Codes against Online Adversaries

B. K. Dey* S. Jaggi† M. Langberg‡

November 18, 2008

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 $\mathbf{x} = (x_1, \ldots, x_n)$ 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, the decisions of the jammer must be made in an *online* or *causal* manner. Namely, 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 $d \in (0, 1)$, we study the scenario in which the jammer's decision on the corruption of x_i must depend solely on x_j for $j \leq i - dn$.

In this work, we initiate the study of codes for online adversaries, and present a *tight* characterization of the amount of information one can transmit in both the 0-delay and, more generally, the d-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 d we show that the achievable rate can be significantly greater than that of the classical model.

We prove tight results for both *additive* and *overwrite* jammers when the transmitted symbols are assumed to be over a sufficiently large field $\mathbb{F}$. In the additive case the jammer may corrupt information $x_i \in \mathbb{F}$ by adding onto it a corresponding error $e_i \in \mathbb{F}$. In this case the receiver gets the symbol $y_i = x_i + e_i$. In the overwrite case, the jammer may corrupt information $x_i \in \mathbb{F}$ by replacing it with a corresponding corrupted symbol $y_i \in \mathbb{F}$. For positive delay d, symbol x_i may not be known to the adversarial jammer at the time it is being corrupted, hence these two error models, and the corresponding achievable rates, are shown to differ substantially.

Finally, we extend our results to a *jam-or-listen* online model, where the online adversary can *either* jam a symbol *or* eavesdrop on it. This corresponds to several scenarios that arise in practice. We again provide a tight characterization of the achievable rate for several variants of this model.

The rate-regions we prove for each model are informational-theoretic in nature and hold for computationally unbounded adversaries. The rate regions are characterized by "simple" piecewise linear functions of p and d. The codes we construct to attain the optimal rate for each scenario are computationally efficient.

*Department of Electrical Engineering, Indian Institute of Technology Bombay, Mumbai, India, 400 076, email: bikash@ee.iitb.ac.in

†Department of Information Engineering, Chinese University of Hong Kong, Shatin, N.T., Hong Kong, email: jaggi@ie.cuhk.edu.hk

‡Computer Science Division, Open University of Israel, 108 Ravutski St., Raanana 43107, Israel, email: mikel@openu.ac.il

1 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 $\mathbf{x} = x_1, \ldots, x_n$ is considered to be a vector of length n over an alphabet $\mathbb{F}$ 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).

In the classical adversarial channel model, e.g., [6, 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 [10, 1] are known to achieve this optimal rate.

Online adversaries In this work we initiate 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* 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 $\mathbf{x} = x_1, \ldots, x_n$ 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, limited transmit energy, or a need to keep his location secret.

In addition to the online adversaries described above, we also consider the more general scenario in which Calvin's jamming decisions are delayed. That is, for a delay parameter $d \in (0,1)$, Calvin's decision on the corruption of x_i must depend solely on x_j for $j \leq i - dn$. We refer to such adversaries as *d-delay online* adversaries. Such d-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., $d = 0$) and the d-delay model for $d > 0$ 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[1]. These two attacks are significantly 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 d scenario.

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.

[1] 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.

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. 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/d-delay, and additive/overwrite errors.

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 $n \to \infty$). 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 $C^{\mathtt{omni}}(p)$. We denote the capacity of the d-delay online adversarial channels which can change pn characters by $C_d^{\mathtt{add}}(p)$ for the `additive` error model, and $C_d^{\mathtt{ow}}(p)$ for the `overwrite` error model. For the `jam-or-listen` adversary, we denote the corresponding capacities by $C_d^{\mathtt{jl,add}}(p)$ or $C_d^{\mathtt{jl,ow}}(p)$, depending on whether Calvin uses additive or overwrite errors. A more detailed discussion of our definitions and notation is given in Section 2.

