INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Distributed Summation Privacy for Speech Enhancement

Matt O'Connor and W. Bastiaan Kleijn

Victoria University of Wellington

matthew.o.connor@ecs.vuw.ac.nz, bastiaan.kleijn@ecs.vuw.ac.nz

Abstract

Speech privacy in modern sensor network environments is necessary for widespread adoption and public trust of collaborative acoustic signal processing. Most current distributed privacy research deals with ensuring local node observations are not accessible by neighbouring nodes while still solving shared tasks. In this work we develop the concept of distributed task privacy in unbounded public networks, where linear codes are used to create limits on the number of nodes contributing to a distributed summation task, such as beamforming. We accomplish this by wrapping local observations in a linear code and intentionally applying symbol errors prior to transmission. If many nodes join a distributed speech enhancement task, a proportional number of symbol errors are introduced into the aggregated code leading to decoding failure if the code's predefined symbol error limit is exceeded.

Index Terms: Distributed, privacy, audio, linear codes, beamforming

1. Introduction

In recent years many distributed algorithms for wireless sensor networks (WSNs) have been developed, primarily stemming from the advances made in small and energy efficient processing units and battery technology. These systems offer exciting potential for use in speech enhancement [1,2]. As concepts such as the Internet of Things (IoT) [3] mature, the wireless sensor network will become ubiquitous. Nearly any technological device in the near future will have the capability to be a part of the IoT, and potentially contribute to enhancement tasks.

Algorithmic developments in WSNs aim to provide distributed solutions for traditional problems such as acoustic beamforming [4–6] and image enhancement [7]. Distributed sensors are exploited to collaboratively solve tasks in a manner optimal for the data present, by sharing local observations in an unrestricted manner. The rapidly growing field of distributed optimization [8] provides a framework for problems of this type and often allows for the computation of distributed solutions that are equal in performance to their centrally computed alternatives. However, these advances are not without their challenges [9–13].

Many devices absorbed into the IoT, or designed as part of large-scale public sensor networks, will have physical sensors, such as microphones or cameras, offering major concerns for the privacy of device owners, users, and the general public. Current research focuses on privacy-preserving algorithms that prevent nodes other than the observing node from accessing this private observation. We refer to this as Local Data Privacy (LDP). LDP may be further classified into computational security and information-theoretic security. Computationally secure approaches [14–16] attempt to retain privacy by using techniques such as homomorphic encryption [17, 18], but this is often computationally expensive, particularly for the low compute and low energy world of WSNs [19]. Information-theoretically secure methods use noise to protect local data, and include secret sharing [20], differential privacy approaches [21–23], and more recent convex optimisation based methods [24].

In this work we introduce Distributed Task Privacy (DTP) for summations that, in contrast to LDP, aims to ensure the privacy of entire distributed summation tasks from outside eavesdroppers. These summations often occur when performing common enhancement methods such as source separation [25] and beamforming [5]. We consider dealing with effectively unbounded public WSNs where a user may tap the network at any point to designate a query node. The task to be performed is shared to form a task subnet, and information is processed in such a way that distant nodes within the network cannot participate in the task without severely degrading task performance. We accomplish this by wrapping messages in linear codes and applying forced errors to the local initial codewords. When performing aggregation in the codeword domain, we enable speech enhancement near to the query node while guaranteeing data destruction if the task is shared with a subnet larger than the specified threshold. This allows for a public network with privacy, where many different users may independently access the network for processing while maintaining an expected level of security consistent with the physical signal, e.g., it should not be possible to eavesdrop across a large building but it may be possible if you are within the same room. Importantly, the proposed DTP may be used in conjunction with LDP approaches to ensure both local data and distributed summation task privacy.

The following sections are organised as follows: Section 2 introduces the notation, network setup, and summation technique used in this paper; Section 3 summarizes linear codes; Section 4 develops an approach for privacy-preserving data summation; Section 5 presents simulations confirming the validity of our approach; and Section 6 concludes this work.

