Two-Round Perfectly Secure Message Transmission with Optimal Transmission Rate

Nicolas Resch
Informatics’ Institute, University of Amsterdam, The Netherlands

Chen Yuan
School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China

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 \( \ell \) 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 \( \ell \) is asymptotically large. Specifically, we first construct a protocol that allows Alice to communicate an \( \ell \) symbol secret to Bob by transmitting at most \( 2(1 + o(1))n\ell \) 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).

2012 ACM Subject Classification Security and privacy → Mathematical foundations of cryptography

Keywords and phrases Secure transmission, Information theoretical secure, MDS codes

Digital Object Identifier 10.4230/LIPIcs.ITC.2023.1

1 Introduction

Perfectly secure message transmission (PSMT) was first introduced by Dolev et al. in [2]. 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
Two-Round Perfectly Secure Message Transmission with Optimal Transmission Rate

precisely, it is guaranteed that Bob always completely recovers the secret (reliability) and Eve learns absolutely nothing about the secret (privacy).\(^1\) 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 \(c_1\) represents the secret 1 and \(c_0\) 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 \(c_1\) and \(c_0\), unique decoding is not possible.

The original paper in \([2]\) 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 \([5]\) first presented a two-round PSMT achieving transmission rate \(O(n^3)\). Agarwal et al. \([1]\) further improved it to \(O(n)\) which is asymptotically optimal as a lower bound of \(n\) was proved in \([7]\). However, implementing this protocol requires an inefficient exponential-time algorithm. A breakthrough was achieved by Kurosawa and Suzuki \([4]\) whose protocol achieves transmission rate \(6n\), and can be run in polynomial time. Inspired by this protocol, Spini and Zémor \([6]\) 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 \([4]\) 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 \([6]\) 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 \(\ell\) secret with transmission rate \(2(1 + o_{\ell \to \infty}(1))n\).\(^2\) This protocol improves over the state-of-the-art protocol in \([6]\) 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 \([6]\).

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 \(\ell\)-bit secret. Our assumption comes from the observation

---

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

2 Here and throughout, \(o_{\ell \to \infty}(1)\) denotes a quantity which tends to 0 as \(\ell \to \infty\), holding \(n\) fixed.
that all known efficient constructions such as \([1, 4, 6]\) 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 \(\ell\)-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 \(c_1, \ldots, c_{t+1}\) which are picked independently and uniformly at random from a \([n, t+1, n-t]_q\) Reed-Solomon code over \(\mathbb{F}_q\). Alice receives the corrupted codewords \(\tilde{c}_i = c_i + e_i\). She uses the parity check matrix of this Reed-Solomon code to calculate the syndrome vectors \(H\tilde{c}_i = s_i\). 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 \(\sum_{i=1}^{t+1} \lambda_i s_i = 0\). From this one can show \(\sum_{i=1}^{t+1} \lambda_i e_i = 0\) and thus \(\sum_{i=1}^{t+1} \lambda_i c_i = \sum_{i=1}^{t+1} \lambda_i \tilde{c}_i\). To simplify the following expressions, denote \(\bar{c} := \sum_{i=1}^{t+1} \lambda_i c_i = \sum_{i=1}^{t+1} \lambda_i \tilde{c}_i\).

Let \(h \in \mathbb{F}_q^n\) be a vector of weight \(n\) that is not orthogonal to the \([n, t+1, n-t]_q\) Reed-Solomon code. Alice broadcasts \(\lambda_1, \ldots, \lambda_{t+1}\) together with \(h, \bar{c}\) + \(m\) to Bob where \(m\) is the secret; \(\langle h, \bar{c}\rangle\) is a mask for the secret. Bob first uses \(\lambda_1, \ldots, \lambda_{t+1}\) to recover \(\bar{c}\) and then obtains \(m\) by removing the mask \(\langle h, \bar{c}\rangle\) 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 \(s_i = H\tilde{c}_i\) is available to her. This means we can reduce the privacy argument to the last message \(\langle h, \bar{c}\rangle + m\) which is an immediate consequence of the \([n, t+1, n-t]_q\) 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 \(s_1, \ldots, s_{t+1}\) 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 \(\ell\) 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 \(c_1, \ldots, c_{t+\ell}\) which are picked independently and uniformly at random from a \([n, t+1, n-t]_{q}\) Reed-Solomon code over \(\mathbb{F}_q\). Alice receives the corrupted codewords \(\tilde{c}_i = c_i + e_i\) for \(i \in [t+\ell]\). She uses the parity-check matrix of this Reed-Solomon code to calculate the syndrome vectors \(H\tilde{c}_i = s_i\).

