Journal of Privacy and Confidentiality
Vol. 15 (3) 2025

Submitted March 2024
Published December 2025

ACHIEVING PRIVACY UTILITY BALANCE FOR MULTIVARIATE TIME SERIES DATA

GAURAB HORE, TUCKER S. MCELROY, AND ANINDYA ROY

University of Maryland Baltimore County, 1000 Hilltop Cir, Baltimore, MD 21250
e-mail address: gaurabh1@umbc.edu

Research and Methodology Directorate, U.S. Census Bureau, 4600 Silver Hill Road,Washington, D.C. 20233-9100, USA
e-mail address: tucker.s.mcelroy@census.gov

University of Maryland Baltimore County, 1000 Hilltop Cir, Baltimore, MD 21250
e-mail address: anindya@umbc.edu

Abstract. Utility-preserving data privatization is of utmost importance for data-producing agencies. The popular noise-addition privacy mechanism distorts autocorrelation patterns in time series data, thereby marring utility; in response, (MRH23) introduced all-pass filtering (FLIP) as a utility-preserving time series data privatization method. Adapting this concept to multivariate data is more complex, and in this paper we propose a multivariate all-pass (MAP) filtering method, employing an optimization algorithm to achieve the best balance between data utility and privacy protection. To test the effectiveness of our approach, we apply MAP filtering to both simulated and real data, sourced from the U.S. Census Bureau's Quarterly Workforce Indicator (QWI) dataset.

Introduction

With increased digital participation and online engagement, safeguarding sensitive data has become extremely important over the last decade. Researchers have devised innovative approaches for data privacy, and a multitude of privacy measures and implementation mechanisms have been proposed in the literature. However, noise additions are relied upon by most privacy mechanisms, including *differential privacy* (DP) (Dwo06; DR14), the gold standard for privacy implementation. Since privacy measures such as DP are developed primarily for databases with independent entries, the privacy guarantees no longer hold for dependent data (such as time series data). Moreover, when it comes to time series, independent white noise addition (or multiplication) may significantly change the autocorrelation structure, thereby diminishing the quality and utility of such data. To address these problems, (MRH23) developed a predictive privacy measure, called Linear Incremental Privacy (LIP), that is particularly suited for regularly-spaced time series. The

Key words and phrases: All-pass filter; Linear incremental privacy; Multiple time series; Spectral factorization.

www.journalprivacyconfidentiality.org
DOI:10.29012/jpc.916

biggest challenge in implementation of formal privacy mechanisms for complex datasets is the balancing of data utility with privacy objectives. In (WZ10), the authors forcefully argue for maintaining data utility while implementing disclosure avoidance algorithms. In (MRH23), the authors suggested a utility-preserving implementation for the LIP mechanism, called FLIP, that is based on noise filtering instead of noise addition.

For multiple time series, the need to account for utility while ensuring privacy is even more stark. The usual series-by-series privacy evaluation methods ignore a critical component of data utility for multivariate time series, viz. the cross-correlation structure. To our knowledge, there are no privacy procedures that preserve cross-series dependence information along with marginal time series properties while providing desirable privacy guarantees. The present article fills that gap. The main goal of this article is to extend the LIP measure to the multivariate setting and develop a multivariate generalization of FLIP implementation. The task is particularly challenging and nuanced because the utility landscape is much more multi-faceted in the multivariate case, and because the main tool in the FLIP mechanism, the all-pass phase filters, are not available for multivariate time series. This article develops the necessary tool for the multivariate extension of FLIP, including a novel class of multivariate all-pass filters. Based on the new class of filters, the generalization of the FLIP mechanism is established within a formal privacy framework that balances privacy and data utility.

Differential privacy is generally the norm for a formal privacy framework that provides hard privacy guarantees. It has been accepted widely in industry and government data protection plans, including implementation in the decennial U.S. census, probably one of the largest and most complex exercises in data collection and publication; see (AACM+22). But differential privacy is not designed for every data type. It is primarily designed for databases with independent entries (i.e., the mathematical formulation is valid under the independence assumption), and lacks optimality properties for time series data. While some articles ((SWC17) and (SC17)) examined modified DP mechanisms that are applied to time series structures, none provide any optimal balancing of privacy and utility. Thus, dependent data like time series data require a new privacy framework.

However, for new privacy mechanisms it is advantageous that they share the desirable properties of DP wherever possible. Although LIP is a predictive privacy measure that is more apt for time series, the incremental privacy addressed in the LIP framework is similar in spirit to DP, which addresses disclosure avoidance beyond what is available to the attacker. Moreover, LIP can be cast into the generalized DP framework espoused in (GM20), making it an attractive privacy framework for time series data.

For DP-like implementation in time series, researchers define the concept of adjacent databases, a central concept in the DP formulation, as event (window)-level adjacency or user-event-level adjacency. Recently, several articles have looked at forecasting properties of multiple time series after the application of privacy mechanisms. Many are based on deep learning and predictive structures for dynamical models when there is centralized or locally private implementation; see (KMC22; FX13; LXJL15; DGSG22; XYH+22; APPG23; PAK18; FVH19; ZKL22; WRN+20; IHA+20**?**). By examining the forecast properties of privatized series, these approaches do consider data utility, but they do not use any formal framework for balancing privacy and utility. There are several other approaches for univariate time series, developed in different disciplines like economics, cryptography, data mining, and data-streaming (and under different engineering applications such as power-grid) that are available in the literature; see (AGM+12; RN10; C12; LLJP17; LLML21; HGKM13;

SCR+11; EFM15; SST09; FVH19; Sta19; KTK22), and the references therein. However, these approaches do not use any mathematical framework for optimizing the privacy-utility trade-off.

Whereas the incremental privacy measure under LIP can be extended to the multivariate time series context, as is done in Section 1, the concept of all-pass filtering that was the primary tool for implementation of LIP is non-trivial to formulate in the multivariate case. In particular, the filters become matrix-valued, and hence the algebra is no longer commutative, complicating the mathematics. The main goal of this article is to develop the multivariate generalization of FLIP along with a generalization of all-pass filtering for multiple time series. The multivariate LIP measure, called m-LIP, is described in Section 1, and the multivariate all-pass filter is developed in Section 2. Section 3 discusses the optimal balancing of utility and privacy, with important extensions of our framework to non-stationary time series and the situation of a "worst case attacker." Section 4 provides the details for the implementation of m-LIP in practice. Limited numerical studies are given in Section 5 along with a real data application that examines Quarterly Workforce Indicator (QWI) data published by the U.S. Census Bureau. Section 6 provides conclusions and a discussion of future work.

1. Privacy Measure for Multivariate Time Series

Consider a sensitive multivariate time series $\{\mathbf{X}_t\}$ that will be perturbed using a randomized mechanism $\mathcal{M}$ to mitigate disclosure risk. The published series resulting from application of the randomized mechanism $\mathcal{M}$ will be denoted by $\{\mathbf{Y}_t\}$. The objective is to develop a formal privacy measure by which the privacy of the transformed time series $\{\mathbf{Y}_t\}$ can be measured – and hence optimized – to achieve desired privacy, possibly under utility constraints. In alignment with the approach presented in (MRH23), we operate under the assumption that potential adversaries possess prior information about the sensitive series in question, and we introduce auxiliary time series $\{\mathbf{Z}_t\}$ that encapsulate any knowledge that advanced attackers could employ for prediction. The randomized mechanism $\mathcal{M}$ is developed with the awareness of the existence of $\{\mathbf{Z}_t\}$. Each of these time series – the sensitive, the published, and the auxiliary – are multivariate with possibly different dimension.

Hereafter, we employ the following notation: the braces notation $\{\mathbf{X}_t\}$ (the bold font indicates that the time series is multivariate) denotes the entire time series, while $\mathbf{X}_t$ denotes the single random vector at time t. The optimal predictor for $\mathbf{X}_t$ given an information set $\mathcal{I}$ is the conditional expectation $E(\mathbf{X}_t|\mathcal{I})$. In particular, we examine optimal predictors of $\mathbf{X}_t$ under two different information sets, viz. when the adversary has the base knowledge of $\{\mathbf{Z}_t\}$, and with the added knowledge of the published value $\mathbf{Y}_t$ at time t. These predictors are denoted respectively by $E(\mathbf{X}_t|\{\mathbf{Z}_t\})$ and $E(\mathbf{X}_t|\mathbf{Y}_t, \{\mathbf{Z}_t\})$. It is easy to show that the reduction in the prediction variance matrix from the added knowledge of the released value is

$$\mathrm{Var}[\mathbf{X}_t|\{\mathbf{Z}_t\}] - \mathrm{Var}[\mathbf{X}_t|\mathbf{Y}_t, \{\mathbf{Z}_t\}] = \mathrm{Cov}[\mathbf{X}_t, \mathbf{Y}_t|\{\mathbf{Z}_t\}]\,\mathrm{Var}[\mathbf{Y}_t|\{\mathbf{Z}_t\}]^{-1}\,\mathrm{Cov}[\mathbf{Y}_t, \mathbf{X}_t|\{\mathbf{Z}_t\}]. \tag{1.1}$$

Here, the left-hand side involves two conditional variances for the prediction of $\mathbf{X}_t$, considered before and after the publication of $\mathbf{Y}_t$, with the difference indicating incremental vulnerability to the sensitive data. The right-hand side involves a non-negative definite matrix; this quantity equals zero when $\mathbf{Y}_t$ offers no assistance to the attack. To ensure that the release of $\mathbf{Y}_t$ does not provide any incremental gain to the attacker in terms of prediction accuracy

for predicting $\mathbf{X}_t$, the value of the sensitive series, the reduction in prediction MSE from using the knowledge of the published value should be minimal.

In most cases the joint distribution of the time series will not be fully specified, and hence obtaining an analytical expression for the conditional expectation will be an infeasible objective. However, in most of the time series literature (BD) the prediction theory is developed based on linear predictors using the L_2 projection theory, and we will do the same. Therefore, with a slight abuse of notation, the conditional expectation denotes the optimal linear predictors. Of course it coincides with the conditional expectation when the series are jointly Gaussian. Below, we define our privacy measure in terms of the reduction in prediction MSE from using the knowledge of $\mathbf{Y}_t$.

Definition 1 (m-LIP)**.** Let $\{\mathbf{X}_t, \mathbf{Y}_t, \mathbf{Z}_t\}$ be jointly stationary multivariate time series, where the published series $\{\mathbf{Y}_t\}$ is obtained from the sensitive series $\{\mathbf{X}_t\}$ by applying a randomized mechanism $\mathcal{M}$ on $\{\mathbf{X}_t\}$ and $\{\mathbf{Z}_t\}$ is an auxiliary series representing the knowledge of an advanced adversary. Then the *multivariate Linear Incremental Privacy* (m-LIP) of $\mathcal{M}$ at $\{\mathbf{X}_t\}$ given $\{\mathbf{Z}_t\}$ is defined as

$$m\text{-LIP}(\mathcal{M}(\{\mathbf{X}_t\})|\{\mathbf{Z}\}) = 1 - \frac{\det\left[\mathrm{Cov}[\mathbf{X}_t, \mathbf{Y}_t|\{\mathbf{Z}_t\}]\,\mathrm{Var}[\mathbf{Y}_t|\{\mathbf{Z}_t\}]^{-1}\,\mathrm{Cov}[\mathbf{Y}_t, \mathbf{X}_t|\{\mathbf{Z}_t\}]\right]}{\det \mathrm{Var}[\mathbf{X}_t|\{\mathbf{Z}_t\}]}. \tag{1.2}$$

The measure is well-defined unless $\det \mathrm{Var}[\mathbf{X}_t|\{\mathbf{Z}_t\}] = 0$, which corresponds to a trivial case where the attacker already possesses the sensitive information, making privacy unattainable. Otherwise, this measure can be viewed as one minus a function of the multivariate squared conditional correlation, analogous to the familiar R^2 statistic from linear models. The following proposition shows that the privacy measure takes values in $[0, 1]$.

Proposition 1.1. *Let $\{\mathbf{X}_t\}$, $\{\mathbf{Y}_t\}$, and $\{\mathbf{Z}_t\}$ be multivariate time series, and let $\{\mathbf{Y}_t\}$ be produced from $\{\mathbf{X}_t\}$ after applying a randomized mechanism $\mathcal{M}$. If Var$[\mathbf{X}_t|\{\mathbf{Z}_t\}]$ is full rank, then*

$$0 \le m\text{-LIP}(\mathcal{M}(\{\mathbf{X}_t\})|\{\mathbf{Z}\}) \le 1,$$

where m-LIP$(\mathcal{M}(\{\mathbf{X}_t\})|\{\mathbf{Z}\})$ is defined in (1.2).

The maximum attainable m-LIP privacy is one, and it is desirable that a privacy mechanism should have m-LIP values close to one. This motivates the following definition:

Definition 2 ($\delta - m$-LIP)**.** Let $\{\mathbf{X}_t, \mathbf{Y}_t, \mathbf{Z}_t\}$ be as defined in Definition 1. Then for any $0 < \delta < 1$, a randomized mechanism $\mathcal{M}$ such that $\mathcal{M}(\{\mathbf{X}_t\}) = \{\mathbf{Y}_t\}$ is called a $\delta - m$-LIP *privacy mechanism* if m-LIP$(\mathcal{M}(\{\mathbf{X}_t\})|\{\mathbf{Z}\}) > 1 - \delta$.