2. Problem formulation

An effectively unbounded public wireless sensor network (WSN) may be described as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of vertices, or nodes, $\mathcal{V}$ connected via edges $\mathcal{E}$. The node set has cardinality $|\mathcal{V}| = V$. Each node is equipped with an on-board processor, a two-way communication system, a power supply, and a microphone. We assume that communication in our graph is undirected and contains self-loops. Scalars are denoted with lower case regular font x, vectors are boldface lower case $\boldsymbol{x}$, while matrices are upper case boldface $\boldsymbol{P}$. We use subscripts to designate that a specific variable is owned by a node and superscripts to indicate an update index for iterative algorithms, therefore $\boldsymbol{x}_i^k$ describes a vector held by node i at iteration k. $[\boldsymbol{x}]_i$ denotes the ith element of vector $\boldsymbol{x}$, $[\boldsymbol{P}]_{i,j}$ selects out the scalar entry at row i and column j of matrix $\boldsymbol{P}$. We denote selection of multiple elements from a vector as $[\boldsymbol{x}]_e$,

This research was supported by a New Zealand Marsden Fund grant.

http://dx.doi.org/10.21437/Interspeech.2020-1977

where $\boldsymbol{e}$ is the vector of indices from which to select, resulting in a vector with dimensionality equal to the number of elements selected. $\boldsymbol{x}^T$ denotes vector transpose.

Each node $i \in \mathcal{V}$ holds observation data $\boldsymbol{u}_i(t) : \mathbb{Z} \rightarrow \mathbb{R}^l$ at time sample t. When a user wishes to begin a task, they tap a node that will henceforth be considered the *query node* for that task, denoted by the specific subscript index q. For both practical and privacy reasons, the query node spreads the task to a subset $\mathcal{V}_q \subset \mathcal{V}$ with cardinality $|\mathcal{V}_q| = V_q$ of nearby nodes.

The problem we consider is that the subset $\mathcal{V}_q$, which forms its own connected subgraph $\mathcal{G}_q = (\mathcal{V}_q, \mathcal{E}_q)$ with edge set $\mathcal{E}_q \in \mathcal{E}$, must collaboratively solve the user-requested task while also restricting the ability for nodes to join the task that are not within the task subnet. This smaller task subnet $\mathcal{V}_q$ allows for more efficient computations to be performed since information is not required to propagate through the entire public network. For privacy purposes this reduced information travel distance means that expected levels of privacy are more easily retained - nodes that are very distant from a query node should not have access to tasks seeded at the query node. However, enforcing the size of subnet $\mathcal{V}_q$ is not trivial, particularly if some nodes become compromised and actively wish to spread tasks further than intended.

Formally, we consider an aggregation process where information that has an expected level of privacy, such as an acoustic signal that is assumed to decay with distance as it propagates through air, is combined in a distributed manner as the weighted sum

$$\sum_{i \in \mathcal{V}_q} a_i \boldsymbol{u}_i, \tag{1}$$

where $\boldsymbol{u}_i$ is observed data at each node i, the time index t has been omitted due to the aggregate of each sample being computed independently, and a_i is some scalar. Specific examples of these scalars could be $a_i = 1 \; \forall i \in \mathcal{V}_q$, which corresponds to a simple summation, or $a_i = 1/V_q \; \forall i \in \mathcal{V}_q$, which would result in a uniform average of observations.

In order to compute the weighted summation (1) in a distributed manner, each node i may linearly combine data from neighbouring nodes iteratively to produce new estimates. This may be framed generally as

$$\boldsymbol{u}_i^{k+1} = \sum_{j \in \mathcal{N}_i} [\boldsymbol{P}^{k+1}]_{i,j} \boldsymbol{u}_j^k \quad \forall i \in \mathcal{V}_q \tag{2}$$

where $\mathcal{N}_i$ denotes the neighbourhood set of node i, and $\boldsymbol{P}^{k+1}$ is a mixing matrix for iteration $k+1$ constrained to have a sparsity pattern according to the topology of the network, such that $\boldsymbol{P} \in \mathcal{S}$ where $\mathcal{S} = \{\boldsymbol{P} \in \mathbb{R}^{V_q \times V_q} | \{i,j\} \notin \mathcal{E}_q \Rightarrow [\boldsymbol{P}]_{i,j} = 0\}$. Note that $\boldsymbol{P}^{k+1}$ may change for each iteration, and does not necessarily mix information over all neighbourhoods. A specific case of the general mixing (2) frequently seen in the literature, and often useful in practice, are routed protocols [26–29]. Since iterations may be performed at certain nodes before others, routing protocols that remove edges from the query subnet to form a tree topology rooted on the query node may be implemented using (2) by ensuring mixing is performed from leaf to parent nodes. After the routing summation is complete, the query node will have access to the sum of all nodes within the query subnet. In this work, we focus on summations using routed protocols.