---

3 A \([n, k, d]_q\) Reed-Solomon code has block-length \(n\), dimension \(k\) and distance \(d = n - k + 1\).

4 To broadcast \(\lambda \in \mathbb{F}_q\), Alice sends \(\lambda\) through every channel; note that Bob can easily recover \(\lambda\) by choosing the majority symbol.
Assume that the space spanned by $s_1, \ldots, s_{t+\ell}$ has dimension $r$. Let $S \subset [t+\ell]$ be the index set of $s_i$ 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 $i \in [t]$, there exist not all zero coefficients $\lambda_{ij}$ for $j \in S$ such that $s_i = \sum_{j \in S} \lambda_{ij} s_j$. In analogy to what we did in the simpler protocol, we let $\tilde{c}_i := c_i - \sum_{j \in S} \lambda_{ij} c_j = \tilde{c}_i - \sum_{j \in S} \lambda_{ij} \tilde{c}_j$.

Before entering into the second round, we do the same thing as [6] so as to reduce the communication complexity: we spot a corrupted codeword $c_\tilde{i}$ with error weight at least $r$. Alice first broadcasts the index set $S$ together with $\lambda_{ij}$ for $j \in S$ and $\tilde{c}$ 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 $\langle \mathbf{h}, \tilde{c}_i \rangle + m_i$ for $i \in [t]$.

Once Bob receives the messages, he can correctly recover the index set $S$ and $\lambda_{ij}$ for $j \in S$ and $\tilde{c}$ as these messages are broadcasted. By applying the same linear operation on the codewords in $S$, Bob will obtain $c = \sum_{j \in S} \lambda_{ij} c_j$ which is at least distance $r$ away from $\tilde{c}$. Bob then ignores the $r$ channels that cause the inconsistency between $c$ and $\tilde{c}$. 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 $\lambda_{ij}$ can be computed by Eve on her own. Then, the privacy of the secret $m_i$ can be reduced to the privacy of $c_i$ 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 $\ell$-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 $2\mathbf{a}n$ bits so as to securely send the $\ell$-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 $\ell n$ 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

---

5 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 $r \leq \frac{1}{2}$; otherwise the protocol will be little more complicated. Specifically, Alice first broadcasts a corrupted codeword with error weight $\frac{r}{2}$ and then sends all corrupted codewords in $S$ to Bob via a $[n, \frac{r}{2}, n - \frac{r}{2} + 1]$ 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.
known efficient constructions use the same code book in this situation. Their protocol only protects the correctness of the transmission in the second round not the privacy. In the following argument, we assume that Eve knows $b$ if there is no corruption in the first round. Therefore, to achieve perfect security, Alice and Bob must share a private key of size $t$ 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 $a = (a_1, \ldots, a_n)$ 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 implies that such a secret sharing scheme has $t$-privacy. We next show that such secret sharing scheme must have $t+1$-reconstruction.

Let $a_1$ be any share vector of secret $s_1$ and $a_2$ be any share vector of secret $s_2$. If $a_1$ and $a_2$ are within distance $t$, Eve may inject $t$ errors to change $a_1$ to $a_2$. 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 $tn$. Putting it all together, we obtain the desired $2tn$ 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 [4]. They also proposed the idea of generalized broadcast to decrease the communication cost of the second round. Spini and Zémor [6] 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 [4] which greatly simplified the technique. These ideas also appear in our protocol; in particular, the first round of our protocol matches that of [6].

To obtain a more efficient PSMT protocol, we observe that the protocol in [6] 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

---

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.
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 [6] that used a decoding algorithm, which itself was already a key improvement over the protocol in [4]. Furthermore, the approach in [6] 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 [7].

## 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 \( \ell \). 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 \( \text{wt}(\mathbf{x}) := d(\mathbf{x}, \mathbf{0}) \).

The support of \( \mathbf{x} \) is \( \text{supp}(\mathbf{x}) := \{i \in [n] : x_i \neq 0\} \). Note that \( \text{wt}(\mathbf{x}) = |\text{supp}(\mathbf{x})| \) and \( d(\mathbf{x}, \mathbf{y}) = |\text{supp}(\mathbf{x} - \mathbf{y})| \). 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 \( \mathcal{C} \subseteq \mathbb{F}_q^n \); \( n \) is the block-length, \( k = \dim(\mathcal{C}) \) is the dimension and \( d = \min\{\text{wt}(\mathbf{c}) : \mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\}\} \) is the (minimum) distance. We refer to such a code as an \([n, k, d]\) 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,\(^7\) so we will state our results for arbitrary linear MDS codes.

\(^7\) Although in order to implement the protocol efficiently we will use the existence of efficient encoding and decoding algorithms for RS codes.
Any linear code $C$ may be described as the kernel of a matrix, i.e., $C = \{ x \in \mathbb{F}_q^n : Hx = 0 \}$. Such a matrix $H \in \mathbb{F}_q^{(n-k) \times n}$ is called a parity-check matrix.

