Practical Secure OMP Computation and Its Application to Image Modeling

Takayuki Nakachi
NTT Network Innovation Labs.
NTT Corporation
Yokosuka, Kanagawa 239-0847 Japan
nakachi.takayuki@lab.ntt.co.jp

Hitoshi Kiya
School of System Design
Tokyo Metropolitan University
Hino, Tokyo 191-0065 Japan
kiya@tmu.ac.jp

ABSTRACT

In this paper, we propose a secure computation of sparse coding using a random unitary transform. A cloud computing is spreading in many application fields including applications of the sparse coding. This situation raises many new privacy concerns. The proposed scheme computes an Orthogonal Matching Pursuit (OMP) algorithm in an encrypted form, which can be used in practice. We prove that the proposed secure OMP computation has exact the same sparse coefficients estimation performance as the OMP algorithm in an unencrypted form theoretically. Then, we apply it to image modeling based on a image patch model. Finally, we demonstrate its performance both on synthetic data and in an application on natural image.

Keywords

Sparse Coding, Orthogonal Matching Pursuit (OMP), Random Unitary Transform, Secure Computation

1. INTRODUCTION

Early work on sparse coding was based on the efficient coding hypothesis, which assumes that the goal of visual coding is to faithfully represent the visual input with minimal neural activity. The idea goes back to Barlow [1]. It represents observed signals effectively as a linear combination of a small number of bases which are chosen from the basis functions trained by the algorithm. The sparse coding model has found numerous applications in processing such as image/video, audio, biological signal, seismic data and more [2][3].

On the other hand, a cloud computing is spreading in many fields including applications of the sparse coding. However, the cloud computing has some serious issues for end users, such as unauthorized use and leak of data, and privacy compromise, due to unreliability of providers and some accident [4]. Many studies have been reported for processing encrypted data, which are designed using homomorphic encryption [5]. While service providers cannot access directly the content of the encrypted signals, the homomorphic encryption can still process in encrypted form to perform the required signal processing task. Especially, fully homomorphic encryption allow arbitrary computation on encrypted data. However, it requires a specific computation and a high computational complexity.

Our study focus on a practical and backward compatible secure computation, which can use existing algorithms. The proposed scheme is based on a random unitary transform. So far, a lot of studies on generating efficient random unitary matrices have been reported [6]-[7] for biometric template protection. In this manuscript, we propose a secure computation for an Orthogonal Matching Pursuit (OMP) algorithm [9]. It is shown that the secure OMP computation enables us not only to encrypt signals, but also to have the same performance as that of the OMP algorithm in the unencrypted signals.

Then, we apply it to image modeling. Images are widely used in the cloud computing. Effectiveness of the sparse coding for image processing is reported in the area of image compression, image denoising and image separation, etc [2]. These application is based on an image patch model. Here we consider secure sparse coding for images based on the patch model. The security of image data from unauthorized uses is important. Based on the secure OMP computation mentioned above, we propose a secure sparse image modeling based on the patch model. For example, this model can be applied to Encryption-then-Compression (EtC) systems [8].

Finally, we demonstrate its results both on synthetic data and in an application on natural image. The organization of this paper is as follows. Section 2 describes overview of sparse coding. In Sec. 3, we propose secure OMP computation. Section 4 introduce its application to image processing. Section 5 shows simulation results. Conclusion is given in 6.

IHIP2018 Sep. 22-24, 2018, Manchester, UK

DOI: 10.1145/1235

2. SPARSE CODING

2.1 Sparse Representation

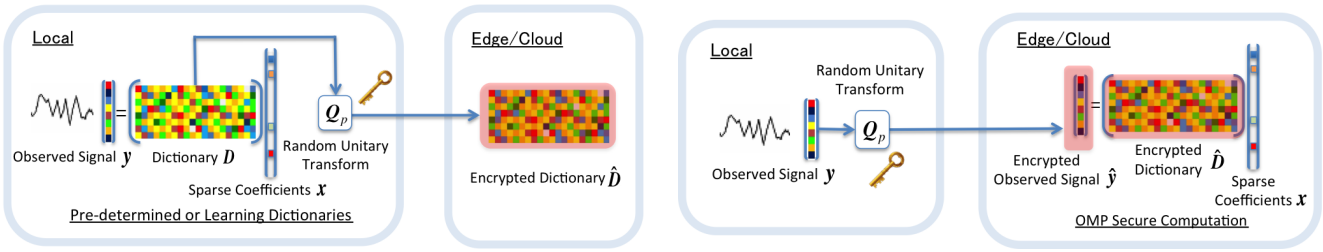
Using an overcomplete dictionary matrix $\boldsymbol{D} = \{\boldsymbol{d}_1, ..., \boldsymbol{d}_K\} \in \mathbb{R}^{n \times K}$ that contains K prototype signal-atoms $\boldsymbol{d}_i$ for columns, a signal $\boldsymbol{y} = \{y_1, , ..., y_n\} \in \mathbb{R}^n$ can be represented as a sparse linear combination of these atoms:

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}. \tag{1}$$

The vector $\boldsymbol{x} = \{x_1, ..., x_K\} \in \mathbb{R}^K$ contains the representation coefficients of the signal $\boldsymbol{y}$.

