Private Outsourced Kriging Interpolation

James Alderman[1], Benjamin R. Curtis[1], Oriol Farràs Ventura[2], Keith M. Martin[1], and Jordi Ribes-González[2]

[1] Information Security Group, Royal Holloway, University of London
{James.Alderman, Benjamin.Curtis.2015, Keith.Martin}@rhul.ac.uk
[2] Universitat Rovira i Virgili, Tarragona, Catalonia, Spain
{oriol.farras, jordi.ribes}@urv.cat

Abstract. Kriging is a spatial interpolation algorithm which provides the best unbiased linear prediction of an observed phenomena by taking a weighted average of samples within a neighbourhood. It is widely used in areas such as geo-statistics where, for example, it may be used to predict the quality of mineral deposits in a location based on previous sample measurements. Kriging has been identified as a good candidate process to be outsourced to a cloud service provider, though outsourcing presents an issue since measurements and predictions may be highly sensitive. We present a method for the private outsourcing of Kriging interpolation using a tailored modification of the Kriging algorithm in combination with homomorphic encryption, allowing crucial information relating to measurement values to be hidden from the cloud service provider.

1 Introduction

Cost-effective third-party (cloud) service providers facilitate the outsourcing of large, potentially sensitive, datasets for both storage *and* processing. In this paper, we discuss approaches to outsourcing a particular computational process known as *Kriging* in an efficient and secure fashion.

Kriging [6,7,9,14] is a well-recognized form of linear interpolation that predicts the value z_0^* of some phenomena at an unobserved location (x_0, y_0) in a two-dimensional region. The quality of a Kriging prediction relies on some *variogram parameters*, which reflect the assumption that measurements taken at nearby locations are more likely to be 'similar' than measurements taken far apart. Such parameters must be carefully selected prior to interpolation. The prediction is then formed as a weighted sum of prior measurements, where measurements taken close to (x_0, y_0) are given a greater weight than those far away. Kriging was designed with geo-statistical applications in mind (*e.g.* to predict the best location to mine based on the mineral deposits found at previous boreholes within a region), but has also found applications in a variety of settings including remote sensing, real-estate appraisal and computer simulations.

Kriging has been identified as a good candidate process to be outsourced, based on the practical and legislative requirements of industrial users (for instance, [1,2]). Many users may need access to a Kriging prediction service (indeed legal frameworks may require such data to be shared amongst relevant authorities [8]). A secured storage server may be preferable to distributing copies of the entire dataset to each authorised user, especially when datasets are large and/or user devices are constrained. Further, Kriging might need to be performed over data owned by multiple organizations, with an independent cloud service provider performing processing duties on behalf of all concerned parties. Centralized outsourcing also makes sense when remote sensors take frequent measurements and push the results to a central database.

Consider a client $\mathcal{C}$ that owns a Kriging dataset (a set of measurements taken at various locations) which it wishes to outsource to an honest-but-curious cloud service provider $\mathcal{S}$. Client $\mathcal{C}$ would like to make use of both the storage *and* computational power of $\mathcal{S}$ to make a Kriging service on its dataset available to multiple users. Further, other *data generating nodes* may be authorised by $\mathcal{C}$ to add/remove data (measurements) to/from the outsourced dataset.

A trivial solution consists of encrypting all data using a symmetric encryption scheme and using the server only for Storage-as-a-Service. To compute a Kriging prediction, all relevant data is retrieved, decrypted and computed on locally. Unfortunately, this solution may not be efficient, particularly if client devices have limited computational power or storage capacity, and require a high bandwidth during queries. This may be an issue if, for example, a surveyor in the field requires an on-line Kriging prediction service; mobile data services may be expensive, intermittently available or slow.

An alternative is to compute the entire Kriging process on encrypted data by encrypting all data using Fully Homomorphic Encryption (FHE)[3]. Unfortunately, Kriging involves several computations that are currently challenging when using FHE, including computing square roots and natural exponentiations. It is possible to outsource the Kriging process and protect *all* information using FHE. However this results in prohibitively high encryption and decryption costs, as well as a large amount of interactivity and local computation, which may diminish the benefits of cloud computing. Preliminary experiments using the SEAL library [4] (admittedly without optimization of code or parameter choices) did not yield promising results when computing a Kriging prediction using a dataset of more than three measurements. Whilst the use of FHE schemes should be explored further in future work, particularly to reflect advances in FHE schemes, we show in this work that such schemes are not strictly required in this setting.

Our proposed solution uses additive homomorphic encryption to outsource Kriging interpolation efficiently. We make a trade-off by protecting only the most sensitive parameters. That is, we protect the prior measurement values in the dataset, the generated Kriging predictions and the variogram parameters chosen by the client. We do not hide locations (of prior measurements or queries), noting that prior measurement locations may well be externally observable (*e.g.* if measurements come from previous mining operations).

Our main contribution is to show that the Kriging process can be adapted such that the sensitive variogram parameters may be 'factored out' from the online computation by $\mathcal{S}$ whilst the remainder of the Kriging computation may be performed on *encrypted* measurement values using an additively homomorphic encryption scheme. We thus gain a practical, efficient and secure solution to outsourced, private Kriging. An outline of our protocol is as follows:

1. $\mathcal{C}$ uploads an encrypted dataset, comprising n measurements, to $\mathcal{S}$. The cost of this step is $O(n)$ due to encryption of the measurement values.
2. $\mathcal{S}$ prepares the Kriging dataset for future queries. This process comprises plaintext operations that are also necessary in an unprotected outsourced Kriging scheme.
3. $\mathcal{C}$ makes a query to $\mathcal{S}$ requesting a Kriging prediction at a location (x_0, y_0); this is done in plaintext with virtually no cost.
4. $\mathcal{S}$ computes the interpolation on encrypted measurements. The cost with respect to an unprotected outsourced Kriging scheme is increased by $O(n)$, due to operations over encrypted data.
5. $\mathcal{C}$ decrypts the result.