Given two vectors $x, y \in \mathbb{F}_q^n$ we define their inner product via $\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$. We will need the following lemma from [6]. It states that there exists an MDS code $C \subseteq \mathbb{F}_q^n$ of dimension $t$ for $n = 2t + 1$ for which one can find a vector $h \in \mathbb{F}_q^n$ such that, even once $t$ coordinates are revealed from a codeword $c \in C$, the inner-product $\langle h, c \rangle \in \mathbb{F}_q$ is completely unconstrained.

**Lemma 2 (Lemma 1 from [6]).** For any $n$ and any $t < n$ there exists a linear MDS code $C$ of parameters $[n, t+1, n-t]$ and a vector $h \in \mathbb{F}_q^n$ such that given a uniformly random codeword $c \in C$, the scalar product $\langle h, c \rangle$ is a uniformly random element of $\mathbb{F}_q$, even when conditioned on any $t$ symbols of $c$. Moreover, $h$ can be found efficiently.

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

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

where the randomness is over the uniformly random $c \in C$.

**Remark 3.** We note that any such vector $h$ must not lie in the dual of $C$, 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.

**Pseudobases**

An important technical tool in our protocols are pseudobases, as introduced in the work of Kurosawa and Suzuki [4]. Before providing the definition, we explain their utility. (A similar discussion of the utility of pseudobases is available in Section 3.2 of [6].) Consider the scenario where Bob has sent a codeword $c \in C$ 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 $\dim C \geq t + 1$. However, by the Singleton bound, that forces the distance of $C$ 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 $c_1, \ldots, c_r$. 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 $\bar{c}$ of which Eve knows at most $t$ coordinates (which in turn means that $\langle h, \bar{c} \rangle$ can effectively mask the secret $m$). Using the concept of pseudobases, it turns out that this is possible (so long as the distance of $C$ is at least $t + 1$, as is the case when $C$ is MDS).

We now provide the formal definition of a pseudobasis.

**Definition 4 (Pseudobasis [4]).** Let $y_1, \ldots, y_s \in \mathbb{F}_q^n$ be vectors. A pseudobasis for $y_1, \ldots, y_s$ is a subcollection $y_{i_1}, \ldots, y_{i_r}$ with $1 \leq i_1 < \cdots < i_r \leq s$ such that $Hy_{i_1}, \ldots, Hy_{i_r} \in \mathbb{F}_q^{n-k}$ is a basis for the linear space span$\{Hy_1, \ldots, Hy_s\}$. 

ITC 2023
In other words, one computes a basis for the space spanned by \( H y_1, \ldots, H y_s \in \mathbb{F}_q^{n-k} \), and then the preimage of the basis vectors in \( \mathbb{F}_q^n \) provides a pseudobasis. Observe that, given access to \( H \), such a pseudobasis can be found in time polynomial in \( n \), and furthermore that it consists of at most \( n - k \) vectors.

**Remark 5.** Note that if we have a code \( C \subseteq \mathbb{F}_q^n \) with parity-check matrix \( H \) and we write \( y_i = c_i + e_i \) for each \( i \in [s] \) with \( c_i \in C \), then as

\[
H y_i = H(c_i + e_i) = Hc_i + He_i = He_i,
\]

we conclude that \( y_1, \ldots, y_r \) forms a pseudobasis for \( y_1, \ldots, y_s \) if and only if \( e_1, \ldots, e_r \) forms a pseudobasis for \( e_1, \ldots, e_s \).

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 \( \tilde{c}_i = c_i + e_i \), where \( e_i \) 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 \( e_i \) that she knows.

### 3 The Protocol

In this section, we present our protocol which allows Alice to privately and reliably transmit an \( \ell \) 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 8, this means of extracting this secret codewords represents our core improvement over [6].

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 \( \ell \) random codewords that are not completely known to Eve. In order to guarantee small transmission rate, we need a few more tricks. As in [6], 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. 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. 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.

---

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

9 More precisely, if the dimension of the syndrome space exceeds \( t/3 \).
Notations for this section. Throughout, $C \subseteq \mathbb{F}_q^n$ denotes an MDS code of dimension $t+1$ and $h \in \mathbb{F}_q^n$ a vector satisfying the conclusion of Lemma 2. Also, $H \in \mathbb{F}_q^{n \times n}$ denotes a parity-check matrix for $C$. The datum $(C, h, H)$ 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 $\tilde{c}$ and $c'$. As we discuss further in Remark 8, this idea is the core of what allows us to go beyond the protocol of [6] 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 $c_p = \sum_{j \neq p} s_j$. As $s_1, \ldots, s_{t+1} \in \mathbb{F}_q^n$ are $t+1$ vectors in a $t$-dimensional space, they must satisfy a nontrivial linear dependence $\sum_{j=1}^{t+1} \lambda_j s_j = 0$. 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 $j \in [t+1] \setminus \{p\}$.

