On one-round reliable message transmission

René Bødker Christensen

Department of Mathematical Sciences, Aalborg University, Denmark.
rene@math.aau.dk

Abstract

In this paper, we consider one-round protocols for reliable message transmission (RMT) when t out of $n = 2t + 1$ available channels are controlled by an adversary. We show impossibility of constructing such a protocol that achieves a transmission rate of less than $\Theta(n)$ for constant-size messages and arbitrary reliability parameter. In addition, we show how to improve two existing protocols for RMT to allow for either larger messages or reduced field sizes.

1. Introduction

The concept of secure message transmission was first introduced in [3], and the term comprises a model where a sender and a receiver are connected via n channels. Up to t of these channels are controlled by a computationally unbounded active adversary who can read and alter the symbols sent across these t channels. More specifically, we consider the setting where $n = 2t+1$. In keeping with cryptographic tradition, we will call the sender 'Alice', the receiver 'Bob', and the adversary 'Eve'. The challenge is to devise a strategy that allows Alice and Bob to communicate securely and reliably in a limited number of transmission rounds. We focus on one-round protocols.

In the original setting of [3], the protocols are required to be perfectly secure, meaning that no matter what Eve might attempt, she will gain no information about the message. They are also required to be perfectly reliable such that Bob will always recover the correct message. Later, [4] relaxed these conditions to allow some small failure probabilities for both security and reliability. Taking this idea even further, [9] considers protocols where the security of the message delivery is *not* required, but only reliable

 The published version can be found at https://doi.org/10.1016/j.ipl.2019.02.011

transmission is of interest. They call this *unconditionally reliable message transmission*, but we will omit 'unconditionally' and write RMT instead.

To asses the efficiency of a message transmission-protocol, it is common to use the *transmission rate* defined as the total number of transmitted bits divided by the bit-length of the message. Hence, a low transmission rate is preferable. As shown in [9, Theorem 3], we cannot do better than $\Omega(1)$ for RMT, and this bound is tight. In Section 3, however, we show that this transmission rate is not achievable for messages of a constant size.

1.1. Related work

RMT has also been studied in [9, 10]. The protocol in [10] is based on list-decoding of folded Reed-Solomon codes, but although it attains the optimal transmission rate, the computational cost for the receiver to recover the message is exponential in the number of channels. The work [9] contains bounds and constructions for both the secure and the reliable-only settings. In addition, they achieve this while tolerating a mixed adversary, giving more fine-grained control of the adversarial assumptions.

Although this paper is only concerned with RMT, we also direct the reader to related works on secure message transmission; that is, protocols that also offer privacy. This additional guarantee comes at a cost. As shown by [3], perfect security for $n = 2t + 1$ requires at least two rounds, and a single-round protocol can only offer security in the case $n \geq 3t + 1$. In the former setting, Agarwal et al. [1] gave a perfectly secure two-round protocol that achieves optimal performance asymptotically, albeit at a high computational cost. A computationally efficient protocol was subsequently achieved by Kurosawa and Suzuki [8] using the concept of pseudobases. This idea was also taken up by [11], who obtained further improvements, reducing the minimally required message size from $\mathcal{O}(n^2 \log n)$ to $\mathcal{O}(n \log n)$.

The setting where privacy is perfect, but reliability is not, was initially handled by [4] under the assumption that channels support multicast. The proposed solution, however, was inefficient for certain values of t and n. This was rectified in [13], where an efficient protocol for these values was given.

2. Preliminaries

2.1. Model assumptions

We assume that Alice and Bob are connected via $n = 2t+1$ *simple* channels, meaning that the channels allow both Alice and Bob to transmit data, but

no additional functionality is assumed. Before the protocol begins, Eve chooses t of these to be under her control. In other words, the adversary in our model is *static* and *active*.

For simple channels, [4] showed that $2t \geq n$ leads to a probability of failure of at least $1/4$. Hence, the setting where $n = 2t + 1$ has the maximal number of corruptions that we can hope to overcome. Since a majority of the channels are honest – i.e. not controlled by the adversary – a naive solution to the RMT-problem is to broadcast the message across all n channels. This leads to a transmission rate of n, but gives perfect reliability. Thus, this is the benchmark performance.

2.2. Universal hash families

The methods we present rely on the concept of ε-almost universal hash families as introduced by [12].

Definition 2.1:
Let $\mathcal{H}$ be a family of hash functions from $\mathcal{M}$ to A, and let $\varepsilon \in \mathbb{R}_+$. Then $\mathcal{H}$ is called ε-almost universal if for any $m \neq m' \in \mathcal{M}$,

$$\Pr_{h \leftarrow \mathcal{H}}[h(m) = h(m')] \leq \varepsilon.$$

Using the concept of ε-almost universal hash families aids in the setting of message transmission, since it gives a relatively easy, yet general, way to analyse the probability that an adversary can successfully tamper with information without being detected.

In particular, we use a hash family based on polynomial evaluation similar to the one used in [2], but generalized to evaluate in several points.

Definition 2.2:
Let $\mathbb{F}$ be a finite field, and $\mathcal{K} \subseteq \mathbb{F}$. For every pair of positive integers $\eta \leq a$, define the map $\mathsf{PEval}^\eta \colon \mathbb{F}^a \times \mathcal{K}^\eta \to \mathbb{F}^\eta$ by

$$\mathsf{PEval}^\eta(\mathbf{m}, \mathbf{k}) = (f_{\mathbf{m}}(k_1), f_{\mathbf{m}}(k_2), \ldots, f_{\mathbf{m}}(k_\eta)),$$

where $f_{\mathbf{m}}(x) = \sum_{i=1}^{a} m_i x^i$. We use the notation $\mathsf{PEval}^\eta_{\mathbf{k}}(\mathbf{m}) = \mathsf{PEval}^\eta(\mathbf{m}, \mathbf{k})$.