Our results In this work, we initiate the study of codes for online adversaries, and present a *tight* characterization of the amount of information one can transmit in both the 0-delay and, more generally, the d-delay online setting. To the best of our knowledge, communication in the presence of an online adversary (with or without delay) has not been explicitly addressed in the literature. 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 Korner [4] (in the section addressing Arbitrary Varying Channels [2]). Various variants of causal adversaries have been addressed in the past, for instance [2, 5, 11, 12, 9] – however the models considered therein differ significantly from ours.

At a high level, we show that for 0-delay adversaries 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 d, that the achievable rate can be significantly greater than those achievable against omniscient adversaries.

We stress that our results are information-theoretic in nature and thus hold even if the adversary is computationally unbounded. The codes we construct to achieve the optimal rates are computationally efficient to design, and for Alice and Bob to implement (i.e., efficiently encodable and decodable). All our results assume that the field size q is significantly larger than n. In some cases it suffices to take $q = \mathrm{poly}(n)$, but in others we need $q = \exp(\mathrm{poly}(n))$. Both settings lend themselves naturally to real-world scenarios, as in both cases a field element x_i can be represented by a polynomial (in n) number of bits.

The exact statements of our results are in Theorems 1, 2, 3 and 4 below. The technical parameters (including rate, field size, error probability, and time complexity) of our results are summarized in Table 1 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` *model. In particular,*

$$C_0^{\mathtt{ow}}(p) = C_0^{\mathtt{add}}(p) = C^{\mathtt{omni}}(p) = (1-2p)^+ = \begin{cases} 1-2p, & p \in [0, 0.5) \\ 0, & p \in [0.5, 1] \end{cases} . \tag{1}$$

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

Next we characterize the capacity of the d-delay online channel under the additive error model.

Theorem 2 (d delay with `additive` error model) *For any $p \in [0,1]$ the capacity $C_d^{\mathtt{add}}(p)$ of the d-delay online channel for $d > 0$ 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 d-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 d-delay adversary.

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

$$C_d^{\mathtt{ow}}(p) = \begin{cases} 1-p, & p \in [0, 0.5), p < d \\ 1-2p+d, & p \in [0, 0.5), p > d \\ 0, & p \in [0.5, 1] \end{cases}. \tag{2}$$

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 d-delay, and additive or overwrite attacks.

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

$$C_d^{\mathtt{jl,add}}(p) = C_d^{\mathtt{add}}(p),\ C_d^{\mathtt{jl,ow}}(p) = C_d^{\mathtt{ow}}(p). \tag{3}$$

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

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 [10, 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 is fairly straightforward, and allows Calvin to enforce a probability of error of size at least $1/4$ 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 $d > 0$, the capacity of the d-delay online channel under the additive error model is $1-p$. Note that this expression is independent of d. 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 d-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 corrupted 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 in 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 d-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 d-delay `overwrite` model. Namely, as before, Alice's encoding scheme comprises of an erasure code along with a hash function 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 is a valid output of the hash function. Hence the design of the hash function for Theorem 3 is more intricate than the corresponding construction in Theorem 2.

Roughly speaking, in the scheme we propose for the d-delay `overwrite` scenario, the redundancy 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 σ_{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 $|i-j| < dn$ 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* 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. We view Bob's efficient decoding algorithm as the main technical contribution 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.

Outline The rest of the paper is organized as follows. In Section 2 we present a detailed description of our adversarial models together with some notation to be used throughout our work. In Section 3 we present the proof of Theorem 2. In Section 4 we present the main technical contribution of this work, the proof of Theorem 3. Theorem 1, although stated first in the Introduction, follows rather easily from the proof of Theorem 3 and is thus presented in Section B of the Appendix. Theorem 4 follows directly from Theorems 1, 2, and 3, and is thus presented in Section C of the Appendix. Some remarks and open problems are finally given in Section 5. The technical parameters of our results are summarized in Table 1 of the Appendix.