If $n < K$ and $\boldsymbol{D}$ is a full-rank matrix, an infinite number of solutions are available for the representation problem. The solution with the fewest number of nonzero coefficients is certainly an appealing representation. This sparsest representation is the solution:

$$(P_0) \quad \min_{\boldsymbol{x}_0} \|\boldsymbol{x}\|_0 \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}. \tag{2}$$



(a) Preparation: generating encrypted dictionary

(b) Running: secure OMP computation

Figure 1: Secure sparse coding computation architecture.

where $\|\cdot\|_0$ is the l_0-norm, counting the nonzero entries of a vector. Extraction of the sparsest representation is a NP-hard problem [10].

2.2 Selection of Dictionary Atoms

A selection of dictionary atoms is typically done by a "pursuit algorithm" that finds an approximate solution:

$$x = \arg\min_x \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{x}\|_0 < \epsilon. \tag{3}$$

The well known pursuit algorithms are the Matching Pursuit (MP) [11] and Orthogonal Matching Pursuit (OMP) [9]. These are greedy algorithms that select the dictionary atoms and calculate the sparse coefficients sequentially.

The OMP is a greedy step-wise regression algorithm. At each stage, this method selects the dictionary atom having the maximal projection onto the residual signal. After each selection, the representation coefficients w.r.t. the atoms selected so far are found via least-squares. Given a signal $\boldsymbol{y} \in \mathbb{R}^n$, and a dictionary $\boldsymbol{D}$ with K l_2-normalized columns $\{d_k\}_{k=1}^K$. The following presents a formal description of OMP algorithm.

Orthogonal Matching Pursuit (OMP)

Initialization: $k = 0$, and set

· The initial solution $\boldsymbol{x}^0 = \boldsymbol{0}$
· The initial residual $\boldsymbol{r}^0 = \boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}^0 = \boldsymbol{y}$
· The initial solution support $S^0 = \emptyset$.

Main Iteration:
Increment k by 1 and perform the following steps:

· **Sweep:** Compute the errors

$$\epsilon(i) = \min_{x_i} \|x_i\boldsymbol{d}_i - \boldsymbol{r}^{k-1}\|_2^2 = \|\boldsymbol{r}^{k-1}\|_2^2 - \frac{(\boldsymbol{d}_i \cdot \boldsymbol{r}^{k-1})^2}{\|\boldsymbol{d}_i\|_2^2}. \tag{4}$$

· **Update Support:** Find the minimizer

$$i_0 = \arg\min_{i \notin S^{k-1}} \{\epsilon(i)\}, S^k = S^{k-1} \cup \{i_0\}. \tag{5}$$

· **Update Provisional Solution:** Compute

$$\bar{\boldsymbol{x}}^k = \arg\min_{\boldsymbol{x}_{S^k}} \|\boldsymbol{y} - \boldsymbol{D}_{S^k}\boldsymbol{x}_{S^k}\|_2^2 = (\boldsymbol{D}_{S^k}^*\boldsymbol{D}_{S^k})^{-1}(\boldsymbol{D}_{S^k}^*\boldsymbol{y}). \tag{6}$$

· **Update Residual:** Compute $\boldsymbol{r}^k = \boldsymbol{y} - \boldsymbol{D}_{S^k}\bar{\boldsymbol{x}}^k$.
· **Stopping Rule:** If $\|\boldsymbol{r}^k\|_2 < \epsilon$, stop. Otherwise, apply another iteration.

Output: The proposed solution $\bar{\boldsymbol{x}}^k$ is obtained after k iterations.

