.

Transmission Rate. In the first round, Bob sends $(t+\ell)n$ symbols. In the second round, Alice first broadcasts $\frac{r \log(t+\ell)}{\log q} + r + n$ symbols and then r'-broadcasts $\ell(r+1)$ symbols, where we recall that r denotes the size of the pseudobasis and $r' = \min\{r, \lfloor t/e \rfloor\}$. 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, then $\frac{N}{\ell}$ is

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

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

Remark 6 *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 2 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 $(2 + o_{\ell\to\infty}(1))n$ 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 2. *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 1, 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 3 the changes that need to be made to Algorithm 2 to obtain the $\sim 2n$ transmission rate.

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

1: **procedure** Round 1: Bob Transmits
2:     Bob performs lines 2-3 from Algorithm 2.
3: **end procedure**
4: **procedure** Round 2: Alice Transmits
5:     Alice performs lines 6-14 from Algorithm 2.
6:     **if** $r = r'$ **then**
7:         Alice performs Line 15 from Algorithm 2.
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 2.
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 2 *Algorithm 3 is a PSMT with transmission rate* $(2 + o_{\ell\to\infty}(1))n$.

Proof. As usual, we first establish reliability, then privacy, and lastly compute the transmission rate. We just indicate the changes required to the proof of Theorem 1 to obtain Theorem 2, as most of the ideas are the same.

Reliability. In light of the reliability of Algorithm 2, in order to verify the reliability of Algorithm 3 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, even if $r > t/3$, we need to ensure that r-generalized broadcast is reliable, i.e., that Bob knows at least r

channels that Eve controls. But this is exactly what is guaranteed by Lemma 2: $\bigcup_{j \in S} \mathrm{supp}(\tilde{\mathbf{c}}_j - \mathbf{c}_j)$ 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 1 (see the justification for the second bullet-point), 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 1 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 2, 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 6, 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. □

4 Lower Bound

In this section, we prove a lower bound on the transmission rate of any two-round PSMT under an assumption which we now formally introduce.

Our starting point is the observation that in our two-round PSMTs from Section 3, we always have Alice broadcast her desired transmission to Bob which completely sacrifices the privacy of her transmission. That is, the adversary completely learns the transmission from the second round. And this is not unique to our protocols: all of the efficient two-round PSMT protocols from the literature [ACdH06,KS08,SZ16] sacrifice the privacy of Alice's transmission.

Therefore, we make the assumption that the adversary learns the entire transmission of the second round and prove a $2n$ lower bound on the transmission rate under this assumption. This argument shows that among all two-round PSMTs satisfying this assumption, the one guaranteed by Theorem 2 is actually optimal. In other words, if one want to design a more efficient PSMT, the second round of this protocol must somehow bypass this assumption and keep something hidden from Eve. In this sense, we prove an inherent limitation for the line of optimizing two-round PSMT protocols [ACdH06,KS08,SZ16].

Assumption 1 *The adversary learns the whole transmission of the second round. More precisely, there is a function mapping the symbols Alice transmits through t of the channels to the symbols she sends through the other channels.*

Theorem 3 *Under Assumption 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 Appendix B the definition of a t-threshold secret-sharing scheme, as well as the observation that the share size must exceed the secret size. The reader that is familiar with these notions may safely proceed.

Proof. First of all, we formalize the behaviours of the sender Alice and the receiver Bob in a two-round PSMT.

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.

Note that if $B(\mathbf{a}, s) = \mathbf{b}$ then we must have $C(\mathbf{b}, \mathbf{a}) = s$, i.e., the protocol must succeed if the adversary Eve injects no errors.

Next, we 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 $\sum_{i=1}^{n}(\log|\mathcal{A}_i| + \log|\mathcal{B}_i|)$.

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. 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. 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. Under Assumption 1, we have that the adversary learns Alice's entire transmission in the second round. However, the ℓ bits of her secret must somehow be transmitted to Bob and kept secret from Eve. Intuitively, this means that Alice must use a perfectly-secure encryption scheme to send her secret message. This implies that Bob and Alice must have shared a private key of length ℓ in the first round, and moreover it must be shared in such a way that the adversary Eve observing t symbols from the transmission cannot learn anything about this private key. We formalize this intuition by showing that Bob's transmission in the first round yields a (t, n)-threshold secret sharing scheme with domain of secrets $[2^\ell]$.