Cryptographically-secured Kriging was previously studied in a different setting, where a *server* owns a dataset and clients may query the dataset at a previously unsampled location [12]: the queried location and resulting prediction should be private from the server, whilst the dataset held by the server should be private from the client. Two solutions are proposed in [12] which, unlike our solution, support only one variogram model and require high communication complexity, interactivity and local computation. The first is based on creating random 'dummy' queries to hide the queried location, and using an oblivious transfer protocol to hide predictions for all but the legitimate query location. The second solution uses the Paillier encryption scheme in an interactive protocol requiring multiple round-trips between client and server. In [13] collaborative private Kriging was investigated, where users combine their datasets to gain more accurate Kriging predictions.

The remainder of this paper is structured as follows. In Section 2 we describe the Kriging interpolation process (additional details may be found in Appendix A). In Section 3 we define our system model and analyse the required security properties of each piece of data in our setting. In Section 4 we introduce the idea of a *canonical* variogram, which we use in our construction to allow the server to compute a Kriging prediction without relying on the sensitive parameters. Our construction is given in Section 5 and we discuss its performance in Section 6. Finally in Section 7 we conclude the article with some final remarks and outline some potential directions for future work.

[3] In fact, it suffices to consider Somewhat Homomorphic Encryption rather than FHE as the functionality is fixed and has a reasonably low multiplicative depth.

2 Kriging Interpolation

This section outlines the background theory of Kriging Interpolation. For more detail, see Appendix A and [6,7,9,14]. There are many variants of Kriging, but we focus on the widely used *Ordinary Kriging* variant.

The Kriging process starts with a set of measurements taken at some locations in a spatial region, and produces predicted measurements at unsampled locations. We denote this spatial region by $R \subset \mathbb{R}^2$ and denote the locations of prior measurements by $P = (r_1, r_2, \ldots, r_n)$, where each $r_i = (x_i, y_i) \in R$. The Euclidean distance between two locations $r_i, r_j \in R$ is denoted by $d(r_i, r_j)$. We refer to the set of taken measurements by $S = (z_1, z_2, \ldots, z_n)$, where z_i is measured at the location $r_i \in P$. The *Kriging dataset* then is the tuple (P, S).

The Kriging process allows a client to query an arbitrary location $r_0 \in R$ in order to receive a prediction z_0^* of the true value z_0 that would be measured at r_0. Informally, Kriging consists of three phases:

1. *Computing the experimental variogram*: one of the underlying assumptions of the Kriging process is that two measurements of a phenomenon will be similar if measured in nearby locations. Using the sampled dataset, one can plot the *experimental variogram* to show the dependence between measurements sampled at locations at certain distances h.
2. *Fitting a variogram model*: unfortunately, the experimental variogram is not usually sufficient to use in the Kriging prediction directly, since there may not be sampled data at every required distance. Therefore, one chooses a parametric *variogram model* and empirically chooses model parameters to fit a curve to the points of the experimental variogram.
3. *Computing the prediction*: using the variogram, one can determine appropriate weights for each measurement (based on the distance between each measurement and the queried location). The Kriging prediction is then computed as a weighted sum of the measured samples.

Let $N(h) = \{(z_i, z_j) : d(r_i, r_j) \in (h - \mathit{\Delta}, h + \mathit{\Delta})\}$ be the set of all pairs of measurements taken approximately distance h apart[4]. The *experimental variogram* γ^* plots, for every distance h such that $N(h) \neq \emptyset$:

$$\gamma^*(h) = \frac{1}{2N(h)} \sum_{(z_i, z_j) \in N(h)} (z_i - z_j)^2.$$

A suitable variogram function $\gamma : \mathbb{R}^{\geq 0} \to \mathbb{R}$, in phase 2, must satisfy a set of conditions [6,7]; the most commonly used models require that $\gamma(0) = 0$, that $\gamma(h)$ is positive and bounded, and the existence of the limits $\lim_{h \to 0^+} \gamma(h)$ and $\lim_{h \to \infty} \gamma(h)$. These models are parametrized by the following three variables:

- The *nugget effect* η: The limit of $\gamma(h)$ as $h \to 0^+$.
- The *sill* ν: The limit of $\gamma(h)$ as $h \to \infty$.
- The *range* ρ: Controls how fast $\gamma(h)$ approaches ν as h increases.

Typically, one chooses a variogram model from a set of standard parametric variogram models, and then fits the model to the experimental variogram by empirically adjusting the nugget effect, sill and range parameters. A selection of the most common choices of bounded variogram models are, for $h > 0$:

- The *bounded linear model*: $\gamma(h) = \nu - (\nu - \eta)\left(1 - \frac{h}{\rho}\right) 1_{(0,\rho)}(h)$.
- The *exponential variogram model*: $\gamma(h) = \nu - (\nu - \eta) e^{-h/\rho}$.
- The *spherical variogram model*: $\gamma(h) = \nu - (\nu - \eta)\left(1 - \frac{3h}{2\rho} + \frac{h^3}{2\rho^3}\right) 1_{(0,\rho)}(h)$.
- The *Gaussian variogram model*: $\gamma(h) = \nu - (\nu - \eta) e^{-h^2/\rho^2}$.

where $1_I(x) = 1$ if $x \in I$, and $1_I(x) = 0$ otherwise.

Let γ be one of the above variogram models instantiated with empirically chosen parameters. To construct the best unbiased linear predictor of the phenomenon at a queried location $r_0 = (x_0, y_0) \in R$, we first form the *Kriging matrix* $K \in \mathbb{R}^{(n+1)\times(n+1)}$ with elements:

[4] The approximation tolerance $\mathit{\Delta}$ can be increased when the Kriging dataset does not include enough sample points at a close enough distance.

- $K_{i,j} = \gamma(d(r_i, r_j))$ for $1 \leq i, j \leq n$,
- $K_{n+1,i} = K_{i,n+1} = 1$ for $i \neq n+1$, and
- $K_{n+1,n+1} = 0$.