To retain privacy when performing iterations such as (2), recent methods [22–24] aim to perform neighbourhood mixing in such a way that nodes do not directly observe data other than their own. This maintains the privacy of observed signals between nodes, since local observations are not explicitly shared with neighbours, but *does not limit the distance that these mixtures travel within the larger public WSN $\mathcal{V}$.* If, for example, the mixture $\boldsymbol{u}_i^{k+1}$ in (2) was an acoustic speech enhancement estimate, then this information would be allowed to travel unbounded within the network, allowing for distant eavesdropping.

3. Linear codes over prime fields

Linear error correcting codes, or simply linear codes, e.g., [30,31], are an important class of forward error correcting codes used to protect information transmission or storage from symbol errors by using redundancy. They are defined over a finite vector space $\mathbb{F}_r^n$, where $\mathbb{F}_r$ is a finite field of order r. In this work we limit ourselves to prime fields, where $r = p$. The following definition holds:

Definition 1. *(Linear code). A linear code C is a code in $\mathbb{F}_p^n$ for which, whenever $\boldsymbol{x}, \boldsymbol{y} \in C$, then $a\boldsymbol{x} + b\boldsymbol{y} \in C$, for all $a, b \in \mathbb{F}_p$, i.e., C is a linear subspace of $\mathbb{F}_p^n$.*

A linear code C defines an encoder map $E_{l,n}^p : \mathbb{F}_p^l \rightarrow \mathbb{F}_p^n$ from an l-dimensional message $\boldsymbol{m} \in \mathbb{F}_p^l$ to an n-dimensional codeword $\boldsymbol{c} \in \mathbb{F}_p^n$. This encoder map is usually computed using the generator matrix $\boldsymbol{G} \in \mathbb{F}_p^{l \times n}$, where encoding is performed as

$$\boldsymbol{c} = \boldsymbol{G}\boldsymbol{m}. \tag{3}$$

The corresponding decoder map $D_{n,l}^p : \mathbb{F}_p^n \rightarrow \mathbb{F}_p^l$ recovers the original message $\boldsymbol{m}$ from the codeword $\boldsymbol{c}$. Linear codes are typically denoted as $[n, l, d]$ codes, where n is the length of the codeword, l is the length of the message to be encoded, and d refers to the minimum Hamming distance between any two codewords. Every linear code satisfies the Singleton bound $l + d \leq n + 1$, where $1 \leq l \leq n$. Given the Hamming distance d, a linear code may correctly decode a corrupted codeword $\tilde{\boldsymbol{c}}$ provided that fewer than $d/2$ symbol errors occur.

We assume the observation $\boldsymbol{u}_i \in \mathbb{R}^l$ at each node $i \in \mathcal{V}$ has entries bounded in the range $-y \preceq \boldsymbol{u}_i \preceq y$. Given that the observed data may be continuous (or stored using quantization at a far finer level than our transmission rate would allow, so as to appear effectively continuous), a quantization step may be required prior to coding. The observations are quantized using a uniform p-level quantizer $Q_{l,p}$ resulting in p equally spaced values over the range $-y$ to y. We refer to the quantized observations as messages $\boldsymbol{m}_i = Q_{l,p}(\boldsymbol{u}_i) \in \mathbb{F}_p^l \; \forall i \in \mathcal{V}$, and their decoded approximations as $\hat{\boldsymbol{u}}_i = R_{l,p}(Q_{l,p}(\boldsymbol{u}_i)) \in \mathbb{R}^l \; \forall i \in \mathcal{V}$. In the remainder of this work we refer to Q as the quantizer and R as the dequantizer (rather than as the decoder, to avoid confusion with the linear code decoder $D_{n,l}^p$).

4. Distributed private summation

In this section, we exploit linear codes in a novel way to guarantee distributed privacy when performing processing over a WSN, where a processing task is defined over the subnet $\mathcal{V}_q$ of nodes originating from a query node q. We firstly present the algorithm, and then provide an analysis on the mutual information decay that occurs when decoding failure occurs.