2 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. Throughout this work we assume that the field size q is exponential in poly(n) (although some of our results will only need a polynomial in n sized q) and that our parameters p and d are constant. For any integer i let $[i]$ denote the set $\{1, \ldots, i\}$. Let $R \geq 0$ be Alice's *rate*. An $[n, nR]_q$*-code* is defined by Alice's encoder and Bob's corresponding decoder, as defined below.

Alice: Alice's message u is assumed to be an element of $[q^{nR}]$. In our schemes, Alice will also hold a uniformly distributed *secret* r which is assumed to be a number of elements (say ℓ) of $[q]$. Alice's secret is assumed to be unknown to *both* Bob and Calvin prior to transmission. Alice's *encoder* is a deterministic function mapping every (w, r) in $[q^{nR}] \times [q]^\ell$ to a vector $\mathbf{x} = (x_1, \ldots x_n)$ in $\mathbb{F}^n$.

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/d-delay:* Calvin's knowledge set K_i is a subset of $[i - dn]$. *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. 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 for at most pn values in $[i]$.

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

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 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. Note that these definitions imply that a successful decoding scheme allows a *worst case* promise. Namely, it implies high success probability no matter which message u was chosen by Alice.

The rate R is said to be *achievable* if for every $\varepsilon > 0$, $\delta > 0$ 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 ε. The supremum of the achievable rates is called the *capacity* and is denoted by C. We denote the capacity of the d-delay online adversarial channels under the `additive` error model by $C_d^{\mathtt{add}}(p)$ and under the `overwrite` error model by $C_d^{\mathtt{ow}}(p)$. For a `jam-or-listen` adversary we denote the corresponding capacities by $C_d^{\mathtt{jl,add}}(p)$ and $C_d^{\mathtt{jl,add}}(p)$.

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.

Remark 2.1 *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 are 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* $\mathbf{x}$.

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 $\mathbf{x} = x_1, \ldots, x_n$, 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 $\mathbb{F}_{q'}$.

3 Proof of Theorem 2

We consider block length n large enough so that $d > 1/n$. Throughout, to simplify our presentation, we assume that expressions such as pn or dn are integers. We first prove that $1-p$ is an upper bound on $C_d^{\mathtt{add}}(p)$ by showing a "random-add" strategy for Calvin. Namely, consider an adversary who chooses elements of $\mathbb{F}_q$ uniformly at random and adds them to the first pn symbols in Alice's transmissions. Thus the first pn symbols Bob receives are uniformly distributed random elements of $\mathbb{F}_q$, and carry no information at all. It is not hard to verify that such an adversarial strategy allows communication between Alice and Bob at rate at most $1 - p$. This concludes our discussion for 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. In the first phase, 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 $\mathbf{v}$. In the second phase, 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 (a positive-delay additive adversary) 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 [10, 1] can be used). In what follows we give our code construction in detail.

Let q be sufficiently large (to be specified explicitly later in the proof). Let $m = n^2 + 2n$. As mentioned in Section 2, Alice treats each symbol of a codeword $\mathbf{x} = x_1, \ldots, x_n$ 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 ith 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 σ_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 $mnR = (m-2n)(1-p)n = (1-p)n^3$ 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 $n^2 i - 1$.

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.

We now define our hash family $\mathcal{H}$ and show that with high probability any corrupted symbol $y_i \neq x_i$

will not pass Bob's authentication check. 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 $y_i = x_i + e_i$. 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 the sets $\hat{w}_i$, $\hat{r}_i$ and $\hat{\sigma}_i$ so to satisfy $w'_i = w_i + \hat{w}_i$, $r'_i = r_i + \hat{r}_i$ and $\sigma'_i = \sigma_i + \hat{\sigma}_i$ (addition is performed by element-wise addition over $\mathbb{F}_{q'}$ of corresponding sub-symbols in each set). 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 $\Pr_{r_i}[H_{r'_i}(w'_i) = \sigma'_i \mid H_{r_i}(w_i) = \sigma_i]$ is sufficiently small. Or equivalently, $\Pr_{r_i}[H_{r_i+\hat{r}_i}(w_i+\hat{w}_i) = \sigma_i + \hat{\sigma}_i \mid H_{r_i}(w_i) = \sigma_i] = \Pr_{r_i}[H_{r_i+\hat{r}_i}(w_i+\hat{w}_i) - H_{r_i}(w_i) = \hat{\sigma}_i]$ is sufficiently small.