Now, the important observation is that since the code $C$ has distance $t+1$, we have $c' = \tilde{c}$. Indeed, first note that $\tilde{c} \in C$, as

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

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

Thus, in particular, $(h, c') = (h, \tilde{c})$, so $m' - (h, c') = m + (h, \tilde{c}) - (h, c') = m$, 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 $C$ has dimension $t+1$ and is MDS, Eve learns only these $|E|$ symbols from $c_1, \ldots, c_{t+1}$.

In the second round, Eve sees $(p, \lambda_j : j \neq p)$. However, she already knows $e_1, \ldots, e_{t+1}$ and $H$ and, using the fact that $s_j = H\tilde{c}_j = Hc_j$ for $j \in [t+1]$, $(p, \lambda_j : j \neq p)$ can be computed from $e_1, \ldots, e_{t+1}$ and $H$. 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 $c_1, \ldots, c_{t+1}$. In particular, Eve only knows $t$ symbols of $c' = \tilde{c} = \tilde{c}_p - \sum_{j \neq p} \lambda_j \tilde{c}_j$ which is a codeword distributed uniformly at random in $C$, and so Lemma 2 guarantees that Eve has no information on $(h, \tilde{c})$. Thus, even after observing $m + (h, \tilde{c})$, 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 \(\ell\) 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 [4, 6]. 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 6 (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 \(C_r\) 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 \(c \in C_r\), and sending the \(i\)-th symbol of \(c\) 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 7.** 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 [8].

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 \(\ell\), we will have Bob transmit \(t + \ell\) uniformly random codewords in the first round, and Alice receives the corrupted codewords \(c_j = c_j + e_j\), where the support of each \(e_j\) is contained in the \(t\) channels Eve controls, \(E\).
Now, if Alice were aware that $e_j$ has large weight for some $j$, then she could just broadcast $\tilde{c}_j$ and the index $j$ to Bob. Bob could then compute the set $\text{supp}(\tilde{c}_j - c_j)$ and subsequently ignore the transmissions sent through those channels. However, one problem is that there might not be an $e_j$ that has sufficiently large weight. More concerning, Alice does not actually know $e_1, \ldots, e_{t+\ell}$.

Dealing with the first issue, note that it actually suffices to find multipliers $\lambda_j$ such that $\sum_j \lambda_j e_j$ has large weight: then Alice can broadcast the $\lambda_j$’s and $y := \sum_j \lambda_j \tilde{c}_j$, and then Bob can compute $\text{supp} \left( y - \sum_j \lambda_j c_j \right)$ 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 $\ell$. It will turn out that we can find such a vector $y$ which is a linear combination of a pseudobasis for the vectors $\tilde{c}_1, \ldots, \tilde{c}_{t+\ell}$. Recalling that the dimension of the syndrome space is at most $t$, this guarantees that we don’t need to transmit too many multipliers $\lambda_j$.

However, we still haven’t addressed the issue that Alice does not have direct access to the $e_j$’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 $\sum_j \lambda_j \tilde{c}_j$ that is far from every codeword. So, in particular, it will be far from $\sum_j \lambda_j c_j$, as required.

Specifically, if $r \leq t/3$ and $y_1, \ldots, y_r \in \mathbb{F}_q^t$ are vectors such that the syndromes $Hy_1, \ldots, Hy_r \in \mathbb{F}_q^t$ are linearly independent, then Algorithm 4 finds a vector $y$ in the span of $y_1, \ldots, y_r$ that satisfies $d(y, C) \geq r$. This procedure and its analysis are presented in Appendix D.

▶ Remark 8. There is a procedure in [6] 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 [6]; 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 $m_1, \ldots, m_t$, the most efficient protocol of [6] 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 $\langle h, y_{p_i} \rangle$; $z_2^{(i)}$ uses the mask $\langle h, \tilde{c}_{p_i} \rangle$ where $\tilde{c}_{p_i}$ is the decoding of $y_{p_i}$, 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 $\tilde{c}_{p_i} = c^{(i)}_{p_i}$ to yield the masks $\langle h, \tilde{c}_i \rangle$ 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 $\ell$ symbol secret: the details are in Algorithm 2.