Here, we define the atom $\boldsymbol{d}_i$ as follows:

$$\boldsymbol{d}_i = \boldsymbol{D}\boldsymbol{\delta}_i \tag{7}$$

where $\boldsymbol{\delta}_i = [(0, \cdots, 0, \delta(i), 0, \cdots, 0)]^T$ that is equal to zero everywhere except one point (i-th element is 1). The approximation errors $\epsilon(i)$ of Eq. (4) can be expressed as follows:

$$\epsilon(i) = \min_{x_i} \|x_i\boldsymbol{D}\boldsymbol{\delta}_i - \boldsymbol{r}^{k-1}\|_2^2 = \|\boldsymbol{r}^{k-1}\|_2^2 - \frac{(\boldsymbol{D}\boldsymbol{\delta}_i \cdot \boldsymbol{r}^{k-1})^2}{\|\boldsymbol{D}\boldsymbol{\delta}_i\|_2^2}. \tag{4'}$$

3. SECURE COMPUTATION OF SPARSE CODING

3.1 Secure Computation Architecture

Figure 1 illustrates architectures of the secure sparse coding computation. Figure 1(a) shows a preparation in the local site.

A dictionary $\boldsymbol{D}$ is predetermined (overcomplete wavelets, short-time-Fourier transforms, curvelets, etc.) or designed by training algorithms such as MOD [12], K-SVD [13] algorithms. Then a transform function $T(\cdot)$ is applied to the dictionary $\boldsymbol{D}$ to generate encrypted dictionary $\hat{\boldsymbol{D}}$. Then the encrypted dictionary $\hat{\boldsymbol{D}}$ are sent to a edge/cloud site and stored into a database.

Figure 1(b) shows a secure computation of a sparse coefficients selection. The same transform function $T(\cdot)$ is applied to a observed signal $\boldsymbol{y}$ to generate an encrypted observed signal $\hat{\boldsymbol{y}}$ in the local site. Then the encrypted signal $\hat{\boldsymbol{y}}$ is sent to a edge/cloud site. In the edge/cloud site, by using the encrypted signal $\hat{\boldsymbol{y}}$ and the stored dictionary $\hat{\boldsymbol{D}}$ sent in advance, a secure sparse coding computation is carried out.

3.2 Random Unitary Transform

Generally, a vector $\boldsymbol{f}_i$ $(i = 1, \cdots, L) \in \mathbb{R}^N$ is encrypted by a unitary matrix $\boldsymbol{Q}_p \in \mathbb{C}^{N\times N}$ with a key p as

$$\hat{\boldsymbol{f}}_i = T(\boldsymbol{f}_i, p) = \boldsymbol{Q}_p\boldsymbol{f}_i, \tag{8}$$

where $\hat{\boldsymbol{f}}_i$ is an encrypted vector, L is the number of vectors. Note that the unitary matrix $\boldsymbol{Q}_p$ satisfies is encrypted by a unitary matrix $\boldsymbol{Q}_p \in \mathbb{C}^{N\times N}$ with a parameter p as

$$\boldsymbol{Q}_p^*\boldsymbol{Q}_p = \boldsymbol{I} \tag{9}$$

where $[\cdot]^*$ and $\boldsymbol{I}$ mean the Hermitian transpose operation and the identity matrix respectively. In addition to the unitarity, $\boldsymbol{Q}_p$ needs to have randomness for generating the encrypted signal. The Gram-Schmidt orthogonalization is a typical method for generating $\boldsymbol{Q}_p$. Furthermore, the encrypted vector has following properties [6].
· Property 1: Conservation of the Euclidean distances.

$$\|\boldsymbol{f}_i - \boldsymbol{f}_j\|_2^2 = \|\hat{\boldsymbol{f}}_i - \hat{\boldsymbol{f}}_j\|_2^2 \tag{10}$$

· Property 2: Conservation of inner products.

$$\boldsymbol{f}_i^*\boldsymbol{f}_j = \hat{\boldsymbol{f}}_i^*\hat{\boldsymbol{f}}_j \tag{11}$$

· Property 3: Conservation of correlation coefficients.

$$\frac{\boldsymbol{f}_i\boldsymbol{f}_j}{\sqrt{\boldsymbol{f}_i\boldsymbol{f}_j}\sqrt{\boldsymbol{f}_i\boldsymbol{f}_j}} = \frac{\hat{\boldsymbol{f}}_i\hat{\boldsymbol{f}}_j}{\sqrt{\hat{\boldsymbol{f}}_i\hat{\boldsymbol{f}}_j}\sqrt{\hat{\boldsymbol{f}}_i\hat{\boldsymbol{f}}_j}} \tag{12}$$

3.3 Secure OMP Computation

The proposed secure sparse coding computation generates an encrypted signal $\hat{\boldsymbol{y}}$ and a dictionary $\hat{\boldsymbol{D}}$ by the following transform:

$$\hat{\boldsymbol{y}} = T(\boldsymbol{y}, p) = \boldsymbol{Q}_p \boldsymbol{y} \tag{13}$$

$$\hat{\boldsymbol{D}} = T(\boldsymbol{D}, p) = \boldsymbol{Q}_p \boldsymbol{D}. \tag{14}$$