Proposition 2.3:
Let $\mathbb{F}$ be a finite field, and $\mathcal{K} \subseteq \mathbb{F}$. For any $\mathbf{m} \neq \mathbf{m}' \in \mathbb{F}^a$, and $1 \leq \eta \leq a$, the hash function from Definition 2.2 satisfies

$$P_{\mathbf{k} \leftarrow \mathcal{K}^\eta}[\mathsf{PEval}^\eta_{\mathbf{k}}(\mathbf{m}) = \mathsf{PEval}^\eta_{\mathbf{k}}(\mathbf{m}')] \leq \frac{a^\eta}{|\mathcal{K}|^\eta}.$$

In other words, $\mathcal{H}^{\eta}_{\mathsf{PEval}}$ is $(a/|\mathcal{K}|)^{\eta}$-almost universal.

PROOF: Sampling the key $\mathbf{k}$ uniformly from $\mathcal{K}^{\eta}$ corresponds to sampling η keys $k_1, k_2, \ldots, k_{\eta}$ uniformly and independently from $\mathcal{K}$. Hence, the event

$$\mathsf{PEval}^{\eta}_{\mathbf{k}}(\mathbf{m}) = \mathsf{PEval}^{\eta}_{\mathbf{k}}(\mathbf{m}')$$

reduces to the intersection of events

$$\bigcap_{i=1}^{\eta} \left(\mathsf{PEval}^{1}_{k_i}(\mathbf{m}) = \mathsf{PEval}^{1}_{k_i}(\mathbf{m}')\right),$$

Since the keys k_i are independent random variables, it follows that the individual events in the intersection are independent as well, and each happens with probability at most $\frac{a}{|\mathcal{K}|}$, see [2, Claim 2.5]. Hence, the intersection of the events happens with probability at most $(a/|\mathcal{K}|)^{\eta}$, proving the proposition.■

3. Constant-size messages

One could hope that the overall optimal transmission rate $\Theta(1)$ is achievable for constant-size messages. As we show in Proposition 3.2, however, this is not possible for arbitrary reliability parameters. The proof of the proposition relies on the following result from [4, Theorem 5.1].

Theorem 3.1:
Assume that $n \leq 2t$, and denote by $\mathcal{M}$ the message space. Then any reliable message transmission protocol fails with probability at least $\frac{1}{2}(1 - 1/|\mathcal{M}|)$.

Proposition 3.2:
Let $n = 2t + 1$, and consider the RMT-problem for a message of size $\Theta(1)$ bits. Then it is impossible to construct a protocol attaining a transmission rate lower than $\Theta(n)$ for arbitrary reliability parameters.

PROOF: Assume for contradiction that $\mathcal{P}$ is such a protocol. We show the existence of an adversarial strategy such that $\mathcal{P}$ will fail with a probability greater than a constant.

Note that if all n available channels are used, at least n bits will be transmitted during the protocol. Hence, $\mathcal{P}$ can use at most $n - 1$ channels. Let $X \in \{1, 2, \ldots, n\}$ be a random variable describing the unused channel. No assumptions are made about the probability distribution of X; it simply depends on $\mathcal{P}$. Consider an adversarial strategy where the corrupt channels

are chosen uniformly at random. Equivalently, we can assume that the honest channels are given by the set $\{I_1, I_2, \ldots, I_{t+1}\}$, where $I_1 \in \{1, 2, \ldots, n\}$ is a uniformly random variable. The remaining I_j are defined in a recursive manner such that for any $a \in \{1, 2, \ldots, n\}$,

$$\Pr[I_j = a] = \begin{cases} \frac{1}{2t+2-j} & \text{if } I_1 \neq a, I_2 \neq a, \ldots, I_{j-1} \neq a \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

It may be shown that in fact $\Pr[I_j = a] = 1/(2t+1)$ for any j and a; see Appendix 1.

Denote by E the event that Alice leaves out one of the honest channels when following $\mathcal{P}$; that is, $X = I_j$ for some $j \in \{1, 2, \ldots, t+1\}$. Since Alice does not know the outcomes of $I_1, I_2, \ldots, I_{t+1}$, it follows that X is independent from these variables. Using this fact and the fact that the events $X = I_1, X = I_2, \ldots, X = I_{t+1}$ are disjoint, we obtain that

$$\begin{aligned} \Pr[E] &= \Pr[X = I_1 \vee \cdots \vee X = I_{t+1}] = \sum_{j=1}^{t+1} \Pr[X = I_j] \\ &= \sum_{j=1}^{t+1} \sum_{k=1}^{n} \Pr[X = k] \Pr[I_j = k] = \sum_{j=1}^{t+1} \frac{1}{2t+1} \sum_{k=1}^{n} \Pr[X = k] = \frac{t+1}{2t+1}. \end{aligned}$$

If E occurs, it follows from Theorem 3.1 that the probability of protocol failure is at least $\frac{1}{2}(1 - 1/|\mathcal{M}|)$, where $\mathcal{M}$ is the message space. Otherwise, the protocol $\mathcal{P}$ gives a contradiction to Theorem 3.1 since for $n = 2t$, we could introduce a 'dummy channel', discard it, and then mimic protocol $\mathcal{P}$ to obtain a lower probability of failure.

