File Updates Under Random/Arbitrary Insertions And Deletions

Qiwen Wang*, Viveck Cadambe†, Sidharth Jaggi*, Moshe Schwartz‡, Muriel Médard§
*Department of Information Engineering, The Chinese University of Hong Kong
†Department of Electrical Engineering, The Pennsylvania State University
‡Department of Electrical and Computer Engineering, Ben-Gurion University
§Research Laboratory of Electronics, Massachusetts Institute of Technology
Email: {wqw010, viveck, jaggi, schwartz, medard}@ie.cuhk.edu.hk, @engr.psu.edu, @ie.cuhk.edu.hk, @ee.bgu.ac.il, @mit.edu

***Abstract*—A client/encoder edits a file, as modeled by an insertion-deletion *(InDel)* process. An old copy of the file is stored remotely at a data-centre/decoder, and is also available to the client. We consider the problem of throughput- and computationally-efficient communication from the client to the data-centre, to enable the server to update its copy to the newly edited file. We study two models for the source files/edit patterns: the random pre-edit sequence left-to-right random InDel (RPES-LtRRID) process, and the arbitrary pre-edit sequence arbitrary InDel (APES-AID) process. In both models, we consider the regime in which the number of insertions/deletions is a small (but constant) fraction of the original file. For both models we prove information-theoretic lower bounds on the best possible compression rates that enable file updates. Conversely, our compression algorithms use *dynamic programming (DP)* and entropy coding, and achieve rates that are approximately optimal.**

I. Introduction

As the paradigm of cloud computing becomes pervasive, storing and transmitting files and their edited versions consumes a huge amount of resources (storage, bandwidth, computation) in client-datacentre channels, and intra-datacentre traffic.

If a file is "lightly edited", storing and transmitting the entire new file from clients to servers wastes a significant amount of space and bandwidth. For example, data-backup systems such as Dropbox and Time Machine keep regular snapshots of users' files. In revision-control software such as CVS, Git and Mercurial, users are likely to periodically commit and store their code after a small number of edits. Currently, many online-backup services use *delta encoding* (also known as *delta compression*), and only upload the edited pieces of files. However, to the best of our knowledge, none of these techniques provide information-theoretically optimal compression guarantees, and indeed this is the primary contribution of our work.

There are many other types of edits besides symbol insertions and deletions (for instance block insertions/deletion, substitutions, transpositions, *etc.* – see for example [2], [3], [6], [10]–[12]). Since these other edit models are in general a combination of symbol insertions and deletions, we focus on the "base case" of symbol insertions-deletions.

A. Our work/contributions

In this work, we study the problem of one-way communication of file updates to a data-centre. The client (encoder) has a file $\mathbf{X}$ (pre-edit source sequence PreESS) drawn from some distribution, and edits it according to some process – we shortly describe both the source and the edit process in more detail – to generate the new file (post-edit source sequence PosESS) $\mathbf{Y}$. The encoder has both the old file $\mathbf{X}$ and the edited version of the file $\mathbf{Y}$.[1] The encoder transmits a function of $\mathbf{X}, \mathbf{Y}$ to the data-centre (decoder). The pre-edit source sequence $\mathbf{X}$ is available at the decoder as side-information. The goal of communication is for the decoder to reconstruct $\mathbf{Y}$. A "good" communication scheme manages to achieve this while requiring minimal communication from the encoder to the decoder.

There are many possible combinations of different PreESS processes and edit processes. Some of them in the literature include: arbitrary input process [4], [6], random input process [7]–[10], (partial) permutations [11], duplications [13]; arbitrary edit process [1], [2]; random edit process [4], [6]–[8], [10], Markov edit process [9]. In this work, we consider two models. In the random pre-edit sequence left-to-right random InDel (RPES-LtRRID) process, a file is modeled as a sequence of symbols drawn i.i.d. uniformly at random from an alphabet $\mathcal{A}$. The new file is obtained from the old file through a *left-to-right random InDel process*, which is modeled as a Markov chain of three states: the "insertion state", the "deletion state", and the "no-operation state".[2]

We also study an arbitrary pre-edit sequence arbitrary InDel (APES-AID) process. In this model, the old file is modeled as an arbitrary sequence over an arbitrary alphabet $\mathcal{A}$. The post-edit source sequence $\mathbf{Y}$ is generated from the pre-edit source sequence $\mathbf{X}$ through an *arbitrary/"worst-case" InDel process* – we require that the number of edit operations is at most a small (but possibly constant) fraction of the file length n. The sequence of edits (insertions and deletions) is arbitrary up to an upper bound on the total number, occurs in arbitrary positions, and inserts arbitrary symbols from $\mathcal{A}$ for edits corresponding to insertions. Both the RPES-LtRRID and the APES-AID models are described formally in Section II-A.

In both models, we consider arbitrary alphabet sizes. We first prove information-theoretic lower bounds on the compression rate needed so that the decoder is able to reconstruct $\mathbf{Y}$ for both models. To do so we build non-trivially on recent work on

The work of Qiwen Wang and Sidharth Jaggi was supported by a grant from University Grants Committee of the Hong Kong Special Administrative Region, China (Project No. AoE/E-02/08). The work of Muriel Médard was supported by the National Science Foundation (NSF) under Grant No. CCF-1409228.

[1] The encoder may ALSO have access to the actual edit process, but as we shall see this doesn't necessarily help in our problem.