4.1. Distributed private summation algorithm

As a result of Definition 1, for any linear code the sum or difference of any two codewords is also a codeword. Since messages are encoded by a matrix multiplication with the generator $\boldsymbol{G}$, the codeword associated with the sum of two messages $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ is the same as the sum of the two separate encodings

$$\begin{aligned} & E_{l,n}^{p}(\boldsymbol{m}_1 + \boldsymbol{m}_2) = \boldsymbol{G}(\boldsymbol{m}_1 + \boldsymbol{m}_2) \\ & = \boldsymbol{G}\boldsymbol{m}_1 + \boldsymbol{G}\boldsymbol{m}_2 = E_{l,n}^{p}(\boldsymbol{m}_1) + E_{l,n}^{p}(\boldsymbol{m}_2), \end{aligned} \tag{4}$$

where addition and matrix multiplication are performed using finite field arithmetic over the field $\mathbb{F}_p$, i.e., modulo p.

From (4), summations over the network may be performed on *codewords*, rather than messages, with forced errors present. If a node wishes to output an estimate of the private aggregation procedure, only then will it decode the forcibly corrupted codeword mixture. By controlling the number of errors introduced and the Hamming distance of the linear code used, we effectively bound the overall number of nodes able to participate in a distributed task.

Algorithm 1 describes the Distributed Private Summation (DPS) procedure. For this scenario, we require a message quantizer/dequantizer to map between $\mathbb{R}^l$ and $\mathbb{F}_p^l$, and a linear encoder/decoder to map between $\mathbb{F}_p^l$ and $\mathbb{F}_p^n$. The parameter l is the dimensionality of the observation. A predefined code length n and a predefined number of symbol errors λ are also necessary. Given these requirements, each node determines λ codeword symbol indices that will be corrupted by error.

Nodes observe signals $\boldsymbol{u}_i \ \forall i \in \mathcal{V}$, and a summation task is defined over a task subnet $\mathcal{V}_q \subset \mathcal{V}$. A set of edges $\mathcal{T}_q^0$ is determined that converts the general task graph into a tree graph rooted at the query node q. Messages $\boldsymbol{m}_i \ \forall i \in \mathcal{V}$ are formed by mapping observations to the finite field $\boldsymbol{F}_p^l$, where message values must satisfy $|\mathcal{V}_q|\max(\|\boldsymbol{m}_0\|_\infty, \ldots, \|\boldsymbol{m}_{|\mathcal{V}_q|}\|_\infty) \leq p$ to guarantee summation overflow does not occur. Initial codewords $\boldsymbol{c}_i^0$ are computed by encoding messages $\boldsymbol{m}_i, \ \forall i \in \mathcal{V}$, and λ symbol errors are applied to each codeword randomly and independently. We then begin iteratively summing through the tree, from leaf nodes to the root. At each iteration k we use the tree edges $\mathcal{T}_q^k$ to define a leaf node set $\mathcal{L}_q^k$, a set of leaf parent nodes $\mathcal{P}_q^k$, and a set of all edges connected to leaf nodes denoted $\mathcal{F}_q^k$. Each leaf parent stores the sum of its own codeword and the codewords of all its leaf neighbours (defined as the union of the leaf parent's neighbours and the current leaf nodes) as $\boldsymbol{c}_i^{k+1}$. The tree edge set is then updated by removing the current leaf edge set $\mathcal{F}_q^k$ from the current tree edge set. The final output at the query node q is the decoded and dequantized codeword after summation termination.

4.2. Mutual information decay

Let $I(\boldsymbol{m}, \hat{\boldsymbol{m}})$ denote the mutual information between true message $\boldsymbol{m}$ and decoded message $\hat{\boldsymbol{m}}$, $H(\boldsymbol{m})$ as the entropy of message $\boldsymbol{m}$, $\mathbb{E}[\cdot]$ as the expected value operator, and $S[\cdot]$ as the Heaviside step function defined to be zero when its argument is non-positive, and one when it is positive. The following proposition holds, where the proof is omitted to conserve space:

Proposition 1. *Given a network of V_q nodes using a MDS linear code with codeword length n, message length l, and prime field order p, each node introduces λ independent codeword errors, then the mutual information between true aggregated message $\boldsymbol{m}$ and the decoded message aggregate $\hat{\boldsymbol{m}}$ is given by*

$$\begin{aligned} I(\boldsymbol{m}, \hat{\boldsymbol{m}}) = H(\boldsymbol{m})\big(S\big[-\mathbb{E}[\Lambda] + (n-l+1)/2\big] \\ + S\big[\mathbb{E}[\Lambda] - (n-l+1)/2\big](1 - \frac{\mathbb{E}[\Lambda]}{n})\big), \end{aligned} \tag{5}$$

where

$$\mathbb{E}[\Lambda] = (1 - \frac{1}{p})n(1 - (1 - \frac{1}{n})^{\lambda V_q}). \tag{6}$$