To complete our proof we present our hash family $\mathcal{H}$. Recall that w_i consists of n^2 sub-symbols in $\mathbb{F}_{q'}$. Let W_i represent w_i when arranged as a $n \times n$ 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 $\sigma_\mathbf{i}$ 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 $(W_i + \hat{W}_i)(\mathbf{r}_i + \hat{\mathbf{r}}_\mathbf{i}) - (W_i\mathbf{r}_i) = \hat{\sigma}_\mathbf{i}$. Here $\hat{W}_i$ is the matrix representation of $\hat{w}_i$ and $\hat{\mathbf{r}}_\mathbf{i}, \hat{\sigma}_\mathbf{i}$ correspond to $\hat{r}_i, \hat{\sigma}_i$. Namely, the corrupted symbol received by Bob is authenticated only if $\hat{W}_i\mathbf{r}_i = \hat{\sigma}_\mathbf{i} - (W_i + \hat{W}_i)\hat{\mathbf{r}}_\mathbf{i}$.

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 $\hat{W}_i\mathbf{r}_i = \hat{\sigma}_\mathbf{i} - (W_i + \hat{W}_i)\hat{\mathbf{r}}_\mathbf{i}$, the left hand side depends on r_i while the right hand side does not. Hence 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 $(q')^m = \exp(\mathrm{poly}(n))$. ■

4 Proof of Theorem 3

Proof of Upper bound: We start by addressing the three cases in the upper bound on the capacity $C_d^{\mathtt{ow}}(p)$. First, if $p < d$, 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. Second, if $p \geq 1/2$ and the rate $R > 0$ is positive, Calvin picks a codeword $\mathbf{x}'$ uniformly at random from Alice's codebook. With probability at least $1-q^{-Rn}$, Alice's *true* codeword $\mathbf{x}$ is distinct from the codeword $\mathbf{x}'$. Calvin then flips an unbiased coin, and depending on the outcome he corrupts either the first half or the second half of $\mathbf{x}$. This corruption is done by replacing the symbols of $\mathbf{x}$ by the corresponding symbols of $\mathbf{x}'$. If indeed $\mathbf{x} \neq \mathbf{x}'$, Bob has no way of determining whether Alice transmitted $\mathbf{x}$ or $\mathbf{x}'$. Thus, Bob's probability of decoding incorrectly is at least $\frac{1}{2}(1 - q^{-Rn}) \geq \frac{1}{4}$ for large enough q and/or n.

Finally, if $d < p < 1/2$, we present a "wait-and-attack" strategy for Calvin to prove that $1 - 2p + d$ is an upper bound on $C_d^{\mathtt{omni}}(p)$. Suppose not, and that rate $R = 1 - 2p + d + \varepsilon$ is achievable for some $\varepsilon > 0$. Then there are q^{Rn} possible messages in Alice's codebook. Calvin starts by eavesdropping on, but not corrupting, the first $(R - \varepsilon)n$ symbols Alice transmits. He then overwrites the next dn symbols with symbols chosen uniformly at random from $\mathbb{F}_q$. These dn locations convey no information to Bob. At this point (after Alice transmits $(R + d - \varepsilon)n$ symbols), the d-delay Calvin only knows the value of the first $(R-\varepsilon)n$ symbols of $\mathbf{x}$. It can be verified that with probability at least $1-q^{-\varepsilon n/2}$ over Alice's codebook, after Alice's first $(R+d-\varepsilon)n$ transmitted symbols, the set $\mathcal{S}$ of codewords consistent with what Bob and Calvin have observed thus far is of size at least $q^{\varepsilon n/2}$. Calvin then picks a random $\mathbf{x}'$ from $\mathcal{S}$. With probability at least $1 - q^{-\varepsilon n/2}$, $\mathbf{x}'$ is distinct from Alice's $\mathbf{x}$. Calvin then flips an unbiased coin, and depending on the outcome he corrupts either the first half or the second half of the remaining $(1-(R+d-\varepsilon))n = 2(p-d)n$ symbols of $\mathbf{x}$. This corruption is done by replacing the symbols of $\mathbf{x}$ by the corresponding symbols of $\mathbf{x}'$. If indeed $\mathbf{x} \neq \mathbf{x}'$, Bob has no way of determining whether Alice transmitted $\mathbf{x}$ or $\mathbf{x}'$. Thus Bob's probability

