

# Codes against online adversaries, part I: Large alphabets

B. K. Dey, S. Jaggi, M. Langberg

## Abstract

In this work we consider the communication of information in the presence of an *online* adversarial jammer. In the setting under study, a sender wishes to communicate a message to a receiver by transmitting a codeword  $\mathbf{x} = (x_1, \dots, x_n)$  symbol-by-symbol over a communication channel. The adversarial jammer can view the transmitted symbols  $x_i$  one at a time, and can change up to a  $p$ -fraction of them. However, for each symbol  $x_i$  the jammer's decision on whether to corrupt it or not (and on how to change it) must depend only on  $x_j$  for  $j \leq i$ . This is in contrast to the “classical” adversarial jammer which may base its decisions on its complete knowledge of  $\mathbf{x}$ . More generally, for a *delay* parameter  $\delta \in (0, 1)$ , we study the scenario in which the jammer's decision on the corruption of  $x_i$  must depend solely on  $x_j$  for  $j \leq i - \delta n$ .

In this work, the transmitted symbols are assumed to be over a sufficiently large field  $\mathbb{F}$ . We present a *tight* characterization of the amount of information one can transmit in both the 0-delay and, more generally, the  $\delta$ -delay online setting. We show that for 0-delay adversaries, the achievable rate asymptotically equals that of the classical adversarial model. For positive values of  $\delta$ , we consider two types of jamming, *additive* and *overwrite*. We also extend our results to a *jam-or-listen* online model, where the online adversary can *either* jam a symbol *or* eavesdrop on it.

## I. INTRODUCTION

Consider the following adversarial communication scenario. A sender Alice wishes to transmit a message  $u$  to a receiver Bob. To do so, Alice encodes  $u$  into a codeword  $\mathbf{x}$  and transmits it over a channel. In this work the codeword  $\mathbf{x} = x_1, \dots, x_n$  is considered to be a vector of length  $n$  over an alphabet  $\mathbb{F}_Q$  of size  $Q$ . However, Calvin, a malicious adversary, can observe  $\mathbf{x}$  and corrupt up to a  $p$ -fraction of the  $n$  transmitted symbols (*i.e.*,  $pn$  symbols from  $\mathbb{F}_Q$ ).

<sup>0</sup>B. K. Dey is with the Department of Electrical Engineering, Indian Institute of Technology Bombay, Mumbai, India, 400 076, email: bikash@ee.iitb.ac.in,

S. Jaggi is with the Department of Information Engineering, Chinese University of Hong Kong, Shatin, N.T., Hong Kong, email: jaggi@ie.cuhk.edu.hk,

M. Langberg is with the Computer Science Division, Open University of Israel, 108 Ravutski St., Raanana 43107, Israel, email: mikel@openu.ac.il

The work of B. K. Dey was supported in part by Bharti Centre for Communication in IIT Bombay, the work of S. Jaggi was supported by CERG grant 2150581 and the CUHK MoE-Microsoft Key Laboratory of Human-centric Computing and Interface Technologies, and the work of M. Langberg was supported in part by ISF grant 480/08 and by the Open University of Israel's research fund (grant no. 46114). Authors appear in alphabetical order.

In the classical adversarial channel model, *e.g.*, [12], [3], it is usually assumed that Calvin has full knowledge of the entire codeword  $\mathbf{x}$ , and based on this knowledge (together with the knowledge of the code shared by Alice and Bob) Calvin can maliciously plan what error to impose on  $\mathbf{x}$ . We refer to such an adversary as an *omniscient* adversary. For large values of  $Q$  (which is the focus of this work) communication in the presence of an omniscient adversary is well-understood. It is known that Alice can transmit no more than  $(1 - 2p)n$  error-free symbols to Bob when using codewords of block length  $n$ . Further, efficient schemes such as Reed-Solomon codes [16], [1] are known to achieve this optimal rate.

**Online adversaries:** In this work we address the analysis of coding schemes that allow communication against certain adversaries that are weaker than the omniscient adversary. We consider adversaries that behave in an *online* or *causal* manner. Namely, for each symbol  $x_i$ , we assume that Calvin decides whether to change it or not (and if so, how to change it) based on the symbols  $x_j$ , for  $j \leq i$  alone, *i.e.*, the symbols that he has already observed. In this case we refer to Calvin as an *online* adversary.

Online adversaries arise naturally in practical settings, where adversaries typically have no *a priori* knowledge of Alice's message  $u$ . In such cases they must simultaneously learn  $u$  based on Alice's transmissions, and jam the corresponding codeword  $\mathbf{x}$  accordingly. This *causality* assumption is reasonable for many communication channels, both wired and wireless, where Calvin is not co-located with Alice. For example consider the scenario in which the transmission of  $\mathbf{x} = x_1, \dots, x_n$  is done during  $n$  channel uses over time, where at time  $i$  the symbol (or packet)  $x_i$  is transmitted over the channel. Calvin can only corrupt a packet when it is transmitted (and thus its error is based on its view so far). To decode the transmitted message, Bob waits until all the packets have arrived. As in the omniscient model, Calvin is restricted in the number of packets  $pn$  he can corrupt. This might be because of limited processing power or limited transmit energy.

In addition to the online adversary described above, we also consider the more general scenario in which Calvin's jamming decisions are delayed. That is, for a delay parameter  $\delta \in (0, 1)$ , Calvin's decision on the corruption of  $x_i$  must depend solely on  $x_j$  for  $j \leq i - \delta n$ . We refer to such adversaries as  $\delta$ -*delay online* adversaries. Such  $\delta$ -delay online adversaries correspond, for example, to the scenario in which the error transmission of the adversary is delayed due to certain computational tasks that the adversary needs to perform. We show that the 0-delay model (*i.e.*,  $\delta = 0$ ) and the  $\delta$ -delay model for  $\delta > 0$  display different behaviour, hence we treat them separately.

**Error model:** We consider two types of attacks by Calvin.

- An *additive* attack is one in which Calvin can add  $pn$  error symbols  $e_i$  to Alice's transmitted symbols  $x_i$ . Thus  $y_i$ , the  $i$ 'th symbol Bob receives, equals  $x_i + e_i$ . Here addition is defined over the finite field  $\mathbb{F}_Q$  with  $Q$  elements.
- An *overwrite* attack is one in which Calvin overwrites  $pn$  of Alice's transmitted symbols  $x_i$  by the symbols  $y_i$  received by Bob.

Note that in the 0-delay case these two attacks are equivalent. This is because in both cases Calvin can change an  $x_i$  into an arbitrary  $y_i$ ; an additive Calvin can choose  $e_i = y_i - x_i$ , whereas an overwriting Calvin directly uses  $y_i$ . These two attacks are different if we assume that at the time Calvin is corrupting  $x_i$  he has no knowledge of its value – this is exactly the positive-delay  $\delta$  scenario. In what follows we demonstrate that the best rates of communication in fact differ for these two attacks. The two attacks we study are intended to model different physical models of Calvin’s jamming. For instance, in wired packet-based channels Calvin can directly replace some transmitted packets  $x_i$  with some fake packets  $y_i$ , and therefore behave like an overwriting adversary. On the other hand in wireless networks, Bob’s received signal is usually a function of both  $x_i$  and the additive error  $e_i$ .

Lastly we consider the *jam-or-listen* online adversary. In this scenario, in addition to being an online adversary, if Calvin jams a symbol  $x_i$  then he has no idea what value it takes (e.g., even after the entire codeword has been transmitted, the value  $x_i$  of jammed symbols is unknown to Calvin). This model is again motivated by wireless transmissions, where a node can typically either transmit or receive, but not both. For this model, we consider all four combinations of 0-delay/ $\delta$ -delay, and additive/overwrite errors.

Roughly speaking, a rate  $R$  is said to be *achievable* against an adversary Calvin if it is possible for Alice to transmit a message  $u$  of at least  $Rn$  symbols of  $\mathbb{F}_Q$  over  $n$  channel uses to Bob (with probability of decoding error going to zero as  $n \rightarrow \infty$ ). The *capacity*, when communicating in the presence of a certain adversarial model, is defined to be the supremum of all achievable rates. Thus, the capacity characterizes the rate achievable in the adversarial model under study. We denote the capacity of the classical omniscient adversarial channel which can change  $pn$  characters by  $C^{\text{omni}}(p)$ . We denote the capacity of the  $\delta$ -delay online adversarial channels which can change  $pn$  characters by  $C_\delta^{\text{add}}(p)$  for the additive error model, and  $C_\delta^{\text{ow}}(p)$  for the overwrite error model. For the *jam-or-listen* adversary, we denote the corresponding capacities by  $C_\delta^{\text{j1,add}}(p)$  or  $C_\delta^{\text{j1,ow}}(p)$ , depending on whether Calvin uses additive or overwrite errors. We note that it is crucial for our proofs that Alice’s encoders are stochastic, *i.e.*, depend on private randomness that is available to *neither* Bob *nor* Calvin. A detailed discussion of our definitions and notation is given in Section IV.

**Related work:** In this work, we study codes for online adversaries, and present *tight* characterizations of the amount of information one can transmit in both the 0-delay and, more generally, the  $\delta$ -delay online setting. To the best of our knowledge, communication in the presence of an online adversary (with or without delay) has only been addressed in recent works of ours (jointly with Anand Sarwate) [9], [6] on online channels over *small* alphabets (e.g., binary). Nevertheless, we note that the model of online channels, being a natural one, has been “on the table” for several decades and the analysis of the online channel model appears as an open question in the book of Csiszár and Körner [5] (in the section addressing

Arbitrary Varying Channels [2]<sup>1</sup>). Various variants of causal adversaries have been addressed in the past, for instance [2], [8], [18], [19], [15] – however the models considered therein differ significantly from ours. A recent work [4] considers  $\delta$ -delay additive adversaries for the extreme value of  $\delta = 1$  in the context of *secret sharing* schemes and *fuzzy extractors*. Although the motivation and context of [4] differ substantially from ours, their proof techniques (based on certain authentication schemes) bear some resemblance with ours.

As mentioned above, in recent works of ours [9], [6] we study the 0-delay and  $\delta$  delay (for constant  $\delta$ ) additive channel when communicating over binary alphabets. In [9] we present upper bounds on the binary, 0-delay, additive error capacity; whereas in [6] we show that allowing constant delay  $\delta$  the additive error capacity is  $1 - H(p)$  which matches that of the Binary Symmetric Channel. The results of [6] are existential in nature and generalize to additional error models and larger alphabets as well. The majority of proof techniques used in both [9] and [6] along with its results differ substantially from the work at hand. A preliminary version of the results in this paper were presented at [10].

## II. MAIN RESULTS

At a high level, we show in this work that for 0-delay adversaries over large alphabets, the achievable rate equals that of the classical “omniscient” adversarial model. This may at first come as a surprise, as the online adversary is weaker than the omniscient one, and hence one may suspect that it allows a higher rate of communication. We then show, for positive values of the delay parameter  $\delta$ , that the achievable rate can be significantly greater than those achievable against omniscient adversaries.

We stress that our results do not assume any computational limitations on the adversarial jammer Calvin. The codes we construct to achieve the optimal rates are computationally efficient to design, encode and decode. All our results assume that the field size  $Q$  is significantly larger than  $n$ . In some cases it suffices to take  $Q = \text{poly}(n)$ , but in others we need  $Q = \exp(\text{poly}(n))$ . Both settings lend themselves naturally to real-world scenarios, as in both cases a field element  $x_i$  from  $\mathbb{F}_Q$  can be represented by a packet of  $\lceil \log(Q) \rceil$  bits.

The exact statements of our results are in Theorems 1, 2, 3 and 4 below. The capacities given by these theorems are plotted in Fig. 1. The technical parameters (including rate, field size, error probability, and time complexity) of our results are summarized in Table II of the Appendix. We start by showing that in the 0-delay case, the capacity of the online channel equals that of the stronger omniscient channel model.