▶ Theorem 9. 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 Appendix B is valid: indeed, $r = |S| \leq t$ since a pseudobasis has size at most $t$, so there are at least $\ell$ elements in $[t + \ell] \setminus S$. Also, we note that $z = \sum_{j \in S} \lambda_j c_j \in \mathcal{C}$, so since $y$ is at distance at least $r'$ from $\mathcal{C}$, we have $|\text{supp}(z - y)| = d(z, y) \geq r'$, as stated in Appendix B. Furthermore, as $y = \sum_{j \in S} \lambda_j \tilde{c}_j$, if $E \subseteq [n]$ denotes the set of channels that Eve controls, then $\text{supp}(y - z) \subseteq E$. Hence, for each $i \in [\ell]$, the transmission from Alice to Bob of $(\lambda_{ij} : j \in S)$ and $(h, \tilde{c}_{pi}) + m_i$ 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 $\tilde{c}_{pi} = c'_{pi}$. And once again, the argument proceeds by demonstrating that both $\tilde{c}_{pi}$ and $c'_{pi}$ are elements of $\mathcal{C}$. This is clear for $c'_{pi}$; for $\tilde{c}_{pi}$, we use the parity-check matrix $H$:

$$H \tilde{c}_{pi} = H \left( \tilde{c}_{pi} - \sum_{j \in S} \lambda_j \tilde{c}_j \right) = s_{pi} - \sum_{j \in S} \lambda_j s_j = 0.$$ 

Now, since $\text{supp}(c_j - \tilde{c}_j) \subseteq E$ for each $j \in [t + \ell]$, we also have

$$\text{supp}(c'_{pi} - \tilde{c}_{pi}) = \text{supp} \left( \left( c_{pi} - \sum_{j \in S} \lambda_j c_j \right) - \left( \tilde{c}_{pi} - \sum_{j \in S} \lambda_j \tilde{c}_j \right) \right) \subseteq E,$$

which implies $d(c'_{pi}, \tilde{c}_{pi}) \leq |E| \leq t$. As $\mathcal{C}$ has distance $t + 1$, it follows that $c'_{pi} = \tilde{c}_{pi}$. In particular, we have $(h, c'_{pi}) = (h, \tilde{c}_{pi})$.

Hence, for each $i \in [\ell]$, $m'_i = \langle h, c'_{pi} \rangle = m_i + \langle h, \tilde{c}_{pi} \rangle = \langle h, c_{pi} \rangle = m_i$, demonstrating reliability.

Privacy. First, we describe Eve’s view of the protocol. In the first round, she observes $(c_1) |_{[E]} , \ldots, (c_{t+\ell}) |_{[E]}$. In the second round, she first observes $(S, (\lambda_j : j \in S), y)$. Then, for each $i \in [\ell]$, she observes $(\lambda_{ij} : j \in S)$ and $m'_i = \langle h, c'_{pi} \rangle$.

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, $(h, \tilde{c}_{pi})$ is a uniformly random element of $\mathbb{F}_q$. And to do this, according to Lemma 2, it suffices to show that from Eve’s perspective, $c_{pi}$ is a uniformly random codeword from which Eve has observed only $t$ coordinates.

First of all, as $c_1, \ldots, c_{t+\ell}$ 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 $(c_j) |_{[E]}$ for each $j \in [t + \ell]$.

Next, we consider the second round. We begin by noting that Eve can compute $S$ from $H$ and $e_1, \ldots, e_{t+\ell}$, which she knows. Indeed, as $s_j = H e_j = H e_j$, 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 $\lambda_{ij}$’s are a function of the sets $S$ and $T$ and the syndromes $s_1, \ldots, s_{t+\ell}$, to which she has access.

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

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

- The codewords $(c_j : j \in S)$;
- The coordinates of all the codewords indexed by $E$, i.e., $(c_j) |_{[E]}$ for $j \in [t + \ell]$. 
Recall that, for each \(i \in [\ell]\), \(c_{p_i} = c'_{p_i} = c_{p_i} - \sum_{j \in S} \lambda_{ij} c_j\). On the one hand, from the two pieces of data above, we have shown that Eve can compute exactly \(\sum_{j \in S} \lambda_{ij} c_j\). On the other hand, as the \(c_j\)'s are sampled independently, the above data reveals nothing about \(c_{p_i}\) other than the coordinates indexed by \(E\). Thus, from Eve’s perspective, \(c_{p_i} = c_{p_i} - \sum_{j \in S} \lambda_{ij} c_j\) is a uniformly random codeword from which she has only observed the coordinates indexed by \(E\). Therefore the messages \(m'_i = m_i + \langle h, c_{p_i} \rangle\) 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. If Alice first broadcasts \(\frac{r \log(t + \ell)}{\log q} + r + n\) symbols and then \(r'\)-broadcasts \((r + 1)\ell\) 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 \(\frac{N}{r}\) is

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

(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 \(\frac{N}{r} \sim 4n\), as promised.

**Remark 10.** Note that if we had been in the case that \(r = r'\), i.e., \(r \leq \frac{t}{4}\), 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 \(r > \frac{t}{4}\). This is what we do in the following subsection.