By applying the law of total probability, we obtain

$$\begin{aligned} \Pr[\mathcal{P} \text{ fails}] &= \Pr[\mathcal{P} \text{ fails} \mid E] \Pr[E] + \Pr[\mathcal{P} \text{ fails} \mid \bar{E}] \Pr[\bar{E}] \\ &\geq \Pr[\mathcal{P} \text{ fails} \mid E] \Pr[E] \\ &\geq \frac{1}{2} \left(1 - \frac{1}{|\mathcal{M}|}\right) \frac{t+1}{2t+1} > \frac{1}{4} \left(1 - \frac{1}{|\mathcal{M}|}\right). \end{aligned}$$

In conclusion, it is not possible to obtain arbitrarily levels of reliability with a transmission rate of less than $\Theta(n)$ for constant size messages.

It is worth pointing out that this result is true for any RMT-protocol; not only one-round ones.

4. A method based on list-decoding

As part of a protocol for robust secret sharing, [2] introduced the notion of a 'robust distributed storage'. Their method for achieving this can easily be converted to a one-round protocol for RMT. However, since the original authors only need the asymptotical performance, they base their method on the list-decoding algorithm of Sudan, and use messages of size at most $\lfloor n/8 \rfloor + 1$. This may be increased to $\lfloor n/5 \rfloor + 1$ with no penalty in reliability by applying the Guruswami-Sudan algorithm [5] instead. The full protocol description is given in Protocol 1 on page 17, and it relies on the following theorem from [6, Theorem 6.16 (iii)].

Theorem 4.1:
Let $\mathcal{C}$ be a Reed-Solomon code of length n and dimension a. If at most $n - \sqrt{(1+\rho)(a-1)n}$ errors occur, it is possible to perform list decoding using a list of size $\mathcal{O}\big(\rho^{-1}\sqrt{n/(a-1)}\big)$.

The adversary can introduce up to $t < \frac{n}{2}$ errors, meaning that we can correct all t errors if we require that $\sqrt{(1+\rho)(a-1)n} \leq \frac{n}{2}$. If we further write $(a-1) = Rn$ for some $0 < R < 1$, we obtain

$$\sqrt{(1+\rho)Rn^2} \leq \frac{n}{2} \quad \Longleftrightarrow \quad (1+\rho)R \leq \frac{1}{4}.$$

Fixing a desired 'rate' R, allows us to fix the value of ρ; that is, both are constant values. Thus, by Theorem 4.1 the list size returned by the decoding algorithm will be

$$\mathcal{O}\left(\rho^{-1}\sqrt{\frac{n}{Rn}}\right) = \mathcal{O}(1).$$

For concreteness, we give explicit list-decoding parameters for $R = 1/5$. In order for list decoding of τ errors to be possible, [7, p. 131] gives the two conditions

$$\tau \leq n - \frac{L(a-1)}{s} \tag{2}$$

$$\tau < \frac{n(2L-s+1)}{2(L+1)} - \frac{L(a-1)}{2s}, \tag{3}$$

where s is the multiplicity parameter passed to the algorithm. Since we wish to correct $\tau = t$ errors out of $n = 2t+1$ shares, we can substitute these

into the inequalities, together with $a - 1 = Rn = 2t/5 + 1/5$. For (2) this yields

$$t \leq 2t + 1 - \frac{L\left(\frac{2}{5}t + \frac{1}{5}\right)}{s} \quad \Longleftrightarrow \quad \frac{2L}{5s}t + \frac{L}{5s} \leq t + 1.$$

Here, it is seen that $L = 5$ and $s = 2$ is a possible solution. Rewriting (3) in the same way, and substituting the aforementioned values of L and s, we see that this inequality is satisfied as well. Thus, Protocol 1 is capable of sending messages of length $a = \left\lfloor \frac{n}{5} \right\rfloor + 1$ rather than $a = \left\lfloor \frac{n}{8} \right\rfloor + 1$ as in [2]. The output list is 5 codewords in this case.

4.1. Protocol reliability

If the family of hashes $\mathcal{H}$ applied in Protocol 1 is ε-almost universal, the reliability of the protocol is $nL\varepsilon$, as shown in [Preprint of 2, Theorem 4.3]. This is stated formally below.

Proposition 4.2:
Protocol 1 fails with probability at most $nL\varepsilon$, when the list decoding algorithm returns a list of at most L elements, and the hash family applied is ε-almost universal.

The crux of the argument is the observation that the original message $\mathbf{m}$ must appear on the list returned by the decoding algorithm. Hence, in order for the algorithm to fail, there must be some other message $\mathbf{m}_i$ that agrees with the hash key/value (k_j, v_j) of some uncorrupted channel j. By the ε-almost universality, this happens with probability at most ε, and the overall reliability follows from a union bound argument.

4.2. Number of bits transmitted

Each channel receives a single field element s_i, a key k_i and a hash value v_i. Thus, the total number of $\mathbb{F}$-symbols sent during Protocol 1 is

$$n(1 + |\mathcal{K}| + |\mathcal{V}|),$$

where $|\mathcal{K}|$ and $|\mathcal{V}|$ denote the number of field symbols necessary to represent the key k_i and the hash v_i, respectively.

4.3. Using polynomial evaluation

To be more concrete, we consider the use of the hash family $\mathcal{H}^{\eta}_{\mathsf{PEval}}$. In this case, it follows that $|\mathcal{K}| = |\mathcal{V}| = \eta$, which gives a total of $n(1 + 2\eta)$

field elements transmitted. In order to determine the transmission rate we assume that η is of constant size, and consider messages of size $a = \Theta(n)$. In this case, the transmission rate is $\frac{\Theta(n)}{\Theta(n)} = \Theta(1)$. Thus, Protocol 1 reaches the optimal transmission rate for messages of size $\Theta(n)$.