[2] Roughly speaking, these three states correspond to the cursor moving "from left to right", and at each point, either a uniformly random symbol is inserted, the symbol at the cursor is deleted, or the cursor jumps ahead without changing the previous symbol.

Ref	[1]Prob Description	[2] $\mathcal{A}$ size	[3]Pre-ESS	[4]Edits	[5]Edit Ops	[6] #(Edits) as $(\epsilon+\delta)n$	[7] Explicit Info Theo LB	[8]Algo	[9]Comp	[10] P_e	[11] #(bits) transmitted	[12] Remarks
[1]O93	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	2	Arb	Arb	Ins,Del,etc.	$\mathcal{O}(n)$	Y	R	$\mathcal{O}(e^n)$	0	$(1+\epsilon+\delta)H(\frac{\epsilon+\delta}{1+\epsilon+\delta})n+\mathcal{O}(\log n)$	Upper bound on total # of ins & del, $(\mathbf{X},\mathbf{Y})$ balanced pair *only for edits not changing runs
	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightarrow Dec$							D	-		$(\epsilon+\delta)n\log(1+\epsilon+\delta)n+o(n\log n)$*	
	$Enc \Leftarrow \{\mathbf{X},\mathbf{Y}\}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightarrow Dec$							D	$\mathcal{O}(n^2)$		$(\epsilon+\delta)n[\log((1+\epsilon+\delta)n)+2]$	
[2]OV01	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	2	Arb	Arb	Ins,Del,etc.	$\mathcal{O}(n)$	N	-	$\mathcal{O}(n\log n)$	ε	$2(\epsilon+\delta)n\log n(2\log n+$ $\log\log n+\log(\epsilon+\delta)-\log P_e)$	Theoretical upper bd $(\epsilon+\delta)n\log n$
[3]CPC$^+$00	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	$\lvert\mathcal{A}\rvert$	Arb	Arb	Ins,Del,Sub	$\mathcal{O}(n)$	Y	R	$\mathcal{O}(n\log n)$	ε	$\Theta((\epsilon+\delta)n\log^2 n)$	LZ distance, block edits
[4]VZR10	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	2	Arb	Ran	Ins,Del	$o(\frac{n}{\log n})$	-	D	$\mathcal{O}(n)$	ε	$\mathcal{O}((\epsilon_n+\delta_n)n\log n)$	build on VT code [5]
		$\lvert\mathcal{A}\rvert$									$\mathcal{O}((\log\lvert\mathcal{A}\rvert)(\epsilon_n+\delta_n)n\log n)$	
[6]VSR13	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	2	Arb	Ran	Ins,Del,Sub	$o(n)$	-	D	$\mathcal{O}(n)$	ε	$\Theta((\epsilon_n+\delta_n)^{2/3}n\log n)$	
		$\lvert\mathcal{A}\rvert$									$\Theta((\log\lvert\mathcal{A}\rvert)(\epsilon_n+\delta_n)^{2/3}n\log n)$	
[7]YD12	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	2	Ran	Ran	Del	$\mathcal{O}(n)$	-	D	$\mathcal{O}(n^4)$	ε	$\mathcal{O}((-\delta\log\delta)n)$	
[8]BD13	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	$\lvert\mathcal{A}\rvert$	Ran	Ran	Ins,Del	$\mathcal{O}(n)$	-	D	-	ε	$\mathcal{O}((-(\epsilon+\delta)\log(\epsilon+\delta))n)$	$\mathcal{A}$ can be non-uniform
[9]MRT11	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightarrow Dec$	2	Ran	Markov	Del	$\mathcal{O}(n)$	Y	-	-	ε	$(-\delta\log\delta+\delta(\log 2e-1.29)$ $+\mathcal{O}(\delta^{2-\tau}))n$	In [4] Ran is a special case of Markov
[10]MRT12	$Enc \Leftarrow \{\mathbf{X},\mathbf{Y}\}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightarrow Dec$	2	Ran	Ran	Ins,Del,Sub	$\mathcal{O}(n)$	N	D	$\mathcal{O}(n^2)$	0	$(\lim_{n\to\infty} H(\mathbf{X}\lvert\mathbf{Y})/n+$ $\mathcal{O}(\max(\epsilon,\delta)^{2-\tau}))n$	
This work	$Enc \Leftarrow \{\mathbf{X},\mathbf{Y}\}$ $Dec \Leftarrow \mathbf{X}$ $Enc \rightarrow Dec$	$\lvert\mathcal{A}\rvert$	Arb	Arb	Ins,Del	$\mathcal{O}(n)$	Y	D	$\mathcal{O}(n^2)$	0	$(H(\delta)+H(\epsilon)+\epsilon\log\lvert\mathcal{A}\rvert$ $+\overline{\lambda}_A\cdot\epsilon^2)n$	
			Ran	Ran	Ins,Del	$\mathcal{O}(n)$	Y	D	$\mathcal{O}(n^2)$	ε	$(H(\delta)+H(\epsilon)+\epsilon\log\lvert\mathcal{A}\rvert$ $+\overline{\lambda}_R\cdot\max(\epsilon,\delta)^{2-\tau(\epsilon,\delta)})n$	

