Fifty-Fifth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
October 3-6, 2017

Reconciling Similar Sets

Ryan Gabrys*, Farzad Farnoud (Hassanzadeh)†

*Spawar Systems Center San Diego
ryan.gabrys@navy.mil

†University of Virginia
farzad@virginia.edu

***Abstract*—In this work, we study the problem of synchronizing two sets of data where the size of the symmetric difference between the sets is small and, in addition, the elements in the symmetric difference are related through the Hamming distance metric. In our initial work [16], we provided communication-efficient methods for this problem when the elements in the symmetric difference were within a given Hamming distance. We now extend that work by considering sets whose symmetric difference consists of many blocks, each composed of elements that are within a given Hamming distance. We develop new encoding and decoding algorithms to address the complexities arising from allowing multiple difference blocks.**

Keywords. Distributed databases, Coding theory

I. Introduction

Suppose two hosts, A and B, each have a set of length-n binary strings. Let $\mathcal{S}^A$ denote the set of strings on Host A and let $\mathcal{S}^B$ denote the set of strings on Host B. The ***set reconciliation*** problem is to determine the minimum information that must be sent from Host A to Host B with a single round of communication so that Host B can compute their symmetric difference $\mathcal{S}^A \triangle \mathcal{S}^B = (\mathcal{S}^A \setminus \mathcal{S}^B) \cup (\mathcal{S}^B \setminus \mathcal{S}^A)$ where $|\mathcal{S}^A \triangle \mathcal{S}^B| \leq t$.

This problem has been the subject of study in many works such as [6], [7], [8], [10], [13], and [14]. The work in [6] provides an approach to set reconciliation that uses polynomial interpolation. In [7] and [8], coding schemes were studied that were based upon error-correcting codes and polynomial interpolation. In [10] and [13] algorithms based upon Bloom filters were considered. In [15], the authors take into account the similarities between objects; however, the problem studied is to minimize the earth movers distance between two sets given an upper bound on the communication cost.

In this paper, we consider a variant of the traditional set reconciliation problem whereby the elements in the symmetric difference $\mathcal{S}^A \triangle \mathcal{S}^B$ are related. In particular, we consider the setup where this symmetric difference can be partitioned into *blocks* such that elements in each of these blocks are within a certain Hamming distance of each other. Our goal will be to design transmission schemes that minimize the amount of information exchanged between two hosts.

This model is motivated by the scenario where two hosts are storing a large number of (potentially large) documents, and where information is never deleted so that each database contains many different versions of the documents. The contribution of this work will be to consider coding schemes for reconciling sets of related strings that extend the work in [16], where the symmetric difference consisted only of one block. The paper is organized as follows. In Section II, we formally define our problem and introduce some useful notation. Section III provides an overview of the encoding/decoding algorithm along with an overview of the main ideas. The encoding process is described more formally in Section IV and the decoding process is given in Section V. We note that due to lack of space, many of the proofs are deferred to an extended version of the work.

II. Model and Preliminaries

For two strings $\boldsymbol{x}, \boldsymbol{y} \in GF(2)^n$, let $d_H(\boldsymbol{x}, \boldsymbol{y})$ denote their Hamming distance. We denote the Hamming weight of $\boldsymbol{x}$ as $\mathrm{wt}(\boldsymbol{x})$. For a set $I \subseteq [n]$, and a vector $\boldsymbol{x} \in \mathrm{GF}(2)^n$, let $\boldsymbol{x}_I$ denote the vector that results by discarding the components of $\boldsymbol{x}$ outside I. For example, if $\boldsymbol{x} = (1,0,1,0)$, then $\boldsymbol{x}_{\{1,3\}} = (1,1)$. For a set of vectors $S \subseteq \mathrm{GF}(2)^n$, let S_I denote the set of vectors that results from discarding the components of each element in S outside I.

The aim of this paper is to describe an approach to synchronize two sets of data $\mathcal{S}^A$ and $\mathcal{S}^B$ where the symmetric difference between the sets has the following structure. For given t, h, and ℓ,

$$\mathcal{S}^A \triangle \mathcal{S}^B = \bigcup_{i=1}^{j} \mathcal{B}_i,$$

where $\mathcal{B}_i = \{\boldsymbol{x}_{i,1}, \dots, \boldsymbol{x}_{i,k_i}\}$ and the following hold:

1) $j \leq t$;
2) for $1 \leq i \leq j$, $k_i \leq h$;
3) for any $\boldsymbol{u}, \boldsymbol{w} \in \mathcal{B}_i$, we have $d_H(\boldsymbol{u}, \boldsymbol{w}) \leq \ell$; and
4) $\exists I \subseteq [n]$ such that:
    a) For any $i_1 \neq i_2$ and any $\boldsymbol{x} \in \mathcal{B}_{i_1}, \boldsymbol{y} \in \mathcal{B}_{i_2}$, we have $\boldsymbol{x}_I \neq \boldsymbol{y}_I$; and
    b) For any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{B}_i$, we have $\boldsymbol{x}_I = \boldsymbol{y}_I$.

Each set $\mathcal{B}_i$ is referred to as a ***(difference) block***.

If the sets $\mathcal{S}^A$ and $\mathcal{S}^B$ satisfy the conditions $1) - 4)$, then $\mathcal{S}^A$, $\mathcal{S}^B$ are called (t, h, ℓ)-sets. We note that our original definition for (t, h, ℓ)-sets in [16] did not require condition 4). The reason for including it here is that it is used during the encoding/decoding to group the elements in $S^A \triangle S^B$. For the case where $t = 1$, which was considered [16], there exists at most one difference block and so no grouping was required.

Assuming that the elements of the symmetric difference are chosen at random but with the constraint that conditions 1)-

978-1-5386-3266-6/17/$31.00 ©2017 IEEE

3) are satisfied, a simple argument shows that condition 4) is violated with probability at most

$$\frac{th^2\ell|I|}{n}+\frac{t^2h^2}{2^{|I|}}$$

which approaches 0 for $|I|=\lg(n)$ provided that $t^2h^2\ell=o(n/\lg n)$. Furthermore, given the setup where database documents are being synchronized, the set I could be derived from the document's unique identifier for instance. Thus, the (t,h,ℓ)-sets considered here could arise in several different ways.