By Propositions 2.3 and 4.2, the probability of protocol failure is bounded by $nL\varepsilon = nL(a/|\mathbb{F}|)^\eta$. To obtain some desired reliability δ, we can rewrite this expression to obtain a lower bound on the field size. Doing so yields

$$nL\frac{a^\eta}{|\mathbb{F}|^\eta} \leq \delta \quad \Longrightarrow \quad |\mathbb{F}| \geq \left(\frac{nL}{\delta}\right)^{\frac{1}{\eta}} a. \tag{4}$$

In particular, we see that for $\eta = 1$ and $a = \Theta(n)$ the required field size has a quadratic dependence on n. Increasing η allows for smaller field sizes, but increases the number of field symbols transmitted.

5. A method based on erasure decoding

In the following, we will describe the one-round RMT-protocol given in [9] in the language of Reed-Solomon codes and hash families. In this representation, the original authors are essentially relying only on the erasure correcting capabilities of the codes. We show that a careful choice of parameters allows correction of errors as well, causing the required field size to be quadratic rather than cubic in n.

The message we consider is an $a \times b$-matrix M over a finite field $\mathbb{F}$. Each row of this message is encoded by means of an $[n, b]$ Reed-Solomon code, yielding an $a \times n$-matrix S where each row is a codeword. Across the i'th channel, Alice sends the i'th column $\mathbf{s}_i$ of S. Since Bob needs to determine if Eve modified some of these columns during transmission, Alice also computes n verification tags $\{v_{i1}, v_{i2}, \ldots, v_{in}\}$ for each $\mathbf{s}_i$ by applying uniformly sampled hash functions from some family $\mathcal{H}$. Denote the keys of these functions by $\{k_{i1}, k_{i2}, \ldots, k_{in}\}$. Across the i'th channel, Alice then sends $\{\mathbf{s}_i\} \cup \{k_{ji}, v_{ji}\}_{j=1}^n$. That is, each channel will transmit the codeword entries $\mathbf{s}_i$, and a key/tag-pair (k_{ji}, v_{ji}) for every channel j.

When Bob receives the possibly modified values $\{\mathbf{s}'_i\} \cup \{k'_{ji}, v'_{ji}\}_{j=1}^n$, he will check the integrity of $\mathbf{s}'_i$ by computing the hash value $h_{k'_{ij}}(\mathbf{s}'_i)$ and comparing the result with the received tag v'_{ij}. He will do so for each received key/tag-pair, and if more than t tags disagree with the computed values, Bob will mark $\mathbf{s}'_i$ as modified and treat it as an erasure when recovering the message.

With large probability, these checks performed by Bob reveal a considerable part of the corrupt channels delivering erroneous information. This causes a number of columns in S' to be marked as erasures. However, some

small number e of corrupted channels may have passed the checks, meaning that the remaining entries in S' may still contain errors. In fact, each row of S' may contain up to $t - e$ erasures and e errors. If the parameter b agreed upon by Alice and Bob is sufficiently small, Bob may nevertheless correct these erasures and errors in S'. Since the rows of S' are codewords of an $[n, b]$ Reed-Solomon code which has minimal distance $n - b + 1$, Bob can recover the correct message if

$$2e + t - e < n - b + 1 \quad \Longrightarrow \quad b \leq n - (t + e) = t + 1 - e.$$

Thus, after verifying the received values, Bob can determine if the message can be recovered by simply counting the number of non-erased columns and computing syndromes. The complete description of our protocol is given in Protocol 2 on page 18. The correctness of the protocol follows from essentially the same arguments as used by [9], albeit with the following modification.

Lemma 5.1:
If at least $t - e$ columns of S' are marked as erasures in step 4. of Protocol 2, Bob will recover the correct message.

PROOF: Let $u \geq t - e$ be the number of erased columns, meaning that each row of S' contains at most $t - u$ errors. The minimal distance of the code is $d = n - b + 1$, which means that u erasures and $t - u$ errors can be corrected if $2(t - u) + u < d$. This is true because

$$2(t - u) + u = 2t - u \leq t + e \leq n - b,$$

where the last inequality follows from the requirement $e \leq t + 1 - b$ given in the protocol specification. ■

5.1. Protocol reliability

Under the assumption that the hash family $\mathcal{H}$ applied in the protocol is ε-almost universal, we can bound the probability that Bob cannot recover the correct message.

Proposition 5.2:
If $\mathcal{H}$ is an ε-almost universal family of hash functions, then

$$\Pr[\text{Protocol 2 fails}] \leq \frac{t(t+1)\varepsilon}{e+1}.$$

PROOF: By Lemma 5.1, at least $e+1$ of the channels modified by Eve must pass the integrity check performed by Bob. To achieve this, it is necessary that the hash value of the modified $\mathbf{s}'_i$ matches at least one verification tag v_{ij} sent across an honest channel.

The ε-almost universality of $\mathcal{H}$ implies that $\Pr_{h\leftarrow\mathcal{H}}[h(\mathbf{s}_i) = h(\mathbf{s}'_i)] \leq \varepsilon$ whenever $\mathbf{s}_i \neq \mathbf{s}'_i$. Hence, ε is an upper bound on the probability that a single corrupt channel agrees with a single honest channel. Since there are $t+1$ honest channels, the probability for a modified channel to be consistent with at least one honest can be bounded above by $(t+1)\varepsilon$.

Let X be the random variable counting the number of modified but uncaught channels. Since the hash keys k_{ij}, $k_{i'j'}$ are independent whenever $(i,j) \neq (i',j')$, the integrity checks of the modified channels can be considered as t independent Bernoulli trials, each with a success probability of at most $(t+1)\varepsilon$. Thus, X follows a binomial distribution, and has expected value $\mathbb{E}[X] \leq t(t+1)\varepsilon$. The Markov inequality now gives

$$\Pr[X \geq e+1] \leq \frac{\mathbb{E}[X]}{e+1} \leq \frac{t(t+1)\varepsilon}{e+1},$$

and the result follows.