Table 1: (Related work) The content of each column is as follows – [1] Two aspects of each communication model are shown here. The first aspect concerns what information is available to which party. Depending on the specific model considered, either the original file (the pre-edit source sequence) $\mathbf{X}$, or the new file (the post-edit source sequence) $\mathbf{Y}$, or both may be available at the encoder and the decoder. The second aspect considered is whether interactive/two-way transmissions between the encoder and decoder are allowed, or only the encoder is allowed to transmit (one-way communication). [2] The size of the source alphabet – 2 denotes a binary source alphabet, and $|\mathcal{A}|$ denotes a general non-binary alphabet. [3] 'Arb' represents an arbitrary ("worst-case") pre-edit source sequence; 'Ran' represents the pre-edit sequences drawn i.i.d. from the alphabet. [4] 'Arb' represents the positions and contents of the edits being arbitrary; 'Ran' represents random positions and contents of edits; 'Markov' represents the edit process being a Markov chain. [5] Here 'Ins','Del' and 'Sub' respectively represent insertion, deletion and substitution edit operations. [6] Upper bounds on the number of edits in each work, as a function of n (length of the pre-edit source sequence $\mathbf{X}$). [7] Whether an explicit information-theoretical lower bound is presented, where 'Y' and 'N' stands for 'Yes' and 'No' respectively. [8] Whether the algorithm is deterministic ('D') or random ('R'). [9] The complexity of the algorithm, as a function of n (length of the pre-edit source sequence $\mathbf{X}$). [10] Whether the algorithm has "small" error – ε-error, or zero error. [11] The number of bits transmitted. In our notation, ε stands for the fraction (of n) of insertions, and δ for the fraction of deletions. In [4], [6]–[8], the fractions of insertions and deletions vanish with n, hence the corresponding variables are denoted ϵ_n and δ_n. [12] This column has additional remarks on specific works.

the deletion channel [14] in the random pre-edit sequence/edit model (see Theorem 1), and provide a combinatorial argument in the arbitrary pre-edit source/edit model (see Theorem 2). We then design "universal" computationally-efficient achievability schemes based on dynamic programming and entropy coding (see Theorems 3 & 4). The compression rates achieved by the scheme is an explicitly computable additive term away from the lower bounds for almost all alphabet-sizes[3], and number of edits. In the regime wherein the number of edits is a small (but possibly constant) fraction of the length of $\mathbf{X}$ and the alphabet-size is large, this term is small (Section III).

B. Related work

Various models of the file-synchronization problem have been considered in the literature – see Table 1 for a summary. Our work here differs from each of those works in significant ways. For instance, in our model the encoder knows both files, hence we design one-way communication protocols (rather than the multi-round protocols required in the models where the encoder and the decoder each has one version of the file as in [2]–[4], [6]–[8]); hence our protocols are information-theoretically near-optimal (however for two-way communication model, computationally efficient schemes which achieve rates within constant factors of the lower bounds are already challenging). The one-way file synchronization model studied in [9], [10] is the closest to our RPES-LtRRID model. For the information-theoretical lower bound, we differ from [9] by considering both insertions and deletions, and arbitrary alphabet. The achievability scheme in [10] matches the lower bound up to first order term for the random source/edit model, whereas our scheme is "universal" for both RPES-LtRRID and APES-AID models in this work. The literature on insertion/deletion channels and error-correcting codes is also quite closely related – indeed, we borrow significantly from techniques in [14]–[16].

[3]In the random source/edit model, we actually have no restriction on the alphabet-size; in the arbitrary source/edit model, for technical reasons, our bounds hold only for alphabets of size at least 3.

II. Model

A. Edit Process

1) Random Pre-Edit Sequence Left-to-Right Random InDel (RPES-LtRRID) Process: As noted in the introduction, different stochastic models for source- and edit-processes have been considered in the literature. We study a *RPES-LtRRID* process, which is motivated by the Markov deletion model in [9].[4]

Pre-edit source sequence (PreESS): The source initially has a *pre-edit source sequence* $\bar{\mathbf{X}} = (\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_n)$, a length-$n$ sequence of symbols drawn i.i.d. uniformly at random from the source alphabet $\mathcal{A} = \{0, \ldots, a-1\}$. Finally, we append an *end of file* symbol $\bar{X}_{n+1} = \text{eof}$ to the end of $\bar{\mathbf{X}}$. We denote the distribution of the pre-edit source sequence by $p(\bar{\mathbf{X}})$.

InDel process: The InDel process is a Markov chain with three states as shown in Figure 1: the "insertion state" $\bar{\iota}$ inserts (writes) a symbol uniformly drawn from $\mathcal{A}$; the "deletion state"

[4]Our results should in general translate over to other stochastic models as well in the regime wherein there is a small number of insertions and deletions.

$\bar{\Delta}$ reads one symbol rightwards in $\bar{\mathbf{X}}$ and deletes the symbol; the "no-operation state" $\bar{\eta}$ reads one symbol rightwards in the pre-edit source sequence $\bar{\mathbf{X}}$, and do nothing.

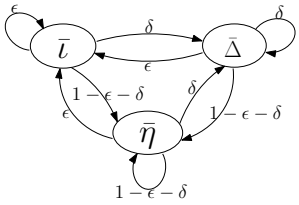

Fig. 1: *Left-to-Right Random InDel (LtRRID)* process: starting in front of the first symbol of $\bar{\mathbf{X}}$, at each step, insert a symbol uniformly drawn from $\mathcal{A}$ with probability ϵ, read one symbol rightwards and delete it with probability δ, read one symbol rightwards and do nothing with probability $1-\epsilon-\delta$. Note that an inserted symbol is never deleted in this process. Whereas, a deleted symbol might be inserted back right away, with probability $\epsilon\frac{1}{|\mathcal{A}|}$. The process stops when it reaches the end of file $\bar{X}_{n+1} = \text{eof}$.