Claim. Fix $\mathbf{b}_0 \in \mathcal{B}_1 \times \cdots \times \mathcal{B}_n$. Consider the following pair of algorithms:

- Share, on input $s \in [2^\ell]$, samples according to $A(\ell)$ a random $\mathbf{a} \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ conditioned on the event $B(\mathbf{a}, s) = \mathbf{b}_0$, and outputs $\mathbf{a}$.
- Recon, on input $(T, \mathbf{a}_T)$ with $T \subseteq [n]$ of size $|T| \geq t+1$ and $\mathbf{a}_T \in \prod_{i \in T} \mathcal{A}_i$, finds the unique $s \in [2^\ell]$ for which there exists $\mathbf{a}^* \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ agreeing with $\mathbf{a}_T$ on the coordinates in T and satisfying $C(\mathbf{b}_0, \mathbf{a}^*) = s$.

Then $(\mathsf{Share}, \mathsf{Recon})$ provides a (t, n)-threshold secret-sharing scheme with domain of secrets $[2^\ell]$ and share space $\mathcal{A}_1 \times \cdots \times \mathcal{A}_n$.

Proof. We verify privacy and reconstruction.

- t-Privacy. Let $s_1, s_2 \in [2^\ell]$ and let $\mathbf{a}^{(j)} := \mathsf{Share}(s_j)$, for $j = 1, 2$, denote the random sharings of the secrets. Let $T \subseteq [n]$ be any set with $|T| \leq t$. As $|T| \leq t$, we can consider an adversary Eve corrupting the channels indexed by T.
  For $j = 1, 2$, consider an execution of the PSMT protocol where Bob sends $\mathbf{a}^{(j)}$ and Eve introduces no corruptions in the first round. By the definition of Share, this means that Alice responds with $\mathbf{b}^{(0)}$. Therefore Eve sees $\mathbf{a}_T^{(j)}$ in the first round, and then $\mathbf{b}_T^{(0)}$ in the second round. By Assumption 1, we have that $\mathbf{b}^{(0)}$ is completely revealed to Eve. Thus, by the privacy of the PSMT protocol it must be that for any fixed vector $\mathbf{a}_T \in \prod_{i \in T} \mathcal{A}_i$,

  $$\Pr[\mathbf{a}_T^{(0)} = \mathbf{a}_T | \text{Alice transmits } \mathbf{b}^{(0)}] = \Pr[\mathbf{a}_T^{(1)} = \mathbf{a}_T | \text{Alice transmits } \mathbf{b}^{(0)}].$$

  This establishes the privacy of the secret-sharing scheme.
- $(t+1)$-Reconstruction. We must verify that Recon is well-defined. That is, if $\mathbf{a}$ is output by Share on input s and $B \subseteq [n]$ has size at least $t+1$, then there is indeed a unique $s \in [2^\ell]$ such that one can find $\mathbf{a}^* \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ agreeing with $\mathbf{a}_T$ on the coordinates in T and satisfying $C(\mathbf{a}^*, \mathbf{b}_0) = s$. Once we establish this property, it is clear that $\Pr[\mathsf{Recon}(T, \mathsf{Share}(s)_T) = s] = 1$, as required.
  Assume not and there are two distinct $x, y \in [2^\ell]$ with $\mathbf{x}, \mathbf{y} \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ such that $\mathbf{x}_T = \mathbf{y}_T$, $\mathsf{Recon}(T, \mathbf{x}_T) = x$ and $\mathsf{Recon}(T, \mathbf{y}_T) = y$. Then, the vectors $\mathbf{x}$ and $\mathbf{y}$ differ in at most t coordinates. Let $E \subseteq [n]$ denote the coordinates where they disagree, and suppose the adversary Eve corrupts the channels in E. Consider an execution of the protocol where Bob first

tries to transmit $\mathbf{x}$. The adversary Eve, controlling the channels in E, can change the vector $\mathbf{x}$ to the vector $\mathbf{y}$. Then, if Alice wants to send the secret y to Bob, she will transmit $B(\mathbf{y}, y) = \mathbf{b}_0$ to Bob; assume Eve does not corrupt this transmission. When Bob receives $\mathbf{b}_0$, the algorithm $C(\mathbf{x}, \mathbf{b}_0)$ will output x instead of y. This contradicts the reliability of the PSMT.

□

As mentioned earlier (and proved in Appendix B), in any (t, n)-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. Lastly, 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. □

Acknowledgement

NR is supported in part by ERC H2020 grant No. 74079 (ALGSTRONGCRYPTO). CY is supported in part by the National Natural Science Foundation of China under Grant 12101403, the National Natural Science Foundation of China under Grant 12031011, and National Key Research and Development Project 2021YFE0109900. CY would also 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.

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

FW00. Matthew Franklin and Rebecca N Wright. Secure communication in minimal connectivity models. *Journal of Cryptology*, 13(1):9–30, 2000.

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.

A Procedure for Finding a Vector Far from Code

In this section, we present our algorithm for finding a vector that is far from the code.