Instead of Eq. (3), we consider the following optimization problem that $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{D}}$ are given:

$$\hat{\boldsymbol{x}} = \arg\min_{x} \| \hat{\boldsymbol{y}} - \hat{\boldsymbol{D}}\boldsymbol{x} \|_2^2 \quad \text{subject to} \quad \|\boldsymbol{x}\|_0 < \epsilon. \tag{15}$$

We prove that $\hat{\boldsymbol{x}}$ obtained by the secure OMP computation is the same result as the unencrypted version. Giving proof is not straightforward way because the OMP algorithm is an approximate technique. It depend on its algorithm whether the secure OMP computation provides same result as the unencrypted version.

Secure OMP Computation Algorithm

Initialization: $k = 0$, and set

· The initial solution $\boldsymbol{x}^0 = \boldsymbol{0}$
· The initial residual $\hat{\boldsymbol{r}}^0 = \hat{\boldsymbol{y}} - \hat{\boldsymbol{D}}\mathrm{x}^0 = \hat{\boldsymbol{y}} = \boldsymbol{Q}_p \boldsymbol{y}$
· The initial solution support $S^0 = \emptyset$

Main Iteration:
Increment k by 1 and perform the following steps:

· **Sweep:** Compute the errors
In Eq. (4), the dictionary $\boldsymbol{D}$ and the residual $\boldsymbol{r^{k-1}}$ are replaced with $\hat{\boldsymbol{D}}$ and $\hat{\boldsymbol{r}}^{\boldsymbol{k-1}}$. From Eqs. (13) and (14), the initial estimation error can be written as

$$\begin{aligned}\hat{\epsilon}(i) &= \min_{\hat{x}_i} \left\| \hat{x}_i \hat{\boldsymbol{D}}\boldsymbol{\delta}_i - \hat{\boldsymbol{r}}^{k-1} \right\|_2^2 \\ &= \left\| \hat{\boldsymbol{r}}^{k-1} \right\|_2^2 - \frac{(\hat{\boldsymbol{D}}\boldsymbol{\delta}_i \cdot \hat{\boldsymbol{r}}^{k-1})^2}{\left\| \hat{\boldsymbol{D}}\boldsymbol{\delta}_i \right\|^2} \\ &= \left\| \boldsymbol{Q}_p \boldsymbol{r}^{k-1} \right\|_2^2 - \frac{(\boldsymbol{Q}_p \boldsymbol{D}\boldsymbol{\delta}_i \cdot \boldsymbol{Q}_p \boldsymbol{r}^{k-1})^2}{\boldsymbol{\delta}_i^* \hat{\boldsymbol{D}}^* \hat{\boldsymbol{D}} \boldsymbol{\delta}_i}. \end{aligned} \tag{16}$$

From the properties of the unitary transform: $\left\| \boldsymbol{Q}_p \boldsymbol{r}^{k-1} \right\|_2^2 = \left\| \boldsymbol{r}^{k-1} \right\|_2^2$ (norm isometry), $\boldsymbol{Q}_p \boldsymbol{D}\boldsymbol{\delta}_i \cdot \boldsymbol{Q}_p \boldsymbol{r}^{k-1} = \boldsymbol{D}\boldsymbol{\delta}_i \cdot \boldsymbol{r}^{k-1}$ (conservation of inner products), $\hat{\boldsymbol{D}}^* \hat{\boldsymbol{D}} = \boldsymbol{D}^* \boldsymbol{D}$ (conservation of inner products), Eq. (16) can be rewritten as follows:

$$\hat{\epsilon}(i) = \left\| \boldsymbol{r}^{k-1} \right\|_2^2 - \frac{(\boldsymbol{D}\boldsymbol{\delta}_i \cdot \boldsymbol{r}^{k-1})^2}{\| \boldsymbol{D}\boldsymbol{\delta}_i \|_2^2}. \tag{17}$$

Equation (17) is equal to Eq. (4)', i.e. the relation $\hat{\epsilon}(i)$=$\epsilon(i)$ is satisfied.
· **Update Support:** Find the minimizer
From $\hat{\epsilon}(i)$=$\epsilon(i)$, the following relation is also satisfied.

$$\begin{aligned} i_0 &= \arg\min_{i \notin S^{k-1}} \{\hat{\epsilon}(i)\} \\ &= \arg\min_{i \notin S^{k-1}} \{\epsilon(i)\}, S^k = S^{k-1} \cup \{i_0\} \end{aligned} \tag{18}$$