Next, define a real vector $v \in \mathbb{R}^{n+1}$ with $v_i = \gamma(d(r_0, r_i))$ for $1 \leq i \leq n$, and $v_{n+1} = 1$. Let $\lambda = (\lambda_i)_{i=1}^{n+1}$ satisfy $K\lambda = v$. The *(Ordinary) Kriging prediction* z_0^* of the measured phenomena at the location r_0 is computed as the weighted sum of the sampled measurements, with the weights defined by λ. That is,

$$z_0^* = \sum_{i=1}^{n} \lambda_i z_i.$$

The set of linear equations defined by K and v are known as the *Normal Equations.* They are derived by imposing that the induced linear predictor is unbiased (by ensuring that the first n weights sum to one; that is $\sum_{i=1}^{n} \lambda_i = 1$) while minimizing the variance of the induced linear predictor [14].

The resulting minimized variance σ_0^{*2} is called the *(Ordinary) Kriging variance*, and it is described by the following expression

$$\sigma_0^{*2} = \lambda_{n+1} + \sum_{i=1}^{n} \lambda_i \gamma(d(r_0, r_i)).$$

The Kriging variance allows the construction of confidence intervals for each prediction and thus describes the error associated to the prediction. For a reference on the computation of confidence intervals in this context, see [7].

We define a variogram function to be *non-degenerate* if $\eta \neq \nu$ *i.e.* if γ is non-constant for $h > 0$. We restrict our attention to non-degenerate variogram functions. It is easy to see that using the degenerate variogram (also called the *nugget effect* variogram [14]) results in the average Kriging predictor $z_0^* = \sum_{i=1}^{n} z_i/n$ at all unsampled locations $r_0 \notin P$, with Kriging variance $\sigma_0^{*2} = n+1$.

3 Private Outsourced Kriging Interpolation

Consider a system comprising a client $\mathcal{C}$ that owns a Kriging dataset (P, S) along with a choice of variogram γ, a server $\mathcal{S}$ that is willing to perform outsourced Kriging on behalf of the client, and additional users $\mathcal{U}$ that are authorised by $\mathcal{C}$ to make Kriging queries to $\mathcal{S}$. Furthermore, there may be additional data generating nodes (*e.g* other users or remote sensors *etc.*) that may update the outsourced dataset by producing additional measurement data or removing prior (*e.g* outdated) measurements. The requirements of each entity are as follows:

- The data owner must choose the variogram to be used and upload a Kriging dataset, and should be able to update data and request Kriging predictions.
- Data users may request Kriging predictions and update data.
- Data generating nodes should only be able to update data.
- The server should only be able to perform Kriging predictions, and should do so without learning the data used in the computation. We assume that the server $\mathcal{S}$ is honest-but-curious, i.e. it follows the Kriging protocol (indeed, its business model may depend on doing so) but may attempt to learn information about the outsourced data.

Informally, the protocol runs as follows. The data owner $\mathcal{C}$ chooses the variogram to be used and runs the Outsource algorithm to generate the (protected) dataset to be sent to the server, as well as 'keys' that are issued to authorise entities to update the outsourced dataset or to perform Kriging queries respectively. Upon receipt of the protected data, the server may run the Setup algorithm to process the data and perform any necessary precomputation. After this step, the system is ready to accept queries. The data owner or an authorised data user (in possession of the query key) may request a Kriging prediction at a specified location by running the Query algorithm to generate a query token Q. This is sent to the server who runs the Interpolate algorithm using the processed database to generate an encrypted prediction and an encoding of the Kriging variance (the estimation of the error in the prediction). An entity authorised to perform queries may learn the prediction and variance by running the Decrypt algorithm. To dynamically update the outsourced dataset, an authorised entity (in possession of the update key) may run the AddRequest algorithm on a specified location r' and measurement z', or the DeleteRequest algorithm on a specified location r. These algorithms produce an addition token $\alpha_{r',z'}$ or deletion token δ_r respectively that is sent to the server. Upon receipt of such a token, the server may run the Add or Delete algorithm respectively to update the database accordingly.

For the purposes of this paper, we assume that any user authorised to generate a Kriging query is also permitted to update the dataset. If this should not be the case, then the proposed construction can be easily modified to include a digital signature computed on any addition or deletion token, where the signing key is contained in the update key (and not the query key). The server should be trusted to reject any tokens that do not have a valid signature. Then, only users in possession of the private signature key would be able to update the dataset.

Definition 1. *A private outsourced Kriging interpolation scheme comprises the following algorithms:*