Next we discuss how to obtain randomized mechanisms $\mathcal{M}$ such that the published series also honor first and second order utility requirements (to be described in the following sections).

2. Utility-Aware Mechanism: Multivariate All-pass Filters

We develop our proposed perturbation mechanism under a second-order stationary framework, as we are dedicated to preserving second-order utility, i.e., preservation of covariances, as further described below. However, we do provide a generalization to the non-stationary

setting, and also illustrate implementation of the proposed privacy mechanism under a setting where the observed series comprise a non-stationary mean aggregated with a sensitive stationary series. The development of the FLIP mechanism (MRH23) for univariate time series (which helped retain utility of the privatized data) relied upon the mathematical concept of all-pass filtering. Here we make non-trivial extensions to the multivariate setting. In particular, we define the notion of a multivariate all-pass filter, and develop a filter class that is particularly suitable for the privacy application.

2.1. **Multivariate All-Pass Filtering.** Suppose that $\{\mathbf{X}_t\}$ is a second-order stationary multivariate time series of dimension n, with components denoted by $X_{j,t}$ for $1 \leq j \leq n$. Denoting the process' autocovariance function by $\Gamma_{\mathbf{X}}(h) = \mathrm{Cov}(\mathbf{X}_{t+h}, \mathbf{X}_t)$ for $h \in \mathrm{Z}$, its spectral density is defined by $S_{\mathbf{X}}(\lambda) = \sum_h e^{-ih\lambda}\Gamma_{\mathbf{X}}(h)$ for $\lambda \in [-\pi, \pi]$. It is known that $S_{\mathbf{X}}(\lambda)$ has the hermitian property for each λ, i.e., $S_{\mathbf{X}}(\lambda)^* = S_{\mathbf{X}}(\lambda)$, where $\boldsymbol{A}^*$ denotes the conjugate transpose of any complex matrix $\boldsymbol{A}$. The spectral density $S_{\boldsymbol{X}}$ is a complex matrix-valued function from $[-\pi, \pi]$ to $\mathbb{C}^{n\times n}$, such that $S_{\boldsymbol{X}}(\lambda)$ is hermitian and non-negative definite for each $\lambda \in [-\pi, \pi]$. For the sensitive series to be protected, we will further assume the following:

Assumption PD: For each $\lambda \in [-\pi, \pi]$, the spectral density matrix $S(\lambda)$ is positive definite.

Assumption **PD** states that the multiple time series to be protected are not in the frequency domain at particular frequencies. From the perspective of implementation this assumption is not restrictive, since statistical estimation of the spectral density can be constrained so as to guarantee the positive definite property.

Next, we extend the concept of all-pass filtering to the multivariate case. Letting B denote the backshift operator (MP20), $\Psi(B) = \sum_k \Psi_k B^k$ defines a multivariate linear time-invariant filter, where each coefficient Ψ_k is a $n \times n$-dimensional matrix. This filter operates linearly on a time series $\{\mathbf{X}_t\}$, yielding an output time series $\{\mathbf{Y}_t\}$:

$$\mathbf{Y}_t = \Psi(B)\mathbf{X}_t = \sum_k \Psi_k \mathbf{X}_{t-k}. \tag{2.1}$$

Evaluating the filter at $z = e^{-i\lambda}$ yields the frequency response function of the filter, viz. $\Psi(z) = \sum_k \Psi_k z^k$. It follows that the filter output $\{\mathbf{Y}_t\}$ is also second-order stationary so long as the filter's frequency response function has a finite matrix norm at each λ. Then $S_{\mathbf{Y}}$ is related to $S_{\mathbf{X}}$ via (see (Bri01))

$$S_{\mathbf{Y}}(\lambda) = \Psi(z)\, S_{\mathbf{X}}(\lambda)\, \Psi(z^{-1})'. \tag{2.2}$$

When $n = 1$ (the univariate case), $\Psi(z)$ is an all-pass filter if $|\Psi(z)| = 1$ for all λ, and hence $S_{\mathbf{Y}} \equiv S_{\mathbf{X}}$. Extending this concept to the multivariate context ($n > 1$), we say that a matrix filter $\Psi(z)$ is all-pass if $S_{\mathbf{Y}} \equiv S_{\mathbf{X}}$ in (2.2). Though we might conjecture that it is sufficient that $\Psi(z)$ be unitary (i.e., $\Psi(z)\Psi(z)^* = \boldsymbol{I}$, the identity matrix) for (2.2) to hold for each λ, such a condition is too demanding in practice; for the relation (2.2) to hold for any spectral density $S_{\boldsymbol{X}}$, $\Psi(z)$ must commute with every spectral density matrix function (of the same order) at each frequency λ. This occurs if and only if $\Psi(z) = \boldsymbol{I}$. Thus, there are no universal all-pass filters in the matrix case other than the trivial identity filter.

Fortunately, for the data privacy application we only need to filter specific series whose spectral density is known to the data curator. Thus, it suffices to generate a class of filters that act as all-pass filters for a given spectral density $S_{\boldsymbol{X}}$. Given this background, we can

state the definition of the desired multivariate all-pass filter for a specified spectral density S as the following.

Definition 3 (S-Multivariate All-pass or S-MAP)**.** Given a spectral density matrix function S, a linear time invariant filter $\Psi(B)$ is said to be *S-Multivariate All-pass* (or S-MAP for short) if the relation

$$S(\lambda) = \Psi(z)\, S(\lambda)\, \Psi(z)^*$$

holds for all $\lambda \in [-\pi, \pi]$.

In view of Definition 3 and equation (2.2), if $\{\boldsymbol{X}_t\}$ is a second-order stationary time series with spectral density $S_{\boldsymbol{X}}$, and if $\boldsymbol{Y}_t = \Psi(B)\boldsymbol{X}_t$ is the filtered series, then the spectral density $S_{\mathbf{Y}}$ of $\{\boldsymbol{Y}_t\}$ equals $S_{\mathbf{X}}$ provided Ψ is $S_{\mathbf{X}}$-MAP. If Ψ is $S_{\mathbf{X}}$-MAP, then it implies that the autocovariances of $\{\mathbf{Y}_t\}$ are the same as those of $\{\mathbf{X}_t\}$. Clearly, given an $n-$dimensional spectral density $S_{\boldsymbol{X}}$, $\Psi(z) = \boldsymbol{I}$ is a trivial $S_{\mathbf{X}}$-MAP filter, but there are many more choices.

2.2. **$S_{\mathbf{X}}$-MAP and Second-Order Utility.** The class of $S_{\mathbf{X}}$-MAP is infinite dimensional. For specific applications, it will be advantageous to find suitable special cases for which closed-form solutions are readily available. We next develop a special case that will be useful in our more general treatment. Suppose that $\{\mathbf{X}_t\}$ is a white noise time series of covariance matrix $\boldsymbol{I}$, so that $S_{\mathbf{X}}(\lambda) = \boldsymbol{I}$. Then the all-pass condition becomes

$$\boldsymbol{I} = \Psi(z)\, \Psi(z^{-1})' \tag{2.3}$$

for $z = e^{-i\lambda}$, and all $\lambda \in [-\pi, \pi]$ (i.e., $\Psi(z)$ is unitary for all λ). One way to parameterize such unitary functions is through the matrix *cepstral* representation discussed in (HMW17). Consider a matrix Laurent series $\Omega(z) = \sum_{k\in \mathrm{Z}} \Omega_k z^k$ that is related to $\Psi(z)$ via the matrix exponential, viz.

$$\Psi(z) = \exp\{\Omega(z)\}. \tag{2.4}$$

Then $\Omega(z)$ is the cepstral representation of $\Psi(z)$, and the Ω_k are the matrix cepstral coefficients. Then (2.3) implies that the all-pass condition is

$$\boldsymbol{I} = \exp\{\Omega(z)\}\, \exp\{\Omega(z^{-1})'\},$$

using the transpose property of the matrix exponential. Recall that $z = e^{-i\lambda}$, so $z^{-1} = e^{i\lambda} = \bar{z}$. If $\Omega(z) = -\Omega(\bar{z})'$, then (since $\Omega(z)$ and $-\Omega(z)$ commute)

$$\exp\{\Omega(z)\}\, \exp\{\Omega(z^{-1})'\} = \exp\{\Omega(z)\}\, \exp\{-\Omega(z)\} = \exp\{\Omega(z) - \Omega(z)\} = \exp\{0\} = \boldsymbol{I}.$$

This condition on $\Omega(z)$ means that $\Omega_k = -\Omega'_{-k}$ for $k \in \mathrm{Z}$, implying Ω_0 is a skew-symmetric matrix. Hence, anti-symmetric cepstral coefficients correspond to a unitary filter $\Psi(z)$.

We will use the parameterization of the unitary operators in terms of its cepstral representation to generate a suitable parametric class of S-MAP filters for any specified spectral density S. Under the positive definiteness assumption, at each frequency $\lambda \in [-\pi, \pi]$, the spectral density matrix $S_{\boldsymbol{X}}(\lambda)$ admits a non-singular square root $S^+_{\boldsymbol{X}}(\lambda)$ (McE18), i.e., for each $\lambda \in [-\pi, \pi]$ we can find a full rank matrix $S^+_{\boldsymbol{X}}(\lambda)$ such that

$$S_{\boldsymbol{X}}(\lambda) = S^+_{\boldsymbol{X}}(\lambda)\, S^+_{\boldsymbol{X}}(\lambda)^*. \tag{2.5}$$

If the filter $\Psi(z)$ is also non-singular, then by the relation (2.2), $S_{\boldsymbol{Y}}(\lambda)$ is also positive definite at each frequency, and hence admits non-singular square roots $S^+_{\boldsymbol{Y}}(\lambda)$. Thus

$$S^+_{\boldsymbol{Y}}(\lambda)\, S^+_{\boldsymbol{Y}}(\lambda)^* = \Psi(z)\, S^+_{\boldsymbol{X}}(\lambda)\, S^+_{\boldsymbol{X}}(\lambda)\, \Psi(z)^*.$$

For $\Psi(z)$ to be $S_{\mathbf{X}}$-MAP, a sufficient condition is $S^+_{\boldsymbol{Y}}(\lambda) = S^+_{\boldsymbol{X}}(\lambda)$ for all $\lambda \in [-\pi, \pi]$. Hence we seek $\Psi(z)$ such that

$$S^+_{\boldsymbol{X}}(\lambda)\, S^+_{\boldsymbol{X}}(\lambda)^* = (\Psi(z)\, S^+_{\boldsymbol{X}}(\lambda))\, (\Psi(z)\, S^+_{\boldsymbol{X}}(\lambda))^*.$$

This condition implies that $S^+_{\boldsymbol{X}}(\lambda)\, U(z) = \Psi(z)\, S^+_{\boldsymbol{X}}(\lambda)$ for some unitary matrix $U(z)$ (since for nonsingular matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, $\boldsymbol{A}\,\boldsymbol{A}^* = \boldsymbol{B}\,\boldsymbol{B}^*$ if and only if $\boldsymbol{A}\,\boldsymbol{U} = \boldsymbol{B}$ for some unitary matrix $\boldsymbol{U}$). Therefore, $\Psi(z) = S^+_{\boldsymbol{X}}(\lambda)\, U(z)\, S^+_{\boldsymbol{X}}(\lambda)^{-1}$ must hold. Thus, for a given spectral density S, a class of S-MAP filters is given by

$$\Psi(z) = S^+(\lambda)\, U(z)\, S^+(\lambda)^{-1}. \tag{2.6}$$

The implications of (2.6) are substantial. It means that given a spectral density S, that we could select the desired all-pass filters from a rich class of S-MAP filters obtained by rotating the expression in (2.6) over the unitary group, and everything can be computed in closed-form. This provides flexibility in the selection of the privacy mechanism while optimizing privacy measures to attain a privacy-utility balance. Based on the parameterization of the unitary operator through the cepstral representation, a general class of S-MAP filters for a given n-dimensional positive definite spectral density function S can thus be defined as

$$\mathcal{F}_S = \{S^+(\lambda)\, U(z)\, S^+(\lambda)^{-1} : U(z) = \exp\{\sum_{k\in\mathbb{Z}} \Omega_k z^k\},\ \Omega_{-k} = -\Omega_k'\}, \tag{2.7}$$

where $S^+(\lambda)$ is a square root of $S(\lambda)$ for each $\lambda \in [-\pi, \pi]$.

The preservation of the autocorrelation structure of $\{\mathbf{X}_t\}$ is referred to as *second-order utility*, and mathematically is the requirement that $\Gamma_{\mathbf{X}}(h) = \Gamma_{\mathbf{Y}}(h)$ for all $h \in \mathrm{Z}$. This is equivalent to the requirement that $S_{\mathbf{Y}} = S_{\mathbf{X}}$; clearly, one such privacy mechanism that preserves second-order utility is all-pass filtering via $S_{\mathbf{X}}$-MAP filters belonging to class (2.7).

