# OIRS: Outsourced Image Recovery Service from Compressive Sensing with Privacy Assurance [Extended Abstract]

Cong Wang
City University of Hong Kong
congwang@cityu.edu.hk

Zhen Xu, Bingsheng Zhang, Kui Ren
University at Buffalo
{zxu8,bzhang26,kuiren}@buffalo.edu

Janet Wang
University of Arizona at Tucson
wml@ece.arizona.edu

## 1 Introduction

There is a fast-growing trend to outsource the large-scale image management systems to cloud today, where abundant computing resources [1] can be leveraged to efficiently and effectively store, process, and share images among data owners and users [7]. However, for the image service outsourcing to be truly successful, there are still fundamental challenges yet to overcome. Firstly, because the cloud is a public environment operated by external third-parties usually outside the data owner/users trusted domain, the outsourcing design has to be privacy-protecting and sometimes mandatorily provide legal compliance to various privacy regulations [4]. Secondly, due to the high-dimensionality and large-scale of the image datasets, it is both necessary and desirable for the outsourcing design to be as efficient and less resource-consuming as possible in order to keep the cloud economically attractive.

To address these fundamental challenges, we investigate a novel outsourced image recovery service (OIRS) architecture in this paper, which exploits techniques from different domains and takes security, complexity, and efficiency into consideration from the very beginning of the service flow. Particularly, OIRS is designed under the compressive sensing framework [3], a recent data sensing/sampling paradigm known for its simplicity of unifying the traditional sampling and compression for image acquisition. As shown in Fig. 1, data owners only need to outsource compressed image samples to cloud for reduced storage overhead and simplified local sensing, while data users can leverage the cloud's computing resources to securely reconstruct images without revealing information from the image samples or the underlying image content. OIRS has the benefit of saving owner/users' workloads in the image recovery computation. In addition, it also incurs comparable amount of storage overhead and computational complexity at cloud as current mechanism does without security consideration. Below we introduce the main idea of OIRS design, covering the cases of sparse data, non-sparse data, and sampling with noises. Our preliminary analysis and experiments validate the security and efficiency of our design.

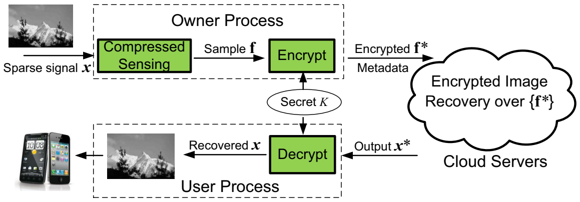


Figure 1: The architecture of OIRS

***Preliminaries***: Compressive sensing exploits the sparsity of natural data. Given a sparse data $\mathbf{b} \in \mathbb{R}^n$ and an orthonormal basis $\mathbf{V} \in \mathbb{R}^{n \times n}$, then the coefficient $\mathbf{x} \in \mathbb{R}^n$, satisfying $\mathbf{b} = \mathbf{V}\mathbf{x}$, has $s \ll n$ nonzero entries. Taking compressed samples [3] is done by multiplying an $m \times n$, $s < m \ll n$, selecting matrix $\mathbf{R}$ with full row rank to $\mathbf{b}$ to derive an $m \times 1$ sample vector $\mathbf{f} = \mathbf{R}\mathbf{b} = \mathbf{R}\mathbf{V}\mathbf{x} = \mathbf{A}\mathbf{x}$, where $\mathbf{A} = \mathbf{R}\mathbf{V}$. If $\mathbf{A} \in \mathbb{R}^{m \times n}$ satisfies *Restricted Isometry Property* (RIP) [3] with $m = 2s$ and $\delta_{2s} < \sqrt{2} - 1$, then when $\mathbf{x}$ is $s$-sparse, it can be recovered exactly from $\mathbf{f}$ by solving an $\ell_1$ minimization problem,

$$\min \quad \|\mathbf{x}\|_1, \quad \text{s.t.} \quad \mathbf{f} = \mathbf{A}\mathbf{x}. \tag{1}$$

Here $\delta_{2s}$ is the $2s$-th *restricted isometry constant* of $\mathbf{A}$ [2]. In practice, one can form $\mathbf{R}$ or $\mathbf{A}$ for RIP by sampling i.i.d. entries from normal distribution [3].