*Theorem 1 (0-delay model):* For any  $p \in [0, 1]$ , communicating against a 0-delay online adversary channel under both the `overwrite` and `additive error` models equals the capacity under the `omniscient`

<sup>1</sup>The Arbitrary Varying Channel (AVC) model is a broad framework for modeling channels, which encapsulates our online model. For a nice survey on AVCs see [11].

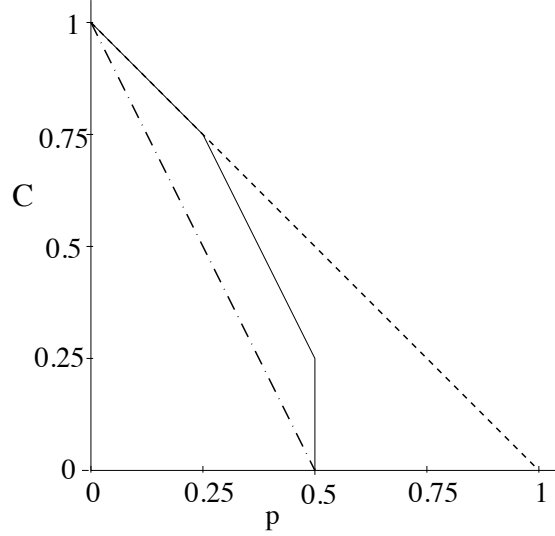


Fig. 1. Capacity of various channels as per Theorems 1–3. The dashed line shows  $C_{\delta}^{\text{add}}(p)$  for  $\delta > 0$ , the firm line shows  $C_{0.25}^{\text{ow}}(p)$ , and the dot-dashed line shows  $C_0^{\text{ow}}(p)$ ,  $C_0^{\text{add}}(p)$  and  $C^{\text{omni}}(p)$ .

model. In particular,  $C_0^{\text{ow}}(p)$ ,  $C_0^{\text{add}}(p)$  and  $C^{\text{omni}}(p)$  all equal

$$(1 - 2p)^+ = \begin{cases} 1 - 2p, & p \in [0, 0.5) \\ 0, & p \in [0.5, 1]. \end{cases} \quad (1)$$

Moreover, the capacity can be attained by an efficient encoding and decoding scheme.

Next we characterize the capacity of the  $\delta$ -delay online channel under the additive error model.

*Theorem 2 ( $\delta$  delay with additive error model):* For any  $p \in [0, 1]$  the capacity  $C_{\delta}^{\text{add}}(p)$  of the  $\delta$ -delay online channel for  $\delta > 0$  under the additive error model is  $1 - p$ . Moreover, the capacity can be attained by an efficient encoding and decoding scheme.

We then turn to study the  $\delta$ -delay online channel under the overwrite error model. The capacity we present is at least as large as that achievable against an additive or overwrite 0-delay adversary who changes  $pn$  symbols. However, it is sometimes significantly lower than that achievable against an additive  $\delta$ -delay adversary.

*Theorem 3 ( $\delta$  delay with overwrite error model):* For any  $p$  and  $\delta$  in  $[0, 1]$  the capacity of the  $\delta$ -delay online channel under the overwrite error model is

$$C_{\delta}^{\text{ow}}(p) = \begin{cases} 1 - p, & p \in [0, 0.5), p \leq \delta \\ 1 - 2p + \delta, & p \in [0, 0.5), p > \delta \\ 0, & p \in [0.5, 1] \end{cases} . \quad (2)$$

Moreover, the capacity can be attained by an efficient encoding and decoding scheme.

Lastly, we show that the optimal rates achievable against a jam-or-listen online adversary equal the corresponding optimal rates achievable against an online adversary, for each of the four combinations of 0- or  $\delta$ -delay, and additive or overwrite attacks.

*Theorem 4 (jam-or-listen model): For any  $p$  and  $\delta$  in  $[0, 1]$  the capacity of the  $\delta$ -delay online channel under the jam-or-listen error model is equal to that of the  $\delta$ -delay online channel:*

$$C_{\delta}^{\text{j1,add}}(p) = C_{\delta}^{\text{add}}(p), \quad C_{\delta}^{\text{j1,ow}}(p) = C_{\delta}^{\text{ow}}(p). \quad (3)$$

*Moreover, the capacity can be attained by the same efficient encoding and decoding schemes as in Theorems 1, 2 and 3.*

The proofs of Theorems 2 and 3 are given in Sections V and VI respectively. The upper bound of Theorem 1 follows from Theorem 3 as a special case (when  $\delta = 0$ ), and thus we omit its proof. The proof of Theorem 4 follows from the proofs of Theorems 2 and 3 and is also omitted. For completeness, a brief discussion addressing Theorem 4 appears in the upcoming Section III.

### III. OUTLINE OF PROOF TECHNIQUES

The proofs of Theorems 1, 2, 3 and 4 require obtaining several non-trivial upper and lower bounds on the capacity of the corresponding channel models. The lower bounds are proved constructively by presenting efficient encoding and decoding schemes operating at the optimal rates of communication. The upper bounds are typically proven by presenting strategies for Calvin that result in a probability of decoding error that is strictly bounded away from zero regardless of Alice and Bob's encoding/decoding schemes.

Theorem 1 states that communication in the presence of a 0-delay online adversary is no easier than communicating in the presence of (the more powerful) omniscient adversary. There already exist efficient encoding and decoding schemes that allow communication at the optimal rate of  $1 - 2p$  in the presence of an omniscient adversary [16], [1]. Thus our contribution in this scenario is in the design of a strategy for Calvin that does not allow communication at a higher rate. The scheme we present allows Calvin to enforce a constant probability of error whenever Alice and Bob communicate at a rate higher than  $1 - 2p$ . Roughly speaking, Calvin uses a two-phase *wait and attack* strategy. In the first phase (whose length depends on  $p$ ), Calvin does not corrupt the transmitted symbols but merely eavesdrops. He is thus able to reduce his ambiguity regarding the codeword  $\mathbf{x}$  that Alice transmits. In the second phase, using the knowledge of  $\mathbf{x}$  he has gained so far, Calvin designs an error vector to be imposed on the remaining part of the codeword that Alice is yet to transmit.

Theorem 2 states that for  $\delta > 0$ , the capacity of the  $\delta$ -delay online channel under the additive error model is  $1 - p$ . Note that this expression is independent of  $\delta$ . In fact, even if Calvin's attack is delayed by just a *single* symbol, the rate of communication achievable between Alice and Bob is strictly greater

than in the corresponding scenario in Theorem 1! The upper bound follows directly from the simple observation that Calvin can always add  $pn$  random symbols from  $\mathbb{F}_Q$  to the first  $pn$  symbols of  $\mathbf{x}$ , and therefore the corresponding symbols received carry no information. The lower bound involves a non-trivial code construction. In a nutshell, we show a reduction between communicating over the  $\delta$ -delay online channel under the additive error model and communicating over an *erasure* channel. In an erasure channel, the receiver Bob is assumed to know which of the  $pn$  elements of the transmitted codeword  $\mathbf{x}$  were erased by Calvin. As one can efficiently communicate over an erasure channel with rate  $1 - p$ , e.g., [3], we obtain the same rate for our online channel. The main question is now: “In our model, how can Bob detect that a received symbol  $y_i$  was corrupted by Calvin?” The idea is to use authentication schemes which are information theoretically secure, and lend themselves to the adversarial setting at hand. Namely, each transmitted symbol will include some internal redundancy, a signature, which upon decoding will be authenticated. As Calvin is a positive delay adversary, it is assumed that he is unaware of both the symbol being transmitted and its signature. It is enough that the signature scheme we construct be resilient against such an adversary.

In Theorem 3 both the lower and upper bound on the capacity require novel constructions. For the upper bound we refine the “wait-and attack” strategy for Calvin outlined in the discussion above on Theorem 1, to fit the  $\delta$ -delay scenario. For the lower bound, we change Alice and Bob’s encoding/decoding schemes, outlined in the discussion above on Theorem 2, to fit the  $\delta$ -delay *overwrite* model. Namely, as before, Alice’s encoding scheme comprises of an erasure code along with a signature used to authenticate individual symbols. However, in general, an *overwrite* adversary is more powerful than an *additive* adversary. This is because an *overwriting* adversary can substitute any symbol  $x_i$  by a new symbol  $y_i$ . Thus Calvin can choose to replace  $x_i$  with a symbol  $y_i$  that includes a valid signature, and will thus pass the authentication done on Bob’s end. Hence the design of the signature scheme for Theorem 3 is more intricate than the corresponding construction in Theorem 2.

Roughly speaking, in the scheme we propose for the  $\delta$ -delay *overwrite* scenario, the redundancy (i.e., signature) added to each symbol  $x_i$  contains information that allows *pairwise* authentication (via a pairwise independent hash function). Namely, each symbol  $x_i$  contains  $n$  signatures  $\sigma_{ij}$  (one for each symbol  $x_j \in \mathbf{x}$ ). Using these signatures, some pairs of symbols  $x_i$  and  $x_j$  can be mutually authenticated to check whether exactly one of them has been corrupted. (For instance, symbols  $x_i$  and  $x_j$  such that  $|i - j| < \delta n$  can be used for mutual authentication, since when Calvin corrupts either one of them he does not yet know the value of the other.) This allows Bob to build a *consistency graph*  $G$  containing a vertex corresponding to each received symbol, and an edge connecting mutually consistent symbols. Bob then analyzes certain combinatorial properties of this consistency graph to extract a maximal set of mutually consistent symbols. He finally inverts Alice’s erasure code to retrieve her message. Naïvely, such a decoding algorithm would take time exponential in the size of the consistency graph  $G$  (as Bob needs to find large cliques in the consistency graph). However, using additional ideas we are able to present an

efficient decoding algorithm. We view Bob’s efficient decoding algorithm as one of the main technical contributions of this work.

Lastly, Theorem 4 states that a `jam-or-listen` adversary is still as powerful as the previously described online adversaries. This is interesting because a `jam-or-listen` adversary is in general weaker than an online adversary, since he *never* finds out the values of the symbols he corrupts. This theorem is a corollary of Theorems 1, 2 and 3 as follows. The code constructions corresponding to the lower bounds are the same as in Theorems 1, 2 and 3. As for the upper bounds, we note that the attacks described for Calvin in Theorems 1, 2 and 3 actually correspond to a `jam-or-listen` adversary, and hence are valid attacks for this scenario as well.

The rest of the paper is organized as follows. In Section IV we present a detailed description of our adversarial models together with some notation to be used throughout our work. In Section V we present the proof of Theorem 2. In Section VI we present the proof of Theorem 3. Some remarks and open problems are finally given in Section VII. The technical parameters of our results are summarized in Table II of the Appendix.

#### IV. DEFINITIONS AND NOTATION

For clarity of presentation we repeat and formalize the definitions presented earlier. Let  $Q$  be a power of some prime integer, and let  $\mathbb{F}_Q$  be the field of size  $Q$ . Let  $\text{poly}(n)$  denote any polynomial of constant degree and with constant valued coefficients (here, a value is referred to as *constant* if it does not depend on the variable  $n$ ). Throughout this work we assume that  $Q$  is exponential in  $\text{poly}(n)$  (although some of our results will only need a  $\text{poly}(n)$  sized  $Q$ ). Also, all logarithms are binary. For any integer  $i$ , let  $[i]$  denote the set  $\{1, \dots, i\}$ . Let  $R \geq 0$  be Alice’s *rate*. An  $[n, Rn]$ -code  $\mathcal{C}$  is defined by Alice’s encoder and Bob’s corresponding decoder, as below.