The edit process starts in front of $\bar{X}_1$ and ends when it reaches the end of file $\bar{X}_{n+1} = \text{eof}$. This means that in our model, the total number of deletions plus no-operations equals exactly n. In addition there are a potentially unbounded number of insertions (though in our model the expected number of insertions is bounded). The number of deletions and insertions are random variables K_D and K_I respectively. We describe the *edit pattern* of the InDel process by a pair of sequences $\bar{\mathbf{E}} = (\bar{O}^{n+K_I}, \bar{C}^{K_I})$, where the edit operation pattern is $\bar{O}^{n+K_I} \in \{\bar{\iota}, \bar{\Delta}, \bar{\eta}\}^{n+K_I}$ and the insertion content is $\bar{C}^{K_I} \in \mathcal{A}^{K_I}$. The transition probabilities (as shown in Figure 1) set the *random* (ϵ, δ)*-InDel process* to be an i.i.d. InDel process with $P(\bar{\iota}) = \epsilon$, $P(\bar{\Delta}) = \delta$, and $P(\bar{\eta}) = 1-\epsilon-\delta$.

Post-edit source sequence (PosESS): The *post-edit source sequence* $\bar{\mathbf{Y}} = \bar{\mathbf{Y}}(\bar{\mathbf{X}}, \bar{\mathbf{E}})$ is a sequence obtained from $\bar{\mathbf{X}}$ through the InDel process $\bar{\mathbf{E}} = (\bar{O}^{n+K_I}, \bar{C}^{K_I})$.

Post-edit set: Given any PreESS $\bar{\mathbf{X}}$, any PosESS $\bar{\mathbf{Y}}$ in $\mathcal{A}^*$ (any sequence over $\mathcal{A}$ of any length) might be in its post-edit set, albeit with possibly "very small" probability. In fact, for any $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$, there may be multiple edit patterns that generate $\bar{\mathbf{Y}}$ from $\bar{\mathbf{X}}$. We use $p(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$ to denote the probability that the output of the random left-to-right InDel process generates $\bar{\mathbf{Y}}$ from $\bar{\mathbf{X}}$ (via any edit pattern).

Runs: We use the usual definition (see, for example [17]) of a *run* being a maximal block of contiguous identical symbols. Since we shall be interested in runs of several different sequences, to avoid confusion about the parent sequence we use $\mathbf{S}$*-run* to denote a run in a sequence $\mathbf{S}$.

2) Arbitrary Pre-Edit Sequence Arbitrary InDel (APES-AID) Process: We say that the PreESS is arbitrary if it can equal any length-n sequence over $\mathcal{A}$. Similarly, we say the InDel process is arbitrary if at most ϵn insertions and δn deletions occur in the edit pattern. We refer the reader to Section II-B2 of [18] for a detailed description.[5]

B. Communication Model

The system diagram for the *RPES-LtRRID process* model is presented in Figure 2. The code $\bar{\mathcal{C}}_n^{\epsilon,\delta}$ comprises the encoder-decoder pair $(\bar{\text{Enc}}, \bar{\text{Dec}})$. The *average rate* $\bar{R}$ of the code $\bar{\mathcal{C}}_n^{\epsilon,\delta}$ is

[5]Note that in the *APES-AID process*, the order of insertions and deletions in the edit process is in general arbitrary. However, we can simplify the model by separating the insertions and deletions. (See Fact 1 in [18].)

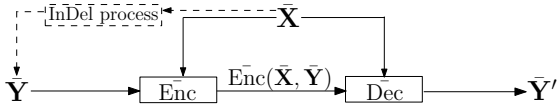

Fig. 2: Communication model: The source has both the random PreESS $\bar{\mathbf{X}}$ and the random PosESS $\bar{\mathbf{Y}}$, as discussed in Section II-A1. The sequence $\bar{\mathbf{Y}}$ is obtained from $\bar{\mathbf{X}}$ through the random (ϵ, δ)-InDel process discussed in Section II-A1. The source encodes the source sequences $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ into a transmission $\bar{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ and sends it to the decoder through a noiseless channel. The arbitrary PreESS $\bar{\mathbf{X}}$ is available at the decoder as side-information. The decoder receives $\bar{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$, and regenerates the arbitrary PosESS $\bar{\mathbf{Y}}'$ from $(\bar{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}}), \bar{\mathbf{X}})$. Here the bar superscript is used to denote the fact that the source sequences and edit process are as described in Section II-A1 rather than Section II-A2. The communication model for the APES-AID model discussed in Section II-A2 is similar, except that the quantity $\{\bar{\mathbf{X}}, \bar{\mathbf{Y}}, \bar{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}}), \bar{\mathbf{Y}}'\}$ are replaced with $\{\mathbf{X}, \mathbf{Y}, \text{Enc}(\mathbf{X}, \mathbf{Y}), \mathbf{Y}'\}$.

the average number of bits transmitted by the encoder, defined as $\sum_{\bar{\mathbf{X}}\in\mathcal{A}^n, \bar{\mathbf{Y}}\in\mathcal{A}^*} p(\bar{\mathbf{X}}, \bar{\mathbf{Y}}) \log |\bar{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})|$, where, as in Figure 2, $\bar{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ denotes the transmission and $|\bar{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})|$ denotes its length. A code $\bar{\mathcal{C}}_n^{\epsilon,\delta}$ is "$(1-P_e)$-good" if the *average probability of error*, defined as $\Pr_{\bar{\mathbf{X}}\in\mathcal{A}^n, \bar{\mathbf{Y}}\in\mathcal{A}^*}\{(\bar{\mathbf{X}}, \bar{\mathbf{Y}}) : \bar{\text{Dec}}(\bar{\text{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}}), \bar{\mathbf{X}}) \neq \bar{\mathbf{Y}}\}$, is less than P_e. A rate $\bar{R}_{\epsilon,\delta}$ is said to be *achievable on average* if for any $P_e > 0$ there is a code for sufficiently large n which is $(1-P_e)$-good. The infimum (over all n and corresponding $\bar{\mathcal{C}}_n^{\epsilon,\delta}$) of all achievable rates $\bar{R}_{\epsilon,\delta}^*$ is called the *optimal average transmission rate*.