## 2 The Proposed OIRS Design

For security, OIRS needs to protect the image samples *before* outsourcing. The protected image samples should support image recovery as needed, while the recovered images at cloud should still be in an protected form. For these purposes, we study the secure transformation based approaches. Note that the $\ell_1$-min of Prob (1) is essentially a linear program (LP) [2]:

$$\min \quad \mathbf{1^T} \cdot \mathbf{r}, \quad \text{s.t.} \quad \mathbf{f} = \mathbf{A}\mathbf{x}, -\mathbf{r} \leq \mathbf{x} \leq \mathbf{r}. \tag{2}$$

Here $\mathbf{r}$ is an $n \times 1$ vector of variables. Let $\mathbf{x} + \mathbf{r} = \mathbf{2s}$ and $\mathbf{x} - \mathbf{r} = \mathbf{2t}$. We denote $\mathbf{x} = \mathbf{s} - \mathbf{t}$ and $\mathbf{r} = \mathbf{s} + \mathbf{t}$ in Prob. (2):

$$\min \ \mathbf{1^T} \cdot \mathbf{s} + \mathbf{1^T} \cdot \mathbf{t}, \ \text{s.t.} \ \mathbf{f} = \mathbf{A}(\mathbf{s} - \mathbf{t}), \mathbf{s} \geq \mathbf{0}, \mathbf{t} \geq \mathbf{0}.$$

Let $\mathbf{y} = [\mathbf{s^T}, \mathbf{t^T}]^\mathbf{T} \in \mathbb{R}^{2n}$, we rewrite the above LP as

$$\min \quad \mathbf{1^T} \cdot \mathbf{y}, \quad \text{s.t.} \quad \mathbf{f} = \mathbf{\Lambda}\mathbf{y}, \mathbf{y} \geq \mathbf{0}, \tag{3}$$

where $\mathbf{\Lambda}$ is the $m \times 2n$ matrix $[\mathbf{A}, -\mathbf{A}]$. Denote this problem as $\Phi = (\mathbf{\Lambda}, \mathbf{f}, \mathbf{I}, \mathbf{1^T})$. Hereafter, we use $\mathbf{x}$ and $\mathbf{y}$ as well as $\mathbf{A}$ and $\mathbf{\Lambda}$ interchangeably. With this formulation, we want formally a transformation algorithm $\mathsf{Trans}$ that takes as input the secret key $K$ and the original LP problem $\Phi$ and outputs the transformed problem $\Phi_k$. For efficiency, we are interested in a secure linear transformation $\mathsf{Trans}$ such that the transformed problem $\Phi_k$ is still an LP. Hence, $\Phi_k$ can be solved by cloud via a standard LP solver, and the OIRS design can be non-interactive. The security strength relies on the adversary's advantage of guessing $\Phi$ given $\Phi_k$. As OIRS works on real number and aims for an efficiency/security balance, we follow the security definition below.

**Definition** We say that a transform $\mathsf{Trans}$ is secure if

$$\forall \Phi_0, \Phi_1 : \left| \Pr\left[ \begin{array}{l} K \leftarrow \mathsf{Gen}(1^\kappa) : \\ \mathsf{Trans}(K, \Phi_0) = \Phi_k \end{array} \right] - \Pr\left[ \begin{array}{l} K \leftarrow \mathsf{Gen}(1^\kappa) : \\ \mathsf{Trans}(K, \Phi_1) = \Phi_k \end{array} \right] \right| \leq \mu(\kappa),$$

where $\mu(\cdot)$ is a negligible function.