(over the message set and over the choice of Calvin) of decoding incorrectly is at least $\frac{1}{2}(1-q^{-\varepsilon n/2})^2 \geq \frac{1}{4}$.

Proof of Lower bound: We now prove that the rate $C_d^{\mathtt{ow}}(p)$ specified in Theorem 3 is indeed achievable with a computationally tractable code. The scheme we present covers all positive rates in the rate-region specified in Theorem 3, *i.e.*, whenever $p < 1/2$. In particular the rate R of our codes equal $1-p$ if $d > p$, and equals $1-2p+d$ if $d < p$. Our scheme follows roughly the ideas that appear in the scheme of Section 3. Namely, Alice's encoding scheme comprises of an erasure code along with a hash function used for authentication. However, in general, an `overwrite` adversary is more powerful than an `additive` adversary, because it can be directly shown that 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 3. 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.

Using notation similar to that of Section 3, let u be the message Alice would like to transmit to Bob, and $\mathbf{v} = v_i, \ldots, v_n$ be the encoding of u via an efficiently encodable and decodable erasure code (here 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 2, Alice treats each symbol of a codeword $\mathbf{x} = x_1, \ldots, x_n$ 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 ith 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 arranged into n sets of n^2 sub-symbols each, denoted by the sets σ_{ij} for each $j \in [n]$, and consist of the pairwise hashes of the symbols x_i and x_j. We define σ_{ij} to be $H_{r_{ij}}(w_j)$, where $H_{r_{ij}}$ is taken from (a slight variation to) a *pairwise independent* family $\mathcal{H}$ (known in advance to all parties). Namely, σ_{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 $(w_i, \{r_{ij}\}_j, \{H_{r_{ij}}(w_j)\}_j)$. Here $j = 1, \ldots, n$.

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 $C_d^{\mathtt{ow}}(p)$ specified in Theorem 3. Note that R equals C asymptotically in n and m. Thus the message u she wishes to transmit to Bob has $mRn = (m-2n^3)Cn = Cn^5$ sub-symbols over $\mathbb{F}_{q'}$. Alice uses an erasure code (resilient to $(1-C)n^5$ erasures) to transform these sub-symbols of u into the vector $\mathbf{v}$ comprising of n^5 sub-symbols over $\mathbb{F}_{q'}$. She then denotes consecutive blocks of n^4 sub-symbols of $\mathbf{v}$ by corresponding w_i's. More specifically, w_i consists of the sub-symbols in $\mathbf{v}$ in locations $n^4(i-1)+1$ through $n^4 i$. Here $i = 1, \ldots, n$.

The remainder of the proof is as follows. We first discuss the property of the family $\mathcal{H}$ of hash functions in use, needed for our analysis. We then describe and analyze Bob's decoding algorithm.

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 $H_{r_{ij}}(w'_j)$ equals $H_{r_{ij}}(w_j)$ is sufficiently small. Such functions are common in the literature (e.g., see [8, 7]). 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 $\sigma_{\mathbf{ij}} = W_i \mathbf{r}_{ij}$. Note that $H_{r_{ij}}(w'_j) = H_{r_{ij}}(w_j)$ means that $W'_j \mathbf{r}_{ij} = W_j \mathbf{r}_{ij}$, 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 at least 1. Thus a random r_{ij} satisfies $(W'_j - W_j)\mathbf{r}_{ij} = \mathbf{0}$ with probability $\leq 1/q'$.