The APES-AID process model is structurally similar to the RPES-LtRRID model, except that we require zero-error reconstruction at the decoder, and we measure worst-case communication cost. We again refer the reader to Section II-C in [18] for details.

III. Main Results

Theorem 1 (RPES-LtRRID Lower Bound)**.** *For any $\tau > 0$, and for all sufficiently small ϵ, δ, the optimal achievable transmission rate $\bar{R}_{\epsilon,\delta}^*$ for the RPES-LtRRID process satisfies $\bar{R}_{\epsilon,\delta}^* \geq H(\delta) + H(\epsilon) + \epsilon\log|\mathcal{A}| - (\delta+\epsilon)C_{|\mathcal{A}|} - \underline{\lambda}_R \cdot \max(\epsilon,\delta)^{2-\tau}$, where $C_{|\mathcal{A}|} = \sum_{l=1}^{\infty}\left(\frac{1}{|\mathcal{A}|}\right)^{l-1}\left(1-\frac{1}{|\mathcal{A}|}\right)^2 l\log l$ is a constant that depends only on the alphabet size $|\mathcal{A}|$, and $\underline{\lambda}_R \leq 56$ is a universal positive constant.*

Theorem 2 (APES-AID Lower Bound)**.** *For all sufficiently small ϵ and δ, and for all alphabet-sizes $|\mathcal{A}| \geq 3$, the optimal achievable transmission rate $R_{\epsilon,\delta}^*$ for the APES-AID process satisfies $R_{\epsilon,\delta}^* \geq H(\delta)+H(\epsilon)+\epsilon\log(|\mathcal{A}|-2)-\underline{\lambda}_A\cdot\max(\epsilon,\delta)^2$, where $\underline{\lambda}_A \leq 4\log e$ is a universal positive constant.*

Theorem 3 (RPES-LtRRID Achievability)**.** *For any $\tau > 0$, and for all sufficiently small ϵ, δ, the optimal achievable transmission rate $\bar{R}_{\epsilon,\delta}^*$ for the RPES-LtRRID process satisfies $\bar{R}_{\epsilon,\delta}^* \leq H(\delta) + H(\epsilon) + \epsilon\log|\mathcal{A}| + \overline{\lambda}_R \cdot \max(\epsilon,\delta)^{2-\tau}$, where $\overline{\lambda}_R \leq 2\log|\mathcal{A}| + 2\log e + 4$ is a positive constant that depends only on the alphabet size $|\mathcal{A}|$.*

Theorem 4 (APES-AID Achievability)**.** *For all sufficiently small ϵ and δ, the optimal achievable transmission rate $R_{\epsilon,\delta}^*$ for the APES-AID process satisfies $R_{\epsilon,\delta}^* \leq H(\delta) + H(\epsilon) + \epsilon\log|\mathcal{A}| + \overline{\lambda}_A \cdot \epsilon^2$, where $\overline{\lambda}_A \leq 2\log e$ is a universal positive constant.*

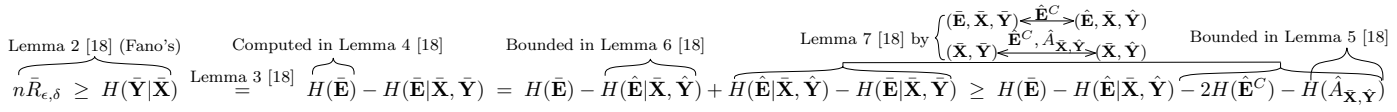

Fig. 3: Flowchart of the proof (Theorem 1): The lower bound of the amount of information that the encoder needs to send to the decoder is given by the conditional entropy $H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$ (Lemma 2 [18]), which we show in Lemma 3 [18] equals to the amount of information to describe the edit pattern $H(\bar{\mathbf{E}})$ subtracts an amount called "nature's secret" $H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})$. We calculate $H(\bar{\mathbf{E}})$ in Lemma 4 [18]. To characterize nature's secret $H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})$, we perturb the edit pattern $\bar{\mathbf{E}}$ to a "typicalized" edit pattern $\hat{\mathbf{E}}$. We show in Lemma 7 [18] that nature's secret $H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})$ is within at most an order $\mathcal{O}(\max(\epsilon,\delta)^{2-\tau})$ distance from the "typicalized nature's secret" $H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}})$, which we characterize in Lemma 6 [18].

IV. Lower Bound

A. RPES-LtRRID Process (Theorem 1)

This is probably the most challenging part of this work. Details are in the long version [18]. We present intuition here.