Algorithm 4 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. $\mathrm{wt}(\mathbf{e}_i^*) \geq r$ **then** $\mathbf{e} \leftarrow \mathbf{e}_i^*$
    else
        for $j = 2, 3, \ldots, r$ **do**
            if $\mathrm{wt}\left(\sum_{i=1}^{j} \mathbf{e}_i^*\right) \geq r$ **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. *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 4 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 $\operatorname{supp}(\mathbf{e}_i^*) \subseteq ([t] \setminus R) \cup \{j_i\}$, so $\operatorname{wt}(\mathbf{e}_i^*) \leq t - r + 1$. Thus, if we are in the case that for some $i \in [r]$ we have $r \leq \operatorname{wt}(\mathbf{e}_i^*)$, we can just return the vector $\mathbf{e}_i^*$.

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

– $\operatorname{wt}\left(\sum_{i=1}^{j} \mathbf{e}_i^*\right) \geq r$;

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

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

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

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 $2r \le t+1-r$ follows from $r \le t/3$. □

B Background on Secret-Sharing Schemes

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 $\mathbf{x} = (x_1, \ldots, x_n)$ and a subset $B \subseteq [n]$, we denote by $\mathbf{x}_B = (x_i : i \in B) \in \prod_{i \in B} \mathcal{X}_i$ the vector projected onto the coordinates indexed by the set B. We now provide the definition of a t-threshold secret-sharing scheme.

Definition 3 (t-Threshold Secret-Sharing Scheme) *Let t, n be integers satisfying $1 \le t < n$, and let $\mathcal{S}, \mathcal{X}_1, \ldots, \mathcal{X}_n$ be finite sets with $|\mathcal{S}| \ge 2$. A (t,n)-*threshold secret-sharing scheme *consists of:*

– *a randomized algorithm* Share *which takes as input a secret $s \in \mathcal{S}$, and outputs a vector of shares $\mathbf{x} \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$;*
– *a deterministic algorithm* Recon *which takes as input a subset $B \subseteq [n]$ satisfying $|B| \ge t+1$ and a vector of shares $\mathbf{x}_B \in \prod_{i \in B} \mathcal{X}_i$, and outputs a secret $s \in \mathcal{S}$ or a failure symbol $\perp$.*

The algorithms satisfy the following properties:

– *t-privacy: Given any subset $T \subseteq [n]$ with $|T| \le t$ and any two secrets $s_1, s_2 \in \mathcal{S}$, we have*
$$\Pr[\mathsf{Share}(s_1)_T = \mathbf{x}_T] = \Pr[\mathsf{Share}(s_2)_T = \mathbf{x}_T],$$
where the probability is over the randomness of the algorithm Share.
– *$(t+1)$-reconstruction: Given any subset $T \subseteq [n]$ with $|T| \ge t+1$ and any $s \in \mathcal{S}$, we have*
$$\Pr[\mathsf{Recon}(T, \mathsf{Share}(s)_T) = s] = 1,$$
where the probability is over the randomness of the algorithm Share.

We call $\mathcal{S}$ the domain of secrets *and $\mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ the* share space.

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 1 *Let* (Share, Recon *be a* (t, n)*-threshold secret-sharing scheme with domain of secrets* $\mathcal{S}$ *and share space* $\mathcal{X}_1 \times \cdots \times \mathcal{X}_n$*. Then for all* $i \in [n]$*,* $|\mathcal{X}_i| \geq |\mathcal{S}|$*.*

For completeness, we include the brief justification for this fact.

Proof. Suppose that for some $i \in [n]$, $|\mathcal{X}_i| < |\mathcal{S}|$, and let $T \subseteq [n] \setminus \{i\}$ be any set of size t. Let $\mathbf{x} \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ be a any tuple such that for some secret $s \in \mathcal{S}$, $\Pr[\mathsf{Share}(s) = \mathbf{x}] > 0$.

For each $y_i \in \mathcal{X}_i$, we denote by $\mathbf{x}_T||y_i \in \prod_{j \in T \cup \{i\}} \mathcal{X}_i$ the vector obtained by adding coordinate y_i to the vector $\mathbf{x}_B$. By $(t+1)$-reconstruction, we have a function $\varphi : \mathcal{X}_i \to \mathcal{S} \cup \perp$ defined by sending $y_i \in \mathcal{X}_i$ to the output of $\mathsf{Recon}(T \cup \{i\}, \mathbf{x}_T||y_i)$. As $|\mathcal{X}_i| < |\mathcal{S}|$, there is a secret s' not in the image of φ. But this then means that

$$\Pr[\mathsf{Share}(s')_T = \mathbf{x}_T] = 0 \neq \Pr[\mathsf{Share}(s)_T = \mathbf{x}_T],$$

contradicting t-privacy.