### 3.3 Protocol with \((2 + o_{t \to \infty}(1)) n\) 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 \(\frac{(r + 1)n}{r' + 1}\) term which comes from the \(\ell r'\)-generalized broadcasts from Appendix B; as \(r' \leq \frac{4}{3}\) 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 \(y\) 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 \(r > \frac{1}{2}\). First, Alice broadcasts \((y, S, \lambda_j : j \in S)\) as before (see Appendix B); 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 \(\tilde{c}_j\) 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.
Two-Round Perfectly Secure Message Transmission with Optimal Transmission Rate

Lemma 11. Let $\tilde{c}_j = c_j + e_j$ for $j \in S$ with $c_j \in C$ and put $s_j = H\tilde{c}_j = He_j$. Assume that $\dim(\text{span}\{s_j : j \in S\}) = r$. Then $\left|\bigcup_{j \in S}\text{supp}(e_j)\right| \geq r$.

Proof. Let $d_i \in \mathbb{F}_q^n$ denote the vector whose $i$-th coordinate is 1 and the remaining coordinates are 0. Let $R = \bigcup_{j \in S}\text{supp}(e_j)$; then clearly $\text{span}\{d_i : i \in R\} \supseteq \text{span}\{e_j : j \in S\}$, so also $\text{span}\{Hd_i : i \in R\} \supseteq \text{span}\{He_j : j \in S\} = \text{span}\{s_j : j \in S\}$.

As $\dim(\text{span}\{Hd_i : i \in R\}) \leq |R|$, we conclude $|R| \geq \dim(\text{span}\{s_j : j \in S\}) = r$, as desired. ▶

Thus, suppose Alice reliably transmits to Bob the vectors $\tilde{c}_j$ for $j \in S$. From this, Bob can compute the set $\bigcup_{j \in S}\text{supp}(e_j - \tilde{c}_j) = \bigcup_{j \in S}\text{supp}(e_j)$; 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 $\ell$ transmissions $(\lambda_{ij} : j \in S)$ and $(h, \tilde{c}_n)$, as in Appendix B.

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 9, one can see that even if Eve learns of $\tilde{c}_j$ for $j \in S$, the inner-product $\langle h, \tilde{c}_n \rangle$ 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.

Theorem 12. Algorithm 3 is a PSMT with transmission rate $(2 + o_{\ell \to \infty}(1))n$.

Proof. The proof is omitted due to page limit. ▶

4 Lower Bound

In this section, we prove a lower bound on the transmission rate of any two-round PSMT under an assumption about the protocol 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 [1, 4, 6] 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 12 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 [1, 4, 6].

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 13. Under Assumption 1, any two-round perfectly secure message transmission of an $\ell$-bit secret requires communicating $2n\ell$ bits.
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 $a = (a_1, \ldots, a_n) \in A_1 \times \cdots \times A_n$ where the randomness is only available to Bob. Bob sends $a$ to Alice such that $a_i$ is sent through the $i$-th channel.

2. Alice receives the corrupted vector $\tilde{a}$ and runs the algorithm $B(\tilde{a}, s)$ to generate the message $b = (b_1, \ldots, b_n) \in B_1 \times \cdots \times B_n$ where $s \in [2^\ell]$ is the secret. Then Alice sends $b$ to Bob such that $b_i$ is sent through the $i$-th channel.

3. Bob receives the corrupted vector $\tilde{b}$ and runs the algorithm $C(\tilde{b}, a)$ to recover the secret.

The protocol succeeds if $C$ outputs $s$ and Eve learns nothing about the secret.

Note that if $B(a, s) = b$ then we must have $C(b, a) = s$, i.e., the protocol must succeed if the adversary Eve injects no errors. We defer the formal proof to the full version. ▷

References


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 $c_1, \ldots, c_{t+1} \in \mathbb{C}$ independently and uniformly at random.
3: For $j = 1, \ldots, t + 1$, Bob transmits the $i$-th coordinate of $c_j$ 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 $\tilde{c}_j$ where $d(c_j, \tilde{c}_j) \leq t$.
7: For $j = 1, \ldots, t + 1$, Alice computes $s_j = H\tilde{c}_j \in \mathbb{F}_q^t$.
8: Alice finds a coordinate $p \in [t + 1]$ such that $s_p \in \text{span}\{s_j : j \neq p\}$.
9: Alice finds $\lambda_j \in \mathbb{F}_q$ for $j \in [t + 1] \setminus \{p\}$ such that $s_p = \sum_{j \neq p} \lambda_j s_j$.
10: $\tilde{c} \leftarrow \tilde{c}_p - \sum_{j \neq p} \lambda_j \tilde{c}_j$
11: Alice broadcasts $p, (\lambda_j : j \neq p)$ and the symbol $m' \leftarrow m + \langle h, \tilde{c} \rangle$.
12: end procedure

13: procedure Output Phase
14: Bob receives $p, (\lambda_j : j \neq p)$ and the symbol $m'$.
15: $c' \leftarrow c_p - \sum_{j \neq p} \lambda_j c_j$
16: return $m' - \langle h, c' \rangle$.
17: end procedure
B Algorithm 2

