Privacy-Assured Outsourcing of Image Reconstruction Service in Cloud

CONG WANG¹ (Member, IEEE), BINGSHENG ZHANG² (Member, IEEE), KUI REN² (Senior Member, IEEE), AND JANET M. ROVEDA³ (Senior Member, IEEE)

¹Department of Computer Science, City University of Hong Kong, Hong Kong
²Department of Computer Science and Engineering, The State University of New York at Buffalo, Buffalo, NY 14214 USA
³Department of Electrical and Computer Engineering, University of Arizona at Tucson, Tucson, AZ 85721 USA

CORRESPONDING AUTHOR: C. WANG (congwang@cityu.edu.hk)

This work was supported in part by grants from City University of Hong Kong under Projects 6000439, 7003033, and 7200320, and by the U.S. National Science Foundation under Grants CNS-1262277, CNS-1262275, CNS-1116939, and IIA-0926278.

ABSTRACT Large-scale image data sets are being exponentially generated today. Along with such data explosion is the fast-growing trend to outsource the image management systems to the cloud for its abundant computing resources and benefits. How to protect the sensitive data while enabling outsourced image services, however, becomes a major concern. To address these challenges, we propose outsourced image recovery service (OIRS), a novel outsourced image recovery service architecture, which exploits different domain technologies and takes security, efficiency, and design complexity into consideration from the very beginning of the service flow. Specifically, we choose to design OIRS under the compressed sensing framework, which is known for its simplicity of unifying the traditional sampling and compression for image acquisition. Data owners only need to outsource compressed image samples to cloud for reduced storage overhead. In addition, in OIRS, data users can harness the cloud to securely reconstruct images without revealing information from either the compressed image samples or the underlying image content. We start with the OIRS design for sparse data, which is the typical application scenario for compressed sensing, and then show its natural extension to the general data for meaningful tradeoffs between efficiency and accuracy. We thoroughly analyze the privacy-protection of OIRS and conduct extensive experiments to demonstrate the system effectiveness and efficiency. For completeness, we also discuss the expected performance speedup of OIRS through hardware built-in system design.

INDEX TERMS Compressed sensing, security and privacy, cloud computing, image reconstruction.

I. INTRODUCTION

With the advancement of information and computing technology, large-scale datasets are being exponentially generated today. Examples under various application contexts include medical images [28], remote sensing images [2], satellite image databases, etc. Along with such data explosion is the fast-growing trend to outsource the image management systems to cloud and leverage its economic yet abundant computing resources [25] to efficiently and effectively acquire, store, and share images from data owners to a large number of data users [24].

Although outsourcing the image services is quite promising, in order to become truly successful, it still faces a number of fundamental and critical challenges, among which security is the top concern. This is due to the fact that the cloud is an open environment operated by external third-parties who are usually outside of the data owner/users’ trusted domain [12], [17]. On the other hand, many image datasets, e.g., the medical images with diagnostic results for different patients, are privacy-sensitive by its nature [1]. Thus, it is of critical importance to ensure that security must be embedded in the image service outsourcing design from the very beginning, so that we can better protect owners’ data privacy without sacrificing the usability and accessibility of the information. Besides, due to the high-dimensionality and large-scale of the image datasets [24], it is both necessary and desirable that the image service outsourcing design should be as efficient and less resource-consuming as possible, in terms of bandwidth and storage cost on cloud.
Traditionally, to establish such an image acquisition and sharing service, the data owner follows the Nyquist sampling theorem and often needs to acquire massive amounts of data samples, e.g., for high resolution images. Prior to transmission and image reconstruction, it is highly desirable to further pass these massive data through a compression stage for efficient usage of storage and bandwidth resources. Such a framework of large data acquisition followed by compression can be very wasteful, and often poses a lot of complexity on the data acquisition mechanism design at data owner side. For example, increasing the sampling rate can be very expensive in modern imaging systems like medical scanners and radars [30].

Compressed sensing [8], [10], [14] is a recently proposed data sampling and reconstruction framework that unifies the traditional sampling and compression process for data acquisition, by leveraging the sparsity of the data.\(^1\) With compressed sensing, data owners can easily capture compressed image samples via a simple non-adaptive linear measurement process from physical imaging devices, and later easily share them with users. In addition to simplified image acquisition and sharing, one can also apply compressed sensing, i.e., the process of taking non-adaptive linear measurements, over any existing large-scale image dataset, for the purpose of storage overhead reduction [13]. Specifically, as later shown in Section III-C, because the size of the sample vectors is almost always much less than the original image data, simply storing the compressed sample vectors rather than the actual image data can help save the storage cost as much as 50% [13]. Understanding these benefits of compressed sensing is pivotal, because it would allow us to explore new possibilities of establishing secure and privacy-assured image service outsourcing in cloud computing, which aims to take security, complexity, and efficiency into consideration from the very beginning of the service flow.