**Alice:** Alice’s message  $u$  is assumed to be an element in the alphabet  $\mathcal{U} = [Q^{Rn}]$ . Alice’s encoder is *stochastic*. Namely, for encoding, Alice will hold a uniformly distributed *secret*  $r$  which is assumed to be a number of elements (say  $t$ ) of  $[Q]$ . Alice’s secret is assumed to be unknown to *both* Bob and Calvin prior to transmission. Alice’s encoder maps every  $(u, r)$  in  $[Q^{nR}] \times [Q]^t$  to a vector  $\mathbf{x} = (x_1, \dots, x_n)$ .

**Calvin/Channel:** We assume that Calvin is online, namely at the time that the character  $x_i$  is transmitted Calvin has the knowledge of  $\{x_i\}_{i \in K_i}$ . Here the *knowledge set*  $K_i$  is a subset of  $[i]$  that is defined below according to the different jamming models we study. Using his *jamming function* Calvin either replaces Alice’s transmitted symbol  $x_i$  in  $\mathbb{F}_Q$  with a corresponding symbol  $y_i$ , or adds an error  $e_i$  to  $x_i$  such that Bob receives  $y_i = x_i + e_i$ .

In this work, Calvin’s knowledge sets must satisfy the following constraints.

- *Causality/ $\delta$ -delay:* Calvin’s knowledge set  $K_i$  is a subset of  $[i - \delta n]$ .
- *Jam-or-listen:* If Calvin is a `jam-or-listen` adversary,  $K_i$  is inductively defined so that it does not contain  $j \leq i$  such that  $y_j \neq x_j$ . That is, Calvin has no knowledge of any  $x_i$  he corrupts.

Calvin's jamming function must satisfy the following constraints.

- *Knowledge*: For each  $i$ , Calvin's jamming function, and in particular the corresponding *error symbol*  $e_i \in \mathbb{F}_Q$ , depends solely on the set  $\{x_i\}_{i \in K_i}$ , Alice's encoding scheme, and Bob's decoding scheme.
- *Additive/Overwrite*: If Calvin is an additive adversary,  $y_i = x_i + e_i$ , with addition defined over  $\mathbb{F}_Q$ . If Calvin is an overwrite adversary,  $y_i = e_i$ .
- *Power*: Bob's received symbol  $y_i$  differs from Alice's transmitted symbol  $x_i$  in at most  $pn$  locations  $i$ .

**Bob**: Bob's *decoder* is a (potentially) probabilistic function solely of Alice's encoder and the received vector  $\mathbf{y}$ . It maps every vector  $\mathbf{y} = (y_1, \dots, y_n)$  in  $\mathbb{F}^n$  to an element  $u'$  of  $\mathcal{U}$ .

**Code parameters**: Bob is said to make a *decoding error* if the message he decodes  $u'$  differs from that encoded by Alice,  $u$ . The *probability of error* for a given message  $u$  is defined as the probability, over Alice's encoder, Alice's secret  $r$ , Calvin's randomness, and Bob's randomness, that Bob decodes incorrectly. The probability of error of the coding scheme is defined as the *maximum* over all  $u$  of the probability of error for message  $u$ . Results similar to those shown in this work can also be shown for average error. We choose to focus on maximal error since it shows that Calvin, in fact, is powerful enough to cause an error for *every* message Alice transmits to Bob, rather than just on average over her messages.

A rate  $R$  is said to be *achievable* if for every  $\varepsilon, \Delta > 0$  and every sufficiently large  $n$  there exists a *computationally efficient*  $[n, n(R - \Delta)]_Q$ -code that allows communication with probability of error at most  $\varepsilon$  (here  $Q$  may depend on the block size  $n$ ). The supremum of the achievable rates is called the *capacity* and is denoted by  $C$ . We denote the capacity of the  $\delta$ -delay online adversarial channels under the additive error model by  $C_\delta^{\text{add}}(p)$  and under the overwrite error model by  $C_\delta^{\text{ow}}(p)$ . For a jam-or-listen adversary we denote the corresponding capacities by  $C_\delta^{\text{j1,add}}(p)$  and  $C_\delta^{\text{j1,ow}}(p)$ .

We put no computational restrictions on Calvin. This is because our proofs are information-theoretic in nature, and are valid even for a computationally unbounded adversary. However, our schemes provide computationally efficient schemes for Alice and Bob.

**Packets**: For several of our code constructions (specifically those in Theorems 2 and 3), it is conceptually and notationally convenient to view each symbol from  $\mathbb{F}_Q$  as a "packet" of symbols from a smaller finite field  $\mathbb{F}_q$  of size  $q$  instead. In particular, we assume  $q^m = Q$ . Here  $m$  is an integer code-design parameter to be specified later. For a codeword  $\mathbf{x} = x_1, \dots, x_n$ , Alice treats each symbol (or packet)  $x_i$  in  $\mathbb{F}_Q$  as  $m$  sub-symbols  $x_{i,1}$  through  $x_{i,m}$  from  $\mathbb{F}_q$ . Similarly, she treats her secret  $r$  as  $m$  sub-symbols  $r_1$  through  $r_m$  from  $\mathbb{F}_q$ .

We note that, throughout our work, we can allow Calvin to be even stronger than outlined in the model above. In particular, Calvin's jamming function can also depend on Alice's message  $u$ , and our theorems and corresponding proofs remain unchanged. The crucial requirement is that each of Calvin's jamming functions be independent of Alice's secret  $r$ , conditioned on the symbols in the corresponding knowledge set. That is, the only information Calvin has of Alice's secret, he gleans by observing  $\mathbf{x}$ .

## V. PROOF OF THEOREM 2

We consider block length  $n$  large enough so that  $\delta > 1/n$ . Throughout, to simplify our presentation, we assume that expressions such as  $pn$  or  $\delta n$  are integers. We first prove that  $C_\delta^{\text{add}}(p) \leq 1 - p$  by showing a “random-add” strategy for Calvin. Namely, suppose Calvin chooses elements of  $\mathbb{F}_Q$  uniformly at random and adds them to the first  $pn$  transmitted symbols. Thus the first  $pn$  received symbols are i.i.d. uniform over  $\mathbb{F}_Q$ , and carry no information. It can be verified that such an adversarial strategy allows communication between Alice and Bob at rate at most  $1 - p$ . This proves the upper bound.

We now describe how Alice and Bob achieve a rate approaching  $1 - p$  with computationally tractable codes. Alice’s encoding is in two phases. First, roughly speaking, she uses an erasure code to encode the approximately  $(1 - p)n$  symbols of her message  $u$  into an erasure-codeword  $\mathbf{v}$  with  $n$  symbols. The erasure code allows  $u$  to be retrieved from any subset of at least  $(1 - p)n$  symbols of the erasure-codeword  $\mathbf{v}$ . Next, Alice uses  $n$  “short” random keys and corresponding hash functions to transform each symbol  $v_i$  of the erasure-codeword  $\mathbf{v}$  into the corresponding transmitted symbol  $x_i$ . This hash function is carefully constructed so that if Calvin corrupts a symbol  $x_i$ , with high probability Bob is able to detect this in a computationally efficient manner by examining the corresponding received  $y_i$ . Bob’s decoding scheme is also a two-phase process. In the first phase he uses the hash scheme described above to discard the symbols he detects Calvin has corrupted – there are at most  $pn$  such symbols. In the second phase Bob uses the remaining  $(1 - p)n$  symbols and the decoder of Alice’s erasure code to retrieve her message. We assume Alice’s erasure code is efficiently encodable and decodable (for instance Reed-Solomon codes [16], [1] can be used). In what follows we give our code construction in detail.

**Alice:** Let  $Q$  be sufficiently large (to be specified explicitly later in the proof). Let  $m = n^2 + 2n$ . As mentioned in Section IV, Alice treats each symbol of a codeword  $\mathbf{x} = x_1, \dots, x_n$  as a packet, by breaking each  $x_i$  into  $m$  sub-symbols  $x_{i,1}$  through  $x_{i,m}$  from  $\mathbb{F}_q$ . She partitions  $x_{i,1}$  through  $x_{i,m}$  into three consecutive sequences of sub-symbols of sizes  $n^2$ ,  $n$  and  $n$  respectively. The sub-symbols  $x_{i,1}$  through  $x_{i,n^2}$  are denoted by the set  $w_i$ , and correspond to the sub-symbols of  $v_i$ , the  $i$ th symbol of the erasure-codeword  $\mathbf{v}$  generated by Alice. The next  $n$  sub-symbols are denoted by the set  $r_i$ , and consist of Alice’s secret for packet  $i$ , namely,  $n$  sub-symbols chosen independently and uniformly at random from  $\mathbb{F}_q$ . For each  $i$ ,  $r_i$  is chosen independently. The final  $n$  sub-symbols are denoted by the set  $\sigma_i$ , and consist of the hash (or signature) of the information  $w_i$  by the function  $H_{r_i}$ . Here,  $H_{r_i}$  is taken from a family  $\mathcal{H}$  of hash functions (known to all parties in advance) to be defined shortly. All in all, each transmitted symbol  $x_i$  of Alice consists of the tuple  $(w_i, r_i, H_{r_i}(w_i))$ .

We now explicitly demonstrate the construction of each  $w_i$  from Alice’s message  $u$ . Alice chooses  $R = (1 - 2n/m)(1 - p)$ . Thus the message  $u$  she wishes to transmit to Bob has  $m n R = (m - 2n)(1 - p)n = (1 - p)n^3$  sub-symbols over  $\mathbb{F}_q$ . Alice uses an erasure code (resilient to  $pn^3$  erasures) to transform these sub-symbols of  $u$  into the vector  $\mathbf{v}$  comprising of  $n^3$  sub-symbols over  $\mathbb{F}_q$ . She then denotes consecutive

blocks of  $n^2$  sub-symbols of  $\mathbf{v}$  by the corresponding  $w_i$ 's. More specifically,  $w_i$  consists of the sub-symbols in  $\mathbf{v}$  in locations  $n^2(i-1)$  through  $n^2i-1$ .

**Bob:** Before completing the description of Alice's encoder by describing the hash family  $\mathcal{H}$ , we outline Bob's decoder. Bob first authenticates each received symbol  $y_i = (w'_i, r'_i, \sigma'_i)$  by checking that  $H_{r'_i}(w'_i) = \sigma'_i$ . He then decodes using the decoding algorithm of the erasure code on the sub-symbols on  $w'_i$  of all symbols  $y_i$  that pass Bob's authentication test.

**Hash function:** We now define our hash family  $\mathcal{H} = \{H_r\}$  and show that with high probability any corrupted symbol  $y_i \neq x_i$  will not pass Bob's authentication check. We note that  $\mathcal{H}$  is part of the code design and is thus known to all parties Alice, Bob, and Calvin. More specifically, we study only corrupted symbols  $y_i \neq x_i$  for which  $w'_i \neq w_i$ . (If  $w'_i = w_i$ , the erasure decoder described above will not make an error.) Let  $e_i$  be the error imposed by Calvin in the transmission of the  $i$ 'th packet  $x_i$ . Hence for an additive adversary Calvin,  $e_i$  is defined by  $y_i = x_i + e_i$  (here, addition is taken over  $\mathbb{F}_Q$ ). Analogously to the corresponding sub-divisions of  $x_i$  and  $y_i$ , we decompose  $e_i$  into the tuple  $(\hat{w}_i, \hat{r}_i, \hat{\sigma}_i)$ . In particular, we *define*  $\hat{w}_i$ ,  $\hat{r}_i$  and  $\hat{\sigma}_i$  so to satisfy  $w'_i = w_i + \hat{w}_i$ ,  $r'_i = r_i + \hat{r}_i$  and  $\sigma'_i = \sigma_i + \hat{\sigma}_i$ . In the latter additions, our operations are done element wise over  $(\mathbb{F}_q)^{n^2}$ ,  $(\mathbb{F}_q)^n$ , and  $(\mathbb{F}_q)^n$  correspondingly. For Bob to decode correctly, the property that  $y_i$  fails Bob's authentication test if  $\hat{w}_i \neq 0$  needs to be satisfied with high probability. More formally, noting that  $r_i$  is not known to Calvin and thus independent of  $\hat{w}_i$ , we need for all  $i$  and all  $e_i$  such that  $\hat{w}_i \neq 0$ , that

$$\begin{aligned} \Pr_{r_i}[H_{r'_i}(w'_i) = \sigma'_i \mid H_{r_i}(w_i) = \sigma_i] &= \Pr_{r_i}[H_{r_i+\hat{r}_i}(w_i + \hat{w}_i) = \sigma_i + \hat{\sigma}_i \mid H_{r_i}(w_i) = \sigma_i] \\ &= \Pr_{r_i}[H_{r_i+\hat{r}_i}(w_i + \hat{w}_i) - H_{r_i}(w_i) = \hat{\sigma}_i] \end{aligned}$$

is sufficiently small.