3. Privacy-Utility Optimization for Multiple Time Series

3.1. **The Stationary Case.** Let $\{\mathbf{X}_t\}$, $\{\mathbf{Y}_t\}$, and $\{\mathbf{Z}_t\}$ be jointly weakly stationary multivariate time series. The notation for the autocovariances and cross-covariances is $\Gamma_{\mathbf{UV}}(h) = \mathrm{Cov}(\mathbf{U}_{t+h}, \mathbf{V}_t)$ for $\mathbf{U}, \mathbf{V} \in \{\mathbf{X}, \mathbf{Z}, \mathbf{Y}\}$. The spectral density matrix of the joint process at any given frequency $\lambda \in [-\pi, \pi]$ is given by

$$S_{\mathbf{X},\mathbf{Y},\mathbf{Z}}(\lambda) = \begin{pmatrix} S_{\mathbf{XX}}(\lambda) & S_{\mathbf{XY}}(\lambda) & S_{\mathbf{XZ}}(\lambda) \\ S_{\mathbf{YX}}(\lambda) & S_{\mathbf{YY}}(\lambda) & S_{\mathbf{YZ}}(\lambda) \\ S_{\mathbf{ZX}}(\lambda) & S_{\mathbf{ZY}}(\lambda) & S_{\mathbf{ZZ}}(\lambda) \end{pmatrix}, \tag{3.1}$$

where $S_{\mathbf{UV}}(\lambda) = \sum_h e^{-ih\lambda}\Gamma_{\mathbf{UV}}(h)$ for $\mathbf{U}, \mathbf{V} \in \{\mathbf{X}, \mathbf{Z}, \mathbf{Y}\}$ is the cross-spectral density (BD). For convenience we will denote the individual spectral densities using a single subscript, i.e., as $S_{\mathbf{X}}(\lambda)$, $S_{\mathbf{Y}}(\lambda)$, and $S_{\mathbf{Z}}(\lambda)$. We suppose that the spectral matrix $S_{\mathbf{X},\mathbf{Z}}$ is well-known to both the data-publishing agency and potential adversaries engaged in what we term an "augury" attack. This is the scenario described in Section 1, wherein the adversary possesses an external source of information $\{\mathbf{Z}_t\}$, and the publishing agency applies a privacy mechanism $\mathcal{M}$ to $\{\mathbf{X}_t\}$, thereby producing $\{\mathbf{Y}_t\}$, which is viewed as a proxy for the sensitive data that preserves some features of interest.

To establish a more convenient form of the privacy measure when the randomized mechanism is an $S_{\mathbf{X}}$-MAP filter Ψ, we first identify the components of the privacy measure in terms of the spectral density of the joint process $\{\mathbf{X}_t, \mathbf{Y}_t, \mathbf{Z}_t\}$. The following proposition

connects the elements of the m-LIP measure to the components of the spectral density of $\{\mathbf{X}_t, \mathbf{Y}_t, \mathbf{Z}_t\}$. We employ the following notation: $\langle u \rangle = (2\pi)^{-1} \int_{-\pi}^{\pi} u(\lambda) d\lambda$.

Proposition 3.1. *Let $\{\mathbf{X}_t\}$, $\{\mathbf{Y}_t\}$, and $\{\mathbf{Z}_t\}$ be jointly weakly stationary multivariate time series, with positive definite spectral density (3.1). Then the conditional spectral densities $S_{\mathbf{X}|\mathbf{Z}}$, $S_{\mathbf{Y}|\mathbf{Z}}$, and $S_{\mathbf{XY}|\mathbf{Z}}$ have formulas*

$$S_{\mathbf{X}|\mathbf{Z}} = S_{\mathbf{X}} - S_{\mathbf{XZ}} S_{\mathbf{Z}}{}^{-1} S_{\mathbf{ZX}}, \quad S_{\mathbf{Y}|\mathbf{Z}} = S_{\mathbf{Y}} - S_{\mathbf{YZ}} S_{\mathbf{Z}}{}^{-1} S_{\mathbf{ZY}}, \quad S_{\mathbf{XY}|\mathbf{Z}} = S_{\mathbf{XY}} - S_{\mathbf{XZ}} S_{\mathbf{Z}}{}^{-1} S_{\mathbf{ZY}}.$$

Since the variance of a stationary process equals the weighted integral of its spectral density, it immediately follows from Proposition 3.1 that

$$\mathrm{Var}[\mathbf{X}_t | \{\mathbf{Z}_t\}] = \langle S_{\mathbf{X}|\mathbf{Z}} \rangle, \quad \mathrm{Var}[\mathbf{Y}_t | \{\mathbf{Z}_t\}] = \langle S_{\mathbf{Y}|\mathbf{Z}} \rangle, \quad \mathrm{Cov}[\mathbf{X}_t, \mathbf{Y}_t | \{\mathbf{Z}_t\}] = \langle S_{\mathbf{XY}|\mathbf{Z}} \rangle.$$

In the case that the published series $\{\mathbf{Y}_t\}$ is generated by application of a linear filter $\Psi(B)$, as in 2.1, application of Proposition 3.1 provides an expression for the m-LIP measure as

$$m\text{-LIP}(\Psi(\{\mathbf{X}_t\}) | \{\mathbf{Z}_t\}) = 1 - \frac{\det\left[\langle S_{\mathbf{X}|\mathbf{Z}} \Psi^* \rangle \langle \Psi S_{\mathbf{X}|\mathbf{Z}} \Psi^* \rangle^{-1} \langle \Psi S_{\mathbf{X}|\mathbf{Z}} \rangle\right]}{\det \langle S_{\mathbf{X}|\mathbf{Z}} \rangle}. \tag{3.2}$$

The value of zero occurs when $\mathrm{Var}[\mathbf{X}_t | \mathbf{Y}_t, \{\mathbf{Z}_t\}]$ is singular, corresponding to complete predictability of $\mathbf{X}_t$ on the basis of $\mathbf{Y}_t$ and $\{\mathbf{Z}_t\}$; since $S_{\mathbf{X}|\mathbf{Z}}$ is positive definite, it follows that $\mathrm{Var}[\mathbf{X}_t | \{\mathbf{Z}_t\}]$ is non-singular, so that the culprit in disclosing $\mathbf{X}_t$ is $\mathbf{Y}_t$, and not $\{\mathbf{Z}_t\}$. On the other hand, when m-LIP equals one it must be the case that $\langle S_{\mathbf{X}|\mathbf{Z}} \Psi^* \rangle \langle \Psi S_{\mathbf{X}|\mathbf{Z}} \Psi^* \rangle^{-1} \langle \Psi S_{\mathbf{X}|\mathbf{Z}} \rangle$ is singular, i.e., that $\mathrm{Var}[\mathbf{X}_t | \{\mathbf{Z}_t\}] - \mathrm{Var}[\mathbf{X}_t | \mathbf{Y}_t, \{\mathbf{Z}_t\}]$ is singular. This means that $\mathbf{Y}_t$ incurs no additional ability to predict certain linear combinations of $\mathbf{X}_t$ over and above what is already furnished by $\{\mathbf{Z}_t\}$.

In the context of the augury solution, any $S_{\mathbf{X}}$-MAP filter Ψ guarantees perfect second-order utility. Consequently, the selection of Ψ should primarily align with the maximum privacy requirements. In particular, we seek an "optimal" Ψ to maximize the privacy metric $m\text{-LIP}(\Psi, S_{\mathbf{X}|\mathbf{Z}})$:

$$\Psi_{opt} = \arg\max_{\Psi} m\text{-LIP}(\Psi(\{\mathbf{X}_t\}) | \{\mathbf{Z}_t\}). \tag{3.3}$$

The optimization is over the class of $S_{\mathbf{X}}$-MAP filters. Given that the objective function is a nonlinear non-convex function of the filter, the optimization is rendered feasible by narrowing the class of all-pass filters. We use the parameterized class $\mathcal{F}_S$ in (2.7) as the set over which the objective function is optimized. Thus, given a conditional spectral density $S_{\boldsymbol{X}|\boldsymbol{Z}}$, the optimal filter is defined as

$$\Psi_{opt} = \arg\min_{\Psi \in \mathcal{F}_{S_{\boldsymbol{X}|\boldsymbol{Z}}}} \frac{\det\left[\langle S_{\mathbf{X}|\mathbf{Z}} \Psi^* \rangle \langle \Psi S_{\mathbf{X}|\mathbf{Z}} \Psi^* \rangle^{-1} \langle \Psi S_{\mathbf{X}|\mathbf{Z}} \rangle\right]}{\det \langle S_{\mathbf{X}|\mathbf{Z}} \rangle}. \tag{3.4}$$

Since the S-MAP filters in $\mathcal{F}_S$ are defined with respect to unitary matrices, the optimization effectively reduces to a search over the set of unitary operators $U(z)$. Consequently, parameterizing unitary operators via their cepstral representation (2.4), we can perform the optimization over Euclidean space.

3.2. **Extension of the Framework to Non-stationary Time Series.** The development in Proposition 3.1 is restricted to stationary series. It is possible to generalize unto a scenario where the time series are difference stationary, i.e., where there exist polynomials $\delta_X(z)$ and $\delta_Z(z)$ such that $\underline{\mathbf{X}}_t = \delta_X(B)\mathbf{X}_t$ and $\underline{\mathbf{Z}}_t = \delta_Z(B)\mathbf{Z}_t$ are stationary, and all the roots of the polynomials have unit modulus. However, in order for our notion of privacy to be tractable we require that the projection error $\mathbf{X}_t - \mathbf{X}_t|\{\mathbf{Z}_t\}$ is a stationary process, and this places a restriction on the joint process $(\{\mathbf{X}_t\}, \{\mathbf{Z}_t\})$. One framework that satisfies such a requirement is that of signal extraction, where it is assumed that the attacker's information consists of the private data as a signal, but obfuscated by noise. Formally, we assume in this subsection that

$$\mathbf{Z}_t = \Lambda(B)\mathbf{X}_t + \mathbf{W}_t,$$

where $\Lambda(z)$ is some matrix Laurent series (which can be rectangular, allowing for the possible difference in dimension between $\mathbf{Z}_t$ and $\mathbf{X}_t$) and $\{\mathbf{W}_t\}$ is a non-stationary noise process with differencing polynomial $\delta_W(z)$. We adopt the classical signal extraction assumptions discussed in (Bel84): the scalar polynomials $\delta_X(z)$ and $\delta_W(z)$ are relatively prime, and the differenced series $\{\underline{X}_t\}$ and $\{\underline{W}_t\}$ are uncorrelated both with one another and with the initial values of the $\{\mathbf{Z}_t\}$ process. The case of multivariate signal extraction is treated in (MT15), and applying those results we find that $\mathbf{X}_t|\{\mathbf{Z}_t\} = \Phi(B)\mathbf{Z}_t$, with

$$\Phi(e^{-i\lambda}) = S_{\mathbf{X}}(\lambda)\,\Lambda(e^{-i\lambda})^*\,[S_{\mathbf{Z}}(\lambda)]^{-1}\,|\delta_W(e^{-i\lambda})|^2,$$

where $S_{\mathbf{X}}$ and $S_{\mathbf{Z}}$ are the spectral densities of $\{\mathbf{X}_t\}$ and $\{\mathbf{Z}_t\}$, respectively. An application of the Sherman-Woodbury identity yields

$$\Phi(e^{-i\lambda}) = M(\lambda)^{-1}\,\Lambda(e^{-i\lambda})^*\,[S_{\mathbf{W}}(\lambda)]^{-1}\,|\delta_W(e^{-i\lambda})|^2,$$

where $S_{\mathbf{W}}$ is the spectral density of $\{\mathbf{W}_t\}$ and

$$M(\lambda) = [S_{\mathbf{X}}(\lambda)]^{-1}\,|\delta_X(e^{-i\lambda})|^2 + \Lambda(e^{-i\lambda})^*\,[S_{\mathbf{W}}(\lambda)]^{-1}\,\Lambda(e^{-i\lambda})\,|\delta_W(e^{-i\lambda})|^2.$$

With these formulas, it is straightforward to check that the projection error $\mathbf{X}_t - \mathbf{X}_t|\{\mathbf{Z}_t\}$ is stationary, uncorrelated with $\{\mathbf{Z}_t\}$, and has spectral density $S_{\mathbf{X}|\mathbf{Z}}(\lambda) = M(\lambda)^{-1}$. If we apply a linear filtering mechanism $\Psi(B)$ as in (2.1), then it follows that the same formula (3.2) for m-LIP holds, but now with $S_{\mathbf{X}|\mathbf{Z}} = M^{-1}$. In this way, we can generalize from the stationary case.