5.2. Number of bits transmitted

When Protocol 2 is used to transmit a message, the total number of $\mathbb{F}$-symbols transmitted is

$$n(a + n|\mathcal{V}| + n|\mathcal{K}|),$$

where $|\mathcal{V}|$ and $|\mathcal{K}|$ denote the number of field symbols necessary to represent v_{ij} and k_{ij}, respectively.

5.3. Using polynomial evaluation

For concreteness, we analyse the reliability of Protocol 2 when $\mathcal{H}^{\eta}_{\mathsf{PEval}}$ is applied with $\mathcal{K} = \mathbb{F}$. Here, both the keys and the verification tags consist of η field elements. Hence, the total number of transmitted field elements is $2\eta n^2 + an$. Depending on the message size, this can give various transmission rates, but under the assumption that η is some constant value, the optimal transmission rate of $\Theta(1)$ is obtained when both a and b are $\Theta(n)$. That is, when the message is of size $\Theta(n^2)$.

Since the hash family is $(a/|\mathbb{F}|)^\eta$-almost universal, it follows from Proposition 5.2 that we must require

$$\frac{t(t+1)a^\eta}{(e+1)|\mathbb{F}|^\eta} \leq \delta \quad \Longrightarrow \quad |\mathbb{F}| \geq a\left(\frac{t(t+1)}{(e+1)\delta}\right)^{\frac{1}{\eta}}.$$

in order to obtain reliability δ. In particular, we note that for $\eta = 1$, the original protocol by [9] requires $|\mathbb{F}| \geq n^3/\delta$. In the proposed protocol, we can set both b and e to be $\Theta(n)$ and obtain the requirement $|\mathbb{F}| \geq \Theta(n^2/\delta)$. In other words, by reducing the second dimension of the message, the required field size is reduced by a factor of n asymptotically. Furthermore, introducing the parameter η highlights the trade-off between the number of $\mathbb{F}$-symbols transmitted and the required field size.

6. Comparisons

6.1. Comparison with existing protocols

In order to compare the RMT-protocols proposed in Sections 4 and 5 to those already in the literature, we will restrict ourselves to the hash family $\mathcal{H}^\eta_{\mathsf{PEval}}$ from Definition 2.2 with $\mathcal{K} = \mathbb{F}$ and $\eta = 1$.

For five protocols, Table 1 gives an overview of the required field size given δ; the message size in $\mathbb{F}$-symbols; whether the protocol attains the optimal transmission rate; and whether it is computationally efficient. Here, *efficient* means polynomial in the number of available channels. We use the Θ-notation to keep the presentation as clear and self-contained as possible.

For the protocol of Section 5, we remark that $a = \Theta(n)$ was chosen even though it is in principle possible to use any value smaller than $|\mathbb{F}|$. Choosing greater values, however, also increases the required field size. We shall refrain from doing such analysis here since Table 1 already shows the desired improvement.

As the table indicates, the first two protocols are better suited for small message sizes. Although both have the same asymptotic performance, the modification suggested in Section 4 allows a larger message size. The remaining three protocols all have $\Theta(n^2)$ as the optimal message size, which suggests that they should fare better when transmitting larger messages. It may be noted that the protocol proposed in Section 5 achieves this while reducing the required field size by a factor of n asymptotically.

Even though Table 1 gives an overview of the general properties of each protocol, it does not reveal how they will perform in concrete problem instances. If the message size and the number of channels have already

Protocol	Field size	Message size	Optimal	Computational efficiency
[2, Sec. 4.1]	$\Theta(n^2/\delta)$	$\lfloor n/8 \rfloor + 1$	✓	✓
Protocol 1	$\Theta(n^2/\delta)$	$\lfloor n/5 \rfloor + 1$	✓	✓
[9, Sec. 4]	n^3/δ	$\Theta(n^2)$	✓	✓
[10, Sec. 3.1]	$\Theta(n^4)$	$\Theta(n^2)$	✓	✗
Protocol 2	$\Theta(n^2/\delta)$	$\Theta(n^2)$	✓	✓

Table 1: Comparison of one-round RMT-protocols. The second column shows the minimal field size given a desired reliability parameter δ. The third column gives the message size (in terms of $\mathbb{F}$-elements) that leads to an optimal transmission rate, and the fourth indicates whether such an optimal transmission rate is achievable. The final column states whether the computational cost is at most polynomial in the number of channels.

been fixed, a separate analysis is needed to determine the protocol that will perform the best.

6.2. Comparing Protocols 1 and 2

In Sections 4 and 5 we saw that Protocols 1 and 2 attain the optimal transmission rate for messages of size $\Theta(n)$ and $\Theta(n^2)$, respectively. Furthermore, both protocols have a quadratic relation between $|\mathbb{F}|$ and n when choosing the parameters appropriately. Loosely speaking, this indicates that Protocol 1 is suited for smaller messages, and Protocol 2 for larger messages.

In a setting where a steady stream of characters is to be transmitted rather than a single message, the distinction between 'small' and 'large' messages looses its importance. Instead, the aim is to achieve as low a transmission rate as possible and then pack the stream characters in appropriately sized messages. In the case $\eta = 1$, we show that the right choice of parameters causes Protocol 2 to outperform Protocol 1 in both transmission rate and field size.