To complete our proof we present our hash family  $\mathcal{H} = \{H_r\}$ . Recall that  $w_i$  consists of  $n^2$  sub-symbols in  $\mathbb{F}_q$ . Let  $W_i$  represent  $w_i$  when arranged as an  $n \times n$  matrix. Let  $\mathbf{r}_i$  be a column vector of  $n$  symbols corresponding to  $r_i$ . We define the value of the hash  $H_{r_i}(w_i)$  as the length- $n$  column vector  $\sigma_i$  defined as  $W_i \mathbf{r}_i$ . Thus for the corresponding errors  $\hat{w}_i \neq 0$ ,  $\hat{r}_i$ ,  $\hat{\sigma}_i$  defined above,  $H_{r_i+\hat{r}_i}(w_i + \hat{w}_i) - H_{r_i}(w_i) = \hat{\sigma}_i$  iff  $(W_i + \hat{W}_i)(\mathbf{r}_i + \hat{\mathbf{r}}_i) - (W_i \mathbf{r}_i) = \hat{\sigma}_i$ . Here  $\hat{W}_i$  is the matrix representation of  $\hat{w}_i$  and  $\hat{\mathbf{r}}_i$ ,  $\hat{\sigma}_i$  correspond to  $\hat{r}_i$ ,  $\hat{\sigma}_i$ . Namely, the corrupted symbol received by Bob is authenticated only if

$$\hat{W}_i \mathbf{r}_i = \hat{\sigma}_i - (W_i + \hat{W}_i) \hat{\mathbf{r}}_i. \quad (4)$$

For Calvin to corrupt Alice's transmission, we assume that  $\hat{w}_i \neq 0$  or equivalently  $\hat{W}_i \neq 0$ , therefore the rank of  $\hat{W}_i$  is at least 1. Now, in (4), the left hand side depends on  $r_i$  while the right hand side does not. Hence, it is not difficult to verify that the equation is satisfied by at most  $q^{n-1}$  values for the vector  $\mathbf{r}_i$ . Since  $\mathbf{r}_i$  is uniformly distributed over  $(\mathbb{F}_q)^n$  and unknown to Calvin, the probability of a decoding error is at most  $1/q = o(n^{-1})$  if  $q$  is chosen to be  $n \cdot \omega(1)$ .

All in all, our communication scheme succeeds if each corrupted symbol with  $\hat{w}_i \neq 0$  fails the authentication test. This happens with probability at least  $1 - n/q = 1 - o(1)$  as desired. Taking  $m = n^2 + 2n$  the rate of the code is  $(1 - o(1))(1 - p)$  and the field size needed is  $q^m = \exp(\text{poly}(n))$ .  $\square$

## VI. PROOF OF THEOREM 3

**Proof of Upper bound:** We start by addressing the three cases in the upper bound on the capacity  $C_\delta^{\text{ow}}(p)$ . First, if  $p \leq \delta$ , Calvin corrupts the first  $pn$  symbols uniformly at random as in the proof of Theorem 2 to attain an upper bound of  $1 - p$  on the achievable rate. In the second and third cases we have that  $\delta < p$ . In what follows we present the proof for the second case in which  $p < 1/2$ . The third case will follow from a similar (and simpler) proof.

For  $\delta < p < 1/2$ , we present a “wait-and-attack” strategy for Calvin to prove that  $C_\delta^{\text{ow}}(p) \leq R = 1 - 2p + \delta + \varepsilon$  for any  $\varepsilon > 0$ . Namely, Calvin eavesdrops, without corrupting, the first few symbols Alice transmits. He then overwrites the next  $\delta n$  symbols with symbols chosen uniformly at random from  $\mathbb{F}_Q$ . These  $\delta n$  locations convey no information to Bob. Finally, Calvin carefully corrupts a portion of the remaining symbols Alice transmits. In what follows, we specify this final corruption step (*i.e.*, the “attack” from the ‘wait-and-attack’ strategy) and show that any attempt of Alice and Bob to communicate at rate higher than  $1 - 2p + \delta$  will result in a decoding error with positive probability that is independent of the block length  $n$ . This will suffice to prove our theorem.

We start with some notation. Recall that Alice’s encoding function is stochastic. Namely, for encoding, Alice will hold a uniformly distributed *secret*  $r$  which is assumed to be a number of elements (say  $t$ ) of  $[Q]$ . Alice’s secret is assumed to be unknown to *both* Bob and Calvin prior to transmission. Alice’s encoder maps every  $(u, r)$  in  $[Q^{nR}] \times [Q]^t$  to a vector  $\mathbf{x} = (x_1, \dots, x_n)$ .

Let  $U$  be the random variable corresponding to Alice’s message and  $p_U$  its distribution (which is in our setting uniform). To strengthen our upper bound, in what follows we assume that Alice’s encoder, for every message  $u$ , may have an arbitrary distribution (and not necessarily a uniform distribution obtained via the uniformly distributed secret  $r$  as suggested above). Namely, let  $\mathcal{X}$  be Alice’s codebook.  $\mathcal{X}$  is a collection  $\{\mathcal{X}(u)\}$  of subsets of  $[Q]^n$ . For each subset  $\mathcal{X}(u) \subset \mathcal{X}$ , there is a corresponding codeword random variable  $\mathbf{X}(u)$  with codeword distribution  $p_{\mathbf{X}(u)}$  over  $\mathcal{X}(u)$ . For any value  $U = u$  of the message, Alice’s encoder chooses a codeword from  $\mathcal{X}(u)$  randomly from the distribution  $p_{\mathbf{X}(u)}$ . Alice’s message distribution  $p_U$ , codebook  $\mathcal{X}$ , and all the codebook distributions  $p_{\mathbf{X}(u)}$  are all known to both Bob and Calvin, but the values of the random variables  $U$  and  $\mathbf{X}(\cdot)$  are unknown to them. If  $\mathcal{X}(u) = \{\mathbf{x}(u, r) : r \in \Lambda_u\}$ , then the transmitted codeword  $\mathbf{X}(U)$  has the probability distribution given by  $\Pr[\mathbf{X}(U) = \mathbf{x}(u, r)] = p_U(u)p_{\mathbf{X}(u)}(\mathbf{x}(u, r))$ . Let  $p$  be the overall distribution of codewords  $\mathbf{x} = \mathbf{x}(u, r)$  of Alice. It holds that  $p(\mathbf{x}(u, r)) = p_U(u)p_{\mathbf{X}(u)}(\mathbf{x})$  and  $p(\mathbf{x}) = \sum_U p_U(u)p_{\mathbf{X}(u)}(\mathbf{x})$ .

We now prove the following technical Lemma that we use in our proof. Let  $q$  be an arbitrary probability distribution over an index set  $I = \{1, \dots, k\}$ . Let  $\mathbf{A}_1, \dots, \mathbf{A}_k$  be arbitrary discrete random variables with

probability distributions  $q_1, \dots, q_k$  over alphabets  $\mathcal{A}_1, \dots, \mathcal{A}_k$  respectively. Let  $\mathbf{A}$  be a random variable that equals the random variable  $\mathbf{A}_i$  with probability  $q(i)$ . Then the following Lemma describing an elementary property of the entropy function  $H(\cdot)$  will be useful in the upcoming section.

*Lemma 6.1:* The entropies of  $\mathbf{A}, \mathbf{A}_1, \dots, \mathbf{A}_k$  and  $q$  satisfy  $H(\mathbf{A}) \leq \sum_{i=1}^k q(i)H(\mathbf{A}_i) + H(q)$ , with equality if and only if for each  $i, i'$  for which both  $q(i)$  and  $q(i')$  are positive it holds that  $\Pr_{q_i, q_{i'}}[\mathbf{A}_i = \mathbf{A}_{i'}] = 0$ .

*Proof:* Let  $Y$  denote the random variable having the distribution  $q$  over  $\{1, 2, \dots, k\}$ , so that  $\mathbf{A} = \mathbf{A}_Y$ . Then

$$\begin{aligned} H(\mathbf{A}) &\leq H(\mathbf{A}, Y) \\ &= H(\mathbf{A}|Y) + H(Y) \\ &= \sum_{i=1}^k q(i)H(\mathbf{A}|Y=i) + H(q) \\ &= \sum_{i=1}^k q(i)H(\mathbf{A}_i) + H(q) \end{aligned}$$

■

For any  $\varepsilon > 0$ , let  $R = 1 - 2p + \delta + \varepsilon$ . We recall Calvin's attack. Calvin first *passively* waits until Alice transmits  $\ell = (R - \varepsilon)n$  characters over the channel. Let  $\mathbf{x}^\ell = x_1, \dots, x_\ell$ . He then corrupts the next  $\delta n$  characters uniformly at random. Namely, let  $\mathbf{e} = e_1, \dots, e_{\delta n}$  be a uniformly chosen element in  $[Q]^{\delta n}$ . The first  $\ell + \delta n$  characters of the transmission that Bob receives is  $\mathbf{x}^{\ell + \delta n} = (\mathbf{x}^\ell, \mathbf{e}) \in [Q]^{\ell + \delta n}$ . Notice that the  $\delta n$  symbols corrupted by Calvin in  $\mathbf{x}^{\ell + \delta n}$  are uniformly distributed and thus convey no information to Bob. This fact will be used later in our proof.

After listening and corrupting randomly, Calvin considers the set of codewords  $\mathbf{x}(u, r)$  *consistent* with the observed  $\mathbf{x}^\ell$ . Here and throughout this section, we denote codewords by their corresponding message  $u$  and index  $r$  in  $\mathcal{X}(u)$ . As it may be that  $\mathbf{x}(u, r)$  is exactly the same codeword as  $\mathbf{x}(u', r')$ , the sets in the definitions to follow and in this section are *multisets*. Namely, Calvin constructs the set  $\mathcal{X}|_{\mathbf{x}^\ell} = \{\mathbf{x}(u, r) = x_1, \dots, x_n \in \mathcal{X} \mid x_1, \dots, x_\ell = \mathbf{x}^\ell\}$ . Let  $p(\mathbf{x}^\ell) = p(\mathcal{X}|_{\mathbf{x}^\ell})$  be the probability, under the probability distribution  $p$ , corresponding to the event that Calvin observes  $\mathbf{x}^\ell$  in the first  $\ell$  transmissions. Let  $p_{U|_{\mathbf{x}^\ell}}$  and  $p_{X(u)|_{\mathbf{x}^\ell}}$  be the probability distributions  $p_U$  and  $p_{X(u)}$  also respectively conditioned on the same event. Calvin then chooses an element  $\mathbf{x}'(u', r') \in \mathcal{X}|_{\mathbf{x}^\ell}$  with probability  $p_{U|_{\mathbf{x}^\ell}}(u')p_{X(u')|_{\mathbf{x}^\ell}}(\mathbf{x}'(u', r'))$ . Now, to complete the definition of Calvin's actions, he randomly picks either the first half or the second half of the remaining  $(1 - (R + \delta - \varepsilon))n = 2(p - \delta)n$  symbols of  $\mathbf{x}$  and replaces it by the corresponding symbols of  $\mathbf{x}'$ . If (as we will show) the message  $u$  corresponding to  $\mathbf{x}$  differs from  $u'$  corresponding to  $\mathbf{x}'$ , Bob has no way of determining whether Alice transmitted  $u$  or  $u'$ . Thus as we prove below, Bob will decode incorrectly with some constant probability that depends only on  $\varepsilon$ .