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

1: procedure Round 1: Bob Transmits
2: Bob samples $c_1, \ldots, c_{t+\ell} \in \mathcal{C}$ independently and uniformly at random.
3: For $j = 1, \ldots, t + \ell$, Bob transmits the $i$-th symbol of $c_j$ through the $i$-th channel.
4: end procedure

5: procedure Round 2: Alice Transmits
6: For $j = 1, \ldots, t + \ell$, Alice receives the vectors $\tilde{c}_j$ where $d(c_j, \tilde{c}_j) \leq t$.
7: For $j = 1, \ldots, t + \ell$, Alice computes $s_j = H_{\tilde{c}_j} \in \mathbb{F}_t q$.
8: Alice computes a pseudobasis for $\tilde{c}_1, \ldots, \tilde{c}_{t+\ell}$. Let $S \subseteq [t + \ell]$ index the elements of the pseudobasis.
9: $r \leftarrow |S|$ and $r' \leftarrow \min\{r, \lfloor t/3 \rfloor\}$.
10: Let $S' \subseteq S$ denote a subset of size $r'$.
11: Let $y = (\tilde{c}_j : j \in S')$; write $y = \sum_{j \in S} \lambda_j \tilde{c}_j$. ▷ Of course, for $j \in S \setminus S'$, we may put $\lambda_j = 0$.
12: Let $T = \{p_1, \ldots, p_\ell\}$ denote the $\ell$ smallest elements of $[t + \ell] \setminus S$.
13: For $i \in [\ell]$, choose coefficients $\lambda_{ij} \in \mathbb{F}_q$ such that $s_{p_i} = \sum_{j \in S} \lambda_{ij} s_j$, and define $c_{p_i} = \tilde{c}_{p_i} - \sum_{j \in S} \lambda_{ij} \tilde{c}_j$.
14: Alice broadcasts the information $(S, (\lambda_{ij} : j \in S), y)$.
15: For each $i \in [\ell]$, Alice $r'$-broadcasts the data $(\lambda_{ij} : j \in S)$ and $m'_i \leftarrow m_i + \langle h, c_{p_i} \rangle$.
16: end procedure

17: procedure Output Phase
18: Bob recovers $(S, (\lambda_{ij} : j \in S), y)$ and defines $z = \sum_{j \in S} \lambda_j c_j$. He also lets $T = \{p_1, \ldots, p_\ell\}$ denote the $\ell$ smallest elements of $[t + \ell] \setminus S$.
19: Bob ignores the channels in the set $\text{supp}(y - z)$, a set of cardinality at least $r'$.
20: For each $i \in [\ell]$, Bob recovers the information $(\lambda_{ij} : j \in S)$ and $m'_i$, defines $c'_{p_i} = c_{p_i} - \sum_{j \in S} \lambda_{ij} c_j$, and then defines $m_i \leftarrow m'_i - \langle h, c'_{p_i} \rangle$.
21: return $(m_1, \ldots, m_\ell)$.
22: end procedure
Algorithm 3

Our final protocol for transmitting an \( \ell \)-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 \( r > r' = \min\{r, [t/3]\} \).

**procedure** Round 1: Bob Transmits

Bob performs lines 2-3 from Algorithm 2.

**end procedure**

**procedure** Round 2: Alice Transmits

Alice performs lines 6-14 from Algorithm 2.

if \( r = r' \) then

Alice performs Appendix B from Algorithm 2.

else

Alice \( r' \)-broadcasts \( \tilde{c}_j \) for each \( j \in S \).

For each \( i \in [\ell] \), Alice \( r \)-broadcasts the data \((\lambda_{ij} : j \in S) \) and \( (h, \tilde{c}_{p_i}) + m_i \).

**end if**

**end procedure**

**procedure** Output Phase

Bob performs lines 18-19 from Algorithm 2.

Let \( r \leftarrow |S| \).

if \( r \leq t/3 \) then Bob performs line 20

else

Bob recovers \( \tilde{c}_j \) for each \( j \in S \).

Bob ignores the channels in the set \( \bigcup_{j \in S} \text{supp}(\tilde{c}_j - c_j) \), which has cardinality at least \( r \).

For each \( i \in [\ell] \), Bob recovers the information \((\lambda_{ij} : j \in S) \) and \( m'_i \), defines \( c'_{p_i} \leftarrow c_{p_i} - \sum_{j \in S} \lambda_{ij} c_j \), and then defines \( m_i \leftarrow m'_i - (h, c'_{p_i}) \).

**end if**

return \((m_1,\ldots,m_\ell)\).

**end procedure**

| D | 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.

**Lemma 14.** Let \( y_1,\ldots,y_r \) have linearly independent syndromes and assume \( r \leq \frac{t}{3} \). Then the vector \( y \) returned by Algorithm 4 has distance at least \( r \) from \( C \).