Before continuing, we provide an example of a (t,h,ℓ)-set.

Example 1. *Suppose* $\mathcal{S}^A,\mathcal{S}^B\in\mathrm{GF}(2)^5$*, where*

$$\mathcal{S}^A=\{(0,0,0,0,0),(1,0,1,1,1)\},$$
$$\mathcal{S}^B=\{(0,0,0,0,0),(1,1,0,0,1)\}.$$

Then we say that $(\mathcal{S}^A,\mathcal{S}^B)$ *are* $(1,2,3)$*-sets since*

$$\mathcal{S}^A\triangle\mathcal{S}^B=\{(1,0,1,1,1),(1,1,0,0,1)\},$$

which can be decomposed into 1 *set of size* 2 *whereby the Hamming distance between any two elements is at most* 3*. Notice here that* $I=\{1,5\}$*.*

III. Overview and Main Ideas

In this section, we give an overview of our method with a fair amount of detail, and postpone the formal presentation of the encoding and decoding algorithms to Sections IV and V, respectively. Similar to the algorithm for $(1,h,\ell)$-sets from [16], the process of synchronizing (t,h,ℓ)-sets will be broken down into 2 main stages:

Stage 1) Determine the differences between the elements in the symmetric difference.

Stage 2) Recover the elements in the symmetric difference.

Notice that under our setup, there exists a set $\mathsf{Cen}\subseteq\mathrm{GF}(2)^n$ of size at most t containing one element $\boldsymbol{c}_i$ from each difference block $\mathcal{B}_i$ such that for each i and any $\boldsymbol{y}\in\mathcal{B}_i$, we have $d_H(\boldsymbol{y},\boldsymbol{c}_i)\le\ell$. We refer to the set Cen as the ***center set*** and to each $\boldsymbol{c}_i$ as a ***block center***.

Our goal during Stage 1) will be to recover the differences between the elements in each block. To this end, we represent the information in the sets as length-N vectors over $\mathrm{GF}(Q)$ where $\mathrm{GF}(Q)$ has characteristic two, denoted $\boldsymbol{z}_1=(z_{1,1},\dots,z_{1,N})\in\mathrm{GF}(Q)^N$, where the values of Q and N are chosen to ensure the existence of two maps

$$M:\mathrm{GF}(2)^n\to[N],$$
$$f:\mathrm{GF}(2)^{|I|}\to\mathrm{GF}(Q)$$

with certain properties that will be described shortly. Both maps are also used in the second stage of synchronization.

The map M is a function that assigns to each $\boldsymbol{x}\in\mathcal{S}$ (where $\mathcal{S}=\mathcal{S}^A$ or $\mathcal{S}=\mathcal{S}^B$) a position in $\boldsymbol{z}_1$, that is, $M(\boldsymbol{x})\in[N]$. This assignment satisfies the following property.

Property 1. *The map* M *is such that if* $d_H(\boldsymbol{x}_1,\boldsymbol{x}_2)\le 2\ell$*, then* $\boldsymbol{x}_1$ *and* $\boldsymbol{x}_2$ *are mapped to different positions, i.e.,*

$$M(\boldsymbol{x}_1)\ne M(\boldsymbol{x}_2).\qquad(1)$$

As a result, no two elements belonging to the same difference block are mapped to the same position.

The map M can be obtained using the parity check matrix $H_M\in\mathrm{GF}(2)^{r\times n}$ for a binary code that has length n, minimum distance $2\ell+1$, and redundancy r, where $N=2^r$. For a number $k\in[N]$, let $B(k)$ denote the r-bit binary representation of $k-1$. For instance, for $r=3$, $B(5)=(0,0,1)$. Clearly the map B is invertible. For a vector $\boldsymbol{x}\in\mathrm{GF}(2)^n$, we define M as:

$$M(\boldsymbol{x})=B^{-1}(H_M\cdot\boldsymbol{x}).$$

Since the minimum distance of the code is $2\ell+1$, the function M has Property 1. The decoder $\mathcal{D}_M:\mathrm{GF}(2)^r\to\mathrm{GF}(2)^n$ for the code is such that for any $\boldsymbol{v}$ with weight at most ℓ, $\mathcal{D}_M(H_M\cdot\boldsymbol{v})=\boldsymbol{v}$.

Let $\boldsymbol{s}_{M,k}=B(k)$ and note that $\boldsymbol{s}_{M,k}\in\mathrm{GF}(2)^r$ and that $M(\boldsymbol{x})=k$ iff $\boldsymbol{s}_{M,k}=H_M\cdot\boldsymbol{x}$. Given the decoder $\mathcal{D}_M$, we are now able to define a function $E:[N]\times[N]\to\mathrm{GF}(2)^n$ that will be used during the decoding. Namely, we let E be defined as

$$E(i,j)=\mathcal{D}_M(\boldsymbol{s}_{M,i}+\boldsymbol{s}_{M,j})=\mathcal{D}_M(B(i)+B(j)).$$

We now turn to discussing the map f. For now, we assume this map has the following property. In Section IV, we show how to construct such maps.