Algorithm 1 Distributed Private Summation

```
Require: Task subnet $\mathcal{G}_q$; prime field characteristic $p$; message quantizer $Q_{l,p}$; message dequantizer $R_{l,p}$; linear encoder $E_{l,n}^p$; linear decoder $D_{n,l}^p$; code length $n$; number of errors $\lambda$

  Symbol error index vectors $\boldsymbol{e}_i = \boldsymbol{a} \sim U(n, \lambda) \ \forall i \in \mathcal{V}$
  Nodes observe $\boldsymbol{u}_i \in \mathbb{R}^l \ \forall i \in \mathcal{V}$
  Summation defined over task subnet $(\mathcal{V}_q, \mathcal{E}_q) \subseteq (\mathcal{V}, \mathcal{E})$
  Construct tree edge set $\mathcal{T}_q^0 \subseteq \mathcal{E}_q$ rooted at query node $q$
  $\boldsymbol{m}_i = Q_{l,p}(\boldsymbol{u}_i), |\mathcal{V}_q|\max(\|\boldsymbol{m}_0\|_\infty, \ldots, \|\boldsymbol{m}_n\|_\infty) \leq p$
  Encode $\boldsymbol{c}_i^0 = E_{l,n}^p(\boldsymbol{m}_i) \in \mathbb{F}_p^n \ \forall i \in \mathcal{V}_q$
  Apply errors $[\boldsymbol{c}_i^0]_{\boldsymbol{e}_i} = \boldsymbol{b} \sim U(p, \lambda) \ \forall i \in \mathcal{V}_q$
  $k = 0$
  while $\mathcal{T}_q^k \neq \emptyset$ do
    Define leaf nodes $\mathcal{L}_q^k$, leaf parents $\mathcal{P}_q^k$, and leaf edges $\mathcal{F}_q^k = \{(i,j) | j \in \mathcal{N}_i \ \forall i \in \mathcal{L}_q^k\}$ using $\mathcal{T}_q^k$
    $[\boldsymbol{P}^{k+1}]_{i,j} = 1$ for $i == j$,
    $[\boldsymbol{P}^{k+1}]_{i,j} = 1$ for $j \in \mathcal{N}_i \cap \mathcal{L}_q^k \ \forall i \in \mathcal{P}_q^k$,
    $[\boldsymbol{P}^{k+1}]_{i,j} = 0$ otherwise.
    $\mathcal{T}_q^{k+1} \leftarrow \mathcal{T}_q^k \setminus \mathcal{F}_q^k$
    $\boldsymbol{c}_i^{k+1} \leftarrow \sum_{j \in \mathcal{N}_i} [\boldsymbol{P}^{k+1}]_{i,j} \boldsymbol{c}_j^k \ \forall i \in \mathcal{V}_q$
    $k \leftarrow k + 1$
  end while

  $\boldsymbol{u}_q^{\mathrm{Sum}} = R_{l,p}(D_{n,l}^p(\boldsymbol{c}_q^k))$
```

5. Simulation experiments

In this section we investigate Algorithm 1 when applied in two simulated scenarios using Reed-Solomon (RS) codes [32] - random number summation and distributed speech enhancement. The network consists of a varying number of nodes uniformly randomly scattered in a $2D$ circular surface with radius 25 m. The nodes have a communication range of 10 m. The binary adjacency matrix for the network graph is then constructed, where connected node pairs are represented with edges of value 1 while unconnected pairs have edge value 0.

5.1. Toy data

In this scenario, nodes are each assigned a random integer message $\boldsymbol{m}_i \in \mathbb{F}_p^l \ \forall i \in \mathcal{V}_q$, where each dimension is drawn from the discrete uniform distribution $\mathcal{U}\{0, p/V_q\}$. We use a prime field with characteristic $p = 251$ so that our messages may be stored within 8 bits. A random node is then selected as the query node q. Firstly, we use a fixed codeword length of $n = 64$ while varying the message dimensionality, and correspondingly the codeword Hamming distance. Three message lengths of $l = 16, 32$, and 48 are implemented in order to compare the point at which decoding breaks down. This gives a redundancy of 300%, 100%, and 33.3%, respectively. Nodes each introduce $\lambda = 1$ symbol errors to their own message. We then double the redundancy for each message length, and also introduce $\lambda = 2$ symbol errors to maintain roughly the same codeword failure point, while increasing the decay rate past this point. Since the generator matrix G for the RS code