3.3. **Worst Case Attacker.** We now consider a special case of Proposition 3.1 where the attacker has almost full information about the sensitive time series. This is of interest because it allows us to compute a privacy measure that does not explicitly depend on an unknown $\{\mathbf{Z}_t\}$, but rather is computed in terms of a particular choice of $\{\mathbf{Z}_t\}$ that is deemed to represent a "worst case scenario." The heuristic is that if we can protect against such a worst case attack, then we will also have privacy protection against other attacks (which are ostensibly less ferocious, or damaging). There are two ways of thinking about this worst case attack. First, we may suppose the attacker has no prior information at all, in which case it may be easy for them to learn something from our release. Hence, the privacy measure would be computed with all conditioning upon $\mathbf{Z}$ removed. Second, we may suppose the attacker knows everything about the sensitive process except $\mathbf{X}_t$ itself, i.e.,

$$\mathbf{Z}_s = \begin{cases} \mathbf{X}_s & \text{if } s \neq t \\ \text{NA} & \text{if } s = t. \end{cases}$$

Note that if the attacker also knew $\mathbf{X}_t$, then privacy is already impossible; our assumption is that the attacker's information stops just short of this undesirable situation. We proceed to calculate (3.2) for this particular worst case scenario, and for simplicity focus upon the case where both time series are stationary. First, we must calculate $\mathbf{X}_t|\{\mathbf{Z}_t\} = \Pi(B)\mathbf{Z}_t$; but this $\Pi(B)$ is the multivariate missing value filter, with formula given by (see (MP22)) $\Pi(e^{-i\lambda}) = \mathbf{I} - \langle S_{\mathbf{X}}^{-1}\rangle^{-1} S_{\mathbf{X}}(e^{-i\lambda})^{-1}$. The conditional variance is $\langle S_{\mathbf{X}|\mathbf{Z}}\rangle = \langle S_{\mathbf{X}}^{-1}\rangle^{-1}$. Applying the same methods, we can compute the related quantity $\mathbf{Y}_t|\{\mathbf{Z}_t\} = \Upsilon(B)\mathbf{Z}_t$, where $\{\mathbf{Y}_t\}$ is jointly stationary with $\{\mathbf{X}_t\}$ and $\{\mathbf{Z}_t\}$, but need not be the output of a linear mechanism. It can be checked that the optimal filter is

$$\Upsilon(e^{-i\lambda}) = \left(S_{\mathbf{YX}}(e^{-i\lambda}) - \langle S_{\mathbf{YX}} S_{\mathbf{X}}^{-1}\rangle \langle S_{\mathbf{X}}^{-1}\rangle^{-1}\right) S_{\mathbf{X}}(e^{-i\lambda})^{-1}.$$

Further calculations reveal that the conditional variance and conditional covariances are

$$\begin{aligned}\langle S_{\mathbf{Y}|\mathbf{Z}}\rangle &= \langle S_{\mathbf{Y}|\mathbf{X}}\rangle + \langle S_{\mathbf{YX}} S_{\mathbf{X}}^{-1}\rangle \langle S_{\mathbf{X}}^{-1}\rangle^{-1} \langle S_{\mathbf{X}}^{-1} S_{\mathbf{XY}}\rangle \\ \langle S_{\mathbf{XY}|\mathbf{Z}}\rangle &= -\langle S_{\mathbf{X}}^{-1}\rangle^{-1} \langle S_{\mathbf{X}}^{-1} S_{\mathbf{XY}}\rangle.\end{aligned}$$

Finally, the scalar privacy measure $1 - \det\left[\langle S_{\mathbf{XY}|\mathbf{Z}}\rangle \langle S_{\mathbf{Y}|\mathbf{Z}}\rangle^{-1} \langle S_{\mathbf{YX}|\mathbf{Z}}\rangle\right] / \det\langle S_{\mathbf{X}|\mathbf{Z}}\rangle$ of Proposition 3.1 has the formula

$$1 - \frac{\det\left[\langle S_{\mathbf{X}}^{-1} S_{\mathbf{XY}}\rangle \left(\langle S_{\mathbf{Y}|\mathbf{X}}\rangle + \langle S_{\mathbf{YX}} S_{\mathbf{X}}^{-1}\rangle \langle S_{\mathbf{X}}^{-1}\rangle^{-1} \langle S_{\mathbf{X}}^{-1} S_{\mathbf{XY}}\rangle\right)^{-1} \langle S_{\mathbf{YX}} S_{\mathbf{X}}^{-1}\rangle\right]}{\det\langle S_{\mathbf{X}}^{-1}\rangle}.$$

For the case of a linear filtering privacy mechanism, we find that m-LIP is

$$1 - \frac{\det\left[\langle \Psi^* \rangle \left(\langle \Psi \rangle \langle S_{\mathbf{X}}^{-1}\rangle^{-1} \langle \Psi^* \rangle\right)^{-1} \langle \Psi \rangle\right]}{\det\langle S_{\mathbf{X}}^{-1}\rangle},$$

which equals zero when $\langle \Psi \rangle$ (coefficient zero of the filter) is invertible. In other words, privatization with a linear prediction mechanism is impossible when the value at the single time point can be reliably predicted by the knowledge of the series at other time points. This is a major challenge in the dependent data situation where predictability of sensitive values based on neighboring values pose significant challenges toward developing formal privacy mechanisms.

4. Feasible Implementation of m-LIP

We assume a framework where each observed series has a time-varying mean function that is not considered sensitive, and only the stationary part of the de-meaned series will be used to build the formal privacy framework. We write the de-meaned processes without a tilde, i.e.,

$$\begin{pmatrix} \bar{\mathbf{X}}_t \\ \bar{\mathbf{Y}}_t \\ \bar{\mathbf{Z}}_t \end{pmatrix} = \begin{pmatrix} \mu_t^{\mathbf{X}} \\ \mu_t^{\mathbf{Y}} \\ \mu_t^{\mathbf{Z}} \end{pmatrix} + \begin{pmatrix} \mathbf{X}_t \\ \mathbf{Y}_t \\ \mathbf{Z}_t \end{pmatrix}, \tag{4.1}$$

where $\{\mathbf{X}_t, \mathbf{Y}_t, \mathbf{Z}_t\}$ are jointly stationary and $\mu_t^{\mathbf{Y}}$ is equal to $\mu_t^{\mathbf{X}}$. For implementation, the non-stationary means are estimated and removed, and the privacy-utility framework based on the $S_{\mathbf{X}}$-MAP filter is applied to the de-meaned process to obtain $\{\mathbf{Y}_t\}$. Finally, the estimated mean of $\bar{\mathbf{X}}_t$ is added back, and the sum $\bar{\mathbf{Y}}_t$ is published.

In practice, selection of an optimal S-MAP filter according to (3.4) is based upon a spectral density S estimated from the available data (or based on prior knowledge). To use the class of S-MAP filters in (2.7), one needs to obtain square roots of a positive definite spectral density. Thus, the spectral density estimation procedure must constrain the estimator to be positive definite. Subsequent to the estimation of the spectral density, the spectral square root factors S^+ need to be computed at each frequency. Then the optimal S-MAP filter is obtained using optimization of the criterion (3.4) over the parametric class (2.7) that is defined based on the estimated spectral factor. Finally, the filter coefficients associated with the optimal filter need to be computed using the inverse Fourier transform of the filter. This section describes the step-by-step process of implementing the m-LIP privacy mechanism.

4.1. **Positive Definite Estimation of Spectral Densities.** For implementation of the m-LIP via spectral density estimation it is imperative that $\hat{S}_{\mathbf{X},\mathbf{Z}}$ – and hence the Schur complement $\hat{S}_{\mathbf{X}|\mathbf{Z}}$ – be positive definite. In particular, with nonparametric approaches we must be careful to ensure this positive definite property is exhibited in the spectral density estimate.

Any such spectral estimator yields a $\hat{S}_{\mathbf{X}}$-MAP filter rather than a S-MAP filter, and thus there will be some degradation of second-order utility due to statistical estimation error of the spectral density; this is different from the univariate case explored in (MRH23), wherein an all-pass filter can be constructed without knowing the spectral density of the input process. However, it can be argued that the practical utility that practitioners care about is based on the finite sample at hand, and the preservation of *sample* autocovariances, i.e., $\hat{\Gamma}_{\mathbf{X}}(h) = \hat{\Gamma}_{\mathbf{Y}}(h)$ for all $h \in \mathrm{Z}$. Such a "sample" – or feasible – second-order utility is equivalent to $\{\mathbf{X}_t\}$ and $\{\mathbf{Y}_t\}$ having the same periodogram. Hence, setting $\hat{S}_{\mathbf{X}}$ to be the periodogram would guarantee feasible second-order utility, but unfortunately the multivariate periodogram is a rank one matrix for all λ, and hence violates our positive definite requirement. Therefore, we recognize there may be some feasible loss of sample utility due to positive definite spectral density estimation; however, as sample size increases these estimates will be consistent for the true $\hat{S}_{\mathbf{X},\mathbf{Z}}$, as will the sample autocovariances for the process' autocovariances, and thus for large sample sizes second-order utility will approximately hold.

Given detrended data $\{\boldsymbol{W}_t\} = \{\boldsymbol{X}_t, \boldsymbol{Z}_t\}$, there are several different options for obtaining positive definite spectral density estimates. One option is to fit a parametric model, such as an order p vector autoregressive process (or VAR(p)), and use the spectral density of that model evaluated at the estimated parameters. Another option is to use a non-parametric estimator that is constrained to be positive definite. In this article, we use the non-parametric kernel estimator of $S_{\mathbf{X},\mathbf{Z}}$ proposed in (Pol11). In (Pol11), the author uses a flat-top kernel because it is an infinite-order kernel, and therefore is capable of achieving higher-order accuracy. The disadvantage of flat-top kernels is that they are not necessarily positive semi-definite. For this reason, the author lets $\epsilon_T > 0$ be some chosen sequence decreasing to zero as $T \to \infty$, and truncates the eigenvalues of the flat-top taper estimator to $[\epsilon_T, \infty)$.

We choose $\epsilon_T = 1/T$ here and employ the flat-top taper method on the sample autocovariances to get a positive definite (PD) estimator. Let $\hat{S}_{\boldsymbol{X},\boldsymbol{Z}}(\lambda)$ be the flat-top taper PD estimator of the spectral density (for the de-meaned process) obtained using $\epsilon_T = 1/T$ for a sample of size T. The top left block of the estimator will be denoted as $\hat{S}_{\boldsymbol{X}}$, and is the PD

estimator of $S_{\boldsymbol{X}}$, and the Schur complement $\hat{S}_{\boldsymbol{X}|\boldsymbol{Z}}$ will be the estimator of the projection error spectral density.

4.2. **Spectral Factorization.** The multivariate spectral factorization problem is fundamental in spectral analysis, wherein the objective is to obtain a vector moving average (VMA) representation of order q that corresponds to a given set of autocovariances, denoted as $\Gamma(0), \ldots, \Gamma(q-1), \Gamma(q)$. The requirement is that $\sum_{|h|\leq q} \Gamma(h) e^{-i\lambda h}$ must be positive definite for all values of the frequency parameter $\lambda \in [-\pi, \pi]$.

There are several available methods for spectral factorization; we follow the method of Bauer (Bau55), as summarized in (McE17). First, we approximate the spectral density $S(\lambda)$ by $\sum_{|h|\leq q} \Gamma(h) e^{-i\lambda h}$ for q large; for simplicity of exposition, suppose this holds exactly, i.e.,

$$S(\lambda) = \sum_{h=-q}^{q} \Gamma(h) e^{-i\lambda h}.$$

Bauer's method first forms the block Toeplitz covariance matrix of a time series sample of length m (where m is taken as large as computationally feasible), and secondly the modified Cholesky decomposition (MCD) is computed. The lower left block row of the Cholesky factor consists (as $m \longrightarrow \infty$) of the autocovariances $\Gamma(q), \Gamma(q-1), \ldots, \Gamma(0)$, as described in (McE17). Then the spectral factorization can be concisely represented as

$$S(z) = S^{+}(z)\, S^{+}(z)^{*} = \Theta(z)\, \Sigma\, \Theta(z)^{*},$$

where the spectral factor $S^{+}(z)$ assumes the form $S^{+}(z) = \Theta(z)\, \Sigma^{1/2}$. Here $\Theta(B) = \sum_{k=0}^{q} \Theta_k B^k$ is an order q matrix polynomial in B such that $\Theta_0 = \boldsymbol{I}$, and whose coefficients are the VMA coefficients. Also, Σ is the covariance matrix of the innovations. The spectral factor $\hat{S}^{+}_{\boldsymbol{X}}(\lambda)$ obtained from using the Bauer algorithm on the flat-top taper PD estimator $\hat{S}_{\boldsymbol{X}}(\lambda)$ is used in the design of S-MAP filters.

4.3. **Parameterization of the S-MAP Class.** Once the estimated spectral factor $\hat{S}^{+}_{\boldsymbol{X}}(\lambda)$ has been obtained, one can construct the parametric class $\mathcal{F}_S$ of S-MAP filters given in (2.7) by setting $S = \hat{S}^{+}_{\boldsymbol{X}}(z)$. The free parameters of the class are obtained from the matrices Ω_k in the cepstral representation $U(z) = \exp\{\Omega(z)\} = \exp\{\sum_{k\in\mathbb{Z}} \Omega_k z^k\}$ of the unitary operator. We can parameterize $\Omega(z)$ by allowing the matrix entries of Ω_k for $k > 0$ to be any real number, and for $k < 0$ we set $\Omega_k = -\Omega'_{-k}$. For $k = 0$, we only need to constrain Ω_0 to be skew-symmetric, which is achieved by freely parameterizing the lower triangular portion of the matrix, and enforcing that the upper triangular portion be equal to the negative transpose of the lower portion (and the diagonal entries are zero). For feasible implementation, we need to truncate the Laurent series $\Omega(z)$ at a finite stage, say r. Thus, the class of filters $\Psi(z)$ that we are choosing to optimize over are of the form

$$\Psi_r(z) = \hat{S}^{+}_{\boldsymbol{X}}(\lambda)\, \exp\{\sum_{k=-r}^{r} \Omega_k z^k\}\, \hat{S}^{+}_{\boldsymbol{X}}(\lambda)^{-1}, \tag{4.2}$$

where $\Omega_{-k} = -\Omega'_k$ for all $k \geq 0$. The truncation stage r has to be chosen by the data curator, and can be done by examining the optimal privacy value for several different choices of r. Given r, the number of free parameters in the class is $n_r = rn^2 + \binom{n}{2}$, which is linear in the cepstral length r and quadratic in n.