We now turn to prove that Calvin's strategy indeed implies a significant decoding error. Namely, we

show that with significant probability the codeword  $\mathbf{x}'(u', r')$  chosen by Calvin has the following two properties: (a) It's corresponding message differs from that corresponding to  $\mathbf{x}(u, r)$  (i.e.,  $u \neq u'$ ), (b) Given  $\mathbf{y}$ , Bob is unable to distinguish whether  $\mathbf{x}(u, r)$  or  $\mathbf{x}'(u', r')$  was transmitted.

In what follows  $H(\cdot)$  is the  $Q$ -ary entropy function

*Claim 6.1:* With probability at least  $\varepsilon/2$  (over the randomness of Alice),  $H(p_{U|\mathbf{x}^\ell}) \geq \varepsilon n/2$ .

*Proof:* Let  $q$  be the probability distribution over  $[Q]^\ell$  for which  $q(\mathbf{x}^\ell) = p(\mathbf{x}^\ell)$  for all possible  $\mathbf{x}^\ell \in [Q]^\ell$ . Now using Lemma 6.1 we obtain

$$H(p_U) \leq \sum_{\mathbf{x}^\ell} q(\mathbf{x}^\ell) H(p_{U|\mathbf{x}^\ell}) + H(q). \quad (5)$$

By our definitions  $H(p_U) = Rn$ . Moreover,  $H(q) \leq \ell = (R - \varepsilon)n$  (since  $q$  is defined over an alphabet of size  $|Q|^\ell$ ). Thus (5) becomes

$$\sum_{\mathbf{x}^\ell} q(\mathbf{x}^\ell) H(p_{U|\mathbf{x}^\ell}) \geq Rn - (R - \varepsilon)n = \varepsilon n.$$

As the average of  $H(p_{U|\mathbf{x}^\ell})$  is at least  $\varepsilon n$ , then  $H(p_{U|\mathbf{x}^\ell}) \geq \varepsilon n/2$  with probability at least  $\varepsilon/2$  (by Markov's inequality, here we use the fact that  $H(p_{U|\mathbf{x}^\ell}) \leq n$ ).  $\blacksquare$

We now assume that Alice has transmitted a codeword  $\mathbf{x}$  such that  $\mathbf{x}^\ell$  satisfies Claim 6.1 above. Under this assumption we show that the message  $u$  of Alice and  $u'$  of Calvin will be distinct with high probability.

*Claim 6.2:* Let  $n$  be sufficiently large. Assuming Claim 6.1, with probability at least  $\varepsilon/4$  it holds that the pair  $(u', r')$  chosen by Calvin satisfies  $u \neq u'$ .

*Proof:* Let  $\mathbf{x}^\ell$  be the first  $\ell$  characters of the message transmitted by Alice. Assume that  $\mathbf{x}^\ell$  satisfies Claim 6.1. Let  $p_{U|\mathbf{x}^\ell}$  be the probability distribution  $p_U$  conditioned on the set  $\mathcal{X}|\mathbf{x}^\ell$ . For a message  $u$  let  $p_{U|\mathbf{x}^\ell}(u)$  be its conditional probability. Alice and Calvin are choosing messages independently at random according to  $p_{U|\mathbf{x}^\ell}$ . Let  $U_A$  and  $U_C$  represent the random variables governing the choice of Alice and Calvin respectively.  $U_A$  and  $U_C$  are identically distributed and independent. Thus, the probability under study is

$$P = \Pr[U_A \neq U_C] = \sum_{u \neq u'} p_{U|\mathbf{x}^\ell}(u) p_{U|\mathbf{x}^\ell}(u')$$

By Claim 6.1, we know that  $H(p_{U|\mathbf{x}^\ell}) = H(U_A) = H(U_C) \geq \varepsilon n/2$ . Now, using Fano's inequality [3], we have that

$$H(U_A|U_C) \leq h(P) + P \log(|\mathcal{U}| - 1)$$

Here,  $h(\cdot)$  is the binary entropy function. As  $H(U_A|U_C) = H(U_A) \geq \varepsilon n/2$ ,  $h(P) \leq 1$ , and  $\log(|\mathcal{U}| - 1) < \log(|\mathcal{U}|) \leq n$ , we have that  $P \geq \frac{\varepsilon n - 2}{2n} \geq \varepsilon/4$  for sufficiently large  $n \geq 4/\varepsilon$ , which concludes our proof.  $\blacksquare$

We are now ready to analyze the error probability of Calvin's "wait and push" scheme. Consider the event  $\Gamma(u, r, u', r')$  in which Alice has chosen  $(u, r)$  and a corresponding  $\mathbf{x}$ , and Calvin has chosen a pair

$(u', r')$  and a corresponding codeword  $\mathbf{x}'$  from  $\mathcal{X}|_{\mathbf{x}^\ell}$ . Let  $p(\mathbf{x}^\ell)$  be the probability that Alice transmits a codeword whose first  $\ell$  characters are  $\mathbf{x}^\ell$ . The probability of  $\Gamma(u, r, u', r')$  is

$$\begin{aligned} & p_U(u)p_{X(u)}(\mathbf{x})p_{U|\mathbf{x}^\ell}(u')p_{X(u')|\mathbf{x}^\ell}(\mathbf{x}') \\ &= p(\mathbf{x}^\ell)p_{U|\mathbf{x}^\ell}(u)p_{X(u)|\mathbf{x}^\ell}(\mathbf{x})p_{U|\mathbf{x}^\ell}(u')p_{X(u')|\mathbf{x}^\ell}(\mathbf{x}') \\ &= p_U(u')p_{X(u')}(\mathbf{x}')p_{U|\mathbf{x}^\ell}(u)p_{X(u)|\mathbf{x}^\ell}(\mathbf{x}) \end{aligned}$$

Namely, the probability of the event  $\Gamma(u, r, u', r')$  is equal to the probability of the event  $\Gamma(u', r', u, r)$  that Alice has chosen  $(u', r')$  and the corresponding  $\mathbf{x}'$  and Calvin has chosen the pair  $(u, r)$  from  $\mathcal{X}|_{\mathbf{x}^\ell}$  and the corresponding codeword  $\mathbf{x}$ . In what follows, we show that this equivalence (or symmetrization, as commonly stated in the study of AVC's) will imply a large decoding error at the receiver Bob.

Consider the event  $\Gamma(u, r, u', r')$ . Recall the corrupting strategy of Calvin, after viewing  $\mathbf{x}^\ell$  and choosing a codeword  $\mathbf{x}' = x'_1, \dots, x'_n$  from  $\mathcal{X}|_{\mathbf{x}^\ell}$ : Calvin choses  $\mathbf{e} = e_1, \dots, e_{\delta n}$  a uniformly distributed element in  $[Q]^{\delta n}$ . Then with probability half Bob will receive the vector:

$$\mathbf{y} = \mathbf{x}^\ell; \mathbf{e}; x_{\ell+\delta n+1}, \dots, x_{(n+\delta n+\ell)/2}, x'_{1+(n+\delta n+\ell)/2}, \dots, x'_n$$

and with probability half Bob will receive the vector:

$$\mathbf{y} = \mathbf{x}^\ell; \mathbf{e}; x'_{\ell+\delta n+1}, \dots, x'_{(n+\delta n+\ell)/2}, x_{1+(n+\delta n+\ell)/2}, \dots, x_n$$

Let  $\mathbf{Y}(u, r, u', r')$  be the distribution on  $\mathbf{y}$  conditioned on the event  $\Gamma(u, r, u', r')$ . It is now not hard to verify that  $\mathbf{Y}(u', r', u, r)$  is identical to  $\mathbf{Y}(u, r, u', r')$ . Namely, given that Alice chooses  $(u', r')$  and a corresponding  $\mathbf{x}'$ , and Calvin chooses  $(u, r)$  and a corresponding  $\mathbf{x}$  from  $\mathcal{X}|_{(\mathbf{x}')^\ell} = \mathcal{X}|_{\mathbf{x}^\ell}$  then for a uniform  $\mathbf{e}$ , with probability half, Bob will receive the vector

$$\mathbf{y} = \mathbf{x}^\ell; \mathbf{e}; x_{\ell+\delta n+1}, \dots, x_{(n+\delta n+\ell)/2}, x'_{1+(n+\delta n+\ell)/2}, \dots, x'_n$$

and with probability half Bob will receive the vector:

$$\mathbf{y} = \mathbf{x}^\ell; \mathbf{e}; x'_{\ell+\delta n+1}, \dots, x'_{(n+\delta n+\ell)/2}, x_{1+(n+\delta n+\ell)/2}, \dots, x_n$$

Here, we use the fact that  $\mathbf{x}^\ell = \mathbf{x}(u, r)^\ell = \mathbf{x}(u', r')^\ell = (\mathbf{x}')^\ell$ .

Let  $\gamma : [Q]^n \rightarrow \mathcal{U}$  be any (potentially probabilistic) decoding function of Bob. We are now ready to analyze the error probability of Calvin's attack. The error probability is defined by the sum

$$\begin{aligned} err &= \sum_{(u,r)} \sum_{(u',r') \in \mathcal{X}|_{\mathbf{x}(u,r)^\ell}} \Pr[\Gamma(u, r, u', r')] \cdot \Pr_{\mathbf{y} \in \mathbf{Y}(u,r,u',r'), \gamma} [\gamma(\mathbf{y}) \neq u] \\ &= \sum_{(u,r)} \sum_{(u',r') \in \mathcal{X}|_{\mathbf{x}(u,r)^\ell}} \Pr[\Gamma(u, r, u', r')] \cdot \frac{1}{2} \left( \Pr_{\mathbf{y} \in \mathbf{Y}(u,r,u',r'), \gamma} [\gamma(\mathbf{y}) \neq u] + \Pr_{\mathbf{y} \in \mathbf{Y}(u,r,u',r'), \gamma} [\gamma(\mathbf{y}) \neq u'] \right) \\ &\geq \frac{1}{2} \sum_{(u,r)} \sum_{(u',r') \in \mathcal{X}|_{\mathbf{x}(u,r)^\ell}} \Pr[\Gamma(u, r, u', r')] \cdot \mathbb{I}_{u \neq u'} \end{aligned}$$

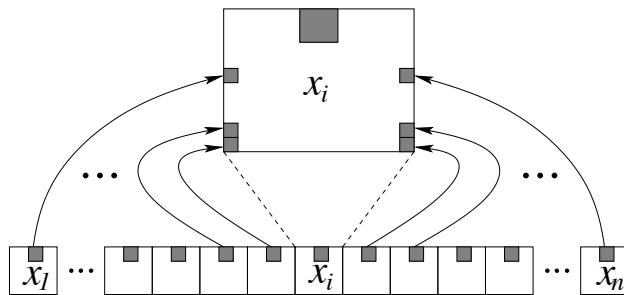


Fig. 2. The authentication scheme for the code for Theorem 3. The character  $x_i$  is schematically “blown up” to show that it includes *pair-wise* authentication information with each and every character  $x_j$  in the codeword.