## 2.1 The Case of Sparse Data

We now propose a preliminary transformation approach, which takes into account the public tuple $\mathbf{I}, \mathbf{1^T}$, in Prob. (3). Firstly, we obfuscate the inequality constraints using a generalized permutation matrix $\mathbf{D}$, which is the product of positive diagonal and permutation matrices.

$$\min \quad \mathbf{1^T} \cdot \mathbf{y}, \quad \text{s.t.} \quad \mathbf{f} = \mathbf{\Lambda y}, \mathbf{Dy} \geq \mathbf{0}.$$

For our choice of $\mathbf{D}$, $\mathbf{Dy} \geq \mathbf{0}$ is equivalent to $\mathbf{y} \geq \mathbf{0}$. Next, we use affine mapping $\mathbf{y} = \mathbf{Mz} - \mathbf{r}$ to protect $\mathbf{y}$, where $\mathbf{M}$ is $2n \times 2n$ invertible matrix and $\mathbf{r}$ is a $2n \times 1$ random vector:

$$\min \quad \mathbf{1^T} \cdot (\mathbf{Mz} - \mathbf{r}), \quad \text{s.t.} \quad \mathbf{\Lambda Mz} = \mathbf{f} + \mathbf{\Lambda r}, \mathbf{DMz} \geq \mathbf{Dr}.$$

Thirdly, we randomly mix the equality and inequality constraints together by first multiplying a random $2n \times m$ matrix $\boldsymbol{\lambda}$ to both sides of equality constraints and then adding them up to both sides of the inequality constraints:

$$\min \mathbf{1^T} \cdot (\mathbf{Mz} - \mathbf{r}),$$
$$\text{s.t.} \; \mathbf{\Lambda Mz} = \mathbf{f} + \mathbf{\Lambda r}, (\mathbf{DM} - \boldsymbol{\lambda}\mathbf{\Lambda M})\mathbf{z} \geq \mathbf{Dr} - \boldsymbol{\lambda}(\mathbf{f} + \mathbf{\Lambda r}).$$

Finally, we multiply a random $m \times m$ invertible matrix $\mathbf{Q}$ to equality constraints and ignore $\mathbf{1^T} \cdot \mathbf{r}$ in objective function:

$$\min \quad \mathbf{c'^T} \cdot \mathbf{z} \quad \text{s.t.} \quad \mathbf{\Lambda'} = \mathbf{f'}, \quad \mathbf{D'z} \geq \mathbf{r'},$$

where $\mathbf{c'} = \mathbf{1^T M}, \mathbf{\Lambda'} = \mathbf{Q \Lambda M}, \mathbf{f'} = \mathbf{Q} \cdot (\mathbf{f} + \mathbf{\Lambda r}), \mathbf{D'} = \mathbf{DM} - \boldsymbol{\lambda}\mathbf{\Lambda M}, \mathbf{r'} = \mathbf{Dr} - \boldsymbol{\lambda} \cdot (\mathbf{f} + \mathbf{\Lambda r})$. We can further let $\mathbf{c'^T} = \mathbf{1^T}$ and $\mathbf{r'} = \mathbf{0}$ to make the randomly transformed LP share the same public structure as $\Phi$ in Eq. (3):

$$\Phi_k = (\mathbf{\Lambda'}, \mathbf{f'}, \mathbf{D'}, \mathbf{r'} = \mathbf{0}, \mathbf{c'^T} = \mathbf{1^T}), \tag{4}$$

Keeping the same structure results in the secret transformation key as $K = (\mathbf{Q}, \mathbf{M}, \mathbf{r}, \mathbf{D}, \boldsymbol{\lambda})$, where the $2n \times 2n$ matrix $\mathbf{M}$ contains $(2n-1) \times 2n$ random elements, and the $2n \times m$ matrix $\boldsymbol{\lambda}$ contains $2n \times (m-1)$ random elements.

Based on the transformation, OIRS can be instantiated as follows: Data owner randomly generates $K$ and a random sampling matrix $\mathbf{R}$. After sampling $\mathbf{f} = \mathbf{RVx} = \mathbf{Ax}$, data owner picks $(\mathbf{Q}, \mathbf{r})$ from $K$ to encrypt $\mathbf{f}$ and outsources $\mathbf{f'}$ to cloud. Whenever data user sends an image recovery request, data owner calls $\mathsf{Trans}(K, \Phi)$ to output $\Phi_K$ to cloud. Cloud solves $\Phi_k$ and outputs $\mathbf{z}$ to data user. Data user then decrypts the original $\mathbf{y}$ via $\mathbf{y} = \mathbf{Mz} - \mathbf{r}$. Note that all the matrices and vectors to be used in the sampling and secret transformation can be generated by using a keyed pseudorandom function with random seeds. Thus, the sharing of $K$ can be easy by sharing small size seeds. We can use different $K$ and sampling matrices $\mathbf{R}$ for different images.