4.4. **Optimal All-pass Filter Selection.** In view of the filters described in (4.2), the criterion (3.4) can be optimized with respect to the n_r free parameters in $\Omega_0, \Omega_1, \ldots, \Omega_r$. However, the complicated nature of the m-LIP objective function precludes an analytical solution, and we instead proceed via non-linear optimization techniques.

Our numerical method leverages an optimization algorithm known as AGMsDR (NGGD21) that is suitable for nonlinear nonconvex optimization. While conventional optimization techniques like Brent or L-BFGS typically yield dependable results, our preference for AGMsDR stems from its specialized capability to address non-convex and non-smooth functions. Although our objective function is not inherently non-smooth, its non-convex nature makes the AGMsDR algorithm particularly attractive. Additionally, this method proves valuable in situations where more commonly employed methods may encounter convergence issues.

Consider the cepstral series $\Omega(z)$ truncated to some order r, so that

$$\Omega(z) = \sum_{k=-r}^{r} \Omega_k z^k = \Omega_0 + \sum_{k=1}^{r} \Omega_k z^k - \sum_{k=1}^{r} \Omega'_k z^{-k}.$$

Let ϑ denote the vector of n_r real parameters corresponding to the entries of the cepstral matrices Ω_k for $k = 0, 1, \ldots, r$. The unitary operator $U(z)$ then becomes a function of the free parameters, and we denote it as $U(z;\vartheta)$. Also, let $\Psi_r(z;\vartheta) = S^{+}_{\mathbf{X}}\, U(z;\vartheta)\, S^{+^{-1}}_{\mathbf{X}}$. Then the solution to the optimal filter problem (3.4) can be re-expressed as

$$\vartheta_{opt} = \arg\min_{\vartheta} \frac{\det\left[\langle S_{\mathbf{X}|\mathbf{Z}} \Psi_r(z;\vartheta)^*\rangle \langle \Psi_r(z;\vartheta)\, S_{\mathbf{X}|\mathbf{Z}}\, \Psi_r(z;\vartheta)^*\rangle^{-1} \langle \Psi_r(z;\vartheta)\, S_{\mathbf{X}|\mathbf{Z}}\rangle\right]}{\det\langle S_{\mathbf{X}|\mathbf{Z}}\rangle}, \tag{4.3}$$

with $\Psi_{opt}(z) = \Psi_r(z;\vartheta_{opt})$. For initialization of the ϑ parameters we draw a random sample of size n_r from the standard normal distribution, and set the initial values equal to the obtained sample. After the optimal filter $\Psi_{opt}(z)$ has been determined, the filter coefficients are obtained by Fourier inversion, viz. $\Psi_k = \langle z^{-k}, \Psi_{opt}(z)\rangle$.

4.5. **Estimation of Trend and Forecast Extension.** Before the application of the estimated filter to the data, the deterministic trend needs to be estimated and removed from the multiple time series. Then after the application of the filter, the estimated trend is added back to the privatized times series. One could apply the non-stationary filters described in section 3.2 to address non-stationary trends, but for the present application we follow the two-stage procedure where the non-stationary trends are estimated, removed, and then added back.

Trend estimation can be done using different available software. For this article, we used the differencing method to achieve the detrended series using the *diff*() function (details in Section 5.2). After the removal of trends from each of the series, we obtain the detrended data, which is then used for filtering. The filter is two-sided and of finite length, say M on each side. To obtain a series with the same length as the original data after filtering, we extend the detrended series by M time points on each side by using one-sided forecasts. Since we are assuming that the spectral density is known for the original series, we use this same spectral density to generate optimum one-sided $h-$step ahead forecasts for $h = 1, \ldots, M$. After we obtain the filtered series by applying the filter S-MAP to the detrended series, we add back the estimated trends. A privatized series with a trend is thereby generated.

4.6. **Realized Utility.** Due to the error that occurred during spectral estimation, and due to finite sample effects, there can be utility loss; we measure this loss through the Frobenius norm. The Frobenius Discrepancy (FD) (see (MR23)) of the two n-variate spectral density matrices $S_{\mathbf{X}}$ and $S_{\mathbf{Y}}$ is the average (over frequencies) of the squared Frobenius norm of their difference, viz.

$$\mathrm{FD}(S_{\mathbf{X}}, S_{\mathbf{Y}}) = \langle \|S_{\mathbf{X}} - S_{\mathbf{Y}}\|_F^2 \rangle,$$

where $\|\cdot\|_F$ is the Frobenius norm (for any complex matrix A, $\|A\|_F = \sqrt{\mathrm{tr}(AA^*)}$). A property of FD is that

$$\mathrm{FD}(S_{\mathbf{X}}, S_{\mathbf{Y}}) = 0 \text{ if and only if } S_{\mathbf{X}} \overset{a.e.}{=} S_{\mathbf{Y}},$$

where "a.e." indicates that the two matrix-valued functions are equal at all frequencies $\lambda \in [-\pi, \pi]$ except for a subset of Lebesgue measure zero. The above property is referred to as the complete equivalency of $S_{\mathbf{X}}$ and $S_{\mathbf{Y}}$; since the discrepancy of the two spectral densities on a set of measure zero does not disrupt the equality of their corresponding autocovariances, it follows that complete equivalency entails second-order utility.

Another expression for $\mathrm{FD}(S_{\mathbf{X}}, S_{\mathbf{Y}})$ is $\sum_{h\in\mathrm{Z}} \|\Gamma_{\mathbf{X}}(h) - \Gamma_{\mathbf{Y}}(h)\|_F^2$, which makes the connection to second-order utility more explicit. When using FD to assess second-order utility (low values corresponding to higher utility), it is convenient to use a normalized measure; to that end, we derive the upper bound

$$\mathrm{FD}(S_{\mathbf{X}}, S_{\mathbf{Y}}) \leq \sum_{h\in\mathrm{Z}} \left(\|\Gamma_{\mathbf{X}}(h)\|_F + \|\Gamma_{\mathbf{Y}}(h)\|_F\right)^2.$$

This is obtained using the triangle inequality for the Frobenius norm. We use this upper bound to normalize the Frobenius discrepancy, obtaining the so-called NFD:

$$\mathrm{NFD}(S_{\mathbf{X}}, S_{\mathbf{Y}}) = \frac{\sum_{h\in\mathrm{Z}} \|\Gamma_{\mathbf{X}}(h) - \Gamma_{\mathbf{Y}}(h)\|_F^2}{\sum_{h\in\mathrm{Z}} \left(\|\Gamma_{\mathbf{X}}(h)\|_F + \|\Gamma_{\mathbf{Y}}(h)\|_F\right)^2}. \tag{4.4}$$

By the triangle inequality the maximum value of NFD is 1. An empirical version of NFD, denoted as $\widehat{\mathrm{NFD}}$, is obtained by substituting sample autocovariances in (4.4). Finally, we define the realized utility measure (RUM) via

$$\mathrm{RUM}(S_{\mathbf{X}}, S_{\mathbf{Y}}) = 1 - \widehat{\mathrm{NFD}}(S_{\mathbf{X}}, S_{\mathbf{Y}}), \tag{4.5}$$

which has the property that high values (close to unity) correspond to high utility (i.e., when the FD is close to zero). Also, because NFD is bounded by one, low values of RUM correspond to low utility.

5. Numerical Illustration

In this section we apply the m-LIP methods to both simulated data and real data – the QWI employment data published by U.S. Census Bureau.

5.1. **Simulated Data.** Here we simulate data from a Vector Autoregressive Moving Average (VARMA) process of order (1,1), a VAR(1) with i.i.d. innovations, and a VAR(1) where the innovations are drawn from an Autoregressive Conditionally Heteroscedastic (ARCH) process of order 1 (for detailed discussion of VARMA and ARCH models, see (BD)). These simulation processes are used to jointly describe $\{\mathbf{X}_t\}$ and $\mathbf{Z}_t$; for the third case, the ARCH(1) innovations correspond to $\{\mathbf{Z}_t\}$.

For obtaining the $S_{\mathbf{X}}$-MAP privatization filter $\Psi(B)$ in each case, we employ the following settings. For spectral density matrix estimation, we use the flat-top taper method described in Section 4. We obtain the spectral factorization for the joint spectral density of the target series that are the focal point of our protective measures. We then solve the minimization problem posed in (4.3), using various choices of the order r of $\Omega(z)$.

When $\Omega(z)$ equals the zero matrix $\mathbf{0}$, corresponding to $U(z) = \boldsymbol{I}$, then the m-LIP criterion equals zero – which makes sense since no privatization actually occurs. The choice $r = 0$ means that only Ω_0 is present, and there are only $\binom{n}{2}$ parameters (the single lower triangular entry in the case of $n = 2$) in ϑ. For a three-dimensional series, the choice $r = 0$ will give three parameters in ϑ. When $r = 1$, there are 4 additional free parameters in Ω_1 for a 2-dimensional series, and 9 additional parameters for a 3-dimensional series. For a 3-dimensional series, we also explore $r = 2$, which renders a total of 21 free parameters. Given the quadratic rate of the number of parameters with respect to the dimension, for higher dimensional cases (such as the 10-dimensional series explored below) we choose the coefficient matrices to be of lower rank. In each of these scenarios, we minimize the criterion to obtain the optimal ϑ and the corresponding filter $\Psi(B)$.

For each of the three cases, we plot the comparisons of the autocorrelation and the cross-correlation functions of the original and the released series. For the plots, we use $r = 1$ to obtain the optimal $S_{\mathbf{X}}$-MAP filter.

5.1.1. *Simulation from VAR(1).* A stationary VAR(p) model for $\{\mathbf{W}_t\}$ is defined as follows:

$$\mathbf{W}_t = A_1\mathbf{W}_{t-1} + A_2\mathbf{W}_{t-2} + \ldots + A_p\mathbf{W}_{t-p} + \varepsilon_t,$$

where $A_1, A_2, \ldots, A_p$ are coefficient matrices for lags 1 through p, and $\{\varepsilon_t\} \sim \mathrm{WN}(0, \Sigma)$. We generate a time series of length $T = 2000$ from a 6-variate VAR(1) model. The AR coefficient matrix is

$$A = \begin{pmatrix} 0.4 & 0.1 & 0.05 & 0.02 & 0.01 & 0.01 \\ 0.2 & 0.4 & 0.15 & 0.07 & 0.03 & 0.02 \\ 0.1 & 0.2 & 0.4 & 0.12 & 0.06 & 0.04 \\ 0.05 & 0.1 & 0.25 & 0.4 & 0.15 & 0.07 \\ 0.02 & 0.05 & 0.1 & 0.2 & 0.4 & 0.1 \\ 0.01 & 0.02 & 0.05 & 0.1 & 0.15 & 0.4 \end{pmatrix},$$

which has all absolute eigenvalues less than 1, thereby ensuring stationarity and causality of the process. The covariance matrix of the noise is assumed to be

$$\Sigma = \begin{pmatrix} 1 & 0.3 & 0.2 & 0 & 0.1 & 0.05 \\ 0.3 & 1 & 0.3 & 0.2 & 0.1 & 0.1 \\ 0.2 & 0.3 & 1 & 0.3 & 0.2 & 0.1 \\ 0 & 0.2 & 0.3 & 1 & 0.3 & 0.2 \\ 0.1 & 0.1 & 0.2 & 0.3 & 1 & 0.3 \\ 0.05 & 0.1 & 0.1 & 0.2 & 0.3 & 1 \end{pmatrix}.$$

We divide the 6-variate VAR(1) process into two parts: the first three components correspond to $\{\mathbf{X}_t\}$, while the latter three components correspond to $\{\mathbf{Z}_t\}$. Generating the process in this fashion serves the purpose of keeping the $\{\mathbf{X}_t\}$ and $\{\mathbf{Z}_t\}$ time series jointly stationary.

We generate the 6-dimensional VAR(1) time series multiple times (100 Monte Carlo copies), and obtain optimal values of ϑ for various choices of r and number of parameters in ϑ. We measure the time complexities, and report the average time complexity in seconds for each case. From the simulations, we observe that even for moderate values of the number of parameters, corresponding to the order $r \geq 1$, we get the maximized privacy value to be 1 or very close to one.