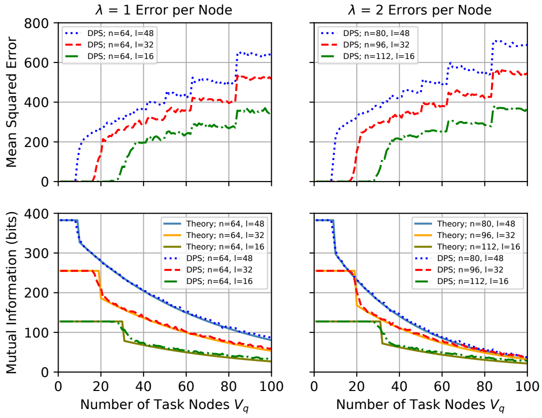

Figure 1: *MSE and Mutual Information versus number of task nodes, for varying levels of codeword redundancy.*

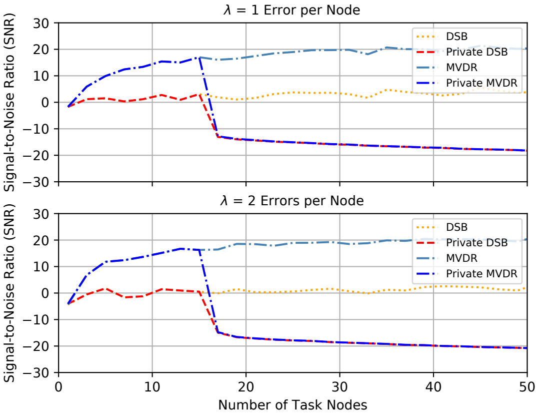

Figure 2: *SNR versus number of task nodes for conventional and private beamformers, with codeword redundancy of* 13.8%.

used is in standard form, it is possible to attempt decoding past the point at which the RS code breaks down by simply reading the first l codeword dimensions. This is used to compute the mean squared error (MSE) and mutual information (MI) after decoding fails.

The left column of Figure 1 shows performance when nodes each introduce a single error, while the right column shows performance when two are added. With a single error, for all code lengths we see decoding error appearing past the expected point of 8, 16, and 24 nodes, respectively, since at this point there are approximately $d/2$ symbol errors in the final summation total when a single error is present (recall d is the Hamming distance of our linear code). We see a constant MI until the point of decoding failure, at which point we see exponential decay. By doubling the redundancy while also doubling errors per node we maintain roughly the same point of decoding failure, but we now accelerate the information decay rate past this point. This simulated data very closely matches the theoretical performance predicted in (5).

5.2. Private speech beamforming

In this private beamforming setup nodes observe acoustic signals originating from a talker located at the centre of the simulated environment surface. A second interfering talker along with independent additive white Gaussian noise at each node is also present. The observed signals at each node, sampled synchronously at 8 kHz, are calculated using the acoustic transfer function vector $\boldsymbol{d}$ computed by assuming a free field model for both talkers. Beamforming is accomplished using an estimated covariance matrix over all nodes $\boldsymbol{R}$ to compute the optimal weight vector $\boldsymbol{w}^* = (\boldsymbol{R}^{-1}\boldsymbol{d})/(\boldsymbol{d}^H\boldsymbol{R}^{-1}\boldsymbol{d})$. The task-specific weighting vector $\boldsymbol{w}^*$ may be computed either centrally or in a distributed manner [8, 33, 34], and we assume that no private data leakage occurs here. For the delay-and-sum (DSB) scenario, the covariance matrix $\boldsymbol{R}$ was assumed to be diagonal, while the MVDR beamformer [35] exploited the full matrix.

We process local signals by taking 50% overlapping time-domain blocks and applying a Hann window prior to taking the short-time Fourier transform. This gives us frequency-domain signals denoted $\boldsymbol{x}_i \in \mathbb{C}^l$ at all nodes $i \in \mathcal{V}_q$ within the query subnet. We denote the stacked collection of these distributed signals as the matrix $\boldsymbol{X}_q \in \mathbb{C}^{V_q \times l}$. The complex weight vector $\boldsymbol{w}^* \in \mathbb{C}^{|\mathcal{V}_q|}$ may then be used to form signal aggregation across the network as