For Protocol 1, we set the dimension of the code to $a = \left\lfloor \frac{n}{5} \right\rfloor + 1$ as in Section 4. The transmission rate is given by

$$\frac{n(2\eta+1)}{\left\lfloor \frac{n}{5} \right\rfloor + 1} = \frac{3n}{\left\lfloor \frac{n}{5} \right\rfloor + 1},$$

which is bounded below by

$$\frac{3n}{\left\lfloor \frac{n}{5} \right\rfloor + 1} \geq \frac{3n}{\frac{n}{5} + 1} = \frac{15n}{n+5}.$$

Noticing that this bound increases for greater values of n, we let $t_{\min}$ be the minimal number of corrupt channels considered. This means that

$$\frac{15(2t_{\min}+1)}{2t_{\min}+6}. \tag{5}$$

gives a lower bound on the transmission rate of Protocol 1 for any $t \geq t_{\min}$. The requirement of the field size is given in (4), and if $\eta = 1$, this leads to the necessary condition that

$$|\mathbb{F}| \geq \frac{anL}{\delta} = \frac{5an}{\delta} > \frac{n^2}{\delta}. \tag{6}$$

In the case of Protocol 2 the transmission rate is given by

$$\frac{n(a+2\eta)}{ab} = \left(\frac{2n}{a}+1\right)\frac{n}{b}. \tag{7}$$

Recall that the requirement on field size is quadratic only if $e = \Theta(n)$. In order to get this quadratic dependence, we assume that $b = \alpha t$ and $e = (1-\alpha)t$ for some $\alpha \in (0,1)$. The transmission rate (7) can then be bounded above by

$$\begin{aligned}\left(\frac{2n}{a}+1\right)\frac{n}{b} &= \left(\frac{2n}{a}+1\right)\frac{2t+1}{\frac{\alpha t_{\min}}{t_{\min}}t} \\ &= \left(\frac{2n}{a}+1\right)\frac{2t_{\min}t+t_{\min}}{\alpha t_{\min}t} \\ &\leq \left(\frac{2n}{a}+1\right)\frac{2t_{\min}+1}{\alpha t_{\min}}.\end{aligned}$$

In terms of field size, we see that for $\eta = 1$,

$$\frac{at(t+1)}{\big((1-\alpha)t+1\big)\delta} \leq \frac{at(t+1)}{(1-\alpha)(t+1)\delta} \leq \frac{an}{2(1-\alpha)\delta},$$

which implies that $|\mathbb{F}| \geq \frac{an}{2(1-\alpha)\delta}$ is a sufficient condition for the field used in Protocol 2. If we set $a = n$ as in [9], we see that for $\alpha < 1/2$ this sufficient condition is less restrictive than the necessary condition of Protocol 1 in (6). In other words, Protocol 2 allows smaller field sizes than Protocol 1 if $\alpha < 1/2$.

By comparing the lower bound in (5) with the upper bound in (7), we may determine $t_{\min}$ as a function of α such that Protocol 2 always outperforms Protocol 1 in these cases. We obtain that

$$\frac{15(2t_{\min}+1)}{2t_{\min}+6} \geq \frac{3(2t_{\min}+1)}{\alpha t_{\min}} \quad \Longrightarrow \quad t_{\min} \geq \frac{6}{5\alpha-2}.$$

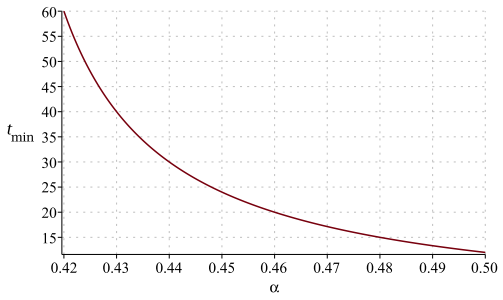

Figure 1: Minimal values of t where Protocol 2 outperforms Protocol 1 as a function of α

A plot of the bound is given in Figure 1. We stress that these considerations relate to the setting where the specific message size is not important.

Bounds can also be made for $\eta \geq 2$, but these do not lead to as meaningful comparisons as above.

6.3. Concrete examples

In order to further highlight the performance differences between the protocols, we will give explicit choices of parameters and compute the transmission rates for each of the two protocols, when the number of channels and the message size are given. In every case, we will require a reliability of 2^{-80}. The examples are given in Tables 2 to 5 on page 19. These are by no means exhaustive. For instance, we have only considered fields of sizes 2^{2^k} for Protocol 2.

It is seen that Protocol 1 often requires a large field size in order to send the entire message in a single run of the protocol. This stems from the restrictive condition on the dimension of the Reed-Solomon codes used. A better option may therefore be to split the message into smaller chunks and run the protocol several times. Of course, some care has to be taken in doing so, since the reliability will be reduced.

When sending 256 bits for $t = 100$, Protocol 1 clearly outperforms 2, while still using a managable field size. Protocol 2 fares better, however, when the message size increases relative to the number of available channels. It is worth noting that the message dimensions given in Table 4 allow much greater message sizes than required. The reason for stating these dimensions rather than something smaller is that a smaller dimension would not lead to better transmission rates. More precisely, it is possible to decrease b, the second dimension, and increase e accordingly, but the requirement for η remains the same, giving exactly the same transmission rate.

7. Acknowledgements

The author extends his gratitude towards Ignacio Cascudo and Diego Ruano for helpful guidance and fruitful discussions.

8. References

[1] Saurabh Agarwal, Ronald Cramer and Robbert de Haan. "Asymptotically Optimal Two-Round Perfectly Secure Message Transmission". In: *CRYPTO 2006*. Springer, Heidelberg, 2006, pp. 394–408. ISBN: 978-3-540-37433-6. DOI: 10.1007/11818175_24.

[2] Allison Bishop, Valerio Pastro, Rajmohan Rajaraman and Daniel Wichs. "Essentially Optimal Robust Secret Sharing with Maximal Corruptions". In: *EUROCRYPT 2016*. 2016, pp. 58–86. DOI: 10.1007/978-3-662-49890-3_3.