- $(C, UK, QK) \xleftarrow{\$} \mathsf{Outsource}(1^\lambda, P, S, \gamma)$*: A probabilistic algorithm run by* $\mathcal{C}$ *which takes as input a security parameter* λ*, the Kriging dataset comprising measurement locations* P *and measurement values* S*, and the chosen variogram* γ*. It produces an outsourceable data set* C *that may be transmitted to the server, an update key* UK *that may used to update the outsourced dataset, and a query key* QK *which may be used to form Kriging queries.*
- $\mathrm{DB} \leftarrow \mathsf{Setup}(C)$*: A deterministic algorithm run by* $\mathcal{S}$ *which takes as input the outsourceable dataset* C*. This algorithm enables* $\mathcal{S}$ *to perform any necessary processing that will enable it to compute Kriging predictions, and produces a processed outsourced dataset* DB*.*
- $Q \xleftarrow{\$} \mathsf{Query}(r_0, QK)$*: A probabilistic algorithm run by* $\mathcal{C}$ *or a data user in* $\mathcal{U}$ *which takes as input a location* $r_0 = (x_0, y_0) \in R$ *for which a Kriging prediction should be computed, and the query key* QK*. It produces a query token* Q *which is sent to* $\mathcal{S}$*.*
- $(\tilde{Z_0}, \tilde{\sigma_0}^{*2}) \leftarrow \mathsf{Interpolate}(Q, \mathrm{DB})$*: A deterministic algorithm run by* $\mathcal{S}$ *that, given a query token* Q *and the database* DB*, returns an encrypted Kriging interpolation* $\tilde{Z_0}$ *and the partially computed Kriging variance* $\tilde{\sigma_0}^{*2}$*.*
- $(z_0^*, \sigma_0^{*2}) \leftarrow \mathsf{Decrypt}(\tilde{Z_0}, \tilde{\sigma_0}^{*2}, QK)$*: A deterministic algorithm run by* $\mathcal{C}$ *or a user in* $\mathcal{U}$ *that takes the Kriging results* $\tilde{Z_0}$ *and* $\tilde{\sigma_0}^{*2}$ *from the server and the query key* QK*, and outputs the Kriging prediction* z_0^* *and the Kriging variance* σ_0^{*2} *at the queried location.*
- $\alpha_{r',z'} \leftarrow \mathsf{AddRequest}(r', z', UK)$*: A deterministic algorithm run by* $\mathcal{C}$*, a data user in* $\mathcal{U}$ *or a data generating node, which takes a location* r'*, a measurement value* z' *and the update key* UK*, and outputs an addition token* $\alpha_{r',z'}$*.*
- $\mathrm{DB}' \leftarrow \mathsf{Add}(\mathrm{DB}, \alpha_{r',z'},)$*: A deterministic algorithm run by* $\mathcal{S}$ *which takes the current outsourced database* DB *and an addition token* $\alpha_{r',z'}$*, and outputs an updated database* DB' *representing the Kriging dataset* $(P \cup \{r'\}, S \cup \{z'\})$*.*
- $\delta_r \leftarrow \mathsf{DeleteRequest}(r, UK)$*: A deterministic algorithm run by* $\mathcal{C}$*, a data user in* $\mathcal{U}$ *or a data generating node. The algorithm takes as input a location* $r \in P$ *and the update key* UK *and outputs a deletion token* δ_r*.*
- $\mathrm{DB}' \leftarrow \mathsf{Delete}(\mathrm{DB}, \delta_r)$*: A deterministic algorithm by the server which takes as input the current database* DB *and a deletion token* δ_r *and outputs an updated database* DB' *representing the Kriging dataset* $(P \setminus \{r\}, S \setminus \{z_r\})$ *where* $z_r \in S$ *is the measurement corresponding to location* $r \in P$ *in* DB*.*

We now analyse the security requirements of each component within a Kriging system; Table 1 summarizes the analysis:

- The measurement values $z_i \in S$ are highly sensitive and business-critical and must be protected at all times.
- In the current work, we consider the coordinates $r_i \in P$ of previous measurements to not be sensitive. This is reasonable, since in some applications they may be externally observable, for instance if they are the locations of previous mining activity.
- The queried location r_0 at which a new prediction should be computed may reveal areas of particular interest to the user. The sensitivity of this relies on the setting and individual user requirements. However, in practice, Kriging queries are often made at *every* location within a

Data	r_i	z_i	(x_0, y_0)	z_0^*	γ model	ρ	ν	η
Protection	✗	✓	✗	✓	✗	✗	✓	✓

Table 1: Data protection offered by our private outsourced Kriging scheme.

region to produce a heat map of a phenomenon, which may limit the sensitivity of individual query locations. Further, the basic assumption of Kriging is that the quality of prediction degrades with distance; thus, the best Kriging results will be obtained when the queried location is broadly within the region of prior (observed) measurements.

- The computed prediction z_0^* is highly sensitive as it may form the basis of future decisions and may be business-critical, and must be protected.
- The choice of variogram model (without the variogram parameters) may reveal something about the overall trend of the spatial dependencies of the measurements. We assume that this is not particularly sensitive information.
- The range parameter ρ of the variogram is a constant scaling of the region R denoting the inter-measurement distance h at which the spatial dependency becomes negligible. For distances $h > \rho$, the variogram approaches the variance of the measurements [14], which is represented by the sill ν.
  The nugget effect η reveals the spatial dependency at very small distances.
  In this work, we assume that the range is not sensitive (as it merely scales the region R), but that information revealed by the nugget and sill may be sensitive. Even in applications where this direct information on the variance and spatial dependency of measurements is deemed non-sensitive, it may be the case that the variogram parameters are commercially sensitive. These parameters must be chosen empirically to best fit the experimental data, a process which may be time-consuming, and the quality of predictions depends on how well the variogram matches the experimental variogram.

4 Our Techniques

In this section we introduce the main concept used in our construction, namely the canonical variogram. We then show how to factor out the variogram parameters in the Normal equations which, ultimately, allows us to remove these parameters from the outsourced dataset and use them only to recover the final prediction on the client side.

The crux of our solution for the private outsourcing of Kriging interpolation is to observe how the Kriging solution varies according to the variogram nugget effect η, the sill ν, and range ρ in the non-degenerate case. We define a *canonical* variogram for each variogram model by arbitrarily fixing the parameters $\eta = \rho = 1$ and $\nu = 0$, although our results clearly translate to other choices.

Since the Kriging process is inherently linear, we show how to 'factor out' the sensitive parameters η and ν from the variogram to leave just the canonical variogram. Using this result and an additively homomorphic scheme, an untrusted server can compute a related Kriging prediction and variance without any knowledge of η, ν and the actual measurements. The variogram parameters can then be efficiently re-added by the client locally to compute the final prediction.

Definition 2 (Canonical variogram). *Let $\gamma(h)$ be a non-degenerate variogram function with nugget effect η, sill ν and range ρ. We define its associated* canonical variogram *as the function $\tilde{\gamma} : \mathbb{R}^{\geq 0} \to \mathbb{R}$ satisfying $\tilde{\gamma}(0) = 0$ and*

$$\tilde{\gamma}(h) = -\frac{1}{\nu - \eta}\gamma(\rho h) + \frac{\nu}{\nu - \eta} \qquad \text{for } h > 0. \tag{1}$$

Note that for any non-degenerate variogram function coming from the parametric variogram models defined in Section 2, the canonical variogram depends only on the considered model itself and not on any parameters.