In this paper, we initiate the investigation for these challenges, and propose a new outsourced image recovery service (OIRS) architecture with privacy assurance. For the simplicity of data acquisition at data owner side, OIRS is specifically designed under the compressed sensing framework. The acquired image samples from data owners are later sent to cloud, which can be considered as a central data hub and is responsible for image sample storage and provides on-demand image reconstruction service for data users. Because reconstructing images from compressed samples requires solving an optimization problem [11], it can be burdensome for users with computationally weak devices, like tablets or large-screen smart phones. OIRS aims to shift such expensive computing workloads from data users to cloud for faster image reconstruction and less local resource consumption, yet without introducing undesired privacy leakages on the possibly sensitive image samples or the recovered image.

\(^1\)To be consistent with the majority work in compressed sensing, we treat images as real-valued signals or data with finite dimensions, which can be represented as a long one-dimensional vector.

content. To meet these challenging requirements, a core part of the OIRS design is a tailored lightweight problem transformation mechanism, which can help data owner/user to protect the sensitive data contained in the optimization problem for original image reconstruction. Cloud only sees a protected version of the compressed sample, solves a protected version of the original optimization problem, and outputs a protected version of the reconstructed image, which can later be sent to data user/owner for easy local postprocessing. Compared to directly reconstructing the image locally, OIRS is expected to bring considerable computational savings to the owner/users. As another salient feature, OIRS also has the benefit of not incurring much extra computational overhead on the cloud side. Our contributions can be summarized as follows.

- To our best knowledge, OIRS is the first image service outsourcing design in cloud that addresses the design challenges of security, complexity, and efficiency simultaneously.
- We show that OIRS not only supports the typical sparse data acquisition and reconstruction in standard compressed sensing context, but can be extended to non-sparse general data via approximation with broader application spectrum.
- We thoroughly analyze the security guarantee of OIRS and demonstrate the efficiency and effectiveness of OIRS via experiment with real world data sets. For completeness, we also discuss how to achieve possible performance speedup via hardware built-in system design.

The rest of this paper is organized as follows. Section II discusses the related work. Section III introduces the system architecture, threat model, system design goals, and some preliminaries. Then Section IV gives the detailed mechanism description, followed by security and efficiency analysis in Section V and further discussions on performance speedup in Section VI. Section VII gives the empirical results. Finally, Section VIII gives the concluding remarks.

II. RELATED WORK

Compressed sensing [8], [10], [14] is a recent data sensing and reconstruction framework well-known for its simplicity of unifying the traditional sampling and compression for data acquisition. Along that line of research, one recent work [13] by Divekar et al. proposed to leverage compressed sensing to compress the storage of correlated image datasets. The idea is to store the compressed image samples instead of the whole image, either in compressed or uncompressed format, on storage servers. Their results show that storing compressed samples offers about 50% storage reduction compared to storing the original image in uncompressed format or other data application scenarios where data compression may not be done. But their work does not consider security in mind, which is an indispensable design requirement in OIRS. In fact, compared to [13] that only focuses on storage reduction, our proposed OIRS aims to achieve a much more ambitious goal, which is an outsourced image service platform and takes into consideration of security, efficiency, effectiveness,
and complexity from the very beginning of the service flow. Another interesting line of research loosely related to the proposed OIRS is about the security and robustness of compressed sensing based encryption [27], [29]. Those works explore the inherent security strength of linear measurement provided by the process of compressed sensing. The authors have shown that if the sensing matrix is unknown to the adversary, then the attempt to exhaustive searching based original data recovery can be considered as computationally infeasible. However, these results are not applicable to OIRS as we intentionally want the cloud to do the image reconstruction job for us, with the challenge of not revealing either the compressed samples or the reconstructed image content.