[3] Danny Dolev, Cynthia Dwork, Orli Waarts and Moti Yung. "Perfectly Secure Message Transmission". In: *J. ACM* 40.1 (Jan. 1993), pp. 17–47. ISSN: 0004-5411. DOI: 10.1145/138027.138036.

[4] Matthew Franklin and Rebecca N. Wright. "Secure Communication in Minimal Connectivity Models". In: *J. Cryptol.* 13.1 (Jan. 2000), pp. 9–30. ISSN: 1432-1378. DOI: 10.1007/s001459910002.

[5] V. Guruswami and M. Sudan. "Improved decoding of Reed-Solomon and algebraic-geometric codes". In: *FOCS 1998*. Nov. 1998, pp. 28–37. DOI: 10.1109/SFCS.1998.743426.

[6] Venkatesan Guruswami. *List decoding of error-correcting codes: winning thesis of the 2002 ACM doctoral dissertation competition*. Vol. 3282. Springer Science & Business Media, 2004.

[7] Jørn Justesen and Tom Høholdt. *A Course in Error-Correcting Codes*. 1st ed. European Mathematical Society, 2004. ISBN: 978-3-03719-001-2.

[8] Kaoru Kurosawa and Kazuhiro Suzuki. "Truly efficient 2-round perfectly secure message transmission scheme". In: *IEEE Trans. Inf. Theory* 55.11 (2009), pp. 5223–5232. DOI: 10.1109/TIT.2009.2030434.

[9] Arpita Patra, Ashish Choudhury, C. Pandu Rangan and Kannan Srinathan. "Unconditionally reliable and secure message transmission in undirected synchronous networks: Possibility, feasibility and optimality". In: *Int. J. Appl. Cryptogr.* 2.2 (2010), pp. 159–197. DOI: 10.1504/IJACT.2010.038309.

[10] Reihaneh Safavi-Naini, Mohammed Ashraful Alam Tuhin and Pengwei Wang. "A General Construction for 1-Round δ-RMT and (0, δ)-SMT". In: *ACNS 2012*. Springer, Heidelberg, 2012, pp. 344–362. ISBN: 978-3-642-31284-7. DOI: 10.1007/978-3-642-31284-7_21.

[11] Gabriele Spini and Gilles Zémor. "Perfectly Secure Message Transmission in Two Rounds". In: *TCC 2016-B*. 2016, pp. 286–304. DOI: 10.1007/978-3-662-53641-4_12.

[12] D.R. Stinson. "Universal hashing and authentication codes". In: *Des. Codes Cryptogr.* 4.3 (1994), pp. 369–380. ISSN: 1573-7586. DOI: 10.1007/BF01388651.

[13] Yongge Wang and Yvo Desmedt. "Secure Communication in Multicast Channels: The Answer to Franklin and Wright's Question". In: *J. Cryptol.* 14.2 (Mar. 2001), pp. 121–135. ISSN: 1432-1378. DOI: 10.1007/s00145-001-0002-y.

Appendix 1

Lemma 1.1:
Let $I_1, I_2, \ldots, I_{t+1}$ be defined as in (1). Then for any $j \in \{1, 2, \ldots, t+1\}$ and any $a \in \{1, 2, \ldots, n\}$, we have

$$\Pr[I_j = a] = \frac{1}{2t+1}.$$

PROOF: We first prove that $\Pr[I_1 \neq a, \ldots, I_j \neq a] = (2t+1-j)/(2t+1)$ for $j \in \{1, 2, \ldots, t+1\}$, and achieve this by induction. The base case $\Pr[I_1 \neq a] = (2t)/(2t+1)$ is obvious. For the induction step, assume that the claim is true for some $k \in \{1, 2, \ldots n-1\}$. We then have

$$\begin{aligned}
&\Pr[I_1 \neq a, I_2 \neq a, \ldots, I_{k+1} \neq a] \\
&\quad = \Pr[I_{k+1} \neq a \mid I_1 \neq a, \ldots, I_k \neq a] \Pr[I_1 \neq a, \ldots, I_k \neq a] \\
&\quad = \frac{2t+1-(k+1)}{2t+1-k} \frac{2t+1-k}{2t+1} \\
&\quad = \frac{2t+1-(k+1)}{2t+1},
\end{aligned}$$

which proves the claim.

To prove the claim of the lemma, we use induction again. The base case $\Pr[I_1 = a] = (1)/(2t+1)$ is trivial. For the induction step, we may use the

law of total probability and the first part of the proof to obtain

$$\begin{aligned}\Pr[I_{k+1} = a] &= \Pr[I_{k+1} = a \mid I_1 \neq a, \ldots, I_k \neq a] \Pr[I_1 \neq a, \ldots, I_k \neq a] + 0 \\ &= \frac{1}{2t+1-k} \frac{2t+1-k}{2t+1} \\ &= \frac{1}{2t+1}.\end{aligned}$$

■

Protocol 1: One-round RMT (using list-decoding)

This protocol allows Alice to reliably send a symbols of a finite field $\mathbb{F}$ to Bob by using $n = 2t + 1$ channels, t of which may be controlled by an adversary. The parameter a must be sufficiently small such that applying the Guruswami-Sudan algorithm on a $[n, a]$ Reed-Solomon code allows correction of t errors with a list of size $L = \mathcal{O}(1)$. The protocol relies on an ε-almost universal hash family $\mathcal{H}$.