Property 2. *The map* f *is an* invertible *function such that*

$$\sum_{\boldsymbol{x}\in\mathcal{X}:\mathcal{X}\subseteq\mathrm{GF}(2)^{|I|},|\mathcal{X}|\le 2t} f(\boldsymbol{x})\ne 0.\qquad(2)$$

For a subset $\mathcal{S}\subseteq\mathrm{GF}(2)^n$ (in particular $\mathcal{S}=\mathcal{S}^A$ or $\mathcal{S}=\mathcal{S}^B$), let $\mathcal{S}_{M,i}=\{\boldsymbol{x}\in\mathcal{S}:M(\boldsymbol{x})=i\}$. The vector $\boldsymbol{z}_1=(z_{1,j})_{j\in[N]}$ is defined as

$$z_{1,j}=\sum_{\boldsymbol{x}\in\mathcal{S}_{M,j}} f(\boldsymbol{x}_I).\qquad(3)$$

The result of Property 2 is that for $d\le t$ and $\boldsymbol{x}_I^{(1)},\boldsymbol{x}_I^{(2)},\dots,\boldsymbol{x}_I^{(d)}\in\mathcal{S}_{M,j}$, we can recover $\boldsymbol{x}_I^{(1)},\boldsymbol{x}_I^{(2)},\dots,\boldsymbol{x}_I^{(d)}$ from their sum $z_{1,j}=\sum_{\boldsymbol{x}\in\mathcal{S}_{M,j}} f(\boldsymbol{x}_I)$.

Let $\boldsymbol{z}_1^A$ and $\boldsymbol{z}_1^B$ be the result of computing $\boldsymbol{z}$ according to (3) on Host A and Host B, respectively. Furthermore, let $\dot{\boldsymbol{z}}=\boldsymbol{z}_1^A+\boldsymbol{z}_1^B$. Each host transmits a compressed version of its $\boldsymbol{z}_1$ vector to the other one and so each can then compute $\dot{\boldsymbol{z}}$. The effect of elements in $\mathcal{S}^A\cap\mathcal{S}^B$ are canceled out in $\dot{\boldsymbol{z}}$ since they contribute to both $\boldsymbol{z}_1^A$ and $\boldsymbol{z}_1^B$ and $\mathrm{GF}(Q)$ is an extension field of $\mathrm{GF}(2)$. Hence,

$$\dot{z}_j=\sum_{\boldsymbol{x}\in(\mathcal{S}^A\triangle\mathcal{S}^B)_{M,j}} f(\boldsymbol{x}_I).\qquad(4)$$

Given $\dot{\boldsymbol{z}}$, from the discussion following (3) and the invertibility of f, for each i we can recover the set $\Big\{M(\boldsymbol{x}):\boldsymbol{x}\in\mathcal{B}_i\Big\}$. Using this information, we can then identify the differences between the elements of each $\mathcal{B}_i$, which is the goal of Stage 1). In particular, consider $\boldsymbol{x}_1,\boldsymbol{x}_2\in\mathcal{B}_i$ such that

$j_1 = M(\boldsymbol{x}_1)$ and $j_2 = M(\boldsymbol{x}_2)$. Let $\boldsymbol{e} = \boldsymbol{x}_1 + \boldsymbol{x}_2$ and note that $\mathrm{wt}(e) \leq \ell$. Now,

$$\begin{aligned} E(j_1, j_2) &= \mathcal{D}_M(\boldsymbol{s}_{M,j_1} + \boldsymbol{s}_{M,j_2}) \\ &= \mathcal{D}_M(H_M \cdot \boldsymbol{x}_1 + H_M \cdot \boldsymbol{x}_2) \\ &= \mathcal{D}_M(H_M \cdot \boldsymbol{e}) \\ &= \boldsymbol{e}. \end{aligned} \tag{5}$$

So based on the preceding discussion, from $\dot{\boldsymbol{z}}$, we can find $\boldsymbol{x}_I$ and $f(\boldsymbol{x}_I)$ for each $\boldsymbol{x} \in \mathcal{S}^A \triangle \mathcal{S}^B$, the number of difference blocks $\mathcal{B}$, the number of elements in each block, and the differences between any two elements in each block.

As mentioned earlier, the hosts do not transmit $\boldsymbol{z}_1^A$ and $\boldsymbol{z}_1^B$ but rather a compressed version of these vectors. Let H_C be a parity check matrix for a code $\mathcal{C}_C$ over $\mathrm{GF}(Q)$ of length N with minimum distance $2th + 1$. Each host computes $\boldsymbol{w}_1 = H_C \cdot \boldsymbol{z}_1$ (resulting in $\boldsymbol{w}_1^A$ and $\boldsymbol{w}_1^B$) and transmits it to the other host. So each host can compute $\dot{\boldsymbol{w}} = \boldsymbol{w}_1^A + \boldsymbol{w}_1^B$. Note that $\dot{\boldsymbol{w}} = H_C \cdot \dot{\boldsymbol{z}}$. Since $\mathrm{wt}(\dot{\boldsymbol{z}}) \leq th$, the hosts can find $\dot{\boldsymbol{z}}$ using a decoder $\mathcal{D}_C$ for the code H_C.

For Stage 2), the idea will be to use the differences between a center set and the remaining elements to encode (and subsequently decode) the elements in the center set only. During the decoding, we will produce the symmetric difference given knowledge of a center set and the differences.

In this stage, we represent our information using the vectors

$$\begin{aligned} \boldsymbol{z}_2^{(0)} &= (z_{2,1}^{(0)}, \ldots, z_{2,N}^{(0)}), \\ \boldsymbol{z}_2^{(1)} &= (z_{2,1}^{(1)}, \ldots, z_{2,N}^{(1)}), \\ &\vdots \\ \boldsymbol{z}_2^{(t-1)} &= (z_{2,1}^{(t-1)}, \ldots, z_{2,N}^{(t-1)}) \in \mathrm{GF}(2^{n-|I|})^N. \end{aligned}$$

For shorthand, let $\bar{n} = n - |I|$ and $\bar{I} = [n] \setminus I$, so that $\boldsymbol{x}_{\bar{I}} = \boldsymbol{x}_{([n]\setminus I)}$. Suppose, as before, we are encoding the set $\mathcal{S}$, where $\mathcal{S} = \mathcal{S}^A$ or $\mathcal{S}^B$. We let

$$z_{2,j}^{(k)} = \sum_{\boldsymbol{x} \in \mathcal{S}_{M,j}} \left(f(\boldsymbol{x}_I)\right)^{2^k} \cdot \boldsymbol{x}_{\bar{I}}$$

for $k \in \{0, 1, \ldots, t-1\}$. For this stage we implicitly make use of a bijection between $\mathrm{GF}(2^{\bar{n}})$ and $\mathrm{GF}(2)^{\bar{n}}$. Further, we assume $\mathrm{GF}(Q) \subseteq \mathrm{GF}(2^{\bar{n}})$.