Now, given a Kriging dataset (P, S) of n measurements, a query position $r_0 \notin P$ and a variogram function γ with nugget effect η, sill ν and range ρ, let $K\lambda = v$ be the corresponding Normal equations as defined in Section 2. Our main result in this stage is that it suffices to consider a canonical version of the Normal equations that depends only on the chosen variogram model, as well as P and the range parameter ρ of γ.

Definition 3. *We define the* canonical Normal equations *as the linear system obtained from the Normal equations $K\lambda = v$ by replacing*

- *every $r_i \in P$ by r_i/ρ,*
- *the query position r_0 by r_0/ρ,*
- *the variogram $\gamma(h)$ by the canonical variogram $\tilde{\gamma}(h)$,*

and we denote the canonical Normal equations by $\tilde{K}\tilde{\lambda} = \tilde{v}$.

Note that, since the canonical variogram is parameterless, the canonical Normal equations involve only the variogram model and the locations in P scaled by the inverse of the range parameter ρ. We make extensive use of this observation in our construction. Indeed, this observation allows us to take advantage of the linearity of the Kriging predictor, in order to protect the measurements and interpolation value, while hiding the sill and nugget parameters ν, η from the server by storing them locally.

The solution to the canonical Normal equations can be described as follows:

Proposition 1. *Let $K, K' \in \mathbb{R}^{(n+1)\times(n+1)}$ be real matrices, and let $v, v' \in \mathbb{R}^{n+1}$ be real vectors such that:*

- *there exist $a, b \in \mathbb{R}$ such that $K'_{i,j} = aK_{i,j} + b$ and $v'_i = av_i + b$ for all $1 \leq i, j \leq n$,*
- *$K_{i,n+1} = K_{n+1,i} = K'_{i,n+1} = K'_{n+1,i} = v_{n+1} = v'_{n+1} = 1$ for all $1 \leq i \leq n$,*
- *$K_{n+1,n+1} = K'_{n+1,n+1} = 0$.*

Then, if $\lambda \in \mathbb{R}^{n+1}$ satisfies $K\lambda = v$, the vector $\lambda' \in \mathbb{R}^{n+1}$ defined by

$$\lambda'_i = \lambda_i \text{ for all } 1 \leq i \leq n,$$
$$\lambda'_{n+1} = a\lambda_{n+1}$$

satisfies $K'\lambda' = v'$.

Proof. Note that $(K'\lambda')_i = av_i + b\sum_{i=1}^{n} \lambda_i$ for $1 \leq i \leq n$, and $(K'\lambda')_{n+1} = 1$. Since $\sum_{i=1}^{n} \lambda_i = 1$ (by the last equation of the system $K\lambda = v$), the result follows. □

This result extends an observation by [7], which states that summing a constant to the variogram does not alter the solutions of the Normal equations, and that such a transformation of the variogram may sometimes be necessary in order to obtain a numerically stable Kriging prediction.

We apply this proposition to the Normal equations with $a = -1/(\nu - \eta)$ and $b = \nu/(\nu - \eta)$, and consider the canonical Normal equations. By the definitions of the Kriging prediction and the Kriging variance in Section 2, we directly obtain the following Corollary.

Corollary 1. *Let z_0^* and $\tilde{z_0}^*$ be the Kriging predictions computed from the Normal and the canonical Normal equations described above, respectively. Denote by σ_0^{*2} and $\tilde{\sigma_0}^{*2}$ the Kriging variance associated to each of the predictors. Then*

$$\tilde{z_0}^* = z_0^* \quad \text{and} \quad \tilde{\sigma_0}^{*2} = -\frac{1}{\nu - \eta}\sigma_0^{*2} + \frac{\nu}{\nu - \eta}.$$

Therefore, in case that the employed variogram is non-degenerate, the Kriging prediction is independent of the sill ν and nugget η parameters of the variogram, whilst the range parameter ρ scales positions. We also see that, when applying a linear transformation to the variogram, the Kriging variance of the obtained Kriging predictor varies according to the same transformation.

5 Our Construction

We now outline the operation of each of the algorithms in Definition 1. Let $\mathcal{H} = (\mathcal{H}.\text{Gen}, \mathcal{H}.\text{Enc}, \mathcal{H}.\text{Dec})$ be an IND-CPA-secure additive homomorphic encryption scheme, such as the Paillier encryption scheme [11]. Then:

- $(C, UK, QK) \overset{\$}{\leftarrow} \mathsf{Outsource}(1^\lambda, P, S, \gamma)$: If γ is a degenerate variogram function, halt and return $\perp$; in this case, our protocol fails. However, if γ is degenerate, the variogram is constant (the so-called 'nugget effect model') and models a purely random variable with no spatial correlation. Hence it is particularly easy to compute predictions in this case: the prediction is $z_0^* = \sum z_i/n$ for $r_0 \notin P$ and the variance is $\sigma_0^{*2} = n + 1$.
  Otherwise, generate a key-pair for the homomorphic encryption scheme:
  $$(pk, sk) \overset{\$}{\leftarrow} \mathcal{H}.\mathrm{Gen}(1^\lambda).$$
  Recall that $P \subseteq \mathbb{R}^2$ is the ordered set of locations $(r_i)_{i=1}^n$ and that $S \subseteq \mathbb{R}$ is the ordered set of measurements $(z_i)_{i=1}^n$. Recall also that the variogram γ comprises three parameters: the nugget η, the sill ν and the range ρ. Let $\tilde{\gamma}$ be the canonical variogram associated to γ, as defined in Section 4. Define
  $$UK = (pk, \rho) \text{ and } QK = (sk, \eta, \nu, \rho).$$
  To account for the factor of ρ in the input to γ in equation 1, compute
  $$\tilde{P} = ((x_i/\rho, y_i/\rho))_{i=1}^n.$$
  Finally, encrypt each measurement in S and define the ordered set
  $$\mathrm{Z} = (\mathcal{H}.\mathrm{Enc}_{pk}(z_i))_{i=1}^n.$$
  Output $C = (\tilde{P}, \mathrm{Z}, \tilde{\gamma})$, along with UK and QK.
- $\mathrm{DB} \leftarrow \mathsf{Setup}(C)$: Instantiate the matrix $\tilde{K}$ from the canonical Normal equations using positions in $r'_i \in \tilde{P}$ and the canonical variogram function $\tilde{\gamma}$:
  - $\tilde{K}_{i,j} = \tilde{\gamma}(d(r'_i, r'_j))$ for $1 \leq i, j \leq n$,
  - $\tilde{K}_{n+1,i} = \tilde{K}_{i,n+1} = 1$ for $i \neq n+1$, and
  - $\tilde{K}_{n+1,n+1} = 0$.