---

1. The message $\mathbf{m} \in \mathbb{F}^a$ is encoded using a $[n, a]$ Reed Solomon code, yielding the codeword $(s_1, s_2, \ldots, s_n)$.
2. For $i = 1, 2, \ldots, n$, Alice samples a random key $k_i \in \mathbb{F}^\eta$, and computes $v_i = h_{k_i}(\mathbf{m})$.
3. Across the i'th channel, Alice transmits $\{s_i, k_i, v_i\}$.
4. Bob receives the possibly modified values $\{s'_i, k'_i, v'_i\}$ for $i = 1, 2, \ldots, n$. He uses the Guruswami-Sudan algorithm on the word $(s'_1, s'_2, \ldots, s'_n)$ to obtain a list of L potential messages $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_L$.
5. For each of the L messages, Bob checks that $\mathbf{m}_i$ agrees with at least $t + 1$ of the hash values v_j. If not, he removes $\mathbf{m}_i$ from the list.
6. If only a single $\mathbf{m}_i$ remains, Bob outputs this message. Otherwise, the protocol has failed.

Protocol 2: One-round RMT-protocol (using erasure-decoding)

This protocol allows Alice to reliably send ab symbols of a finite field $\mathbb{F}$ to Bob in one round by using $n = 2t+1$ channels, t of which may be controlled by an adversary. Beforehand, Alice and Bob have agreed upon a parameter $e \in \mathbb{N}$, which satisfies $e \leq t+1-b$. Additionally, they agree on an ε-almost universal hash family $\mathcal{H}$.

---

1. The message is represented as a matrix $M \in \mathbb{F}^{a \times b}$ and each row is encoded using an $[n, b]$ Reed-Solomon code over $\mathbb{F}$.

2. For each column $\mathbf{s}_i$ of the resulting codewords, Alice samples uniformly and independently n keys $\{k_{i1}, k_{i2}, \ldots, k_{in}\}$ and computes $v_{ij} = h_{k_{ij}}(\mathbf{s}_i)$ for each $j \in \{1, 2, \ldots, n\}$.

3. Across the i'th channel, Alice transmits $\{\mathbf{s}_i\} \cup \{k_{ji}, v_{ji}\}_{j=1,2,\ldots,n}$.

4. Bob receives the possibly modified values $\{\mathbf{s}'_i\} \cup \{k'_{ji}, v'_{ji}\}_{j=1,2,\ldots,n}$ for $i = 1, 2, \ldots, n$. For each i, he compares the tag v'_{ij} received from the j'th channel to the hash value $h_{k'_{ij}}(\mathbf{s}'_i)$. If these disagree for more than t channels $j \in \{1, 2, \ldots, n\}$, he will mark $\mathbf{s}_i$ as modified.

5. For each row in S', Bob computes the syndrome to check if it contains errors. Depending on the result, he proceeds with one of the three following steps.

   a) **The syndrome is zero:** S' contains no errors, meaning that Bob can simply use polynomial interpolation to recover the message.

   b) **The syndrome is nonzero, and S' contains at least $t-e$ erased columns:** Bob uses a decoding algorithm for Reed-Solomon codes to correct the erasures and errors, hereby recovering the message.

   c) **The syndrome is nonzero, and S' contains less than $t-e$ erased columns:** Too many modified channels have passed the integrity checks. The protocol has failed.

Protocol	Field size	Message dimension	Parameters	Bits transmitted	Percent of broadcast
1	2^{2048}	1	$\eta = 1$	18432	300.0 %
2	2^{64}	$(16, 2)$	$\eta = 2, e = 0$	5376	87.5 %
2	2^{32}	$(32, 2)$	$\eta = 3, e = 0$	4800	78.1 %
2	2^{16}	$(64, 2)$	$\eta = 9, e = 0$	5664	92.2 %

Table 2: Examples of performance of Protocols 1 and 2 when sending a message consisting of 2048 bits in the case $t = 1$. All protocols achieve a reliability of 2^{-80}. Broadcast costs 6144 bits.

Protocol	Field size	Message dimension	Parameters	Bits transmitted	Percent of broadcast
1	$2^{8000000}$	1	$\eta = 1$	72000000	300.0 %
2	2^{64}	$(62500, 2)$	$\eta = 2, e = 0$	12002304	50.0 %
2	2^{32}	$(125000, 2)$	$\eta = 6, e = 0$	12003456	50.0 %

Table 3: Examples of performance of Protocols 1 and 2 when sending a 1 megabyte message (8000000 bits) in the case $t = 1$. All protocols achieve a reliability of 2^{-80}. Broadcast costs 24000000 bits.

Protocol	Field size	Message dimension	Parameters	Bits transmitted	Percent of broadcast
1	2^{26}	10	$\eta = 4$	47034	91.4 %
2	2^{16}	$(1, 101)$	$\eta = 6, e = 0$	7760208	1508.1 %
2	2^{8}	$(1, 62)$	$\eta = 11, e = 39$	7112184	1382.2 %

Table 4: Examples of performance of Protocols 1 and 2 when sending a 256 bit message in the case $t = 100$. All protocols achieve a reliability of 2^{-80}. Broadcast costs 51456 bits.

Protocol	Field size	Message dimension	Parameters	Bits transmitted	Percent of broadcast
1	2^{195122}	41	$\eta = 1$	117658566	7.3 %
2	2^{64}	$(1238, 101)$	$\eta = 2, e = 0$	26268288	1.6 %
2	2^{32}	$(2476, 101)$	$\eta = 5, e = 0$	28853952	1.8 %
2	2^{16}	$(5000, 100)$	$\eta = 25, e = 1$	48400800	3.0 %

Table 5: Examples of performance of Protocols 1 and 2 when sending a 1 megabyte message (8000000 bits) in the case $t = 100$. All protocols achieve a reliability of 2^{-80}. Broadcast costs 1608000000 bits.