· **Update Provisional Solution:**
The square error between the encrypted observed signal and the estimation by using the current support $\boldsymbol{x}_{S^k}$ is represented as $E_2 = \left\| \hat{\boldsymbol{y}} - \hat{\boldsymbol{D}}_{S^k} \boldsymbol{x}_{S^k} \right\|_2^2$. From $\frac{\partial E_2}{\partial x_{S^k}} = 0$, $\hat{\boldsymbol{x}}^k$ which provides the minimum square error is represented by

$$\hat{\boldsymbol{x}}^k = (\hat{\boldsymbol{D}}_{S^k}^* \hat{\boldsymbol{D}}_{S^k})^{-1} (\hat{\boldsymbol{D}}_{S^k}^* \hat{\boldsymbol{y}}). \tag{19}$$

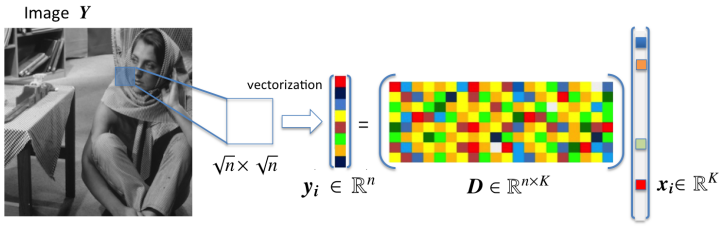


Figure 2: Sparse coding for mage patches.

In addition, from the property 2 in Eq.(11), $\hat{\boldsymbol{D}}_{S^k}^* \hat{\boldsymbol{D}}_{S^k}$ and $\hat{\boldsymbol{D}}_{S^k}^* \hat{\boldsymbol{y}}$ can be also given by $\hat{\boldsymbol{D}}_{S^k}^* \hat{\boldsymbol{D}}_{S^k} = \boldsymbol{D}_{S^k}^* \boldsymbol{D}_{S^k}$, $\hat{\boldsymbol{D}}_{S^k}^* \hat{\boldsymbol{y}} = \boldsymbol{D}_{S^k}^* \boldsymbol{y}$. Therefore, the provisional solution Eq. (19) can be rewritten as

$$\hat{\boldsymbol{x}}^k = (\boldsymbol{D}_{S^k}^* \boldsymbol{D}_{S^k})^{-1} (\boldsymbol{D}_{S^k}^* \boldsymbol{y}). \tag{20}$$

Equation (20) is equal to Eq. (6), i.e. the relation $\hat{\boldsymbol{x}}^k = \bar{\boldsymbol{x}}^k$ is satisfied.
· **Update Residual:**
Residual on the encrypted signals is expressed by $\hat{\boldsymbol{r}}^k = \hat{\boldsymbol{y}} - \hat{\boldsymbol{D}}_{S^k} \hat{\boldsymbol{x}}^k$. From Eqs. (13)-(14) and equality of the provisional residual $\hat{\boldsymbol{x}}^k = \bar{\boldsymbol{x}}^k$, the residual can be rewritten as follows:

$$\begin{aligned} \hat{\boldsymbol{r}}^k &= \boldsymbol{Q}_p \boldsymbol{y} - \boldsymbol{Q}_p \boldsymbol{D}_{S^k} \bar{\boldsymbol{x}}^k = \boldsymbol{Q}_p (\boldsymbol{y} - \boldsymbol{D}_{S^k} \bar{\boldsymbol{x}}^k) \\ &= \boldsymbol{Q}_p \boldsymbol{r}^k. \end{aligned} \tag{21}$$

· **Stopping Rule:**
If $\|\hat{\boldsymbol{r}}^k\|_2 < \epsilon$, stop. From Eq. (21) and the norm isometry of the unitary transform, it can be expressed as follows:

$$\left\| \hat{\boldsymbol{r}}^k \right\|_2 = \left\| \boldsymbol{Q}_p \boldsymbol{r} \right\|_2 = \left\| \boldsymbol{r}^k \right\|_2 < \epsilon \tag{22}$$

The stoping rule is equal to that of the unencrypted version. Unless it is satisfied, apply another iteration.

Output: The proposed solution $\hat{\boldsymbol{x}}^k$ is obtained after k iterations.

From the above analysis, it is shown that the secure computation gives the same result as the non-secure computation.

4. APPLICATION TO IMAGE MODELING

The sparse coding model has found numerous applications. In this section, we consider secure sparse coding for images based on a patch model.