**Proof.** By assumption, we have that the syndromes \( s_i = Hy_i \in \mathbb{F}_q^t \) for \( i = 1,\ldots,r \) are linearly independent. We claim that the vectors \( e_1,\ldots,e_r \in \mathbb{F}_q^n \) are linearly independent. Suppose \( \lambda_1,\ldots,\lambda_r \in \mathbb{F}_q \) are such that \( \sum_{i=1}^r \lambda_i e_i = \mathbf{0} \). Then

\[
0 = \sum_{i=1}^r \lambda_i H e_i = \sum_{i=1}^r \lambda_i H (y_i - x_i) = \sum_{i=1}^r \lambda_i s_i .
\]

As \( s_1,\ldots,s_r \) are linearly independent, this implies \( \lambda_1 = \cdots = \lambda_r = 0 \), as desired.
Algorithm 4 A procedure for Alice to find a vector whose distance from $C$ is at least $r$ for $r \leq \frac{1}{2}$.

1: procedure Many-Errors($y_1, \ldots, y_r$)
2: For $i = 1, \ldots, r$, let $x_i \in C$ denote the codeword agreeing with $y_i$ on the last $t + 1$
   coordinates. $\triangleright$ This is possible, as every subset of $t + 1$ coordinates forms an information
   set for $C$.
3: For $i = 1, \ldots, r$, $e_i \leftarrow y_i - x_i$.
4: Let $M$ denote the matrix in $F_q^{r \times n}$ whose rows are $e_1, \ldots, e_r$.
5: Using Gaussian elimination, put $M$ in reduced row echelon form; let $e_1^*, \ldots, e_r^*$ denote
   the rows.
6: if $\exists i \in [r]$ s.t. $\text{wt}(e_i^*) \geq r$ then $e \leftarrow e_i^*$
7: else
8: for $j = 2, 3, \ldots, r$ do
9: if $\text{wt} \left( \sum_{i=1}^{j} e_i^* \right) \geq r$ then $e \leftarrow \sum_{i=1}^{j} e_i^*$
10: end if
11: end for
12: end if
13: Choose $\lambda_1, \ldots, \lambda_r \in F_q$ such that $e = \sum_{i=1}^{r} \lambda_i e_i$.
14: $y \leftarrow \sum_{i=1}^{r} \lambda_i y_i$
15: return $y$
16: end procedure

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

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

as $\sum_{i=1}^{r} \lambda_i x_i \in C$.

Now, for $e \in \text{span}\{e_1, \ldots, e_r\}$, to ensure $d(e, C) \geq r$, note that it is sufficient to show
that $r \leq \text{wt}(e) \leq t - r + 1$. Indeed, as we have $d(0, e) = \text{wt}(e) \geq r$, it suffices to verify that
for all nonzero codewords $c \in C \setminus \{0\}$ we have $d(e, c) \geq r$. And indeed, this follows as

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

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

Hence, we now show how the algorithm finds a vector $e \in \text{span}\{e_1, \ldots, e_r\}$ which satisfies
$r \leq \text{wt}(e) \leq t - r + 1$. Consider the matrix

$$M = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_r \end{bmatrix} \in F_q^{r \times n}$$

whose rows are given by vectors $e_1, \ldots, e_r$.

Consider putting the matrix $M$ into reduced row echelon form; denote the resulting rows $e_1^*, \ldots, e_r^*$. By the definition of row operations, $\text{span}\{e_1, \ldots, e_r\} = \text{span}\{e_1^*, \ldots, e_r^*\}$, so it suffices to find a vector $e^* \in \text{span}\{e_1^*, \ldots, e_r^*\}$ satisfying $r \leq \text{wt}(e^*) \leq t - r + 1$.

As the vectors $e_1, \ldots, e_r$ 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 $i, p \in [r]$:
\[(e_i)_{jp} = \begin{cases} 1 & \text{if } i = p \\ 0 & \text{otherwise} \end{cases}.\]

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

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

\[\text{wt} \left( \sum_{i=1}^{j} e_i^* \right) \geq r;\]

\[\text{for all } 2 \leq j' \leq j, \text{wt} \left( \sum_{i=1}^{j'} e_i^* \right) < r.\]

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

\[\text{wt} \left( \sum_{i=1}^{j} e_i^* \right) \leq \text{wt} \left( \sum_{i=1}^{j-1} e_i^* \right) + \text{wt}(e_j^*) < r + r \leq t + 1 - r,\]

where the upper bound on the weight of \(\sum_{i=1}^{j-1} e_i^*\) is again by the definition of \(j\) and the upper bound on \(\text{wt}(e_j^*)\) follows from our earlier assumption. That \(2r \leq t + 1 - r\) follows from \(r \leq t/3\). ◢