Since the decoder already has access to PreESS $\bar{\mathbf{X}}$, the entropy of $\bar{\text{Enc}}(\bar{\mathbf{X}},\bar{\mathbf{Y}})$ merely needs to equal $H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$, the conditional entropy of the entire PosESS given the PreESS (Lemma 2 [18]). The challenge is to characterize this conditional entropy in single-letter/computable form, rather than as a "complicated" function of n – indeed the same challenge is faced in providing information-theoretic converses for *any* problems in which information is processed and/or communicated. For scenarios when the relationship from $\bar{\mathbf{X}}$ to $\bar{\mathbf{Y}}$ corresponds to a *memoryless* channel, standard techniques often apply – unfortunately, this is not the case in our file update problem. We follow the lead of [14][6], which noted that for InDel processes that are independent of the sequence being edited (as in our case), characterizing $H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$ is equivalent to characterizing $H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})$. (Recall that $\bar{\mathbf{E}}$ denotes the random variable corresponding to the edit pattern.) In fact $H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$ can be written as $H(\bar{\mathbf{E}})-H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})$ (Lemma 3 [18]). This is because of the aforementioned independence between $\bar{\mathbf{E}}$ and $\bar{\mathbf{X}}$, and the fact that $\bar{\mathbf{Y}}$ is a deterministic function of $\bar{\mathbf{X}}$ and $\bar{\mathbf{E}}$. But the entropy of the edit patterns is exactly equal to the entropy of specifying the locations of deletions, and insertions and their contents. Since multiple edit patterns can take a PreESS $\bar{\mathbf{X}}$ to a PosESS $\bar{\mathbf{Y}}$, the term $H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})$ corresponds to the uncertainty in the edit pattern given both $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$. The intuition is that disambiguating this uncertainty is useless for the problem of file updating, hence this quantity is called "nature's secret" in [9]. For instance, given $\bar{\mathbf{X}}=00000$ and $\bar{\mathbf{Y}}=000$, the decoder doesn't know, nor does it need to know, which specific pattern of two deletions converted $\bar{\mathbf{X}}$ to $\bar{\mathbf{Y}}$; all the encoder needs to communicate to the decoder is that there are two deletions. In general, if a symbol is deleted from a run or the same symbol generating a run is inserted in the run (edits that shorten or lengthen runs in $\bar{\mathbf{X}}$), the encoder doesn't need to specify to the decoder the exact locations of deletions or insertions in $\bar{\mathbf{X}}$-runs.

However, characterizing $H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})$ is still a non-trivial task, since it corresponds to an entropic quantity of "long sequences with memory". One challenge is that it is hard to align $\bar{\mathbf{X}}$-runs and $\bar{\mathbf{Y}}$-runs. In other words, it's in general difficult to tell which run/runs in $\bar{\mathbf{X}}$ lead to a run in $\bar{\mathbf{Y}}$ (we call this run/runs in $\bar{\mathbf{X}}$ *parent run/runs* of the run in $\bar{\mathbf{Y}}$ [14]). We develop the approach in [14]:

- We first carefully "perturb" the original edit pattern $\bar{\mathbf{E}}$ to a *typicalized edit pattern* $\hat{\mathbf{E}}$, by eliminating some "dense" edits. (See Definition 1 and Figure 5 in Section III-A2 in [18] for details and example).
- We compute the *typicalized PosESS* $\hat{\mathbf{Y}}$ corresponding to operating the typicalized edit pattern $\hat{\mathbf{E}}$ on the PreESS $\bar{\mathbf{X}}$.
- We show via non-trivial case analysis (Lemma 5 [18]) that with a "small amount" ($\mathcal{O}(\max(\epsilon,\delta)^2)n$) of additional alignment information $\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$, $\bar{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ can be aligned.
- We show two implications of the above alignment: Lemma 7 [18] shows that $H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}})$ is "close" to $H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})$; Lemma 6 [18] provides a bound on $H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}})$.

Pulling together the implications of the steps above enables us to characterize, up to first order in ϵ and δ, $H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$.

B. APES-AID Process (Theorem 2)

Given an arbitrary pre-edit source sequence $\mathbf{X}\in\mathcal{A}^n$, the $\mathbf{X}$-*post-edit set* $\mathcal{Y}_{\epsilon,\delta}(\mathbf{X})$ denotes the set of all sequences over $\mathcal{A}$ that may be obtained from $\mathbf{X}$ via an arbitrary (ϵ,δ)-InDel process. For zero-error decodability, The encoder needs to send $\log|\mathcal{Y}_{\epsilon,\delta}(\mathbf{X})|$ bits to decoder. The larger the $\mathbf{X}$-post-edit set, the larger the corresponding lower bound on the optimal achievable rate. Hence to find a "good" lower bound on the optimal achievable rate, one needs to find a pre-edit sequence $\mathbf{X}$ with a large $\mathbf{X}$-post-edit set.[7] Below we provide a proof sketch of Theorem 2, by constructing a PreESS $\mathbf{X}_{\text{LB}}$ and a subset of InDel patterns, such that any of the InDel patterns in the subset, applied to $\mathbf{X}_{\text{LB}}$, results in a distinct PosESS $\mathbf{Y}$.