Here,  $\mathbb{I}_{u \neq u'}$  is 1 if  $u \neq u'$  and 0 otherwise; also, we use the equivalence between  $\Gamma(u, r, u', r')$ ,  $\mathbf{Y}(u, r, u', r')$  and  $\Gamma(u', r', u, r)$ ,  $\mathbf{Y}(u', r', u, r)$  (respectively), and the fact that  $\mathbf{x}^\ell = \mathbf{x}(u, r)^\ell = \mathbf{x}(u', r')^\ell = (\mathbf{x}')^\ell$  and thus  $(u', r') \in \mathcal{X}|_{\mathbf{x}(u, r)^\ell}$  iff  $(u, r) \in \mathcal{X}|_{\mathbf{x}(u', r')^\ell}$ . Finally, by Claims 6.1 and 6.2 we have that  $err \geq \varepsilon^2/16$  which concludes the proof of the upper bound of Theorem 3 (case 2).

Our proof for the third and last case in which  $p \geq 1/2$  follows (a simplified version of) that presented above when  $\ell = 0$ . Namely, we assume  $R = \varepsilon$  (implying  $\ell = (R - \varepsilon)n = 0$ ). Calvin starts by considering the entire message set  $\mathcal{U}$  and picks an element  $\mathbf{x}'(u', r') \in \mathcal{X}$  with probability  $p_U(u')p_{X(u')}(\mathbf{x}'(u', r'))$ . Then, Calvin randomly picks either the first half or the second half of the transmitted  $\mathbf{x}$  and replaces it by the corresponding symbols of  $\mathbf{x}'$ . Following the proof above, Bob has no way of determining whether Alice transmitted  $u$  or  $u'$ . Thus, Bob will decode incorrectly with constant probability that depends only on  $\varepsilon$ .

**Proof of Lower bound:** We now prove the achievability of the rate  $C_\delta^{\text{ow}}(p)$  specified in Theorem 3 with a computationally tractable code. The scheme we present covers all positive rates in the rate-region specified in Theorem 3. Our scheme follows roughly the ideas that appear in the scheme of Section V. Namely, Alice will use a uniform encoding scheme comprising of an erasure code along with a hash function used for authentication. However, in this context, an `overwrite` adversary is more powerful than an `additive` adversary, because an `overwriting` adversary can substitute any symbol  $x_i$  by a new symbol  $y_i$  that can pass the authentication scheme used by Bob in Section V. We thus propose a more elaborate authentication scheme in which each symbol  $x_i$  contains information that allows for *pairwise* authentication with every other symbol  $x_j$ . Our scheme is illustrated in Fig. 2.

**Alice:** Using notation similar to that of Section V, let  $u$  be the message Alice would like to transmit to Bob, and  $\mathbf{v} = v_1, \dots, v_n$  be the encoding of  $u$  via an efficiently encodable and decodable linear erasure code (here, as before, we use Reed-Solomon codes). Let  $Q$  be sufficiently large (to be specified explicitly later in the proof). Let  $m = n^4 + 2n^3$  (note that this is significantly larger than in Theorem 2). As mentioned in Section IV, Alice treats each symbol of a codeword  $\mathbf{x} = x_1, \dots, x_n$  as a packet, by breaking each  $x_i$  into  $m$  sub-symbols  $x_{i,1}$  through  $x_{i,m}$  from  $\mathbb{F}_q$ . She partitions  $x_{i,1}$  through  $x_{i,m}$  into three consecutive sequences

of sub-symbols of sizes  $n^4$ ,  $n^3$  and  $n^3$  respectively. The sub-symbols  $x_{i,1}$  through  $x_{i,n^4}$  are denoted by the set  $w_i$ , and correspond to the sub-symbols of  $v_i$ , the  $i$ th symbol of the erasure-codeword  $\mathbf{v}$  generated by Alice. The next  $n^3$  sub-symbols are arranged into  $n$  sets of  $n^2$  sub-symbols each, denoted by the sets  $r_{ij}$  for each  $j \in [n]$ , and consist of Alice's secret for packet  $i$ . That is, each  $r_{ij}$  consists of  $n^2$  sub-symbols chosen independently and uniformly at random from  $\mathbb{F}_q$ . For each  $i$  and  $j$ ,  $r_{ij}$  is chosen independently. The final  $n^3$  sub-symbols are arranged into  $n$  sets of  $n^2$  sub-symbols each, denoted by the sets  $\sigma_{ij}$  for each  $j \in [n]$ , and consist of the pairwise hashes of the symbols  $x_i$  and  $x_j$ . We define  $\sigma_{ij}$  to be  $H_{r_{ij}}(w_j)$ , where  $H_{r_{ij}}$  is taken from (a slight variation to) a *pairwise independent* family  $\mathcal{H} = \{H_r\}_{r \in \mathbb{F}_q}$  (known in advance to all parties). Namely,  $\sigma_{ij}$  is the hash of the information from  $x_j$  using a key from the transmitted symbol  $x_i$ . All in all, each transmitted symbol  $x_i$  of Alice consists of the tuple  $(w_i, (r_{ij})_{j \in [n]}, (H_{r_{ij}}(w_j))_{j \in [n]})$ .

We now explicitly demonstrate the construction of each  $w_i$  from Alice's message  $u$ . Alice chooses  $R = (1 - (2n^3)/m)C$ , where  $C$  is an abbreviation of the capacity  $C_\delta^{\text{ow}}(p)$  specified in Theorem 3. Note that  $R$  equals  $C$  asymptotically in  $n$ . The message  $u$  she wishes to transmit to Bob has  $Cn$  symbols over  $\mathbb{F}_{q^{n^4}}$ . A  $(n, Cn)$  erasure code is used to encode these symbols into  $n$  symbols  $v_1, v_2, \dots, v_n \in \mathbb{F}_{q^{n^4}}$ . This erasure code is resilient to  $(1 - C)n$  erasures. Each symbol  $v_i \in \mathbb{F}_{q^{n^4}}$  of the codeword is then treated as a vector  $w_i \in \mathbb{F}_q^{n^4}$  of  $n^4$  sub-symbols over  $\mathbb{F}_q$ .

**Hash function:** We now discuss the property needed for our analysis of the family  $\mathcal{H}$  of hash functions we use. As mentioned above we use a (variation to a) pairwise independent hash family  $\mathcal{H} = \{H_r\}$  with the property that for all  $w'_j \neq w_j$ , the probability over  $r_{ij}$  that  $H_{r_{ij}}(w'_j) = H_{r_{ij}}(w_j)$  is sufficiently small. Such functions are common in the literature (e.g., see [14], [13]). In fact, we use essentially the same hashes as in Theorem 2, except with different inputs and dimension. Namely, let  $W_i$  and  $W'_i$  represent  $w_i$  and  $w'_i$  respectively arranged as  $n^2 \times n^2$  matrices. Let  $\mathbf{r}_{ij}$  be a length- $n^2$  column vector of symbols corresponding to  $r_{ij}$ . We define the hash  $H_{r_{ij}}(w_j)$  as the column vector  $\sigma_{ij} = W_i \mathbf{r}_{ij}$ . Note that  $H_{r_{ij}}(w'_j) = H_{r_{ij}}(w_j)$  implies  $W'_j \mathbf{r}_{ij} = W_j \mathbf{r}_{ij}$ , which implies that  $(W'_j - W_j) \mathbf{r}_{ij} = \mathbf{0}$ . But by assumption  $w'_j \neq w_j$ , so  $W'_j \neq W_j$ , and so  $W'_j - W_j$  is of rank  $\geq 1$ . Thus a random  $r_{ij}$  satisfies  $(W'_j - W_j) \mathbf{r}_{ij} = \mathbf{0}$  with probability  $\leq 1/q$ .

**Bob:** We now define Bob's decoder. Let  $x_i, x_j$  be two symbols transmitted by Alice, and  $y_i, y_j$  be the corresponding symbols received by Bob. Consider the information  $w_i$ , the secret  $r_{ij}$  and the hash value  $\sigma_{ij}$  in  $x_i$ , and let  $w'_i, r'_{ij}$  and  $\sigma'_{ij}$  be the corresponding (potentially corrupted) values in  $y_i$ . Similarly consider the components of  $x_j$  and  $y_j$ . Bob checks for *mutual consistency* between  $y_i$  and  $y_j$ . Namely, the pair  $y_i$  and  $y_j$  are said to be mutually consistent if both  $\sigma'_{ij} = H_{r'_{ij}}(w'_j)$  and  $\sigma'_{ji} = H_{r'_{ji}}(w'_i)$ . Clearly, if both  $y_i$  and  $y_j$  are uncorrupted versions of  $x_i$  and  $x_j$  respectively, they are mutually consistent. By the analysis above of  $H_{r_{ij}}$ , if Calvin does not know the value of  $r_{ij}$ , does not corrupt  $x_i$  but corrupts  $w_j$ , then the probability over  $r_{ij}$  that  $y_i$  and  $y_j$  are consistent is at most  $1/q$ . This is because  $\sigma'_{ij} = \sigma_{ij} = H_{r_{ij}}(w_j)$ ,  $r'_{ij} = r_{ij}$ , and w.h.p.  $H_{r_{ij}}(w_j) \neq H_{r_{ij}}(w'_j)$ .

We conclude:

*Lemma 6.2:* With probability at least  $1 - 1/q$ , the following  $y_i$  and  $y_j$  are mutually inconsistent.

- 1) *Causality:* If  $i > j$ ,  $x_i = y_i$  and  $w'_j \neq w_j$ .
- 2)  $\delta$ -*delay:* If  $|i - j| < \delta n$ , and Calvin corrupts exactly one of the symbols  $x_i$  and  $x_j$  so that either  $w_i \neq w'_i$  or  $w_j \neq w'_j$ .

Bob decodes via the  $\delta$ -Delay Online Overwriting Disruptive Adversary Decoding ( $\delta$ -DOODAD) Algorithm, described in detail below. We first give a high-level overview of the three major steps of  $\delta$ -DOODAD. Bob's first step is to test pairs of received symbols  $(y_i, y_j)$  for mutual consistency. In particular he considers only pairs of symbols separated by at most  $\delta n$  locations; in this event Lemma 6.2, condition (2) implies that Bob detects the corruption of exactly one of a pair of symbols with high probability.

Based on the  $O(\delta n^2)$  tests in the first step, in the second step he enumerates subsets of  $\{y_1, \dots, y_n\}$  of received symbols as ‘‘candidate subsets’’ for decoding via Alice's erasure code. In particular, each of the candidate subsets satisfies the natural property that it contains at least  $(1 - p)n$  mutually consistent  $y_i$ 's. Naively, this enumeration seems computationally intractable since there may be as many as  $\binom{n}{(1-p)n}$  such sets. However, there is also a more intricate combinatorial property (Step 2c in the  $\delta$ -DOODAD algorithm below) that candidate subsets must satisfy; we discuss this property after presenting the details of the algorithm. The effect of Step 2 is to drastically curtail the number of candidate subsets that Bob needs to consider, to at most  $n^{p/\delta}$ , to make it computationally tractable. (In the special case that  $p < \delta$ , it can be directly verified that in fact Bob only needs to consider a single subset of size  $(1 - p)n$ . If  $p > \delta$ , and in particular if  $p/\delta$  is large, the structure of candidate sets seems intrinsically more complex, and an enumeration of the type we propose seems necessary to deal with all possible adversarial actions.)

In the third step, for each of the candidate subsets generated in the previous step, Bob uses the decoder for Alice's erasure code to generate a set of linear equations that the sub-symbols of her message  $u$  must satisfy. Then we claim that any candidate subset with even one corrupted symbol must generate a set of inconsistent linear equations. Hence Bob decodes the unique candidate subset that generates a consistent set of linear equations. As we will see, the error probability of our scheme will be  $n^2/q$ , which is  $o(1)$  if we set  $Q = \exp(\text{poly}(n))$ .