## 2.2 The Case of General Data

Many physical image sources are not exactly sparse. To broaden the application spectrum of OIRS, a natural question would be: how to extend the application to those non-sparse data? We answer the challenge question by exploring the idea of using sparse data representation to approximate the general data, aiming to achieve a tuneable balance between efficiency and accuracy. This is possible due to the result in [3]. Specifically, if the general data is nearly sparse, the securely recovered image will provide good approximations. Otherwise, OIRS will still recover the image, by reconstruction from the data's $s$ largest coefficients [3].

## 2.3 The Case of Compressive Sensing with Noise

To make OIRS more powerful and robust, we further investigate the case of compressive sensing corrupted with noise, like the errors in transmission channel, the noise brought by the imperfect measuring, etc. Specifically, given corrupted sample $\mathbf{f} = \mathbf{Ax} + \mathbf{e}$, where $\mathbf{e}$ is the unknown errors, how to securely leverage the cloud to recover $\mathbf{x}$? Following the work [2,5] in plaintext compressive sensing, we consider the error $\mathbf{e}$ is a sparse vector. In this case, oversampling, i.e., $m > n$, becomes a must to compensate the errors, and the key to recover $\mathbf{x}$ is to get $\mathbf{e}$ first. Inspired by [2, 5], we propose the following approach: When generating the $m \times n$ sampling matrix $\mathbf{A}$, the data owner also constructs an $n \times m$ matrix $\mathbf{G}$, satisfying $\mathbf{G} \cdot \mathbf{A} = \mathbf{0}$. Note that $\mathbf{G} \cdot \mathbf{f} = \mathbf{G} \cdot (\mathbf{Ax} + \mathbf{e}) = \mathbf{G} \cdot \mathbf{e}$. Now $\mathbf{e}$ is the unknown $m \times 1$ sparse vector, and $\mathbf{G} \cdot \mathbf{f}$ is the $n \times 1$ measurement vector with $n < m$. Thus, by $\ell_1$-min optimization: $\min \|\mathbf{e}\|_1$, s.t. $\mathbf{G} \cdot \mathbf{f} = \mathbf{Ge}$, one can directly solve $\mathbf{e}$ accurately. Thus, it is still possible to apply our previous random transformation to securely outsource the recovery of sparse error vector $\mathbf{e}$ to cloud. With $\mathbf{e}$, the data user can further solve $\mathbf{x}$ from the overdetermined equation $\mathbf{f} - \mathbf{e} = \mathbf{Ax}$.

## 2.4 Security Analysis and Empirical Evaluation

***Security Strength:*** Following our security definition, our preliminary analysis shows that given $\Phi_0, \Phi_1$, the transformed problems are computationally indistinguishable. We assume using finite precision floating numbers and

Table 1: Preliminary efficiency results

| Benchmark | Original | OIRS | | Speedup |
|---|---|---|---|---|
| image block size | $t_{original}$ | $t_{owner}$ | $t_{user}$ | $\frac{t_{original}}{t_{owner}+t_{user}}$ |
| $32 \times 32$ | 1.88 s | 0.44 s | 0.01 s | 4.2 × |
| $48 \times 48$ | 15.37 s | 4.36 s | 0.024 s | 3.5 × |