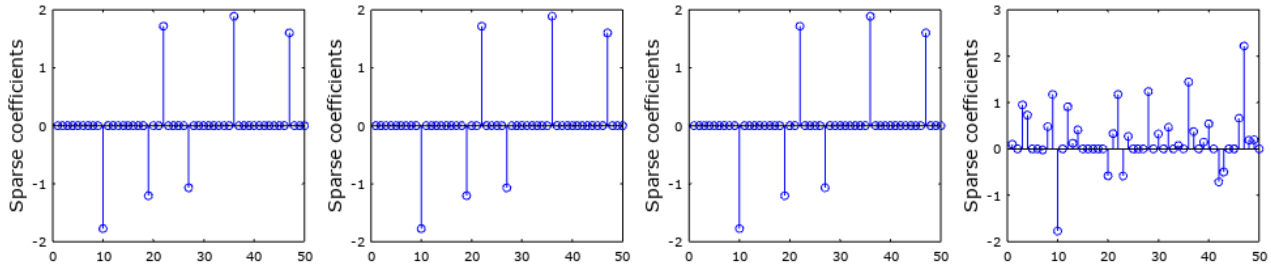
4.1 Sparse Coding for Image Patches

We consider image patches of size $\sqrt{n} \times \sqrt{n}$ pixels, ordered lexicographically as column vectors $\boldsymbol{y}_i \in \mathbb{R}^n$ $(i = 1, \cdots, N)$. The patches are extracted from an image $\boldsymbol{Y}$ as shown in Fig. 2. We assume that every image patch $\boldsymbol{y}_i$ could be represented sparsely over the overcomplete dictionary $\boldsymbol{D} \in \mathbb{R}^{n \times K}$.

$$\boldsymbol{y}_i = \boldsymbol{D}\boldsymbol{x}_i, \tag{23}$$

where $\boldsymbol{x}_i \in \mathbb{R}^K$ $(i = 1, \cdots, N)$ is sparse coefficients, N is the total number of patches. Applications for image compression, image denoising and image separation cited in Ref. [2] is based on the image patch model. In advance, the dictionary $\boldsymbol{D}$ is designed for images by training algorithms such as MOD [12], K-SVD [13] algorithms in the local site.

For example, sparse coding for image patches mentioned above can be applied to Encryption-then-Compression (EtC) systems [8]. In conventional secure image transmission systems, image encryption has to be conducted prior to image compression. On the other



(a) True value (b) OMP (c) Proposed method (d) OMP with non-unitary

Figure 3: Sparse coefficients x (a sample when the cardinality of the true solution is 6).

hand, EtC systems where image encryption can be conducted prior to compression, are expected for privacy protection. This scheme is able to compress images on the cloud while keeping security of image data.

4.2 Generation of Random Unitary Matrices

For each patch $\boldsymbol{y}_i$, secure OMP computation proposed in the previous section is applied. The proposed secure sparse coding computation generages an encrypted signal $\hat{\boldsymbol{y}}_i$ and a dictionary $\hat{\boldsymbol{D}}_i$ by the following transform:

$$\hat{\boldsymbol{y}}_i = T(\boldsymbol{y}_i, p_i) = \boldsymbol{Q}_{pi}\boldsymbol{y}_i \tag{24}$$

$$\hat{\boldsymbol{D}}_i = T(\boldsymbol{D}, p_i) = \boldsymbol{Q}_{pi}\boldsymbol{D} \tag{25}$$

where p_i and $\boldsymbol{Q}_{pi}$ are a key and a random unitary transform for the image patch $\boldsymbol{y}_i$, respectively. For each image patch $\boldsymbol{y}_i$, the sparse coefficient $\hat{\boldsymbol{x}}_i^k$ is estimated.

The image quality of each patch $\boldsymbol{y}_i$ can be controlled by using a sparsity ratio s_i or a threshold ϵ_i. The sparsity ratio s_i is a ratio of the number of nonzero sparse coefficients to the total number of elements of the dictionary $\hat{\boldsymbol{D}}_i$. The threshold ϵ_i determines the stopping condition of the secure OMP algorithm, i.e. $\left\|\boldsymbol{r}_i^k\right\|_2 < \epsilon_i$. In order to keep image quality of each image patch, the same threshold is set: ϵ_i = constant ($i = 1, \cdots, N$).

5. NUMERICAL DEMONSTRATION

In order to evaluate the effectiveness of the proposed secure computation, we demonstrated its performance both on synthetic data and in an application on natural image.