  Return $\mathrm{DB} = (\tilde{K}, C)$.
- $Q \overset{\$}{\leftarrow} \mathsf{Query}(r_0, QK)$: Let $r_0 = (x_0, y_0)$ and, recalling that ρ is contained within QK, return $Q = (x_0/\rho, y_0/\rho)$.
- $(\tilde{Z_0}, \tilde{\sigma_0}^{*2}) \leftarrow \mathsf{Interpolate}(Q, \mathrm{DB})$: Recall that $C = (\tilde{P}, \mathrm{Z}, \tilde{\gamma})$. If $Q \in \tilde{P}$, then the exact measurement is contained in the outsourced dataset and no prediction is required. Let j be the index such that $Q = r_j$, and return $(Z_j, \perp)$, where $\perp$ is a distinguished symbol denoting that the prediction is exact.
  Otherwise, compute the vector $\tilde{v}$ from the canonical Normal equations using the locations $r'_i \in \tilde{P}$, the query position Q and the canonical variogram $\tilde{\gamma}$:
  - $v_i = \tilde{\gamma}(d(Q, r'_i))$ for $1 \leq i \leq n$, and
  - $v_{n+1} = 1$.

  Compute the solution $\tilde{\lambda}$ to the canonical Normal equation $\tilde{K}\tilde{\lambda} = \tilde{v}$; this step essentially computes the Kriging coefficients λ using the canonical variogram and the scaled locations *without* requiring the parameters of the variogram. Then, using the homomorphic property of the encryption, compute:
  $$\tilde{Z_0} = \sum_{i=1}^{n} \tilde{\lambda}_i Z_i \text{ and } \tilde{\sigma_0}^{*2} = \tilde{\lambda}_{n+1} + \sum_{i=1}^{n} \tilde{\lambda}_i \tilde{\gamma}(Q, r'_i).$$
  Return the encrypted prediction $\tilde{Z_0}$ and the partially computed Kriging variance (error estimation) $\tilde{\sigma_0}^{*2}$.
- $(z_0^*, \sigma_0^{*2}) \leftarrow \mathsf{Decrypt}(\tilde{Z_0}, \tilde{\sigma_0}^{*2}, QK)$: First decrypt the Kriging prediction:
  $$\tilde{z_0}^* = \mathcal{H}.\mathrm{Dec}_{sk}(\tilde{Z_0}),$$
  where sk is contained within QK. Then, if $\tilde{\sigma_0}^{*2} = \perp$, set $\sigma_0^{*2} = 0$. Else, compute the Kriging variance
  $$\sigma_0^{*2} = \nu - (\nu - \eta)\tilde{\sigma_0}^{*2}.$$
  This final step essentially adds back in the parameters of the variogram, which were removed for outsourcing, using the result from Corollary 1.

- $\alpha_{r',z'} \leftarrow \mathsf{AddRequest}(r', z', UK)$: Let $r_a = \frac{r'}{\rho}$ and compute the ciphertext

  $$Z_a = \mathcal{H}.\text{Enc}_{pk}(z'),$$

  where ρ and pk are contained within UK. Output the addition token

  $$\alpha_{r',z'} = (r_a, Z_a).$$

- $\text{DB}' \leftarrow \mathsf{Add}(\text{DB}, \alpha_{r',z'})$: Recall that $\alpha_{r',z'} = (r_a, Z_a)$. Compute the updated dataset: if $r_a \in \tilde{P}$ then let j be the index such that $r_j = r_a$ and modify $Z_j \in \mathrm{Z}$ to be Z_a. Otherwise, set $C' = (\tilde{P} \cup \{r_a\}, \mathrm{Z} \cup \{Z_a\}, \tilde{\gamma})$. Return the output of $\mathsf{Setup}(C')$.
- $\delta_r \leftarrow \mathsf{DeleteRequest}(r, UK)$: Return $\delta_r = r/\rho$.
- $\text{DB}' \leftarrow \mathsf{Delete}(\text{DB}, \delta_r)$: If $\delta_r \notin \tilde{P}$, return DB as there is nothing to remove. Otherwise, let j be the index such that $r = r_j$ in $\tilde{P}$. Compute the updated dataset $C' = (\tilde{P} \setminus \{r_j\}, \mathrm{Z} \setminus \{Z_j\}, \tilde{\gamma})$ and return the output of $\mathsf{Setup}(C')$.