The details of  $\delta$ -DOODAD now follow. We define a *connected component*  $\mathcal{G}_i$  of an undirected graph  $\mathcal{G}$  as a connected subgraph of  $\mathcal{G}$  such that there is no edge in  $\mathcal{G}$  between any vertex in  $\mathcal{G}_i$  and any vertex outside it. Also, let  $\mathcal{L}$  be the linear transform of the Reed-Solomon code that takes the length- $Cn$  column vector  $\mathbf{u}$  of Alice's message  $u$  to the length- $n$  column vector of the erasure codeword  $\mathbf{v}$ . Hence  $\mathcal{L}\mathbf{u} = \mathbf{v}$ . Let the column vector of sub-symbols corresponding to  $\mathbf{v}$  in the transmission Bob receives be denoted  $\mathbf{w}'$ . For any subset  $\mathcal{I} \subseteq [n]$  of size  $Cn$ , let  $\mathcal{L}_{\mathcal{I}}$ ,  $\mathbf{v}_{\mathcal{I}}$  and  $\mathbf{w}'_{\mathcal{I}}$  be respectively defined as the restriction of  $\mathcal{L}$  to the  $i$ th rows/indices of  $\mathcal{L}$ ,  $\mathbf{v}$  and  $\mathbf{w}'$  respectively, for all  $i \in \mathcal{I}$ . Since the code is an MDS code, there is a unique codeword with symbols  $\mathbf{w}'_{\mathcal{I}}$  in  $\mathcal{I}$  [12], and that codeword is given by  $\mathcal{L}(\mathcal{L}_{\mathcal{I}}^{-1}\mathbf{w}'_{\mathcal{I}})$ . Hence for any  $\mathcal{K}$  so that  $\mathcal{I} \subseteq \mathcal{K} \subseteq [n]$ ,  $\mathbf{w}'_{\mathcal{K}}$  is a part of a codeword if and only if  $\mathbf{w}'_{\mathcal{K}} = \mathcal{L}_{\mathcal{K}}(\mathcal{L}_{\mathcal{I}}^{-1}\mathbf{w}'_{\mathcal{I}})$ .

**$\delta$ -Delay Online Overwriting Disruptive Adversary Decoding ( $\delta$ -DOODAD) Algorithm :**

- 1) Bob constructs a  $\delta$ -distance mutual consistency graph  $\mathcal{G}$  with vertex set  $\{y_1, \dots, y_n\}$  and edge-set comprising of all mutually consistent pairs  $(y_i, y_j)$  such that  $|i - j| < n\delta$  (but no other edges). Thus  $\mathcal{G}$  comprises of  $\ell \leq n$  connected components  $\{\mathcal{G}_1, \dots, \mathcal{G}_\ell\}$ .
- 2) Let  $\mathcal{K}$  be a subset of  $[\ell]$ . We define the *candidate subset*  $\mathcal{C}(\mathcal{K})$  of  $\mathcal{G}$  as the set  $\{\mathcal{G}_k | k \in \mathcal{K}\}$  of connected components in  $\mathcal{G}$ . If the size of  $\mathcal{K}$  is  $j$ , we say  $\mathcal{C}(\mathcal{K})$  has size  $j$ . Bob enumerates all possible candidate subsets  $\mathcal{C}(\mathcal{K})$  of  $\mathcal{G}$  such that
  - a) The candidate subset  $\mathcal{C}(\mathcal{K})$  has size at most  $c = p/\delta$ .
  - b) The number of vertices in the subgraphs in  $\mathcal{C}(\mathcal{K})$  is at least  $(1 - p)n$ .
  - c) Each pair of vertices  $y_i$  and  $y_j$  in the union of the subgraphs in  $\mathcal{C}(\mathcal{K})$  are mutually (pairwise) consistent. Notice, that consistent  $y_i$  and  $y_j$  that are *far* apart (for which  $|i - j| \geq n\delta$ ) do not share an edge, and thus may belong to different connected components.
- 3) Let  $\bar{\mathcal{K}} \subseteq [n]$  be the set comprising of indices corresponding to all symbols  $y_i$  in the components  $\mathcal{C}(\mathcal{K})$ . Bob picks an arbitrary subset  $\mathcal{I} \subset \bar{\mathcal{K}}$  of size  $Cn$ . If  $\mathcal{L}_{\bar{\mathcal{K}}} \left( (\mathcal{L}_{\mathcal{I}})^{-1} \mathbf{w}'_{\mathcal{I}} \right) = \mathbf{w}'_{\bar{\mathcal{K}}}$ , he decodes  $u$  as the sub-symbols in the vector  $\mathcal{L}_{\mathcal{I}}^{-1} \mathbf{w}'_{\mathcal{I}}$ . Otherwise he discards  $\mathcal{K}$  and returns to the beginning of Step 3 with a new  $\mathcal{K}$ .

*Claim:* The  $\delta$ -DOODAD algorithm decodes Alice's message correctly with probability at least  $1 - n^2/q$ .

*Proof:* Throughout we assume that Lemma 6.2 holds for all corresponding  $y_i$  and  $y_j$  (by the union bound this happens with probability at least  $1 - n^2/q$ ). Thus, corrupted  $y_i$  and uncorrupted  $y_j$  are non-adjacent in  $\mathcal{G}$ . This implies that any connected component  $\mathcal{G}_i$  in  $\mathcal{G}$  contains vertices which are either all corrupted, or all uncorrupted.

**Existence:** We first prove that at least one  $\mathcal{C}(\mathcal{K})$  with only uncorrupted symbols satisfies Steps 2 and 3. We examine the three conditions of Step 2. By the definition of mutual consistency, any set with only uncorrupted symbols satisfies Step 2c. Since Calvin can corrupt at most  $pn$  symbols, there must be some  $\mathcal{C}(\mathcal{K})$  satisfying Step 2b (i.e., take all the components  $\mathcal{G}_i$  that include uncorrupted  $y_i$ ). To prove that this  $\mathcal{C}(\mathcal{K})$  also satisfies Step 2a, we observe the following. If Calvin does not corrupt at least  $\delta n$  consecutive symbols between two uncorrupted symbols  $y_i$  and  $y_j$  (say  $i < j$ ), there must be a sequence of at most  $j - i + 1$  uncorrupted symbols with indices  $i = k_0 \leq k_1 \leq k_2 \leq \dots \leq k_{j-i} = j$  such that any two consecutive symbols in the sequence have indices that differ by less than  $\delta n$ . Then by the definition of  $\mathcal{G}$ , both  $y_i$  and  $y_j$  must be in the same connected component of  $\mathcal{G}$ . But there are at most  $pn$  corrupted symbols, hence there are at most  $c = p/\delta$  disjoint sequences of  $\delta n$  consecutive corrupted symbols (and thus at most  $c$  components in  $\mathcal{C}(\mathcal{K})$ ).

Lastly, we show that any  $\mathcal{C}(\mathcal{K})$  with only uncorrupted symbols and satisfying Step 2 must also satisfy Step 3. To see this, note that any such  $\mathcal{C}(\mathcal{K})$  has at least  $(1 - p)n \geq Cn$  uncorrupted symbols from  $\mathbb{F}_Q$ . Also, since  $\mathcal{C}(\mathcal{K})$  comprises solely of uncorrupted symbols,  $\mathbf{w}'_{\bar{\mathcal{K}}} = \mathbf{v}_{\bar{\mathcal{K}}}$ , hence for any  $\mathcal{I}$ ,  $\mathbf{w}'_{\mathcal{I}} = \mathbf{v}_{\mathcal{I}}$ . But

by the properties of erasure codes,  $\mathcal{L}_{\mathcal{I}}^{-1} \mathbf{v}_{\mathcal{I}} = \mathbf{u}$ , Alice's message vector. Thus  $\mathcal{L}_{\bar{\mathcal{K}}} \left( \mathcal{L}_{\mathcal{I}}^{-1} \mathbf{w}'_{\mathcal{I}} \right) = \mathcal{L}_{\bar{\mathcal{K}}} \mathbf{u} = \mathbf{v}_{\bar{\mathcal{K}}} = \mathbf{w}'_{\bar{\mathcal{K}}}$ .

**Uniqueness:** We now show that there does not exist any  $\mathcal{C}(\mathcal{K}')$  such that the corresponding output of the  $\delta$ -DOODAD algorithm  $u(\mathcal{C}(\mathcal{K}'))$  differs from Alice's real message  $u$ . We prove this by contradiction. Suppose a  $\mathcal{C}(\mathcal{K}')$  passes all the decoding steps of the  $\delta$ -DOODAD algorithm and results in a  $u(\mathcal{C}(\mathcal{K}'))$  distinct from Alice's message  $u$ . We now make a series of observations that successively refine the structure of such a  $\mathcal{C}(\mathcal{K}')$ , resulting in the conclusion that, w.h.p.,  $\mathcal{C}(\mathcal{K}')$  contains no uncorrupted symbols, and therefore  $u(\mathcal{C}(\mathcal{K}')) = u$ .

First, note that  $\mathcal{C}(\mathcal{K}')$  must contain uncorrupted symbols to pass Step 2b, since  $p < 1/2$ . In addition, to pass Step 2c, by Lemma 6.2 condition (1), all the uncorrupted symbols of  $\mathcal{C}(\mathcal{K}')$  must come before all the symbols corrupted by Calvin. Let  $\mathcal{R}$  be the set of consecutive symbols separating the uncorrupted and corrupted symbols in  $\mathcal{C}(\mathcal{K}')$ . Notice that  $\mathcal{R}$  is at least of size  $n\delta$ , if not, Lemma 6.2 (2) would imply that w.h.p.  $\mathcal{C}(\mathcal{K}')$  does not satisfy Step 2c of  $\delta$ -DOODAD. Moreover, the first  $\delta n$  consecutive symbols of  $\mathcal{R}$  must be corrupted by Calvin. Otherwise, the first uncorrupted symbol in  $\mathcal{R}$  will belong to one of the connected components in  $\mathcal{C}(\mathcal{K}')$ , which contradicts the definition of  $\mathcal{R}$ . Notice that if  $\delta > p$  we may conclude our proof at this point, as in this case Calvin cannot corrupt  $\delta n$  consecutive symbols of  $\mathcal{R}$ .

We now observe that there are at most  $(p - \delta)n$  corrupted symbols in  $\bar{\mathcal{K}}'$ . This follows from the fact that  $\mathcal{R}$  contains  $\delta n$  consecutive symbols corrupted by Calvin (not in  $\bar{\mathcal{K}}'$ ), and the fact that Calvin can corrupt at most  $pn$  symbols. This, together with Step 2b of  $\delta$ -DOODAD, implies that the component set  $\mathcal{C}(\mathcal{K}')$  contains a proper subset  $\mathcal{C}(\mathcal{K}'')$  with at least  $Cn$  uncorrupted symbols (recall that  $C = C_{\delta}^{\text{ow}}(p) = 1 - 2p + \delta$ ).

Finally, let  $\mathcal{I}$  be a subset of  $Cn$  uncorrupted symbols in  $\bar{\mathcal{K}}'$ , and suppose that there is at least one corrupted symbol in  $\mathbf{w}'_{\bar{\mathcal{K}}'}$ . Since our erasure code is an MDS code, there is a unique codeword with the same components as  $\mathbf{w}'_{\mathcal{I}}$  in  $\mathcal{I}$  [12], and that codeword is the codeword encoded at the transmitter. This codeword differs from the received symbols in the corrupted components of  $\mathbf{w}'_{\bar{\mathcal{K}}'}$ . So  $\mathbf{w}'_{\bar{\mathcal{K}}'}$  is not a part of any codeword in our erasure code, and thus it does not satisfy  $\mathcal{L}_{\bar{\mathcal{K}}'} \left( \mathcal{L}_{\mathcal{I}'}^{-1} \mathbf{w}'_{\mathcal{I}'} \right) = \mathbf{w}'_{\bar{\mathcal{K}}'}$  for any  $\mathcal{I}' \subseteq \bar{\mathcal{K}}'$  of size  $Cn$ . Hence such a  $\mathcal{K}'$  will be discarded.  $\square$

## VII. CONCLUSION

In this work we characterize the capacity of online adversarial channels and their variants under the additive and overwrite error models. Our results are tight and coding schemes efficient.