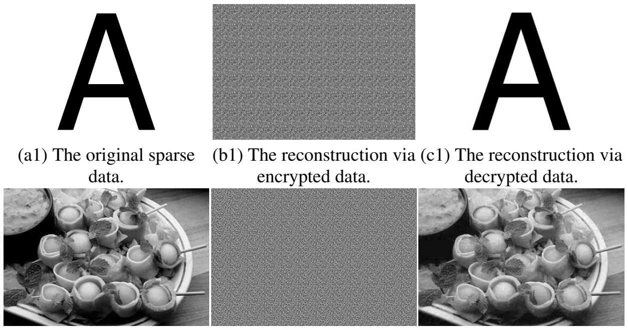
each entry $y_i$ of the original $\mathbf{y}$ should be in range $(-L, L)$, where $L = \text{poly}(\kappa)$ and $\kappa$ is the security parameter. Let the system input $n = \theta(\kappa)$. Denote uniform distribution from $[-2^\kappa, 2^\kappa]$ with fixed precision as $\mathcal{U}(-2^\kappa, 2^\kappa)$. First we show the transformed $\mathbf{z}$ does not reveal $\mathbf{y}$. Recall that $\mathbf{z} = \mathbf{M^{-1}}(\mathbf{y} + \mathbf{r})$. We can uniformly pick each entry $r_i$ of $\mathbf{r}$ from $[-2^\kappa, 2^\kappa]$ with fixed precision. If we pick a random vector $\boldsymbol{\eta}$ with its each entry sampled from $\mathcal{U}(-2^\kappa, 2^\kappa)$, then the distribution of $\mathbf{y} + \mathbf{r}$ is statistically close to $\boldsymbol{\eta}$. This is because for each $(y_i + r_i, \eta_i)$, $i \in \{1, \ldots, 2n\}$, the best distinguishing strategy is to output 0 if the input is from $[-2^\kappa - L, -2^\kappa)$ and $(2^\kappa, 2^\kappa + L]$, and a random guess $b \leftarrow \{0, 1\}$ otherwise. The distinguishing probability is

$$p = \frac{1}{2} + \Pr[y_i + r_i \in [-2^\kappa - L, -2^\kappa)] + \Pr[y_i + r_i \in (2^\kappa, 2^\kappa + L]] \leq \frac{1}{2} + \frac{2L}{2^\kappa} = \frac{1}{2} + \mu'(\kappa),$$

where $\mu'$ is a negligible function. By union bound, the distinguishing probability for $(\mathbf{y} + \mathbf{r}, \boldsymbol{\eta})$ is $\leq \mu(\kappa)$ where $\mu(\kappa) = 2n * \mu'(\kappa)$. Thus, the cloud's view of $\mathbf{z} = \mathbf{M^{-1}}(\mathbf{y} + \mathbf{r})$ and $\mathbf{z}^* = \mathbf{M^{-1}}\boldsymbol{\eta}$ is statistically indistinguishable. Then $\mathbf{z}$ does not reveal $\mathbf{y}$. Similarly, $\mathbf{f}' = \mathbf{Q}(\mathbf{f} + \boldsymbol{\Lambda}\mathbf{r}) = \mathbf{Q}\boldsymbol{\Lambda}(\mathbf{y} + \mathbf{r})$ statistically hides $\mathbf{f}$. Because each entry of $\mathbf{r}$ is sampled from $\mathcal{U}(-2^\kappa, 2^\kappa)$, by previous argument, we can replace $\mathbf{y} + \mathbf{r}$ with $\mathbf{r}$. The cloud's views of $\mathbf{f}'$ and $\mathbf{f}^* = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{r}$ are indistinguishable. Hence, for all $\mathbf{f_0}, \mathbf{f_1}$, the distinguishing probability for $\mathbf{f'_0}, \mathbf{f'_1}$ is at most $\mu(\kappa)$.