We need another matrix to fully describe the encoding process. Let H_F be a $t \times N$ matrix with elements from $\mathrm{GF}(R)$, where $R \geq N^{(2^t - 1)h}$, such that the following property holds.

Property 3. *For any submatrix H_F' of H_F, consisting of any $c \leq th$ nonzero columns from H_F, and for any $\boldsymbol{s} \in \mathrm{GF}(2^{\bar{n}})^t$, there exists at most one choice of a vector $\boldsymbol{v} \in \mathrm{GF}(2^{\bar{n}})^c$ over $\mathrm{GF}(2^{\bar{n}})$ with $\mathrm{rk}(\boldsymbol{v}) \leq t$ that satisfies*

$$H_F' \cdot \boldsymbol{v} = \boldsymbol{s}. \tag{6}$$

Here $\mathrm{rk}(\boldsymbol{v})$ denotes the rank of $\boldsymbol{v}$ over $\mathrm{GF}(2)$ if $\boldsymbol{v}$ is interpreted as an $\bar{n} \times c$ matrix.

Given the $t \times N$ matrix H_F, Host A constructs

$$\boldsymbol{w}_2^A = (\boldsymbol{w}_2^{A,(0)}, \ldots, \boldsymbol{w}_2^{A,(t-1)})$$

where $\boldsymbol{w}_2^{A,(k)} = H_F \cdot \boldsymbol{z}_2^{A,(k)}$, and transmits it to Host B.

We now turn to describe decoding in Stage 2). For clarity of presentation, we assume that the vector $\dot{\boldsymbol{z}} = \boldsymbol{z}_1^A + \boldsymbol{z}_1^B$ from Stage 1) has the following nonzero elements:

$$\begin{aligned} \dot{\boldsymbol{z}}_{j_1} &= \boldsymbol{\sigma}_1, \\ \dot{\boldsymbol{z}}_{j_2} &= \boldsymbol{\sigma}_1 + \boldsymbol{\sigma}_2, \\ \dot{\boldsymbol{z}}_{j_3} &= \boldsymbol{\sigma}_2, \end{aligned}$$

where $\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2 \in \mathrm{GF}(2)^{|I|}$. Notice that under this setup, $t = 2$ and $h = 2$, so that the symmetric difference consists of two blocks, $\mathcal{B}_1$ and $\mathcal{B}_2$, each with two elements. Without loss of generality, suppose that $\mathcal{B}_1 = \{X, X + \boldsymbol{e}_1\}$ and $\mathcal{B}_2 = \{Y, Y + \boldsymbol{e}_2\}$, where

$$\begin{aligned} f(X_I) &= \boldsymbol{\sigma}_1, & M(X) &= j_1, \\ f((X + \boldsymbol{e}_1)_I) &= \boldsymbol{\sigma}_1, & M(X + \boldsymbol{e}_1) &= j_2, \\ f(Y_I) &= \boldsymbol{\sigma}_2, & M(Y) &= j_2, \\ f((Y + \boldsymbol{e}_2)_I) &= \boldsymbol{\sigma}_2, & M(Y + \boldsymbol{e}_2) &= j_3, \end{aligned}$$

and where $\mathrm{wt}(\boldsymbol{e}_1) \leq \ell$ and $\mathrm{wt}(\boldsymbol{e}_2) \leq \ell$. Also note that $(X + \boldsymbol{e}_1)_I = X_I$ and $(Y + \boldsymbol{e}_2)_I = Y_I$. At this point, we still do not know the values of X and Y, but from Stage 1) of the decoding we know the values of $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$:

$$\boldsymbol{e}_1 = E(j_1, j_2), \quad \boldsymbol{e}_2 = E(j_2, j_3). \tag{7}$$

Let $\ddot{\boldsymbol{z}}^{(k)} = \boldsymbol{z}_2^{A,(k)} + \boldsymbol{z}_2^{B,(k)}$ and $\ddot{\boldsymbol{w}}^{(k)} = \boldsymbol{w}_2^{A,(k)} + \boldsymbol{w}_2^{B,(k)}$. When decoding, each node can compute $\ddot{\boldsymbol{w}}^{(k)}$, which equals $H_F \cdot \ddot{\boldsymbol{z}}^{(k)}$.

Because we have mapped the elements in $\mathcal{S}^A \triangle \mathcal{S}^B$ to the same locations in both $\dot{\boldsymbol{z}}$ and $\ddot{\boldsymbol{z}}^{(k)}$, we know $\ddot{\boldsymbol{z}}^{(k)}$ has the following nonzero elements

$$\begin{aligned} \ddot{\boldsymbol{z}}_{j_1}^{(k)} &= \boldsymbol{\sigma}_1^{2^k} \cdot X_{\bar{I}}, \\ \ddot{\boldsymbol{z}}_{j_2}^{(k)} &= \boldsymbol{\sigma}_1^{2^k} \cdot (X + \boldsymbol{e}_1)_{\bar{I}} + \boldsymbol{\sigma}_2^{2^k} \cdot Y_{\bar{I}}, \\ \ddot{\boldsymbol{z}}_{j_3}^{(k)} &= \boldsymbol{\sigma}_2^{2^k} \cdot (Y + \boldsymbol{e}_2)_{\bar{I}}, \end{aligned}$$

for any k. At this point, we still do not know the values of X and Y but do know the values of $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$.

The rank $\mathrm{rk}(\ddot{\boldsymbol{z}}^{(k)})$ of $\ddot{\boldsymbol{z}}$ is at most 3. Our goal now is to decrease the rank of this vector to at most $t = 2$ so that we can use Property 3. To do so, from each block $\mathcal{B}_i$, we arbitrarily pick an element as the block center and as described below, we change every other appearance of an element of $\mathcal{B}_i$ in $\ddot{\boldsymbol{z}}^{(k)}$ to look like the block center. In our current illustration, we pick the element of $\mathcal{B}_1$ mapped to j_1 and the element of $\mathcal{B}_2$ mapped to j_2 as their respective centers, which we have named X and Y.