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 σ_{ij} in x_i, and let w'_i, r'_{ij} and σ'_{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 4.1 *With probability at least $1 - 1/q'$, the following y_i and y_j are mutually inconsistent. (i)* Causality: *If $i > j$, $x_i = y_i$ and $w'_j \neq w_j$. (ii)* d-delay: *If $|i - j| < dn$, 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 d-Delay Online Overwriting Disruptive Adversary Decoding (d-DOODAD) Algorithm, described in detail below. We first give a high-level overview of the three major steps of d-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 dn locations; in this event Lemma 4.1(ii) implies that Bob detects the corruption of exactly one of a pair of symbols with high probability.

Based on the $O(dn^2)$ tests in the first step, in the second step he enumerates subsets of $\{y_1, \ldots, y_n\}$ 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. Naïvely, 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 2(c) in the d-DOODAD algorithm below) that candidate subsets must satisfy; we discuss this property after presenting the details of the algorithm. The effect of Step 2 below is to drastically curtail the number of candidate subsets that Bob needs to consider, to at most $n^{p/d}$, hence ensuring that this step is still computationally tractable.

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 that has even one corrupted symbol must generate a set of inconsistent linear equations. Hence Bob decodes by using the decoder for Alice's erasure code on 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 d-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^5 column vector $\mathbf{u}$ of Alice's message u to the length-n^5 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 $\mathcal{I} \subseteq [n^5]$ of size Cn^5, 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 ith rows/indices of $\mathcal{L}$, $\mathbf{v}$ and $\mathbf{w}'$ respectively, for all $i \in \mathcal{I}$.

d-Delay Online Overwriting Disruptive Adversary Decoding (d-DOODAD) Algorithm :

1. Bob constructs a *d-distance mutual consistency graph* $\mathcal{G}$ with vertex set $\{y_1, \ldots, y_n\}$ and edge-set comprising of all mutually consistent pairs (y_i, y_j) such that $|i - j| < nd$ (but no other edges). Thus $\mathcal{G}$ comprises of $\ell \leq n$ connected components $\{\mathcal{G}_1, \ldots \mathcal{G}_\ell\}$.
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 $c = p/d$. (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 consistent.
3. Let $\bar{\mathcal{K}} \subseteq [n^5]$ be the set comprising of indices in $\mathbf{w}'$ 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^5. If $\mathcal{L}_{\bar{\mathcal{K}}} \left(\left(\mathcal{L}_{\mathcal{I}} \right)^{-1} \mathbf{w}'_{\mathcal{I}} \right) = \mathbf{w}'_{\bar{\mathcal{K}}}$, 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.

Claim 4.1 *The d-DOODAD algorithm decodes Alice's message correctly with probability at least $1 - n^2/q'$.*

Proof: Throughout we assume that Lemma 4.1 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 $\mathcal{G}$. 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 2(c). Since Calvin can corrupt at most pn symbols, there must be some $\mathcal{C}(\mathcal{K})$ satisfying Step 2(b). To prove that $\mathcal{C}(\mathcal{K})$ also satisfies Step 2(a), we observe the following. If Calvin does not corrupt at least dn 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 $i = k_0 \leq k_1 \leq k_2 \leq \ldots \leq k_{j-i} = j$ such that any two consecutive symbols in the sequence have indices that differ by less than dn. 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 $c = p/d$ disjoint sequences of nd 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 $(1-p)n$ symbols from $\mathbb{F}_q$. Thus, by the definitions of m and C for Theorem 3, $\mathcal{C}(\mathcal{K})$ has at least $(1-p)n^5 \geq Cn^5$ uncorrupted sub-symbols over $\mathbb{F}_{q'}$. 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 $\mathcal{L}_{\bar{\mathcal{K}}}\left(\mathcal{L}_{\mathcal{I}}^{-1}\mathbf{w'}_{\mathcal{I}}\right) = \mathcal{L}_{\bar{\mathcal{K}}}\mathbf{u} = \mathbf{v}_{\bar{\mathcal{K}}} = \mathbf{w'}_{\bar{\mathcal{K}}}$.