6 dimensional VAR(1) (100 MC simulations)		
r(no. of parameters)	Privacy value (avg.)	Time in sec. (avg.)
0 (3)	0.7806	232.14
1 (12)	1	1238.97
2 (21)	1	1484.36
2 (10)	0.9998	1432.19
2 (5)	0.9976	1274.44

Table 1: Privacy values and time complexities for VAR(1) simulation

For $r = 0, 1$, we do not fix any parameters and let all the parameters be free. For $r = 2$, we consider three cases. For the first case, all parameters are free and we have 21 free parameters. For the next two cases, we keep the coefficient matrices in the cepstral representation low rank, making the number of parameters 10 and 5. We inspect the average privacy value and average time complexity for each of the cases mentioned; see Table 1. We see that even with a moderate number of parameters, the privacy value attains its maximum value of one; the average times for implementation of the m-LIP procedure are also reported.

To demonstrate utility, we plot sample autocovariances $\hat{\Gamma}_{\mathbf{X}}(h)$ and $\hat{\Gamma}_{\mathbf{Y}}(h)$ for a single simulation in Figure 1, when $r = 1$. For brevity, we only present the correlations for the first two components. The plots show very close agreement between the sample correlations, including the cross-correlations, of the original and the privacy-protected series. However, the trajectories of the two versions are substantially different; they are not shown here since there are 100 Monte Carlo samples. We show the trajectories of the original time series and the filtered time series later in the real data analysis.

We also inspect the privacy guarantee and time complexity for a 10-variate VAR(1) time series. We take six different cases, for the following values of r and number of parameters (in parentheses): 0(2), 0(10), 1(20), 1(40), 2(60), and 2(80). For all the cases our simulation rendered a privacy value of 1 (up to rounding in the fourth decimal). The privacy and average time complexities for different cases are presented in Table 2.

Sensitivity to trend estimation: For the six dimensional VAR(1) case in Section 5.1.1 , we add a quadratic trend to the time series to explore the robustness of the privacy method on the error in estimation of trend. We first generate the series with the trend, then estimate the trend and detrend the series. Following that, we perform the optimization for privacy and obtain the coefficients of the S-MAP filter. For details of the method, refer to Section 4.5. We take $r = 1$ and keep all the parameters free. For the actual series, we achieve the absolute privacy value, i.e., 1, just as in the case where the trend is estimated. The

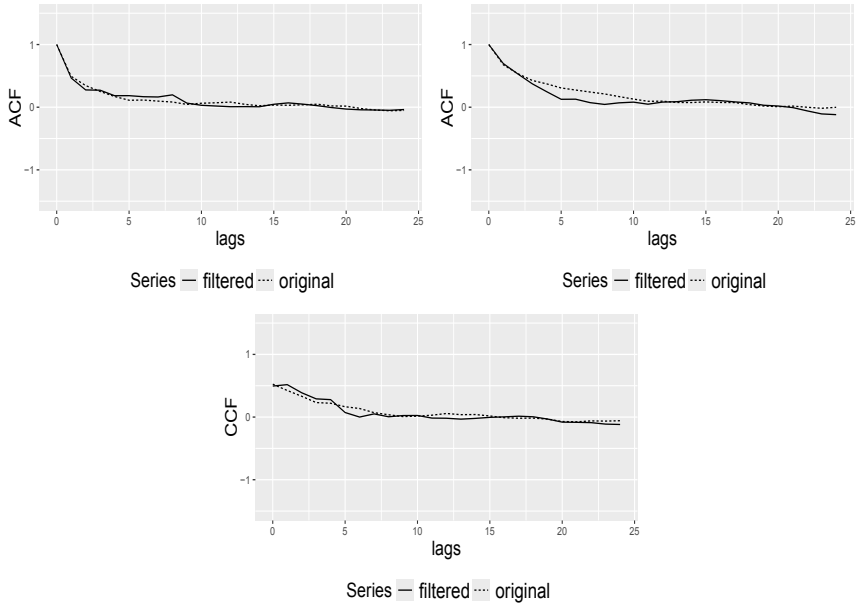

Figure 1: Comparison of sample autocorrelation function (ACF) and the cross-correlation function (CCF) of the original and the filtered copies for the first and second series for the case $r = 1$ (VAR(1)). The top row shows the ACF plots of first and second series while the bottom plot shows the CCF between the first and second series.

10 dimensional VAR(1) (100 MC simulations)		
r (no. of parameters)	Privacy value (avg.)	Time in sec. (avg.)
0 (2)	1	78.15
0 (10)	1	1139.00
1 (20)	1	1227.45
1 (40)	1	1781.28
2 (60)	1	2187.83
2 (80)	1	2085.72

Table 2: Privacy values and time complexities for VAR(1) simulation.

realized utility measure is 0.852, whereas it is 0.843 for the estimated trend case. The time complexities are 1348.97 seconds and 1408.57 seconds for the two cases, respectively. The results are presented in Table 3.

5.1.2. *Simulation from VARMA(1,1).* We generate a 4-variate VARMA(1,1) described by the following equation:

$$\mathbf{W}_t = \Phi \mathbf{W}_{t-1} + \epsilon_t + \Theta \epsilon_{t-1},$$

6 dimensional VAR(1) (100 MC simulations, r=1, all parameters free)			
Trend	Privacy value (avg.)	RUM (avg.)	Time in sec. (avg.)
True	1	0.852	1348.97
Estimated	1	0.843	1408.57

Table 3: Average privacy values, realized utility measures, and time complexities for VAR(1) simulation for estimated and true trends.

where $\{\epsilon_t\}$ is a white noise process with innovation variance-covariance matrix

$$\Sigma = \begin{pmatrix} 0.09 & 0 & 0 & 0 \\ 0 & 0.03 & 0 & 0 \\ 0 & 0 & 0.05 & 0 \\ 0 & 0 & 0 & 0.07 \end{pmatrix}.$$

The coefficient matrices for the Autoregressive (AR) and Moving Average (MA) components are defined respectively as

$$\Phi = \begin{pmatrix} -0.00556 & -0.6353 & 0.2529 & -0.0096 \\ -0.2288 & 0.3506 & 0.2414 & -0.02505 \\ -0.23423 & -1.33007 & 0.517 & -0.1978 \\ 0.1624 & 0.5523 & 0.4042 & -0.1412 \end{pmatrix}$$

and

$$\Theta = \left(\begin{array}{cc|cc} 0.6 & 0.2 & \multicolumn{2}{c}{\multirow{2}{*}{$\mathbf{0}$}} \\ 0 & 0.3 & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}$}} & 0 & 0 \\ & & 0 & 0 \end{array}\right).$$

Both $\{\mathbf{X}_t\}$ and $\{\mathbf{Z}_t\}$ are defined from the VARMA(1,1) process in the same manner as in the previous simulation. We also construct our privatization filter using the same settings, and assess performance in the same way. In Figure 2, we present a comparison of the sample autocovariances $\hat{\Gamma}_{\mathbf{X}}(h)$ and $\hat{\Gamma}_{\mathbf{Y}}(h)$ for a single simulation, for the case $r = 1$. In Table 4, we present the privacy values, average time complexities, and the realized utility measures for the VARMA(1,1) case. The privacy values get closer to the upper bound of one ever with only 5 free parameters. The realized utility is also almost one for that case.

5.1.3. *Simulation of a VAR(1) with ARCH(1) errors.* To check the sensitivity of the procedures to the linearity of the processes, we simulate a non-linear process. We generate a bivariate VAR(1) following the equation

$$\mathbf{Q}_t = A_1\mathbf{Q}_{t-1} + \zeta_t,$$

where $\zeta_{t,1} = \sqrt{h_t}e_t$ and $e_t \sim$ i.i.d. standard normal. $\zeta_{t,1}$ stands for the first component of the innovation series ζ_t, i.e., $\zeta_t = (\zeta_{t,1}, \zeta_{t,2})'$. Here, h_t is defined by

$$h_t = \alpha_0 + \alpha_1\zeta_{t-1,1}^2.$$

For our simulation we set $\alpha_0 = 1$ and $\alpha_1 = 0.5$. We assume $\zeta_{t,2} \sim \mathrm{WN}(0, 1)$ and is drawn independently with respect to $\zeta_{t,1}$. The series $\{\mathbf{Q}_t\}$ serves in the role of $\{\mathbf{X}_t\}$, where $\{\mathbf{Z}_t\}$ is the first component of the $\{\zeta_t\}$. The autocorrelation comparison is plotted in Figure 3. In Table 4, we present the privacy values, average time complexities, and the realized utility

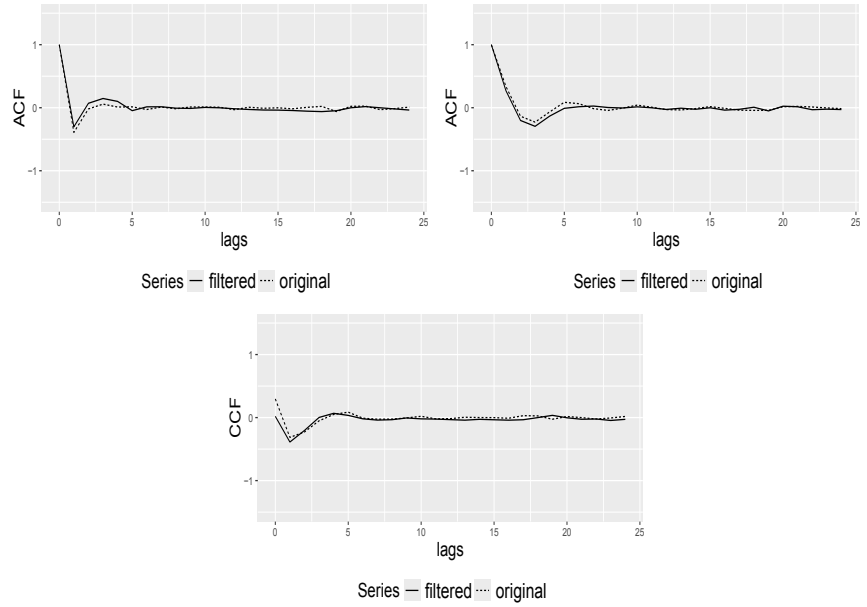

Figure 2: Comparison of sample autocorrelation and the cross-correlation functions of the original and the filtered copies for the first and second series for the case $r = 1$ (VARMA(1,1)). The top row shows the two ACF plots while the bottom plot shows the CCF between the two series.

measures for this case. The privacy values gets closer to the upper bound of one with only 5 free parameters, but they are lower than those obtained for the VARMA(1,1) case. To achieve a desired privacy level, a non-linear process might require more free parameters than a linear process.

	VARMA			ARCH		
Parameters	Privacy	Time	Utility	Privacy	Time	Utility
1	0.785	91.54	0.847	0.754	82.75	0.826
2	0.898	89.391	0.873	0.861	122.431	0.892
5	0.996	432.567	0.93	0.927	323.698	0.967

Table 4: Privacy, time complexity and utility for the VARMA(1,1) and the VAR(1) with ARCH(1) error cases.

5.1.4. *Summary of the simulation results.* Overall, privacy and realized utility both increase with even a moderate increase in the number of free parameters. However, the computation time also increases due to the optimization taking place over a higher dimensional space. In time series of moderate dimension (e.g., 10), even for $r = 2$ there are a large number (nearly 250) potential parameters to optimize over; for such cases, it is sensible to impose reduced rank and sparsity structures to shrink the number of free parameters. What we observe in the repeated simulations is that the desired privacy and utility can be achieved even with a

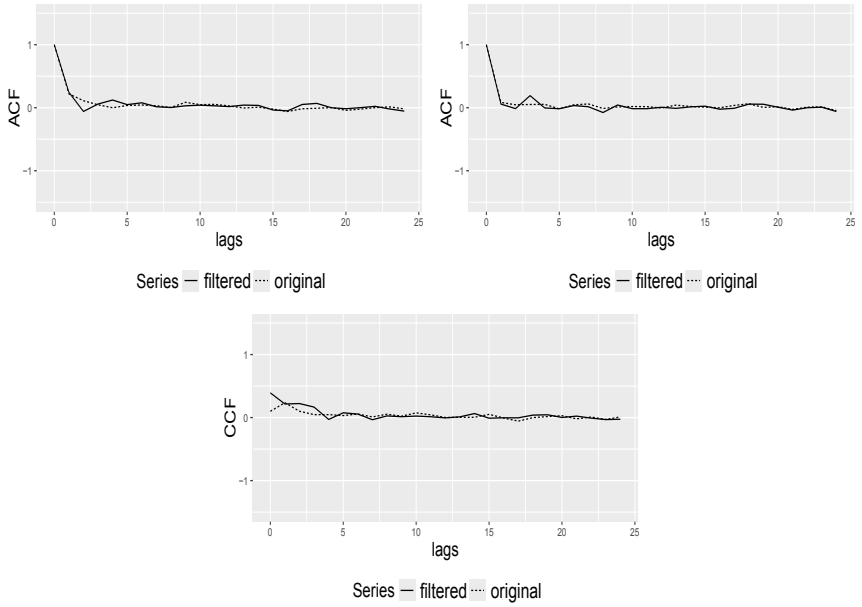

Figure 3: Comparison of sample autocorrelation and the cross-correlation functions of the original and the filtered copies for the first and second series for the case $r = 1$ (VAR(1), ARCH(1) error). The top row shows the two ACF plots while the bottom plot shows the CCF between the two series.

few parameters. Of course, if computation time is not an issue, then values near the upper bound of one can be obtained for both privacy and utility.