Let $\boldsymbol{u} \in \mathrm{GF}(2^{\bar{n}})$ be the all-zero vector except in position j_2, where $u_{j_2} = \boldsymbol{\sigma}_1^{2^k} \cdot (\boldsymbol{e}_1)_{\bar{I}}$. Notice here that we again implicitly use a bijective mapping between $\mathrm{GF}(2^{\bar{n}})$ and $\mathrm{GF}(2)^{\bar{n}}$. We initialize $\hat{S}^{(k)} = \ddot{\boldsymbol{w}}^{(k)} = \boldsymbol{w}_2^{A,(k)} + \boldsymbol{w}_2^{B,(k)}$ and update it by adding $H_F \cdot \boldsymbol{u}$ to it:

$$\begin{aligned} \hat{S}^{(k)} &\leftarrow \ddot{\boldsymbol{w}}^{(k)} + H_F \cdot \boldsymbol{u} \\ &= H_F \cdot (\ddot{\boldsymbol{z}}^{(k)} + \boldsymbol{u}). \end{aligned}$$

Note that the j_2th position of $\ddot{\boldsymbol{z}}^{(k)} + \boldsymbol{u}$, denoted $(\ddot{\boldsymbol{z}}^{(k)} + \boldsymbol{u})_{j_2}$, is

$$\begin{aligned}(\ddot{\boldsymbol{z}}^{(k)} + \boldsymbol{u})_{j_2} &= \ddot{z}^{(k)}_{j_2} + u_{j_2} \\ &= \left(\boldsymbol{\sigma}_1^{2^k} \cdot (X + \boldsymbol{e}_1)_{\bar{I}} + \boldsymbol{\sigma}_2^{2^k} \cdot Y_{\bar{I}}\right) + \left(\boldsymbol{\sigma}_1^{2^k} \cdot (\boldsymbol{e}_1)_{\bar{I}}\right) \\ &= \boldsymbol{\sigma}_1^{2^k} \cdot X_{\bar{I}} + \boldsymbol{\sigma}_2^{2^k} \cdot Y_{\bar{I}}.\end{aligned}$$

So now both elements of $\mathcal{B}_1$ contribute a term of the form $\boldsymbol{\sigma}_1^{2^k} \cdot X_{\bar{I}}$.

We update $\hat{S}^{(k)}$ again by letting $\hat{S}^{(k)} \leftarrow \hat{S}^{(k)} + H_F \cdot \boldsymbol{u}'$, where $u'_{j_3} = \boldsymbol{\sigma}_2^{2^k} \cdot (E(j_2, j_3))_{\bar{I}}$, so that

$$(\ddot{\boldsymbol{z}}^{(k)} + \boldsymbol{u}')_{j_3} = \boldsymbol{\sigma}_2^{2^k} \cdot Y_{\bar{I}}.$$

and that

$$\hat{S}^{(k)} = H_F \cdot V^{(k)},$$

where the non-zero entries in $V^{(k)}$ are contained within the set

$$U^{(k)} = \{\boldsymbol{\sigma}_1^{2^k} \cdot X_{\bar{I}} + \boldsymbol{\sigma}_2^{2^k} \cdot Y_{\bar{I}},\ \boldsymbol{\sigma}_1^{2^k} \cdot X_{\bar{I}},\ \boldsymbol{\sigma}_2^{2^k} \cdot Y_{\bar{I}}\}.$$

Notice that $\text{rk}(V^{(k)}) \leq 2$ whereas $\text{rk}(\ddot{\boldsymbol{z}}) \leq 3$. Thus it is possible now to use Property 3 to recover $V^{(k)}$ for $k \in \{0, 1, 2, \ldots, t-1\}$. In particular, given:

$$\boldsymbol{\sigma}_1^{2^k} \cdot X_{\bar{I}} + \boldsymbol{\sigma}_2^{2^k} \cdot Y_{\bar{I}}$$

for $k \in \{0, 1\}$ along with knowledge of $\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2$ (which we recovered from the first stage of the decoding using the vector $\dot{\boldsymbol{z}}$), we can recover $X_{\bar{I}}$ and $Y_{\bar{I}}$, which allows us to determine the center set $\{X, Y\}$. Then, with X, Y and $\dot{\boldsymbol{z}} = \boldsymbol{z}_1^A + \boldsymbol{z}_1^B$ can recover $\mathcal{B}_1 = \{X, X + \boldsymbol{e}_1\}$ and $\mathcal{B}_2 = \{Y, Y + \boldsymbol{e}_1\}$, so we are able to reconstruct the set $\mathcal{S}^A \triangle \mathcal{S}^B$.

IV. Encoding

In this section, we formally state the encoding algorithm. We present the encoding procedure for Host A, but the same applies to Host B as well.

Recall that for a set S, we let $S_{M,i} = \{\boldsymbol{x} \in S : M(\boldsymbol{x}) = i\}$ and that H_C is a parity check matrix for a code $\mathcal{C}_C$ over $\text{GF}(Q)$ of length N with minimum distance $2th + 1$. The encoding is as follows.