We now show that there does not exist any $\mathcal{C}(\mathcal{K}')$ such that the corresponding output of the d-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 d-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 2(b), since $p < 1/2$. In addition, to pass Step 2(c), by Lemma 4.1(i), all the uncorrupted symbols of $\mathcal{C}(\mathcal{K}')$ must come before all the symbols corrupted by Calvin. Now notice that the uncorrupted and the corrupted symbols in $\mathcal{C}(\mathcal{K}')$ must be separated by a *separating set* $\mathcal{R}$ of at least nd consecutive symbols not in $\mathcal{C}(\mathcal{K}')$. If not, Lemma 4.1(ii) would imply that w.h.p. $\mathcal{C}(\mathcal{K}')$ does not satisfy Step 2(c) of d-DOODAD. Now note that the separating set $\mathcal{R}$ must contain at least dn consecutive symbols corrupted by Calvin. This follows from the fact that $\mathcal{C}(\mathcal{K}')$ consists of connected components. Namely, if $\mathcal{R}$ contains less than dn corrupted symbols, there must exist an uncorrupted symbol y_i and a corrupted symbol y_j, both in $\mathcal{C}(\mathcal{K}')$, satisfying $|j - i| < dn$. But this by Lemma 4.1(ii) would contradict Step 2(c). Notice that if $d > p$ we may conclude our proof at this point.

We now observe that there are at most $(p-d)n$ corrupted symbols in $\mathcal{C}(\mathcal{K}')$. This follows from the fact that $\mathcal{R}$ contains dn consecutive symbols corrupted by Calvin (not in $\mathcal{C}(\mathcal{K}')$), and the fact that Calvin can corrupt at most pn symbols. This, together with Step 2(b) of d-DOODAD, implies that the component set $\mathcal{C}(\mathcal{K}')$ contains a proper subset $\mathcal{C}(\mathcal{K}'')$ with at least Cn uncorrupted symbols. Finally, let $\mathcal{I}$ be any subset of Cn^5 uncorrupted sub-symbols in $\mathcal{C}(\mathcal{K}'')$. Let $\mathcal{I}'$ be any other subset of Cn^5 symbols in $\mathcal{C}(\mathcal{K}'')$. Consider the corresponding message vectors $\mathbf{u} = \mathcal{L}_{\mathcal{I}}^{-1}\mathbf{w'}_{\mathcal{I}}$ and $\mathbf{u'} = \mathcal{L}_{\mathcal{I}'}^{-1}\mathbf{w'}_{\mathcal{I}'}$ that Step 3 of d-DOODAD may decode to. Since $\bar{\mathcal{K}'}$ is of size at least $(1-p)n^5$, by the property of erasure codes [6], if $\mathbf{u'} \neq \mathbf{u}$, then $\mathcal{L}_{\bar{\mathcal{K}'}}\mathbf{u'} \neq \mathcal{L}_{\bar{\mathcal{K}'}}\mathbf{u}$. Thus $\mathcal{L}_{\bar{\mathcal{K}'}}\left(\mathcal{L}_{\mathcal{I}'}^{-1}\mathbf{w'}_{\mathcal{I}'}\right) \neq \mathcal{L}_{\bar{\mathcal{K}'}}\left(\mathcal{L}_{\mathcal{I}}^{-1}\mathbf{w'}_{\mathcal{I}}\right) = \mathcal{L}_{\bar{\mathcal{K}'}}\mathbf{u} = \mathbf{w'}_{\bar{\mathcal{K}'}}$, contradicting Step 3. ■

5 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 communication is over a size q alphabet, assumed to be large compared to the block-length n. An intriguing problem left untouched in this work concerns communication in the online adversarial setting over "small", *e.g.* binary, alphabets. The authentication schemes used extensively in this work depend integrally on the the alphabet size being large. They do not extend naïvely to the binary alphabet case, where new techniques seem to be needed.