5.2. **QWI Employment Data.** Next, we demonstrate the effectiveness of our method upon employment count data obtained from the Quarterly Workforce Indicators (QWI) dataset published by the U.S. Census Bureau. The QWI dataset is derived from a comprehensive collection of job and work location administrative records spanning 49 states, and it is updated quarterly; see (AV11) for full details on the data's construction and publication.

All data used in our analysis were retrieved from the QWI Explorer website (Bur23) on January 28, 2024. Our analysis centers on the quarterly indicator referred to as "Beginning of Quarter Employment: Count," which we will abbreviate as "employment count." The dataset covers the state of Maryland and spans from the first quarter of 1997 (Q1 1997) to the fourth quarter of 2022 (Q4 2022). Specifically, we have gathered data for four distinct counties within Maryland: Baltimore, Frederick, Montgomery, and Howard counties.

We pretend that the Baltimore and Frederick county data is sensitive; our objective is to safeguard the bivariate time series comprising employment counts for Baltimore and Frederick counties, with Montgomery and Howard counties constituting the series that may be known to potential attackers. The employment data spanning 26 years from the aforementioned four counties in Maryland are displayed in Figure 4.

We remove trend and seasonal patterns from the quarterly data by applying the seasonal differencing operator $1 - B^4$. The resulting "annual growth rate" time series is stationary, as is verified through visual inspection of the autocorrelation function and the application of the augmented Dickey-Fuller test on each of the time series.

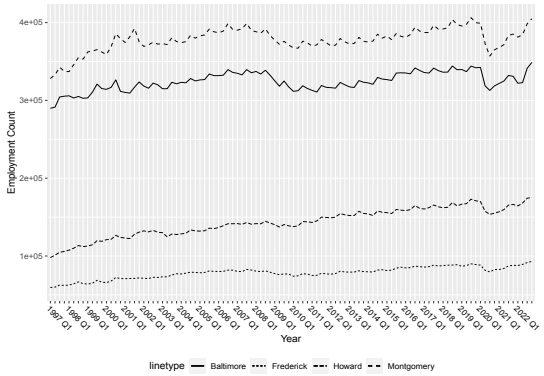

Figure 4: QWI employment count for Maryland counties.

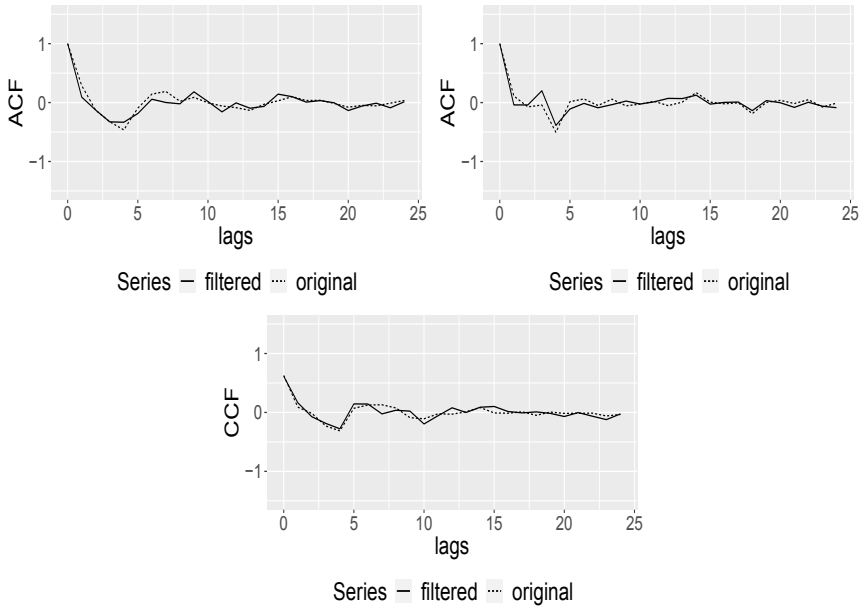

Figure 5: Comparison of sample autocorrelation and the cross-correlation functions of the original and the filtered copies for the first and second detrended series for the QWI data. The top row shows the two ACF plots while the bottom plot shows the CCF between the two series.

We obtain an $S_{\mathbf{X}}$-MAP filter with the choice $r = 1$, and apply the filter to the growth rate data to get the privatized growth rate series. Then we recursively determine modified data in the original scale, inverting the action of the $1 - B^4$ filter. The sample paths for Baltimore County and Frederick County, along with their corresponding filtered counterparts, are displayed in Figure 6. The comparisons of autocovariance and cross-covariance series are depicted in Figure 5.

The plots in Figure 5 show us that the autocorrelation structure of the two series are successfully kept unaltered, preserving utility. Moreover, the cross-correlations are preserved as well – a feature that is not available in current univariate privacy mechanisms. From the

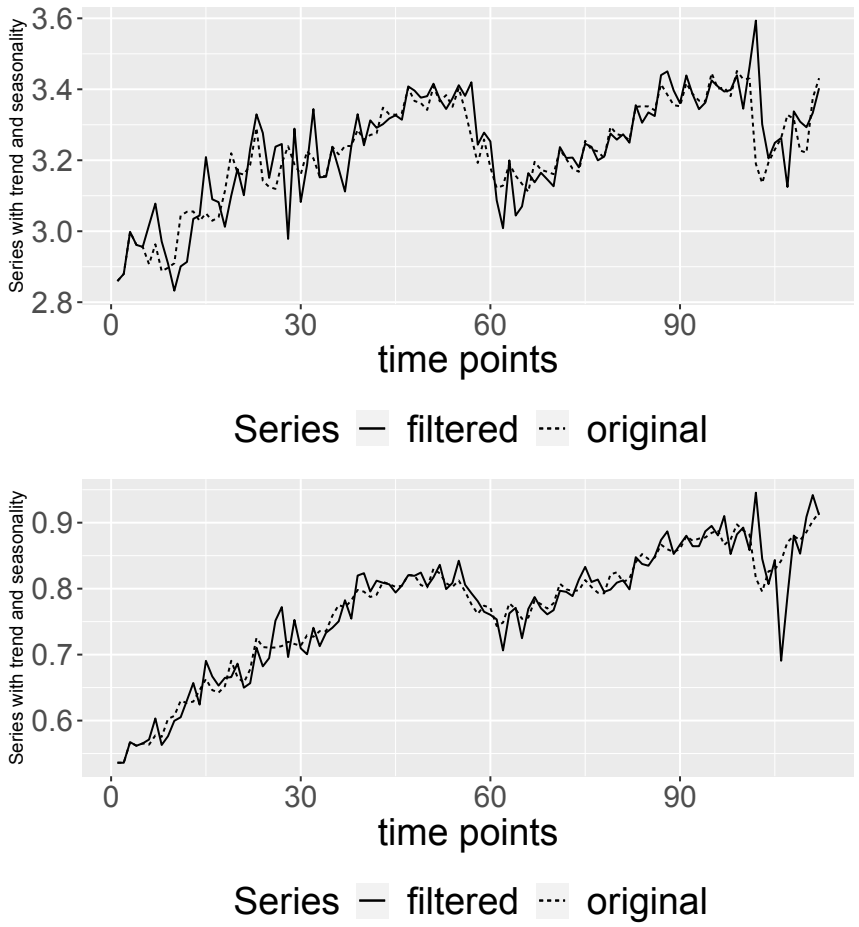

Figure 6: Comparison of standardized sample paths of the original and the filtered copies for the Baltimore (top panel) and Frederick (bottom panel) series. The y-axis is employment count (in units of 10^5).

two plots in Figure 6 it is apparent that the sample paths of the actual series and the released series coincide very rarely, and yet the released series maintains the trend and seasonal structure of the original data. Thus, the released time series serves as a representative proxy for the original time series, striking a balance between privacy and utility.

6. Discussion and Future Work

We have proposed a novel privacy preservation technique for multivariate time series, denoted as m-LIP, which leverages the concept of multivariate all-pass filtering, and provides a formal framework for balancing privacy and multivariate utility objectives. Multivariate all-pass filtering represents a more intricate approach compared to its univariate counterpart, and relies on the spectral density matrix of the target series requiring protection. The proposed framework is a formal privacy-utility framework with respect to the m-LIP measure. In

contrast to the DP framework, the proposed framework treats the entire time series trajectory as a single high-dimensional point.

There are some important similarities between the m-LIP framework and the DP framework. The m-LIP measures the amount of information gained in predicting the sensitive vector $\mathbf{X}$ when the private vector $\mathbf{Y}$ is made available in addition to auxiliary information $\mathbf{Z}$ available to the attacker. This parallels the incremental nature of differential privacy. But there are differences that make the two paradigms distinct. The DP framework considers the distribution of the private series conditional on the sensitive input, whereas the m-LIP framework is concerned with the point prediction of the sensitive series given the private series. In this sense, the m-LIP does not target the entire predictive distribution; but in a time series setting, protecting the entire distribution would require distributional knowledge of the joint behavior of the entire observation vector. That is often not available, and only first- and second-order information (means and covariances) is instead used.

We have implemented the multivariate mechanism after removing deterministic trends from each component. Our numerical sensitivity analysis suggests that our method is robust with respect to trend estimation and removal. However, this implementation is a two-stage procedure that suffers from endemic challenges of multi-stage methods, where errors from previous stages can influence the outcome of subsequent stages. Therefore, it seems desirable to devise a single-stage implementation that constrains multivariate all-pass filters so as to accommodate (and pass) higher order polynomial trends. Such procedures would exclude the macro trends from the privacy budgets, and thereby leave them invariant under the implementation of the multivariate mechanism. To address this issue, we have also proposed an extension of the basic framework that can accommodate difference stationary time series; future work should empirically evaluate the competing merits of the two-stage and single-stage approaches.

We briefly sketch some of the challenges of enforcing such trend-passing constraints: a linear filter Ψ whose application leaves a dth order polynomial unchanged can be constructed by constraining $\Psi(z)$ to be trend-invariant. For $d = 0$ (the case of a constant trend) it is necessary that $\Psi(1) = \Psi(e^{-i\lambda})|_{\lambda=0}$ equals the identity matrix. For $d > 0$, it is required that the dth derivative of $\Psi(e^{-i\lambda})$ (with respect to λ) at $\lambda = 0$ is the zero matrix. In (MRH23) such conditions on the filter were parsed in terms of conditions on the cepstral coefficients. However, in the multivariate case the derivative of $\exp\{\Omega(e^{-i\lambda})\}$ is not easy to compute, due to the fact that the summands $\Omega_k e^{-i\lambda k}$ do not commute with one another. Hence, we cannot directly impose trend-invariant filter constraints on $\Psi(z)$ through conditions on ϑ. This poses a formidable challenge. We intend to explore methods for choosing MAP filters that pass polynomial trends unchanged as a topic of future investigation.

In some applications, it may be reasonable to include the macro features such as trend and seasonality in the privacy budget. For example, if one series has a strikingly different trend, or unique seasonal pattern, it may require disclosure avoidance. We plan to investigate privacy mechanisms applicable to such situations in the future.

Finally, being able to address privacy-utility balancing in a truly multivariate setting, this approach can be leveraged to provide a privacy framework against extensive training of modern forecasting regimes, where the input and output time series can be jointly privatized before passing them through the training module.

Acknowledgment

The authors are thankful to the editor and two anonymous referees for their critical reading and suggestions. Accounting for many of their suggestions has greatly improved the overall exposition of the article.

Disclaimer

Any opinions and conclusions expressed herein are those of the authors and do not represent the views of the U.S. Census Bureau. All results in this paper use publicly available data from Census Bureau websites.

References

[AACM+22] J. M. Abowd, R. Ashmead, R. Cumings-Menon, S. Garfinkel, M. Heineck, C. Heiss, R. Johns, D. Kifer, P. Leclerc, A. Machanavajjhala, B. Moran, W. Sexton, M. Spence, and P. Zhuravlev. The 2020 census disclosure avoidance system topdown algorithm. *Harvard Data Science Review*, (Special Issue 2), 2022. URL: https://systems.cs.columbia.edu/private-systems-class/papers/Abowd2022Census.pdf.

[AGM+12] J. M. Abowd, K. Gittings, K. L. McKinney, B. E. Stephens, L. Vilhuber, and S. Woodcock. Dynamically consistent noise infusion and partially synthetic data as confidentiality protection measures for related time series. *US Census Bureau Center for Economic Studies Paper No. CES-WP-12-13*, 2012. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2159800.

[APPG23] H. H. Arcolezi, C. Pinzón, C. Palamidessi, and S. Gambs. Frequency estimation of evolving data under local differential privacy. *arXiv preprint arXiv:2210.00262*, 2023. URL: https://doi.org/10.48786/edbt.2023.44.

[AV11] J. M. Abowd and L. Vilhuber. National estimates of gross employment and job flows from the quarterly workforce indicators with demographic and industry detail. *Journal of econometrics*, 161(1):82–99, 2011. URL: https://www.sciencedirect.com/science/article/pii/S0304407610001880.