1) Let $\boldsymbol{z}_1^A = (z_{1,1}^A, z_{1,2}^A, \ldots, z_{1,N}^A) \in \text{GF}(Q)^N$ with

$$z_{1,j}^A = \sum_{\boldsymbol{x} \in \mathcal{S}_{M,j}^A} f(\boldsymbol{x}_I),$$

and set

$$\boldsymbol{w}_1^A = H_C \cdot \boldsymbol{z}_1^A.$$

2) Let $\boldsymbol{z}_2^{A,(k)} = (z_{2,1}^{A,(k)}, z_{2,2}^{A,(k)}, \ldots, z_{2,N}^{A,(k)}) \in \text{GF}(2^{\bar{n}})^N$ with

$$z_{2,j}^{A,(k)} = \sum_{\boldsymbol{x} \in \mathcal{S}_{M,j}^A} \left(f(\boldsymbol{x}_I)\right)^{2^k} \cdot \boldsymbol{x}_{\bar{I}}$$

for $k \in \{0, 1, \ldots, t-1\}$ and set

$$\boldsymbol{w}_2^A = (H_F \cdot \boldsymbol{z}_2^{A,(0)}, H_F \cdot \boldsymbol{z}_2^{A,(1)}, \ldots, H_F \cdot \boldsymbol{z}_2^{A,(t-1)}).$$

The information $\boldsymbol{w}^A = (\boldsymbol{w}_1^A, \boldsymbol{w}_2^A)$ is then transmitted to Host B. The size of $(\boldsymbol{w}_1^A, \boldsymbol{w}_2^A)$ is given in Claim 3.

We now provide constructions and some proofs for the maps introduced in earlier in this section.

Claim 1. *There exists a map $M : \text{GF}(2)^n \to [N]$ satisfying Property 1 with $N = n^\ell$.*

Proof: Let $\mathcal{M}$ be an extended BCH code of length n and redundancy $r = \ell \lg n$. This code has minimum distance at least $2\ell + 1$. Furthermore, let H_M be a parity check matrix of the code $\mathcal{M}$. Recall M is such that $M(\boldsymbol{x}) = B^{-1}(H_M \cdot \boldsymbol{x})$, where $B(k)$ is the r-bit binary representation of $k-1$. Suppose $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \text{GF}(2)^n$ are such that $d_H(\boldsymbol{x}_1, \boldsymbol{x}_2) \leq 2\ell$. So we can write $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{e}_1$ and $\boldsymbol{x}_2 = \boldsymbol{x}_0 + \boldsymbol{e}_2$, where $\text{wt}(\boldsymbol{e}_1) \leq \ell$ and $\text{wt}(\boldsymbol{e}_2) \leq \ell$. By the invertibility of B, $M(\boldsymbol{x}_1) \neq M(\boldsymbol{x}_2)$ if and only if $H_M \cdot \boldsymbol{e}_1 \neq H_M \cdot \boldsymbol{e}_2$, which is the case since the minimum distance of $\mathcal{M}$ is at least $2\ell + 1$. ■

The next claim follows using similar logic by using a parity check matrix with minimum distance $2t+1$ along with a hash function similar to [2].

Claim 2. *There exists a map $f : \text{GF}(2)^{|I|} \to \text{GF}(Q)$ satisfying Property 2 where Q is equal to the smallest power of two greater than $2^{2t|I|}$.*

We can apply the previous claim to determine the size of $(\boldsymbol{w}_1^A, \boldsymbol{w}_2^A)$.

Claim 3. *Assuming $|I| < \log n$, $\text{GF}(Q) \subseteq \text{GF}(2^{\bar{n}})$, and $N < 2^{\bar{n}}$, $(\boldsymbol{w}_1, \boldsymbol{w}_2)$ requires at most*

$$t^2 n + 4ht^2 \log n + 2ht\ell \log n$$

bits of information.

Recall that if the method from [6] were used roughly thn bits of information exchange would be required so that the method described here requires less information exchange when $t \ll h$ and n large enough. Recall that in our original work [16], our method for synchronizing $(1, h, \ell)$-sets $n + (h-1)\ell(\log n + 1)$.

We let the matrix H_F be a parity check matrix of the following form:

$$H_F = \begin{bmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_N \\ \gamma_1^2 & \gamma_2^2 & \cdots & \gamma_N^2 \\ \gamma_1^4 & \gamma_2^4 & \cdots & \gamma_N^4 \\ \vdots & & \ddots & \\ \gamma_1^{2^{t-1}} & \gamma_1^{2^{t-1}} & \cdots & \gamma_N^{2^{t-1}} \end{bmatrix},$$

where any subset of elements from $\{\gamma_1, \gamma_2, \ldots, \gamma_N\}$ of size $(2^t - 1)h$ is linearly independent over $\text{GF}(2)$. Hence, we have that $\gamma_i \in \text{GF}(R)$ where $R \geq N^{(2^t-1)h}$ (this statement follows using ideas similar to Claim 1). Recall $\bar{n} = n - |I|$. We assume $\text{GF}(R) \subseteq \text{GF}(2^{\bar{n}})$. For a vector $\boldsymbol{v} \in \text{GF}(2^{\bar{n}})^N$ and an element $\sigma \in \text{GF}(2^{\bar{n}})$, let $\sigma(\boldsymbol{v}) \subseteq [N]$ return the set of positions in $\boldsymbol{v}$ that have value σ. Furthermore, recall that $\text{rk}(\boldsymbol{v})$ denotes the rank of $\boldsymbol{v}$ over $\text{GF}(2)$ if $\boldsymbol{v}$ is interpreted as a $\bar{n} \times N$ matrix over $\text{GF}(2)$. Finally, let $dis(\boldsymbol{v})$ denote the set of non-zero elements in $\boldsymbol{v}$ (the "distinct" elements in $\boldsymbol{v}$). The next lemma can be used to show Property 3 holds.