6 Discussion

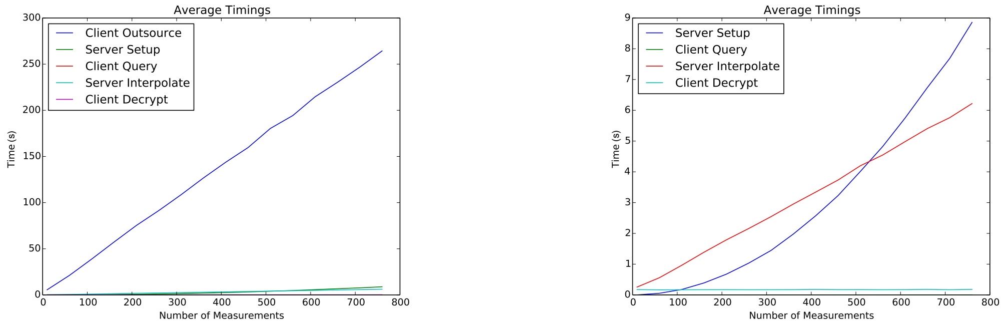
The correctness of the scheme is immediate from Corollary 1 as well as the correctness and homomorphic properties of the encryption scheme $\mathcal{H}$. These homomorphic properties enable addition and scalar multiplication of ciphertexts, whilst ensuring that the results decrypt appropriately. Corollary 1 shows that the Kriging prediction, as well as the Kriging variance, can be computed by applying a linear transformation to the result computed using the canonical (parameterless) variogram. Correctness of the updates is apparent because the addition and deletion tokens format the data in the same way as the original dataset. Since the server is trusted to act honestly (but curiously), it shall modify the dataset correctly; the remainder of the update algorithms then simulate a new setup procedure running Setup on a new Kriging dataset from Outsource.

In terms of security, it is easy to see that the measurement values are always in encrypted form whilst outsourced, and that the leakage is bounded by the variogram model as well as both the queried and observed locations (scaled by the inverse of range parameter ρ). Thus, assuming no collusion between the server and users, the data is confidential from the server. Furthermore, the homomorphic and security properties of the encryption scheme permit the computation to be performed on the measurements whilst they are encrypted; at no point during the computation is the data revealed. The security of the encryption scheme requires each ciphertext to be indistinguishable from a random number, whilst the final prediction $\tilde{Z_0}$ computed by the server comprises a weighted sum of such pseudorandom numbers. Thus, $\tilde{Z_0}$ is a valid ciphertext and is indistinguishable from random, and hence the server cannot learn the prediction from this value.

It is also clear that neither the variogram parameters η and ν, nor any values computed from them, are ever revealed to the server. The final parameter of the variogram, the range ρ, is never explicitly given to the server. However, the server does learn the coordinates of measurements scaled by ρ. Hence, the range could be revealed *if* the server has existing knowledge of the measurement locations. Of the three variogram parameters, we believe that the range is the least sensitive — it reveals how quickly the variogram approaches the sill (*i.e* the distance at which the spatial correlation between measurements becomes negligible) but does not reveal anything relating to the measurement values themselves.

Whilst the queried location is revealed in the plain to the server, we note that the mechanism of Tugrul and Polat [12] may easily be used to gain a weak form of secrecy: during the Query algorithm, the party carrying out the query may choose $q - 1$ additional locations from the region, and scale each by ρ. The query token then comprises q scaled locations, randomly permuted. The server must perform Interpolate for each location, and the client may discard all results except the one it is interested in. Unlike [12], we do not require an oblivious transfer protocol since the querier is authorised to learn as many queries on the dataset as it wishes. However, as in [12], the server may guess the location of interest with probability $1/q$ (but cannot learn the prediction at this location).

Data generating nodes cannot learn Kriging predictions as they do not have the decryption key and $\mathcal{H}$ is assumed to be IND-CPA secure.



(a) All algorithms

(b) Excluding Outsource

Fig. 1: Graphs showing the timing costs of each algorithm.

Regarding the performance evaluation of our scheme, we have implemented our scheme in Python 3.4.3 using the PHE library [3] to provide the Paillier encryption scheme. The implementation is intended as a proof of concept to evaluate the efficiency of the proposed solution. The encryption scheme has not been further optimised beyond that provided by default in the PHE library, and does not use the provided countermeasures to avoid leaking the exponent of floating point numbers. We remark that implementations of Paillier typically manage issues related to fixed-point arithmetic and overflows in a transparent manner; it is not the aim of this article to discuss such issues. All code is executed locally on a t2.micro Amazon EC2 instance with a 2.5GHz Intel Xeon processor and 1GB memory running Ubuntu 14.04.4; in practice, one would expect the server to have a better specification. All timings are averaged over 30 iterations, each on a new randomly generated dataset.

Figures 1a and 1b give some simple timing results using our construction; Figure 1b shows the per-algorithm costs (excluding the update algorithms). The cost of the Outsource algorithm dominates all others (due to the cost of n encryptions); hence, for clarity, Figure 1a shows the same results with the exclusion of the Outsource algorithm. It can be seen that, with the exception of the (high) one-time cost of Outsource (which may be amortised over many queries), the remaining client-side processes are very efficient. The server must perform quadratic work to perform Setup, but this will be required relatively rarely — during initial setup and when the outsourced dataset is updated. The online workload of the client is very low, whilst the server's online work is linear in the size of the dataset and greater than the client's workload (making outsourcing worthwhile). We believe that these experiments are sufficient to demonstrate the performance and scalability of our solution; to our knowledge, the range of the number of measurements is reasonable compared to what may be used in practice — for example, the well-known Meuse dataset [5] (often used to illustrate the Kriging process) comprises 155 measurements.

7 Conclusion

The Kriging interpolation technique describes the best unbiased linear prediction of an observed phenomena in a geographical region, based on a set of measurements, and it is widely used in a wide range of applications. In this article we present a construction that allows for Kriging interpolation to be securely outsourced to a cloud service provider, such that the measurement values and sensitive variogram parameters are withheld from the server.

The proposed construction may be extended in several ways. For example, it would be interesting to protect locations. This can be easily achieved if we increased interactivity, communication complexity and client computation in the query process. However, if most computations should be done by the server, it seems necessary to efficiently compute square roots and natural exponentials over encrypted data which, to the best of our knowledge, remains an open problem. Finally, although we have focused on Kriging due to its current practical applications, it would be interesting to consider whether the techniques presented here could be applied in similar problems such as outsourced polynomial curve fitting and regression techniques such as linear or generalized least squares.