Proof sketch of Theorem 2: Consider a PreESS $\mathbf{X}_{\text{LB}}$ constructed by alternating two symbols, eg: $0101\ldots01$. We describe a subset of arbitrary (ϵ,δ)-InDel patterns that result in a large $\mathbf{X}_{\text{LB}}$-post-edit set. In this subset of InDel patterns, we require that (1) all the δn deletions precede all the ϵn insertions; (2) the deletions, and then the insertions, occur in a "left-to-right manner"; (3) the deletions may delete any δn non-pairwise-contiguous symbols; (4) each insertion may only insert symbols from $\{2,\ldots,|\mathcal{A}|-1\}$. It can be verified that each edit pattern results in a distinct PosESS $\mathbf{Y}$, by noting that given $\mathbf{X}_{\text{LB}}$ and $\mathbf{Y}$, one can reconstruct the edit pattern. (See Section III-B in [18] for a full argument.) The number of such InDel patterns as described above is $\binom{n-\delta n}{\delta n}\binom{n-\delta n+\epsilon n}{\epsilon n}(|\mathcal{A}|-2)^{\epsilon n}$, hence is a lower bound on the number of PosESS $|\mathcal{Y}_{\epsilon,\delta}(\mathbf{X}_{\text{LB}})|$. The

[6] One major difference between our work and the analysis in [14] is that since we consider both insertions and deletions, our case-analysis is significantly more intricate. Another difference is that we explicitly characterize our bounds for sequences over all (finite) alphabet sizes, whereas [14] concerned itself only with binary sequences. Besides the difference in models and techniques, the underlying motivation also differs. The authors of [14] focused on characterizing the capacity of deletion channels (and hence they could choose arbitrary subsets of PreESS). On the other hand we focus on the file update problem (and hence our "channel input" PreESS $\bar{\mathbf{X}}$ is drawn according to source statistics).

[7] In [15], [16], Levenshtein showed the post-edit sets' sizes for the cases with only insertions or deletions. To the best of our knowledge, there is no literature on the bounds for the scenario with both insertions and deletions.

corresponding lower bound on the optimal achievable rate $R^*_{\epsilon,\delta}$ $-\frac{1}{n}\log|\mathcal{Y}_{\epsilon,\delta}(\mathbf{X}_{\mathrm{LB}})|$, is asymptotically $(1-\delta)H\left(\frac{\delta}{1-\delta}\right)+(1-\delta+\epsilon)H\left(\frac{\epsilon}{1-\delta+\epsilon}\right)+\epsilon\log(|\mathcal{A}|-2)$ by Stirling's approximation, which is at least $H(\delta)+H(\epsilon)+\epsilon\log|\mathcal{A}|-\epsilon\frac{2\log e}{|\mathcal{A}|-2}-\underline{\lambda}_A\cdot\max(\epsilon,\delta)^2$ by Taylor expansion, where $\underline{\lambda}_A\leq 4\log e$ is a universal constant independent of $\epsilon,\delta,|\mathcal{A}|$. □

V. Algorithm and Performance

We propose a unified coding scheme for both the RPES-LtRRID and the APES-AID processes.[8] The scheme is a combination of dynamic programming (DP) and entropy coding.[9]

A. Algorithm

In this section we unify the notations for both RPES-LtRRID and APES-AID processes by notations without bars.

The encoder Φ_n takes in the PreESS $\mathbf{X}$ and the PosESS $\mathbf{Y}$ as inputs, and outputs a transmission $\mathrm{Enc}(\mathbf{X},\mathbf{Y})$ as follows. The first subroutine of the encoder runs a dynamic program on the input $(\mathbf{X},\mathbf{Y})$ to output an edit pattern $\tilde{\mathbf{E}}$ with $\tilde{\epsilon}n$ insertions and $\tilde{\delta}n$ deletions. This edit pattern $\tilde{\mathbf{E}}$ satisfies the condition that $(\tilde{\epsilon}+\tilde{\delta})n$ is the minimum number of edits needed to convert $\mathbf{X}$ to $\mathbf{Y}$. This computation dominates the time-complexity, taking time $\mathcal{O}(n^2(\epsilon+\delta))$ (see [19]). The output edit pattern $\tilde{\mathbf{E}}$ is represented as a pair of sequences $(\tilde{O}^{n+\tilde{\epsilon}n},\tilde{C}^{\tilde{\epsilon}n})$, where the edit operation pattern $\tilde{O}^{n+\tilde{\epsilon}n}\in\{\bar{\iota},\bar{\Delta},\bar{\eta}\}^{n+\tilde{\epsilon}n}$ specifies the edit operations and the insertion content pattern $\tilde{C}^{\tilde{\epsilon}n}\in\mathcal{A}^{\tilde{\epsilon}n}$ specifies the content of insertions. The encoder then uses a Lempel-Ziv entropy code to compress $\tilde{O}^{n+\tilde{\epsilon}n}$ and $\tilde{C}^{\tilde{\epsilon}n}$. The decoder decodes $\tilde{O}^{n+\tilde{\epsilon}n}$ and $\tilde{C}^{\tilde{\epsilon}n}$ by a corresponding Lempel-Ziv entropy decoder, and reconstructs $\mathbf{Y}$ from $(\mathbf{X},\tilde{O}^{n+\tilde{\epsilon}n},\tilde{C}^{\tilde{\epsilon}n})$.

B. Performance

In the RPES-LtRRID process, the number of deletions and insertions may exceed the expected values $\frac{\delta}{1-\epsilon}n$ and $\frac{\epsilon}{1-\epsilon}(n+1)$ respectively, in which case more bits may in general need to be transmitted. Moreover, the number of insertions can be unbounded. We show that these events contribute a negligible amount to the expected rate of communication as the block length n tends to infinity, by using the Chernoff bound to show that the probability the number of insertions/deletions is "much more" than the expected value is exponentially small in the block length n, while the amount contributed to the rate is asymptotically negligible in the block length n. See Section IV-B1 in [18] for details. The remainder of our argument holds for both APES-AID and RPES-LtRRID processes.