$$\hat{\boldsymbol{s}} = \boldsymbol{w}^{*T}\boldsymbol{X}_q = \sum_{i \in \mathcal{V}_q} [\boldsymbol{w}^*]_i \boldsymbol{x}_i = \sum_{i \in \mathcal{V}_q} a_i \boldsymbol{u}_i, \qquad (7)$$

where $\boldsymbol{u}_i = \mathrm{real}([\boldsymbol{w}^*]_i \boldsymbol{x}_i) \in \mathbb{R}^l$ is a task weighing at node i applied to the signal at this node to produce observation data $\boldsymbol{u}_i$, and $\hat{\boldsymbol{s}}$ is the enhanced signal sample. These observations are then quantized to give messages $\boldsymbol{m}_i \in \mathbb{F}_p^l \ \forall i \in \mathcal{V}_q$. We use a prime field with characteristic $p = 16381$ so that our messages may be stored within 14 bits. A block size of $l = 224$ is used which allows for a reasonable acoustic window length of 28 ms. With $\lambda = 1$ error introduced we have a codeword length $n = 255$. When doubling the introduced errors we also double the redundancy, giving a codeword length of $n = 286$.

Figure 2 plots the signal-to-noise ratio (SNR) as a function of contributing task nodes. Initially, as more nodes are included in the beamforming task we see an increase in performance of the enhanced signal. This boost in SNR drops after the chosen decoding failure point. In contrast, we see that with no errors applied the DSB and MVDR performance continues to rise as more nodes are included, compromising privacy. We note that the point at which information destruction occurs is entirely controlled by the system designer. This may be set to impose dropoff faster than natural acoustic signal decay, guaranteeing privacy. Importantly, the communication overhead required for our private protocol is minimal (13.8% and 27.7%, respectively) when compared to the block length of the original signal.

6. Conclusion

We conclude that public WSN privacy can be ensured by limiting information propagation throughout an unbounded network, where speech tasks are seeded by user-accessed query nodes. We have applied errors to locally encoded data observations, allowing for distributed aggregation that is performed in a manner that guarantees information destruction when too many nodes contribute to the task. This allows a system designer to enforce a level of acoustic privacy consistent with natural signal expectations. Our approach is efficient, flexible, and scalable, and may be used in combination with other existing protocols that encourage local node privacy.

7. References

[1] Maja Taseska and Emanuël AP Habets, "Informed spatial filtering for sound extraction using distributed microphone arrays," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 7, pp. 1195–1207, 2014.

[2] Vincent M. Tavakoli, Jesper R. Jensen, Richard Heusdens, Jacob Benesty, and Mads G. Christensen, "Distributed max-SINR speech enhancement with ad hoc microphone arrays," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 151–155.

[3] Xiaoyi Cui, "The internet of things," in *Ethical Ripples of Creativity and Innovation*, pp. 61–68. Springer, 2016.

[4] Shmulik Markovich-Golan, Alexander Bertrand, Marc Moonen, and Sharon Gannot, "Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks," *Signal Processing*, vol. 107, pp. 4–20, 2015.

[5] Matt O'Connor, W. Bastiaan Kleijn, and Thushara Abhayapala, "Distributed sparse MVDR beamforming using the bi-alternating direction method of multipliers," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 106–110.

[6] Matt O'Connor and W. Bastiaan Kleijn, "Diffusion-based distributed MVDR beamformer," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 810–814.

[7] Matt O'Connor, W Bastiaan Kleijn, and Thushara Abhayapala, "Distributed TV-L1 image fusion using PDMM," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 3326–3330.

[8] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[9] P. T. Endo, A. V. de Almeida Palhares, N. N. Pereira, G. E. Goncalves, D. Sadok, J. Kelner, B. Melander, and J. Mangs, "Resource allocation for distributed cloud: concepts and research challenges," *IEEE Network*, vol. 25, no. 4, pp. 42–46, July 2011.

[10] S Massoud Amin, "Smart grid: Overview, issues and opportunities. advances and challenges in sensing, modeling, simulation, optimization and control," *European Journal of Control*, vol. 17, no. 5-6, pp. 547–567, 2011.

[11] Feng Chen, Pan Deng, Jiafu Wan, Daqiang Zhang, Athanasios V Vasilakos, and Xiaohui Rong, "Data mining for the internet of things: literature review and challenges," *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, pp. 431047, 2015.

[12] Wenshuang Liang, Zhuorong Li, Hongyang Zhang, Shenling Wang, and Rongfang Bie, "Vehicular ad hoc networks: architectures, research issues, methodologies, challenges, and trends," *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, pp. 745303, 2015.