5.1 Synthetic Data

We create a random matrix $\boldsymbol{D}$ of size 30×50. Each column was normalized to a unit l_2-norm. We generate sparse vectors $\boldsymbol{x}$ with independent and identically-distributed (iid) random supports of candidates in the range [1,10], and non-zero entries drawn as random uniform variables in the range [-2,-1] $\cup$ [1,2]. Once $\boldsymbol{x}$ is generated, we compute $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}$. We perform 1000 such test per each cardinality, and present average results. We present two measures - l_2-error and recovery of the support. The l_2-error is computed as the ratio $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2/\|\boldsymbol{x}\|^2$. Recovery of the support indicates l_2 proximity between the two solutions. Denoting the two supports as $\hat{S}$ and S, we define this distance by

$$dist(\hat{S}, S) = \frac{max\{\hat{S}|, |S|\} - |\hat{S} \cap S|\}}{max\{\hat{S}|, |S|\}}. \tag{26}$$

We apply three algorithms shown below to seek for $\boldsymbol{x}$.

- Method 1: OMP
- Method 2: Secure OMP (Proposed method)
- Method 3: OMP with non-unitary random transform

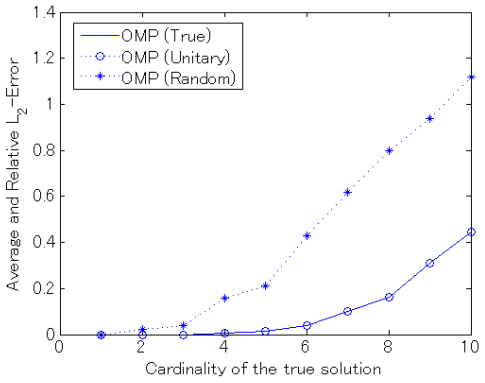


Figure 4: Relative l_2-norm error: $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2/\|\boldsymbol{x}\|^2$.

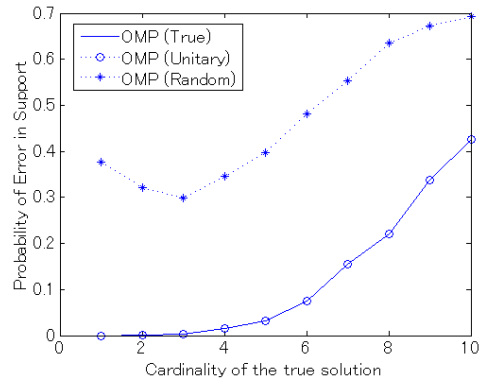


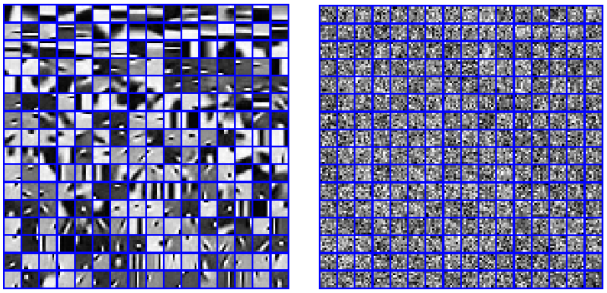
Figure 5: Recovery of the support: dist$(\hat{S}, S)$.

In the method 3, we generage the transformed signal $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{D}}$ by using a non-unitary random transform instead of the unitary random transform $\boldsymbol{Q}_p$. Then apply the OMP to the transformed signals. All these algorithms seek the solution till the residual is below a certain threshold ($\|\boldsymbol{r}^k\|_2^2 \leq 1e-4$).

The results are shown in Figs. 4 and 5. Figures 4 and 5 show that the proposed method gives exactly the same performance to the OMP in both the measures. On the other hand, the OMP with non-unitary random transform performs poorly especially in the area of large cardinality of the true solution. Figure 3 shows a sample of sparse coefficients $\boldsymbol{x}$ when the cardinality of the true solution is 6. It also supports the performance of the proposed method is same as that of the OMP. The unitarity property of the transform proves to be important.

5.2 Image Modeling

We carried out image modeling experiments on natural image



(a) Dictionary $\boldsymbol{D}$ (b) Encrypted dictionary $\hat{\boldsymbol{D}}$

Figure 6: A globally trained dictionary for general images.

(a) Original (b) Encrypted image

Figure 7: Original and encrypted images.

data, trying to show the practicality of the proposed algorithm. We used a pre-trained global dictionary that was trained by the K-SVD algorithm. The dictionary and a corresponding encrypted dictionary are shown in Fig. 6. Then we applied the secure OMP computation for a 256×256 Barbara image. Figure 7 shows an original image and a corresponding encrypted image.