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] I. Csiszár and J. Korner. *Information Theory: Coding Theorems for Discrete Memoryless Systems, 2nd edition.* Akademiai Kiado, New York, NY, 1997.

[5] 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.

[6] F.J. MacWilliams and N.J.A. Sloane. The theory of error-correcting codes. *North-Holland, Amsterdam*, 1977.

[7] M. Mitzenmacher and E. Upfal. *Probability and Computing, Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, Cambridge, UK, 2005.

[8] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, New York, NY, USA, 1995.

[9] 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.

[10] W. W. Peterson. Encoding and error-correction procedures for Bose-Chaudhuri codes. *IRE Transactions on Information Theory*, IT-60:459–470, 1960.

[11] 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.

[12] A. Sarwate. Robust and adaptive communication under uncertain interference. *PhD thesis, Berkeley*, 2008.

A List of parameters of our codes

		Capacity	Minimum q	Complexity	Probability of Error
Theorem 1		$1-2p$	$q > n$	$\mathcal{O}\left(n^2 \log n \log^3 q\right)$	0
Theorem 2		$1-p$	$n^{\Omega(1/\delta^2)}$	$\mathcal{O}\left(n^2 \log n \log^3 q\right)$	$\mathcal{O}\left(nq^{-\delta^2}\right)$
Theorem 3	$d < p < 0.5$	$1-2p+d$	$n^{\Omega(n^2/\delta^2)}$	$\mathcal{O}\left(n^{p/d+2} \log n \log^3 q\right)$	$\mathcal{O}\left(n^2 q^{-\delta^2/n^2}\right)$
	$p < d, p < 0.5$	$1-p$	$n^{\Omega(n^2/\delta^2)}$	$\mathcal{O}\left(n^2 \log n \log^3 q\right)$	$\mathcal{O}\left(n^2 q^{-\delta^2/n^2}\right)$

Table 1: Bounds on the capacity C, alphabet size q required to achieve capacity, computational complexity, and probability of error, of our main results. The bounds are in terms of the parameters p (adversary's power), d (adversary's delay), n (block-length), q (field-size), and δ (difference between the C and rate R).

Table 1 is obtained by careful analysis of the parameters of the algorithms corresponding to Theorems 1, 2 and 3. 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 1 substitute the rate-overhead parameter δ 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 δ parameter; we omitted this degree of freedom in the presentation of the proofs, for ease of exposition. Lastly, no effort has been made to optimize the tradeoffs between the parameters in Table 1; in fact, we have preliminary results on schemes that improve on some of these parameters (work in progress).

B Proof of Theorem 1

As discussed in the Introduction, the lower bound of Theorem 1 follows from known constructions [10, 1]. To complete the proof, then, all that is needed is a corresponding upper bound on the capacity. The required upper bound is novel. However, it is a special case of upper bound of Theorem 3, and follows directly if the parameter d in the corresponding proof is set to zero.

C Proof of Theorem 4

In the `jam-or-listen` online model, Calvin is assumed to be unaware of the value of the symbols x_i that he corrupts. Theorem 4 states that a `jam-or-listen` adversary is still as powerful as the previously described online adversaries, and is actually a corollary of Theorems 1, 2 and 3. First of all, the code constructions corresponding to the lower bounds are the same as in Theorems 1, 2 and 3. As for the upper bounds, it is not hard to verify that the attacks for Calvin outlined in each of the settings addressed in the paper correspond to a `jam-or-listen` adversary, and hence are valid attacks for this scenario as well.