Throughout, we assume that the alphabet size  $Q$  is large compared to the block-length  $n$ . Indeed, the authentication schemes used extensively in this work depend integrally on this assumption – new techniques are needed for “small”, *e.g.* binary, alphabets. Such questions have been addressed partially in [9], [6], which provide upper and lower bounds on the achievable rates of communication.

## REFERENCES

- [1] E. R. Berlekamp. *Algebraic Coding Theory*. McGraw Hill, New York, NY, 1968.
- [2] D. Blackwell, L. Breiman, and A. J. Thomasian. The capacities of certain channel classes under random coding. *The Annals of Mathematical Statistics*, 31(3):558–567, 1960.
- [3] T. M. Cover and J. A. Thomas. *Elements of information theory, 2nd edition*. Wiley-Interscience, New York, NY, USA, 2006.
- [4] R. Cramer, Y. Dodis, S. Fehr, C. Padró, and D. Wichs. Detection of Algebraic Manipulation with Applications to Robust Secret Sharing and Fuzzy Extractors. In *Proc. of Eurocrypt 2008, LNCS 4965/2008*, pages 471–488.
- [5] I. Csiszár and J. Körner. *Information Theory: Coding Theorems for Discrete Memoryless Systems, 2nd edition*. Akademiai Kiado, New York, NY, 1997.
- [6] B. K. Dey, S. Jaggi, M. Langberg, and A. Sarwate. Codes against Delayed Adversaries. In *proceedings of IEEE International Symposium on Information Theory (ISIT) 2010*, 285 - 289.
- [7] W. Feller. An Introduction to Probability Theory and Its Applications, Volume II (2nd ed.). *John Wiley & Sons, New York*, 1972.
- [8] S. Jaggi, M. Langberg, T. Ho, and M. Effros. Correction of Adversarial Errors in Networks. In *proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 1455–1459, 2005.
- [9] M. Langberg, S. Jaggi, and B. K. Dey. Binary Causal-Adversary Channels. *Proceedings of the IEEE International Symposium on Information Theory*, 2009.
- [10] B. K. Dey, S. Jaggi and M. Langberg. Codes against Online Adversaries. *Proceedings of the Forty-Seventh Annual Allerton Conference on Communication, Control, and Computing*, 2009.
- [11] A. Lapidoth and P. Narayan. Reliable Communication under channel uncertainty. *IEEE Transactions on Information Theory*, 44(6), 2148-2177, 1998.
- [12] F.J. MacWilliams and N.J.A. Sloane. The theory of error-correcting codes. *North-Holland, Amsterdam*, 1977.
- [13] M. Mitzenmacher and E. Upfal. *Probability and Computing, Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, Cambridge, UK, 2005.
- [14] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1995.
- [15] L. Nutman and M. Langberg. Adversarial Models and Resilient Schemes for Network Coding. In *proceedings of IEEE International Symposium on Information Theory*, pages 171–175, 2008.
- [16] W. W. Peterson. Encoding and error-correction procedures for Bose-Chaudhuri codes. *IRE Transactions on Information Theory*, IT-60:459–470, 1960.
- [17] P. Moulin S. Ray and M. Médard. On jamming in the wideband regime. In *proceedings of IEEE International Symposium on Information Theory*, 2006.
- [18] A. Sahai and S. Mitter. The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link, Part I: scalar systems. *IEEE Transactions on Information Theory*, 52(8):3369–3395, 2006.
- [19] A. Sarwate. Robust and adaptive communication under uncertain interference. *PhD thesis, Berkeley*, 2008.

## APPENDIX

**List of parameters of our codes:** Table I summarizes a careful analysis of the parameters of the algorithms corresponding to Theorems 1, 2 and 3 (when using Reed-Solomon codes [16], [1] in our constructions). The corresponding values for the scenarios in Theorem 4 are omitted since they are element-wise identical to those in the table. The values in Table I substitute the rate-overhead parameter  $\Delta$  for the packet-size parameter  $m$  used in the proofs of Theorems 2 and 3 since we feel this choice of variables is more “natural” when examining the tradeoffs between code parameters. Also, the algorithms presented in the proofs of Theorems 2 and 3 correspond to a particular setting of the  $\Delta$  parameter; we omitted this degree of freedom in the presentation of the proofs, for ease of exposition. Lastly, we note that in

Section II of this appendix we present an additional hash family to be used in the algorithms presented in the paper. The family we present reduces the overhead  $\Delta$  and improves on the overall performance of our algorithms. Our analysis in Table I addresses both the original hash family used in the body of the paper (referred to as ‘‘M-Hash’’) and the new function family to be presented shortly (referred to as ‘‘P-Hash’’).

## II. AN ALTERNATIVE HASH SCHEME

In this section, we present an alternative family of hash functions  $\{H_r\}$ . For a hash function  $H_r(w)$  in this family,  $w$  is assumed to be a  $t$ -length vector having components  $w_k \in \mathbb{F}_q$ ;  $k = 1, 2, \dots, t$ , and  $r$  is assumed to be a scalar in  $\mathbb{F}_q$ . Let  $P_w(X)$  denote the polynomial  $\sum_{k=1}^t w_k X^{pk+1}$  obtained from  $w$ , where  $p$  denotes the characteristic of  $\mathbb{F}_q$ . Then the hash function  $H_r(w)$  is defined to be the evaluation of the polynomial  $P_w$  at  $r$ , that is,  $H_r(w) = P_w(r)$ .

To satisfy Theorem 2, the hash function needs to satisfy that, for any  $w \in \mathbb{F}_q^t$ ,  $\hat{w} \in \mathbb{F}_q^t \setminus \{0\}$ ,  $\hat{r}, \hat{\sigma} \in \mathbb{F}_q$ , and for a random  $r$  chosen uniformly from  $\mathbb{F}_q$ , the probability  $\Pr_r[P_{w+\hat{w}}(r+\hat{r}) - P_w(r) - \hat{\sigma} = 0]$  is small. Now,

$$\begin{aligned} & P_{w+\hat{w}}(r+\hat{r}) - P_w(r) - \hat{\sigma} \\ &= \sum_{k=1}^t (w_k + \hat{w}_k)(r+\hat{r})^{pk+1} - \sum_{k=1}^t w_k r^{pk+1} - \hat{\sigma}. \end{aligned} \quad (6)$$

We take three cases to show that (6) is a non-zero polynomial of  $r$ .

*Case I:*  $\hat{r} = 0$ : (6) is a non-zero polynomial of  $r$  since  $\hat{w} \neq 0$ .

*Case II:*  $w + \hat{w} = 0$ : Then  $w = -\hat{w} \neq 0$ . So,  $P_{w+\hat{w}} = 0$ , but  $P_w \neq 0$ . Thus (6) is a non-zero polynomial of  $r$ .

*Case III:*  $\hat{r} \neq 0, w + \hat{w} \neq 0$ : Let the degree of  $P_{w+\hat{w}}$  be  $pl + 1$  ( $l \geq 1$ ). Then, the coefficient of  $r^{pl}$  in (6) is the same as the coefficient of  $r^{pl}$  in  $P_{w+\hat{w}}(r+\hat{r})$ . This coefficient is equal to  $\binom{pl+1}{1}(w_l + \hat{w}_l)\hat{r} = (pl+1)(w_l + \hat{w}_l)\hat{r} = (w_l + \hat{w}_l)\hat{r} \neq 0$ . So (6) is a non-zero polynomial of  $r$ .

Since the degree of this non-zero polynomial is at most  $pt + 1$ , the probability (over  $r$ ) that (6) is 0 is at most  $(pt + 1)/q$ .

This hash function can be used to construct a code for proving Theorem 2 as following. Each symbol  $x_i$  of a codeword  $\mathbf{x} = x_1, \dots, x_n$  consists of  $m = n + 2$  sub-symbols  $x_{i,1}$  through  $x_{i,m}$  from  $\mathbb{F}$ . Alice partitions  $x_{i,1}$  through  $x_{i,m}$  into three parts: the vector  $w_i = (x_{i,1}, \dots, x_{i,n})$  and the symbols  $r_i = x_{i,n+1}$  and  $\sigma_i = x_{i,n+2}$ . The sub-symbols  $w_1, w_2, \dots, w_n$  are obtained as a codeword of an erasure code. The sub-symbols  $r_i = x_{i,n+1}$  are generated randomly by Alice, and the sub-symbols  $\sigma_i = x_{i,n+2}$  are generated as  $\sigma_i = H_{r_i}(w_i)$ .

For proving Theorem 3, the hash family needs to satisfy for any  $w \neq w'$  that  $\Pr_r[P_w(r) \neq P_{w'}(r)]$  is small. This follows directly by the fact that  $P_w$  has bounded degree. Now, the hash family can be

used in Theorem 3 as follows. Each symbol  $x_i$  of a codeword  $\mathbf{x} = x_1, \dots, x_n$  consists of  $m = n^2 + 2n$  sub-symbols  $x_{i,1}$  through  $x_{i,m}$  from  $\mathbb{F}$ . Each symbol is partitioned into the following parts. The first  $n^2$  symbols are treated as a vector  $w_i$ . These sub-symbols  $w_1, w_2, \dots, w_n$  are obtained as a codeword of an erasure code. The next  $n$  symbols are generated randomly and independently by Alice and are denoted by  $r_{ij}$  for  $j = 1, \dots, n$ . The last  $n$  symbols are denoted by  $\sigma_{ij}$  for  $j = 1, \dots, n$ , and are generated as  $\sigma_{ij} = H_{r_{ij}}(w_j)$ .

		Capacity		Minimum $Q$	$\Delta$	Complexity	Probability of Error
Theorem 1		$1 - 2p$		$Q > n$	0	$\mathcal{O}(n^2 \log n \log^3 Q)$	0
Theorem 2		$1 - p$	M-hash	$n^{\Omega(n^2)}$	$2/(n+2)$	$\mathcal{O}(n^2 \log n \log^3 Q)$	$\mathcal{O}(nQ^{-1/n^2})$
			P-hash	$n^{\Omega(n)}$	$2/(n+2)$	$\mathcal{O}(n^2 \log n \log^3 Q)$	$\mathcal{O}(n^2 Q^{-1/n})$
Theorem 3	$\delta < p < 0.5$	$1 - 2p + \delta$	M-hash	$n^{\Omega(n^4)}$	$2/(n+2)$	$\mathcal{O}(n^{p/\delta+2} \log n \log^3 Q)$	$\mathcal{O}(n^2 Q^{-1/n^4})$
			P-hash	$n^{\Omega(n^2)}$	$2/(n+2)$	$\mathcal{O}(n^{p/\delta+2} \log n \log^3 Q)$	$\mathcal{O}(n^4 Q^{-1/n^2})$
	$p \leq \delta, p < 0.5$	$1 - p$	M-hash	$n^{\Omega(n^4)}$	$2/(n+2)$	$\mathcal{O}(n^2 \log n \log^3 Q)$	$\mathcal{O}(n^2 Q^{-1/n^4})$
			P-hash	$n^{\Omega(n^2)}$	$2/(n+2)$	$\mathcal{O}(n^2 \log n \log^3 Q)$	$\mathcal{O}(n^4 Q^{-1/n^2})$

TABLE I

LIST OF PARAMETERS FOR OUR CODES. HERE M-HASH AND P-HASH INDICATE RESPECTIVELY THE MATRIX HASH FUNCTION AND THE POLYNOMIAL HASH FUNCTION