[13] Ataul Bari, Jin Jiang, Walid Saad, and Arunita Jaekel, "Challenges in the smart grid applications: an overview," *International Journal of Distributed Sensor Networks*, vol. 10, no. 2, pp. 974682, 2014.

[14] Yoshinori Aono, Takuya Hayashi, Le Trieu Phong, and Lihua Wang, "Privacy-preserving logistic regression with distributed data sources via homomorphic encryption," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 8, pp. 2079–2089, 2016.

[15] Richard C Hendriks, Zekeriya Erkin, and Timo Gerkmann, "Privacy preserving distributed beamforming based on homomorphic encryption," in *21st European Signal Processing Conference (EUSIPCO 2013)*. IEEE, 2013, pp. 1–5.

[16] Chunlei Zhang, Muaz Ahmad, and Yongqiang Wang, "Admm based privacy-preserving decentralized optimization," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 565–580, 2018.

[17] Ronald Cramer, Ivan Bjerre Damgård, and Jesper Buus Nielsen, *Secure multiparty computation*, Cambridge University Press, 2015.

[18] Pascal Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1999, pp. 223–238.

[19] Hai-Ying Zhou, Dan-Yan Luo, Yan Gao, and De-Cheng Zuo, "Modeling of node energy consumption for wireless sensor networks," *Wireless Sensor Network*, vol. 3, no. 1, pp. 18, 2011.

[20] Qiongxiu Li, Ignacio Cascudo, and Mads Græsbøll Christensen, "Privacy-preserving distributed average consensus based on additive secret sharing," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[21] Mahdi Kefayati, Mohammad S Talebi, Babak H Khalaj, and Hamid R Rabiee, "Secure consensus averaging in sensor networks using random offsets," in *2007 IEEE International Conference on Telecommunications and Malaysia International Conference on Communications*. IEEE, 2007, pp. 556–560.

[22] Yilin Mo and Richard M Murray, "Privacy preserving average consensus," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 753–765, 2016.

[23] Nirupam Gupta, Jonathan Katz, and Nikhil Chopra, "Privacy in distributed average consensus," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9515–9520, 2017.

[24] Qiongxiu Li, Richard Heusdens, and Mads Græsbøll Christensen, "Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks," in *45th International Conference on Acoustics, Speech, and Signal Processing*, 2020.

[25] S.R. Mir Alavi and W. Bastiaan Kleijn, "Distributed linear blind source separation over wireless sensor networks with arbitrary connectivity patterns," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 3171–3175.

[26] Wanzhi Qiu, Efstratios Skafidas, and Peng Hao, "Enhanced tree routing for wireless sensor networks," *Ad hoc networks*, vol. 7, no. 3, pp. 638–650, 2009.

[27] Feng Zhao, Leonidas J Guibas, and Leonidas Guibas, *Wireless sensor networks: an information processing approach*, Morgan Kaufmann, 2004.

[28] Kemal Akkaya and Mohamed Younis, "A survey on routing protocols for wireless sensor networks," *Ad hoc networks*, vol. 3, no. 3, pp. 325–349, 2005.

[29] Ahcène Bounceur, Madani Bezoui, Massinissa Lounis, Reinhardt Euler, and Ciprian Teodorov, "A new dominating tree routing algorithm for efficient leader election in iot networks," 01 2018, pp. 1–2.

[30] W Cary Huffman and Vera Pless, *Fundamentals of error-correcting codes*, Cambridge university press, 2010.

[31] Hideki Imai, *Essentials of error-control coding techniques*, Academic Press, 2014.

[32] Irving S. Reed and Gustave Solomon, "Polynomial codes over certain finite fields," *Journal of the society for industrial and applied mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[33] Guoqiang Zhang and Richard Heusdens, "Distributed optimization using the primal-dual method of multipliers," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 173–187, 2017.

[34] Matt O'Connor, Guoqiang Zhang, W Bastiaan Kleijn, and Thushara Dheemantha Abhayapala, "Function splitting and quadratic approximation of the primal-dual method of multipliers for distributed optimization over graphs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 4, pp. 656–666, 2018.

[35] Emanuël AP Habets, Jacob Benesty, Sharon Gannot, and Israel Cohen, "The MVDR beamformer for speech enhancement," in *Speech Processing in Modern Communication*, pp. 225–254. Springer, 2010.