Lemma 1. *Suppose $\boldsymbol{x} \in \mathrm{GF}(2^{\bar{n}})^N$ is such that $\mathrm{rk}(\boldsymbol{x}) \leq t$. If, for every non-zero value $\sigma \in \boldsymbol{x}$, $|\sigma(\boldsymbol{x})| \leq h$, then*

$$H_F \cdot \boldsymbol{x} \neq 0.$$

Proof: Suppose $x_{i_1}, x_{i_2}, \ldots, x_{i_{dis(\boldsymbol{x})}}$ are elements in $\boldsymbol{x}$ that have distinct values. Then can write:

$$H_F \cdot \boldsymbol{x} = \sum_{j=1}^{dis(\boldsymbol{x})} x_{i_j} \cdot \left(\sum_{k \in x_{i_j}(\boldsymbol{x})} \begin{bmatrix} \gamma_{i_k} \\ \gamma_{i_k}^2 \\ \vdots \\ \gamma_{i_k}^{2^{t-1}}. \end{bmatrix} \right)$$

$$= \sum_{j=1}^{dis(\boldsymbol{x})} x_{i_j} \begin{bmatrix} \sum_{k \in x_{i_j}(\boldsymbol{x})} \gamma_{i_k} \\ (\sum_{k \in x_{i_j}(\boldsymbol{x})} \gamma_{i_k})^2 \\ \vdots \\ (\sum_{k \in x_{i_j}(\boldsymbol{x})} \gamma_{i_k})^{2^{t-1}} \end{bmatrix}.$$

Because $\mathrm{rk}(\boldsymbol{x}) \leq t$, clearly $dis(\boldsymbol{x}) \leq 2^t - 1$. Since $|x_{i_j}(\boldsymbol{x})| \leq h$ and any collection of $(2^t - 1)h$ elements from $\{\gamma_1, \ldots, \gamma_N\}$ are linearly independent over GF(2), it follows $\sum_{k \in x_{i_j}(\boldsymbol{x})} \gamma_{i_k} \neq 0$. Using similar reasoning we have that the elements

$$\{ \sum_{k \in x_{i_1}(\boldsymbol{x})} \gamma_{i_k}, \sum_{k \in x_{i_2}(\boldsymbol{x})} \gamma_{i_k}, \ldots, \sum_{k \in x_{i_{dis(\boldsymbol{x})}}(\boldsymbol{x})} \gamma_{i_k} \}$$

are also linearly independent over $GF(2)$. Let H be the $t \times dis(\boldsymbol{x})$ matrix

$$H = \begin{bmatrix} \sum_{k \in x_{i_1}(\boldsymbol{x})} \gamma_{i_k} & \cdots & \sum_{k \in x_{i_{dis(\boldsymbol{x})}}(\boldsymbol{x})} \gamma_{i_k} \\ (\sum_{k \in x_{i_1}(\boldsymbol{x})} \gamma_{i_k})^2 & \cdots & (\sum_{k \in x_{i_{dis(\boldsymbol{x})}}(\boldsymbol{x})} \gamma_{i_k})^2 \\ \vdots & \ddots & \vdots \\ (\sum_{k \in x_{i_1}(\boldsymbol{x})} \gamma_{i_k})^{2^{t-1}} & \cdots & (\sum_{k \in x_{i_{dis(\boldsymbol{x})}}(\boldsymbol{x})} \gamma_{i_k})^{2^{t-1}} \end{bmatrix}.$$

Let $\boldsymbol{x}' = (x_{i_1}, x_{i_2}, \ldots, x_{i_{dis(\boldsymbol{x})}})$. Note that if $H_F \cdot \boldsymbol{x} = \boldsymbol{0}$, then $H \cdot \boldsymbol{x}' = \boldsymbol{0}$. where clearly $\mathrm{rk}(\boldsymbol{x}') = \mathrm{rk}(\boldsymbol{x})$. However, if $H \cdot \boldsymbol{x}' = \boldsymbol{0}$, $\mathrm{rk}(\boldsymbol{x}) = \mathrm{rk}(\boldsymbol{x}') \geq t + 1$ from [3], which is a contradiction. ■

V. Decoding

In this section, we present the decoding algorithm. Let $\mathcal{D}_C$ be a decoder for the code with the parity check matrix H_C so that for any vector $\boldsymbol{v} \in \mathrm{GF}(Q)^N$ where $\mathrm{wt}(\boldsymbol{v}) \leq th$, $\mathcal{D}_C(H_C \cdot \boldsymbol{v}) = \boldsymbol{v}$. Suppose $\boldsymbol{w}^B = (\boldsymbol{w}_1^B, \boldsymbol{w}_2^B)$ is the result of performing steps 1) and 2) in the encoding section using the set $\mathcal{S}^B$ (rather than $\mathcal{S}^A$) where $\boldsymbol{w}_2^B = (\boldsymbol{w}_2^{B,(0)}, \ldots, \boldsymbol{w}_2^{B,(t-1)})$. We will also make use of a map $F : \mathrm{GF}(2)^{|I|} \times \mathrm{GF}(2^{\bar{n}}) \to \mathrm{GF}(2)^n$ that outputs a length n binary vector $\boldsymbol{x}$ where $\boldsymbol{x}_I$ is equal to the first argument and $\boldsymbol{x}_{\bar{I}}$ is equal to the second argument. In the algorithm below, $\widehat{\mathsf{Cen}}$ contains the image of a center set under the map f. For $i \in [N]$, the sets D_i and G_i contain elements in the same difference block. We now detail how to recover $\mathcal{S}^A \triangle \mathcal{S}^B$ given $\boldsymbol{w}^A, \boldsymbol{w}^B$.