Feeding the encrypted dictionary and the encrypted image into the secure OMP computation, we obtained the sparse coefficients $\hat{\boldsymbol{x}}_i$ for each image patch $\boldsymbol{y}_i$. The synthesis image $\dot{\boldsymbol{y}}_i$ is obtained by $\dot{\boldsymbol{y}}_i = \boldsymbol{D}\hat{\boldsymbol{x}}_i$. We carried out the experiments for different stopping conditions. Figure 8 shows the decompressed/decrypted images for two condition cases. Average sparsity ratio is an average of sparsity ratio s_i ($i = 1, \cdots, N$). We confirmed that it provides the same results as the unencrypted version of the OMP algorithm.

Note that our experiments here come only to prove the concept of image modeling using secure sparse coding for images based on patch images. Further study is required to deploy the proposed image modeling to image processing applications.

6. CONCLUSIONS

In this paper, we proposed the secure computation of sparse coding using the random unitary transform. We proved that the proposed secure OMP computation has exact the same performance of sparse coefficients estimation as the OMP algorithm theoretically. We confirmed the estimation performance of the proposed scheme through numerical demonstrations from the viewpoint of the relative l_2-norm error and the recovery of the support measures. Then, we apply it to image modeling based on the image patch model. Experiment results for natural images showed the practicality of the proposed algorithm.

7. REFERENCES

[1] Barlow, H.B. "Possible principles underlying the transformation of sensory messages," Sensory Communication pp. 217-234 (1961).

[2] Michael Elad, "Sparse and redundant representations: from theory to applications in signal and image processing," Springer, 2010.

[3] Michael Elad, “Sparse and redundant representation modeling - what next?,” IEEE Signal Processing Letters, vol. 19, no. 12, pp. 922-928, Dec. 2012.

[4] C. T. Huang, L. Huang, Z. Qin, H. Yuan, L. Zhou, V. Varad-harajan, and C-C. J. Kuo, “Survey on securing data storage in the cloud,” APSIPA Transactions on Signal and Information Processing, vol. 3, e7, 2014.

[5] R. L. Lagendijk, Z. Erkin and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," IEEE Signal Processing Magazine, vol. 30, no. 1, pp. 82-105, Jan. 2013.

[6] I. Nakamura, Y. Tonomura, and H. Kiya, ”Unitary transform-based template protection and its properties,” Proc. European Signal Processing Conference, vol. SIPA-P3.4, pp. 2466-2470, 2015.

[7] I. Nakamura, Y. Tonomura, and H. Kiya, “Unitary transform-based template protection and its application to l2-norm minimization problems,” IEICE Trans. Inf. & Sys., vol. E99-D, no.1, pp. 60-68, Jan. 2016.

[8] W.Sae-Tang,S.Liu,M.Fujiyoshi,andH.Kiya, “A copyright-and privacy-protected Image trading system using fingerprinting in discrete wavelet domain with JPEG2000,” IEICE Trans. Fundamentals, vol. E97-A, no. 11, Nov. 2014.

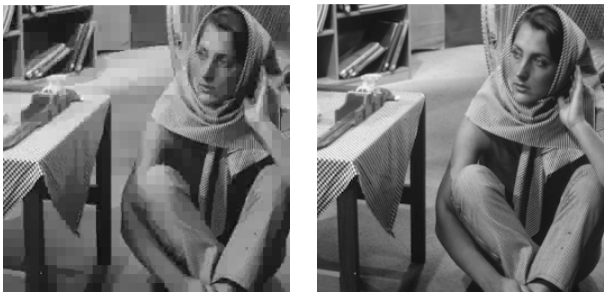
[9] Y. C. Pati, R. Rezaiifar, Y. C. P. R. Rezaiifar and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” Asilomar 1993, pp. 40-44 (1993).

[10] B. K. Natarajan: “Sparse approximate solutions to linear systems”, SIAM J. Comput., 24, 2, pp. 227-234 (1995).

[11] S. Mallat and Z. Zhang. "Matching pursuits with time-frequency dictionary," IEEE Trans. Signal Processing., vol. 41(12), pp. 3397-3415, 1993.

[12] K. Engan, S. O. Aase and J. Hakon Husoy: “Method of optimal directions for frame design,” ICASSP1999, pp. 2443-2446 (1999).

[13] M. Aharon, M. Elad and A. Bruckstein: “K-SVD: An algorithm for designing overcomplete dictionary for sparse representation,” IEEE Transactions on Signal Processing, vol. 54, no. 11, pp. 4311-4322, Nov. 2006.



(a) Average sparsity ratio = 0.049 (PSNR = 29.58 [dB]) (b) Average sparsity ratio = 0.375 (PSNR = 39.12 [dB])

Figure 8: Decompressed/decrypted images.