8 Acknowledgements

Oriol Farràs and Jordi Ribes-González were supported by the European Comission through H2020-ICT-2014-1-644024 “CLARUS” and H2020-DS-2015-1-700540 “CANVAS”, by the Government of Spain through TIN2014-57364-C2-1-R “SmartGlacis” and TIN2016-80250-R “Sec-MCloud”, by the Government of Catalonia through Grant 2014 SGR 537, and by COST Action IC1306. James Alderman was supported by the European Comission through H2020-ICT-2014-1-644024 “CLARUS”. Benjamin R. Curtis was supported by the UK EPSRC through EP/K035584/1 “Centre for Doctoral Training in Cyber Security at Royal Holloway”.

References

1. CLARUS: User centered privacy and security in the cloud. http://clarussecure.eu. Accessed: 11/12/2016.
2. InGeoCloudS: inspired geo-data cloud services. https://www.ingeoclouds.eu/. Accessed: 11/12/2016.
3. python-paillier: a library for partially homomorphic encryption in python, Data61|CSIRO. https://github.com/NICTA/python-paillier. Accessed: 11/12/2016.
4. SEAL: Simple encrypted arithmetic library, cryptography research group, microsoft research. http://sealcrypto.codeplex.com/. Accessed: 11/12/2016.
5. P. A. Burrough, R. McDonnell, R. A. McDonnell, and C. D. Lloyd. *Principles of geographical information systems*. Oxford University Press, 2015.
6. J.-P. Chilès and P. Delfiner. Multivariate methods. *Geostatistics: Modeling Spatial Uncertainty, Second Edition*, pages 299–385, 1999.
7. N. Cressie. Statistics for spatial data. *Terra Nova*, 4(5):613–617, 1992.
8. EU Parliament. Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an infrastructure for spatial information in the European Community (INSPIRE). *Official Journal of the European Union*, 50(L108), 2007.
9. D. Krige. A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6):119–139, 1951.
10. G. Matheron. *Traité de géostatistique appliquée*. Mémoires du Bureau de Recherches Géologiques et Minières. Éditions Technip, 1962-63.
11. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer, 1999.
12. B. Tugrul and H. Polat. Estimating kriging-based predictions with privacy. *International Journal of Innovative Computing, Information and Control, Accepted for publication*, 2013.
13. B. Tugrul and H. Polat. Privacy-preserving kriging interpolation on partitioned data. *Knowledge-Based Systems*, 62:38–46, 2014.
14. H. Wackernagel. *Multivariate geostatistics: an introduction with applications*. Springer Science & Business Media, 2013.

A Additional Details on Kriging

In order to apply the Kriging interpolation technique, the observed phenomena is viewed as a realization of a *random field* which satisfies certain properties related to the observed measurements. A random field generalizes the notion of stochastic process, by allowing the underlying parameter to take values other than real numbers. In the case of spatial interpolation, a random field Z is defined as a collection of real-valued random variables $\{Z(r)\}_{r\in R}$, all defined in the same probability space, and indexed by locations r in a fixed region $R \subseteq \mathbb{R}^2$.

Given a set of n samples S taken at positions P, every sample $z_i \in S$ can be viewed as a realization of the random variable $Z(r_i)$, indexed by the position $r_i \in P$ in a random field Z. Given such realizations, a *linear predictor* Z^* of the random field Z is defined as a random field of the form

$$Z^*(r) = \lambda_0 + \sum_{i=1}^{n} \lambda_i Z(r_i), \qquad \text{where } \lambda_i \in \mathbb{R}.$$

We say a linear predictor Z^* is *unbiased* if the expectation $\mathbb{E}(Z(r) - Z^*(r)) = 0$ for all $r \in R$. Moreover, we say that a linear predictor Z^* is *best* or *optimal* if, for every location $r \in P$, it minimizes the prediction variance $\mathrm{Var}(Z(r) - Z^*(r))$ among all unbiased linear predictors.

The Kriging interpolation technique aims to find a best unbiased linear predictor for the random field Z derived from a Kriging dataset (P, S). In this sense, note that Kriging deals with the same problem as linear least squares in random fields. However, in order to derive such a predictor from sampled values, additional assumptions are usually made on the *stationarity* of the random field. The most widely applied Kriging process is *Ordinary Kriging*. This form of Kriging stems from two stationarity assumptions. The *second-order stationarity* assumption states that the first and second-order moments of the random variables in the random field are shift invariant:

Definition 4. *A random field Z parametrized by elements of a region $R \subseteq \mathbb{R}^2$ is defined to be* second-order stationary *if the following conditions are satisfied:*

- *The mean* $\mathrm{E}(Z(r))$ *does not depend on $r \in R$, and*
- *The covariance* $\mathrm{Cov}(Z(r), Z(r+h))$ *is a function of only the separating vector h for every $r, r+h \in R$.*

The *intrinsic stationarity* assumption considers variance of increments instead of covariance:

Definition 5. *A random field Z parametrized by elements of a region $R \subseteq \mathbb{R}^2$ is defined to be* intrinsic stationary *if the following conditions are satisfied:*

- *The mean* $\mathrm{E}(Z(r))$ *does not depend on $r \in R$, and*
- *The variance of the increments* $\mathrm{Var}(Z(r+h) - Z(r))$ *is a function of only the separating vector h for every $r, r+h \in R$.*

Second-order stationarity implies intrinsic stationarity [14] and thus we restrict our attention to the more general intrinsic stationarity assumption. Our techniques are, however, applicable to Ordinary Kriging in general.

The intrinsic stationarity assumption naturally leads to the notion of *theoretical variogram* [7,10] which models the spatial dependency between the random variables $Z(r)$. Given an intrinsic stationary random field Z, the *theoretical variogram* $\hat{\gamma} : R \to \mathbb{R}$ is defined as the function $\hat{\gamma}(h) = \mathrm{Var}(Z(r+h) - Z(r))$. Under the intrinsic assumption, $\hat{\gamma}(h)$ depends only on the norm of h [14]. Hence, we may view $\hat{\gamma}$ as a function defined over positive real numbers.