Recall that $\boldsymbol{\Lambda} = [\mathbf{RV}, -\mathbf{RV}]$, where $\mathbf{R}$ is randomly sampled for each problem. As the cloud's views of $\mathbf{f}'$ and $\mathbf{f}^*$ are indistinguishable, given $\mathbf{f_b}$, we only need to show the distribution of $\mathbf{f^*_b} = \mathbf{Q_b}\boldsymbol{\Lambda}_\mathbf{b}\mathbf{r_b}$, $\boldsymbol{\Lambda}'_\mathbf{b} = \mathbf{Q_b}\boldsymbol{\Lambda}_\mathbf{b}\mathbf{M_b}$ and $\mathbf{D'_b} = (\mathbf{D_b} - \boldsymbol{\lambda}_\mathbf{b}\boldsymbol{\Lambda}_\mathbf{b})\mathbf{M_b}$ are indistinguishable for $b \in \{0, 1\}$. Indeed, since all the components $\mathbf{Q_b}, \mathbf{r_b}, \boldsymbol{\Lambda}_\mathbf{b}, \mathbf{M_b}, \mathbf{D_b}, \boldsymbol{\lambda}_\mathbf{b}$ used to generate $\mathbf{f^*_b}, \boldsymbol{\Lambda}'_\mathbf{b}$ and $\mathbf{D'_b}$ are randomly sampled for each problem, the distribution of $\mathbf{f^*_b}, \boldsymbol{\Lambda}'_\mathbf{b}$ and $\mathbf{D'_b}$ is indistinguishable for different $\mathbf{f_b}$. Finally, we show that given $\mathbf{f^*_b}$, $\boldsymbol{\Lambda}'_\mathbf{b}$ and $\mathbf{D'_b}$, it is infeasible to solve the key components $\mathbf{Q_b}, \mathbf{r_b}, \boldsymbol{\Lambda}_\mathbf{b}, \mathbf{M_b}, \mathbf{D_b}, \boldsymbol{\lambda}_\mathbf{b}$. Indeed, it is well known that solving the system of nonlinear equation is NP hard. As the problem size $O(n) = \theta(\kappa)$, solving this underdetermined system of non-linear equation takes at least $\exp(\kappa)$ time. Hence, polynomial running time adversary has negligible chance to succeed [6].

***Empirical Evaluation:*** Some preliminary experiment results for the case of sparse and general data are reported in Fig. 2 and Table 1. For sparse data, Fig. 2-(a1) and (c1) denote the original and correctly recovered image. Fig. 2-(b1) containing random noise denotes the recovered image via random vector $\mathbf{z}$. It thus shows that the proposed techniques support secure yet correct image recovery without revealing the underlying image content. For general data, Fig. 2-(a2), (b2), and (c2) show the correctness and privacy-assurance. For ease of experiment, we decompose the original image into multiple image blocks with size 32×32 or $48 \times 48$. Each image block's sampling and recovering is done independently and is later re-assembled together. All experiments are done on the same work station for fair comparison. In Table 1, $t_{original}$, $t_{owner}$, $t_{user}$ denotes the original image recovery time, the transformation time by data owner, and the decryption time by data user, respectively. The speedup captures the efficiency gain via secure image recovery outsourcing. Table 1 shows more than 3× computation savings can be achieved for the experimental setting.



(a1) The original sparse data. (b1) The reconstruction via encrypted data. (c1) The reconstruction via decrypted data.

(a2) The original general data. (b2) The reconstruction via encrypted data. (c2) The reconstruction via decrypted data.

Figure 2: Preliminary effectiveness results.

## 3 Conclusion and Further Remarks

We have summarised OIRS's support on sparse data, general data, and sensing with noise, and presented preliminary security analysis and experiments. We will continue to work on OIRS for its compatibility with other important image services, such as content based image retrieval, while providing extensible service interfaces and possible performance speedup via hardware built-in design.

## References

[1] M. Armbrust et al. A view of cloud computing. *Comm. of the ACM*, 53(4):50–58, 2010.

[2] E. Candès and T. Tao. Decoding by linear programming. *IEEE Trans. Inform. Theory*, 51(12):4203–4215, 2005.

[3] E. Candès and M. Wakin. An introduction to compressive sampling. *IEEE Signal Proc. Mag.*, 25(2):21–30, 2008.

[4] Cloud Security Alliance. Security guidance for critical areas of focus in cloud computing, 2009.

[5] C. Dwork, F. McSherry, and K. Talwar. The price of privacy and the limits of LP decoding. In *Proc. of STOC*, pages 85–94, 2007.

[6] C. Jansson. An np-hardness result for nonlinear systems. *Reliable Computing*, 4(4):345–350, 1998.

[7] M. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. on Multimedia Comp., Comm., and Applications*, 2(1):1–19, 2006.

This work was supported in part by the US National Science Foundation (NSF) under grants CNS-1054317 and CNS-1116939