1) Let $\dot{\boldsymbol{z}} = \mathcal{D}_C(\boldsymbol{w}_1^A + \boldsymbol{w}_1^B) = (\dot{z}_1, \ldots, \dot{z}_N) \in \mathrm{GF}(Q)^N$.
2) For $i \in [N]$, perform the following procedure to generate the sets $D_1, \ldots, D_N \subseteq \mathrm{GF}(Q)$:
   a) If $\dot{z}_i = 0$, then set $D_i = \emptyset$.
   b) Otherwise if $\dot{z}_i = \zeta \in \mathrm{GF}(Q)$, then set $D_i = \{\zeta_1, \zeta_2, \ldots, \zeta_T\} \subseteq \mathrm{GF}(Q)$ where $\sum_{j=1}^T \zeta_i = \zeta$, $T \leq t$.
3) Copy $D_1, \ldots, D_N$ to $G_1, \ldots, G_N$ where $G_1 = D_1, \ldots, G_N = D_N$.
4) For $k \in \{0, 1, \ldots, t-1\}$, set $\ddot{\boldsymbol{w}}^{(k)} = \boldsymbol{w}_2^{A,(k)} + \boldsymbol{w}_2^{B,(k)}$.
5) From $D_1, \ldots, D_N$ update $\ddot{\boldsymbol{w}}^{(0)}, \ldots, \ddot{\boldsymbol{w}}^{(t-1)}$ as follows:
   a) Initialize $i = 0$ and $\widehat{\mathsf{Cen}} = \emptyset$.
   b) Set $i \leftarrow i + 1$. If $i > N$ go to step 6).
   c) If $\exists D_j$ where $|D_i \cap D_j| \neq 0$, do the following:
      i) Let $\sigma \in D_i \cap D_j$.
      ii) For $k \in \{0, 1, \ldots, t-1\}$, update $\ddot{\boldsymbol{w}}^{(k)} = \ddot{\boldsymbol{w}}^{(k)} + H_F \cdot \boldsymbol{u}^{(k)}$ where $\boldsymbol{u}^{(k)}$ is zero except in position j where $u_j^{(k)} = E(i,j)_{\bar{I}} \cdot \sigma^{2^k}$.
      iii) Remove σ from D_j.
      iv) Add (σ, i) to $\widehat{\mathsf{Cen}}$. Repeat step 5c).
   d) If $\nexists D_j$ where $|D_i \cap D_j| > 0$, go to step 5b).
6) For $k \in \{0, 1, \ldots, t-1\}$, do the following: from $\ddot{\boldsymbol{w}}^{(k)}$, compute $\ddot{\boldsymbol{z}}^{(k)}$ such that $\ddot{\boldsymbol{w}}^{(k)} = H_F \cdot \ddot{\boldsymbol{z}}^{(k)}$ where the locations of the non-zero entries in $\ddot{\boldsymbol{z}}^{(k)}$ are equal to the locations of the non-zero entries in $\dot{\boldsymbol{z}}$, and $\mathrm{rk}(\ddot{\boldsymbol{z}}^{(k)}) \leq t$.
7) Initialize $\mathcal{F} = \emptyset$.
8) Add the center set to $\mathcal{F}$ by setting $i = 0$, and doing the following:
   a) Set $i \leftarrow i + 1$. If $i > N$, then go to step 9).
   b) If $G_i = \emptyset$, then go to step 8a).
   c) Suppose $G_i = \{\sigma_1, \sigma_2, \ldots, \sigma_T\}$ where $T \leq t$. Then let
   $$H_i = \begin{bmatrix} \sigma_1 & \sigma_2 & \ldots & \sigma_T \\ \sigma_1^2 & \sigma_2^2 & \ldots & \sigma_T^2 \\ \sigma_1^{2^{t-1}} & \sigma_2^{2^{t-1}} & \ldots & \sigma_T^{2^{t-1}} \end{bmatrix}.$$
   d) Define the $t \times 1$ vector $\boldsymbol{v} = (v_1, \ldots, v_t)$ so that $v_k = \ddot{z}_i^{(k)}$. Let $V = (V_1, \ldots, V_T) = H_i^{-1} \cdot \boldsymbol{v} \in \mathrm{GF}(2^{\bar{n}})^T$.
   e) For every $j \in T$ where $V_j \neq 0$, if $(\sigma_j, i) \in \widehat{\mathsf{Cen}}$ add $F(f^{-1}(\sigma_j), V_j)$ to $\mathcal{F}$. Otherwise, if $(\sigma_j, i) \notin \widehat{\mathsf{Cen}}$, let r be such that $(\sigma_j, r) \in \mathsf{Cen}$. Add $F(f^{-1}(\sigma_j), V_j) + E(r, i)$ to $\mathcal{F}$.
   f) Go to step 8a).

The following theorem can be proven using the ideas introduced in Section III.

Theorem 1. *At the end of the decoding, $\mathcal{F} = \mathcal{S}^A \triangle \mathcal{S}^B$.*

References

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

[2] M. Fredman, M. Komlos, E. Szemeredi," Storing a sparse table with $O(1)$ worst case access time," *Journal of ACM*. vol. 31, no. 3, pp. 538-544, 1984.

[3] È. M. Gabidulin, "Theory of codes with maximum rank distance," *Probl. Peredachi Inf.* vol. 21, no. 1, pp. 3-16, 1985.

[4] R. J. Lipton, "Efficient checking of computations," *STACS*, 1990.

[5] D. B. West, *Introduction to graph theory*, Prentice Hall Upper Saddle River, 2001, vol. 2.

[6] Y. Minsky and A. Trachtenberg, “Practical set reconciliation,” *Tech. Rep.*, Department of Electrical and Computer Engineering, Boston University, 2002.

[7] Y. Minsky, A. Trachtenberg, R. Zippel, “Set reconciliation with nearly optimal communication complexity,” *IEEE Trans. Inform. Theory*, 2003.

[8] M. Karpovsky, L. Levitin, and A. Trachtenberg, “Data verification and reconciliation with generalized error-control codes,” *IEEE Trans. Info. Theory*, July 2003.

[9] R. Roth, *Introduction to coding theory*, Cambridge University Press, 2006.

[10] D. Eppstein, M. Goodrich, F. Uyeda, G. Varghese, “What’s the difference? Efficient set reconciliation without prior context,” *SIGCOMM* 2011.

[11] M. T. Goodrich and M. Mitzenmacher, “Invertible bloom lookup tables,” ArXiv e-prints, 2011.

[12] T. Klove, B. Bose, and N. Elarief, “Systematic, single limited magnitude error correcting codes for flash memories,” *IEEE Trans. Info. Theory*, July 2011.

[13] D. Guo and M. Li, “Set reconciliation via counting bloom filters,” *IEEE Trans. Knowledge and Data Eng.*, 2013.

[14] V. Skachek, M. Rabbat, “Subspace synchronization: a network-coding approach to object reconciliation,” *ISIT*, 2014.

[15] D. Chen, C. Konrad, K. Yi, W. Yu, Q. Zhang, “Robust set reconciliation,” *SIGMOD*, 2014.

[16] R. Gabrys, F. Farnoud, “Reconciling similar sets of data,” *ISIT*, 2015.