It is well-known in the literature (see for instance [19]) that it is possible to use *dynamic programming* to find the the minimal *total number* of insertions and deletions needed to convert one sequence to the other. In our models, it can be further deduced that the number $\tilde{\epsilon}n$ of insertions and the number $\tilde{\delta}n$ of deletions output by the *DP* are both minimized, for the following reason. For each edit pattern that converts $\mathbf{X}$ to $\mathbf{Y}$, the number of insertions (K_I) and the number of deletions (K_D) are jointly subject to the constraint $K_D-K_I=|\mathbf{X}|-|\mathbf{Y}|$, where the lengths of two source sequences $|\mathbf{X}|$ and $|\mathbf{Y}|$ are fixed given the two sequences. Hence, minimizing K_D+K_I over all the edit patterns that converts $\mathbf{X}$ to $\mathbf{Y}$ minimizes both K_D and K_I.

A natural upper bound on the number of bits required to specify an edit sequence with at most $\tilde{\epsilon}n$ insertions and at most $\tilde{\delta}n$ deletions follows by noting (as shown in Appendix C in [18]) that the entropy code used to compress the sequence requires a rate of at most $H(\tilde{\delta})+H(\tilde{\epsilon})+(\log e)\tilde{\epsilon}^2+\mathcal{O}(\tilde{\epsilon}^4)$.

[8] With slight modification to the "run-length" compression scheme in [10] for the RPES-LtRRID process, the rate achieved matches the lower bound up to all first order terms (including the "nature's secret" term). However, in this work we focus on a "universal" scheme for both random and arbitrary models.

[9] Using DP to find the edit distance between two sequences has been considered before in the literature (see for instance [19]). Our contribution here is to demonstrate that for large alphabets and a small number of edits, this algorithmic procedure results in an expected description length that matches information-theoretic lower bounds up to lower-order terms.

References

[1] A. Orlitsky, "Interactive communication of balanced distributions and of correlated files," *SIAM J. Discr. Math.*, vol. 6, no. 4, pp. 548–564, 1993.

[2] A. Orlitsky and K. Viswanathan, "Practical protocols for interactive communication," in *Proc. IEEE Int'l Symp. on Info. Theory*, 2001, p. 115.

[3] G. Cormode, M. Paterson, S. C. Sahinalp, and U. Vishkin, "Communication complexity of document exchange," in *Proc. of the ACM-SIAM Symp. on Discrete algorithms*, Jan. 2000.

[4] R. Venkataramanan, H. Zhang, and K. Ramchandran, "Interactive low-complexity codes for synchronization from deletions and insertions," in *Proc. 48th Allerton Conf. on Com., Control, and Comp.*, 2010.

[5] R. R. Varshamov and G. M. Tenenholtz, "A code for correcting a single asymmetric error," *Autom. Telemekh.*, vol. 26, pp. 288–292, 1965.

[6] R. Venkataramanan, V. N. Swamy, and K. Ramchandran, "Efficient interactive algorithms for file synchronization under general edits," *arXiv preprint arXiv:1310.2026*, 2013.

[7] S. M. Yazdi and L. Dolecek, "Synchronization from deletions through interactive communication," in *Proc. IEEE Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC)*, 2012, pp. 66–70.

[8] N. Bitouze and L. Dolecek, "Synchronization from insertions and deletions under a non-binary, non-uniform source," in *Proc. IEEE Int. Symp. on Information Theory Proceedings (ISIT)*, 2013, pp. 2930–2934.

[9] N. Ma, K. Ramchandran, and D. Tse, "Efficient file synchronization: A distributed source coding approach," in *Proc. IEEE Int. Symp. on Information Theory Proceedings (ISIT)*, 2011, pp. 583–587.

[10] ——, "A compression algorithm using mis-aligned side-information," in *Proc. IEEE Int. Symp. on Information Theory Proceedings (ISIT)*, 2012, pp. 16–20.

[11] L. Su and O. Milenkovic, "Synchronizing rankings via interactive communication," in *Proc. IEEE Int. Symp. on Information Theory Proceedings (ISIT)*, 2014, pp. 1056–1060.

[12] M. C. Davey and D. J. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 687–698, 2001.

[13] S. E. Rouayheb, S. Goparaju, H. M. Kiah, and O. Milenkovic, "Synchronizing edits in distributed storage networks," *arXiv preprint arXiv:1409.1551*, 2014.

[14] Y. Kanoria and A. Montanari, "On the deletion channel with small deletion probability," in *Proc. IEEE Int. Symp. on Information Theory Proceedings (ISIT)*, 2010, pp. 1002–1006.

[15] V. I. Levenshtein, "Efficient reconstruction of sequences from their subsequences or supersequences," *Journal of Combinatorial Theory, Series A*, vol. 93, no. 2, pp. 310–332, 2001.

[16] V. Levenshtein, "Bounds for deletion/insertion correcting codes," in *Proc. IEEE Int. Symp. on Information Theory Proceedings (ISIT)*, 2002, p. 370.

[17] Y. Kanoria and A. Montanari, "Optimal coding for the binary deletion channel with small deletion probability," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6192–6219, 2013.

[18] Q. Wang, V. Cadambe, S. Jaggi, M. Schwartz, and M. Médard, "File updates under random/arbitrary insertions and deletions," *arXiv preprint arXiv:1502.07830*, 2015.

[19] E. Ukkonen, "On approximate string matching," in *Foundations of Computation Theory*. Springer, 1983, pp. 487–495.