[Bau55] F. L. Bauer. Ein direktes iterationsverfahren zur hurwitz-zerlegung eines polynoms. *Archiv der elektrischen Übertragung*, 1955.

[BD] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting*. Springer. URL: https://link.springer.com/book/10.1007/978-3-319-29854-2.

[Bel84] W. Bell. Signal extraction for nonstationary time series. *The Annals of Statistics*, pages 646–664, 1984. URL: https://doi.org/10.1214/aos/1176346512.

[Bri01] D. R. Brillinger. *Time Series: Data Analysis and Theory*. Siam, 2001. URL: https://epubs.siam.org/doi/10.1137/1.9780898719246.

[Bur23] U.S. Census Bureau. Quarterly Workforce Indicator, 2023. [Online; accessed in 2022 and 2023]. URL: https://qwiexplorer.ces.census.gov.

[DGSG22] J. Ding, A. Ghosh, R. Sarkar, and J. Gao. Publishing asynchronous event times with pufferfish privacy. In *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 53–60. IEEE, 2022, pages 53–60. URL: https://doi.org/10.1109/DCOSS54816.2022.00020.

[DR14] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9:211–407, 2014. URL: https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf.

[Dwo06] C. Dwork. Differential privacy. *International Colloquium on Automata, Languages and Programming, part II (ICALP)*, 2006. URL: https://doi.org/10.1007/11787006_1.

[EFM15] M. A. Erdogdu, N. Fawaz, and A. Montanari. Privacy-utility tradeoff for time-series with application to smart-meter data. *Association for the Advancement of Artificial Intelligence*, 2015. URL: https://aaai.org/papers/aaaiw-ws0128-15-10193/.

[FVH19] F. Fioretto and P. Van Hentenryck. Optstream: Releasing time series privately. *Journal of Artificial Intelligence Research*, 65:423–456, 2019. URL: https://doi.org/10.1613/jair.1.11583.

[FX13] L. Fan and L. Xiong. Differentially private anomaly detection with a case study on epidemic outbreak detection. In *2013 IEEE 13th International Conference on Data Mining Workshops*, pages 833–840. IEEE, 2013, pages 833–840. URL: https://doi.org/10.1109/ICDMW.2013.129.

[GM20] R. Gong and X-L. Meng. Congenial differential privacy under mandated disclosure. FODS '20, page 59–70, New York, NY, USA, 2020. Association for Computing Machinery, page 59–70. URL: https://doi.org/10.1145/3412815.3416892.

[HGKM13] S.K. Hong, K. Gurjar, H.S. Kim, and Y.S. Moon. A survey on privacy preserving time-series data mining. *3rd International Conference on Intelligent Computational Systems (ICICS)*, 2013. URL: https://www.researchgate.net/publication/295547124_A_Survey_on_Privacy_Preserving_Time-Series_Data_Mining.

[HMW17] S. Holan, T. S. McElroy, and G. Wu. The cepstral model for multivariate time series: The vector exponential model. *Statistica Sinica*, pages 23–42, 2017. URL: https://www.jstor.org/stable/44114360.

[IHA+20] S. Imtiaz, S-F. Horchidan, Z. Abbas, M. Arsalan, H. N. Chaudhry, and V. Vlassov. Privacy preserving time-series forecasting of user health data streams. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 3428–3437, 2020, pages 3428–3437. URL: https://ieeexplore.ieee.org/document/9378186.

[KMC22] T. Koga, C. Meehan, and K. Chaudhuri. Privacy amplification by subsampling in time domain. In *International Conference on Artificial Intelligence and Statistics*, pages 4055–4069. PMLR, 2022, pages 4055–4069. URL: https://proceedings.mlr.press/v151/koga22a/koga22a.pdf.

[KTK22] M. Katsomallos, K. Tzompanaki, and D. Kotzinos. Landmark privacy: Configurable differential privacy protection for time series. *Conference on Data and Application Security and Privacy (CODASPY)*, 2022. URL: https://doi.org/10.1145/3508398.3511501.

[LLJP17] L. Lyu, Y. W. Law, J. Jin, and M. Palaniswami. Privacy-preserving aggregation of smart metering via transformation and encryption. *IEEE Trustcom/BigDataSE/ICESS, pp. 472–479, IEEE, Sydney, Australia*, 2017. URL: https://ieeexplore.ieee.org/document/8029476.

[LLML21] F. L. Lako, P. Lajoie-Mazenc, and M. Laurent. Privacy-preserving publication of time-series data in smart grid. *Security and Communication Networks*, 2021. URL: https://doi.org/10.1155/2021/6643566.

[LXJL15] H. Li, L. Xiong, X. Jiang, and J. Liu. Differentially private histogram publication for dynamic datasets: an adaptive sampling approach. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1001–1010, 2015, pages 1001–1010. URL: https://doi.org/10.1145/2806416.280644.

[McE17] T. S. McElroy. Recursive computation for block-nested covariance matrices. *Journal of Time Series Analysis*, 2017. URL: https://doi.org/10.1111/jtsa.12267.

[McE18] T. S. McElroy. Recursive computation for block-nested covariance matrices. *Journal of Time Series Analysis*, 39(3):299–312, 2018. URL: https://doi.org/10.1111/jtsa.12267.

[MP20] T. S. McElroy and D. N. Politis. *Time Series: A First Course with Bootstrap Starter*. CRC Press, 2020. URL: https://doi.org/10.1201/9780429109553.

[MP22] T. S. McElroy and D. N. Politis. Optimal linear interpolation of multiple missing values. *Statistical Inference for Stochastic Processes*, 25(3):471–483, 2022. URL: https://doi.org/10.1007/s11203-022-09269-5.

[MR23] T. S. McElroy and A. Roy. Model identification via total frobenius norm of multivariate spectra. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(2):473–495, 2023. URL: https://doi.org/10.1093/jrsssb/qkad012.

[MRH23] T. McElroy, A. Roy, and G. Hore. Flip: A utility preserving privacy mechanism for time series. *Journal of Machine Learning Research*, 2023. URL: https://www.jmlr.org/papers/volume24/22-0734/22-0734.pdf.

[MT15] T. S. McElroy and T. Trimbur. Signal extraction for non-stationary multivariate time series with illustrations for trend inflation. *Journal of Time Series Analysis*, 36(2):209–227, 2015. URL: https://doi.org/10.1111/jtsa.12102.

[NGGD21] Y. Nesterov, A. Gasnikov, S. Guminov, and P. Dvurechensky. Primal–dual accelerated gradient methods with small-dimensional relaxation oracle. *Optimization Methods and Software*, 36:773–810, 2021. URL: https://doi.org/10.1080/10556788.2020.1731747.

[PAK18] V. Perrier, H. J. Asghar, and D. Kaafar. Private continual release of real-valued data streams. *arXiv preprint arXiv:1811.03197*, 2018. URL: https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_07B-5_Perrier_paper.pdf.

[Pol11] D. N. Politis. Higher-order accurate, positive semi-definite estimation of large-sample covariance and spectral density matrices. *Econometric Theory*, 2011. URL: https://www.jstor.org/stable/27975501.

[RN10] V. Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. *International Conference on Management of Data, ACM SIGMOD*, pages 735–746, 2010. URL: https://doi.org/10.1145/1807167.1807247.

[SC17] S. Song and K. Chaudhuri. Composition properties of inferential privacy for time-series data. In *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2017. URL: https://ieeexplore.ieee.

org/document/8262823.

[SCR+11] E. Shi, T-H. H. Chan, E. Rieffel, R. Chow, and D. Song. Privacy-preserving aggregation of time-series data. *In Proceedings of the Network and Distributed System Security Symposium, San Diego, California*, 2011. URL: https://amplab.cs.berkeley.edu/wp-content/uploads/2011/06/Privacy-Preserving-Aggregation-of-Time-Series-Data.pdf.

[SST09] Y. Sang, H. Shen, and H. Tian. Privacy-preserving tuple matching in distributed databases. *IEEE Transactions on Knowledge and Data Engineering, 21(12)*, page 1767–1782, 2009. URL: https://dl.acm.org/doi/abs/10.1109/TKDE.2009.39.

[Sta19] C. Stach. Vault: A privacy approach towards high-utility time series data. *International Conference on Emerging Security Information, Systems and Technologies, pp. 41–46*, 2019. URL: https://www.ipvs.uni-stuttgart.de/departments/as/publications/stachch/securware_19_vault.pdf.

[SWC17] S. Song, Y. Wang, and K. Chaudhuri. Pufferfish privacy mechanisms for correlated data. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1291–1306, 2017, pages 1291–1306. URL: https://dl.acm.org/doi/pdf/10.1145/3035918.3064025.

[WRN+20] S. Wang, C. Rudolph, S. Nepal, M. Grobler, and S. Chen. Part-gan: Privacy-preserving time-series sharing. In *Artificial Neural Networks and Machine Learning–ICANN 2020: 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15–18, 2020, Proceedings, Part I 29*, pages 578–593. Springer, 2020, pages 578–593. URL: https://doi.org/10.1007/978-3-030-61609-0_46.

[WZ10] L. Wasserman and S. Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105:375–389, 2010. URL: https://www.tandfonline.com/doi/abs/10.1198/jasa.2009.tm08651.

[XYH+22] Q. Xue, Q. Ye, H. Hu, Y. Zhu, and J. Wang. Ddrm: A continual frequency estimation mechanism with local differential privacy. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):6784–6797, 2022. URL: https://doi.org/10.1109/TKDE.2022.3177721.

[ZKL22] X. Zhang, M. M. Khalili, and M. Liu. Differentially private real-time release of sequential data. *ACM Transactions on Privacy and Security*, 26(1):1–29, 2022. URL: https://doi.org/10.1145/3544837.

[C12] G. Ács, C. Castelluccia, and R. Chen. Differentially private histogram publishing through lossy compression. *IEEE International Conference on Data Mining*, 2012. URL: https://ieeexplore.ieee.org/document/6413718.

Appendix A: Proofs of Technical Results

Proof of Proposition 1.1. Manipulation of (1.1) shows that $\mathrm{Var}[\mathbf{X}|\mathbf{Y},\mathbf{Z}]$ is composed of the block entries of the matrix $\mathrm{Var}[\mathbf{X},\mathbf{Y}|\mathbf{Z}]$. In particular, $\mathrm{Var}[\mathbf{X}|\mathbf{Y},\mathbf{Z}]$ is the Schur complement of $\mathrm{Var}[\mathbf{X},\mathbf{Y}|\mathbf{Z}]$, and hence is itself non-negative definite. For any positive semi-definite matrices A and B of the same dimension, if $A - B \geq \mathbf{0}$ (i.e., the difference is non-negative definite), then $\det A \geq \det B$. Thus, setting $A = \mathrm{Var}[\mathbf{X}_t|\{\mathbf{Z}_t\}]$ and $B = \mathrm{Var}[\mathbf{X}_t|\{\mathbf{Z}_t\}] -$

$\mathrm{Var}[\mathbf{X}_t|\{\mathbf{Y}_t\},\{\mathbf{Z}_t\}]$ we find that

$$\det \mathrm{Var}[\mathbf{X}|\mathbf{Z}] \geq \det\left[\mathrm{Cov}[\mathbf{X},\mathbf{Y}|\mathbf{Z}]\,\mathrm{Var}[\mathbf{Y}|\mathbf{Z}]^{-1}\,\mathrm{Cov}[\mathbf{Y},\mathbf{X}|\mathbf{Z}]\right] \geq 0,$$

and the stated result follows. □

Proof of Propostion 3.1. Let $E[\mathbf{X}_t|\{\mathbf{Z}_t\}]$ denote the optimal linear predictor of $\mathbf{X}_t$ given the whole process $\{\mathbf{Z}_t\}$. Then this can be expressed as $\Pi(B)Z_t$ for some filter $\Pi(B)$ with frequency response function $\Pi(z) = S_{\mathbf{XZ}}(\lambda)S_{\mathbf{Z}}(\lambda)^{-1}$ by Theorem 8.3.1 of (Bri01). It follows that the residual process $\mathbf{X}_t - E[\mathbf{X}_t|\{\mathbf{Z}_t\}]$ is stationary with spectral density

$$\begin{aligned} S_{\mathbf{X}|\mathbf{Z}}(\lambda) &= S_{\mathbf{X}}(\lambda) - \Pi(z)S_{\mathbf{ZX}}(\lambda) - S_{\mathbf{XZ}}(\lambda)\Pi(z)^* + \Pi(z)S_{\mathbf{Z}}(\lambda)\Pi(z)^* \\ &= S_{\mathbf{X}}(\lambda) - S_{\mathbf{XZ}}(\lambda)S_{\mathbf{Z}}(\lambda)^{-1}S_{\mathbf{ZX}}(\lambda). \end{aligned}$$

The residual process $\mathbf{Y}_t - E[\mathbf{Y}_t|\{\mathbf{Z}_t\}]$ has an analogous expression for its spectral density, and the cross-spectral density between the two residual processes is

$$S_{\mathbf{XY}|\mathbf{Z}}(\lambda) = S_{\mathbf{XY}}(\lambda) - S_{\mathbf{XZ}}(\lambda)S_{\mathbf{Z}}(\lambda)^{-1}S_{\mathbf{ZY}}(\lambda).$$

□