This privacy-preserving image recovery service in OIRS that we propose to explore is also akin to the literature of secure computation outsourcing [3]–[6], [18], [20], [21], which aims to protect both input and output privacy of the outsourced computations. With the breakthrough on fully homomorphic encryption (FHE), a recent work by Gennaro et al. [18] shows that a theoretical solution has already been feasible. The idea is to represent any computation via a garbled combinational circuit [32] and then evaluate it using encrypted input based on FHE. However, such a theoretical approach is still far from being practical, especially when applied in the contexts of image sensing and reconstruction contexts. Both the extremely large circuit and the huge operation complexity of FHE make the general solution impossible to be handled in practice, at least in a foreseeable future. Researchers have also been working on specific designs for securely outsourcing specialized computation tasks, like scientific computations, sequence comparisons, matrix multiplications, modular exponentiations, etc. [3]–[6]. Again, the highly customised design, some of which even involve heavy cryptographic protocols, are also not applicable in OIRS. Another existing list of work that loosely relates to (but is also significantly different from) our work is secure multiparty computation (SMC). Firstly introduced by Yao [32] and later extended by Goldreich et al. [19] and others. SMC allows two or more parties to jointly compute some general function while hiding their inputs to each other. However, schemes in the context of SMC usually impose comparable computation burden on each involved parties, which is undesirable when applied to OIRS model. In short, practically efficient mechanisms with immediate practices for secure image recovery service outsourcing in cloud are still missing.

III. PROBLEM STATEMENT

A. SERVICE MODEL AND THREAT MODEL

The basic service model in the OIRS architecture includes the following: At first, data owner acquires raw image data, in the form of compressed image samples, from the physical world under different imaging application contexts. To reduce the local storage and maintenance overhead, data owner later outsources the raw image samples to the cloud for storage and processing. The cloud will on-demand reconstruct the images from those samples upon receiving the requests from the users. In our model, data users are assumed to possess mobile devices with only limited computational resources.

![FIGURE 1. The OIRS architecture in public cloud.](image)

Fig. 1 demonstrates the basic message flow in OIRS. Let f and y be the signal and its compressed samples to be captured by the data owner (to be elaborated in Section IV). For privacy protection, data owner in OIRS will not outsource y directly. Instead, he outsources an encrypted version y* of y and some associated metadata to cloud. Next, the cloud reconstructs an output f* directly over the encrypted y* and sends f* to data users. Finally, the user obtains f by decrypting f*. We leave the management and sharing of the secret keying material K between the data owner and users in our detailed decryption of OIRS design. In Fig. 1, each block module is considered as the process of a program taking input and producing output. We further assume that the programs are public and the data are private.

Throughout this paper, we consider a semi-trusted cloud as the adversary in OIRS. The cloud is assumed to honestly perform the image reconstruction service as specified, but be curious in learning owner/user’s data content. Because the images samples captured by data owners usually contain data specific/sensitive information, we have to make sure no data outside the data owner/user’s process is in unprotected format.

B. DESIGN GOALS

Our design goals for OIRS under the aforementioned service and threats model consist of the following.

- Security: OIRS should provide the strongest possible protection on both the private image samples and the content of the recovered images from the cloud during the service flow.
- Effectiveness: OIRS should enable cloud to effectively perform the image reconstruction service over the encrypted samples, which can later be correctly decrypted by user.
- Efficiency: OIRS should bring savings from the computation and/or storage aspects to data owner and users, while keeping the extra cost of processing encrypted image samples on cloud as small as possible.
- Extensibility: In addition to image reconstruction service, OIRS should be made possible to support other extensible service interfaces and even performance speedup via hardware built-in design.
C. PRELIMINARIES

To better present OIRS design, we briefly introduce some preliminaries about compressive sensing and notations to be utilised throughout the presentation.

Compressed Sensing: Compressive sensing is a recent data sensing framework known for its simplicity of unifying the traditional sampling and compression for signal acquisition. Consider an \( n \times 1 \) sparse data \( x \). The sampling process, i.e., taking compressed samples [11], is done by multiplying an \( m \times n \), \( m \ll n \), selecting matrix \( R \) with full row rank to \( x \) that derives an \( m \times 1 \) sample vector \( y = Rx \). Note that real world data \( x \) might not always be sparse. But as long as it can be represented as a sparse vector \( f \in \mathbb{R}^n \) under some properly chosen orthonormal basis \( V \in \mathbb{R}^{n \times n} \) via \( x = Vf \), we can still apply the above sensing process and have \( y = Rx = RVf = Af \), where we let \( A = RV \). With out loss of generality, we assume \( f \) is \( s \)-sparse, i.e., the number of non-zero entries in \( f \) is at most \( s \ll n \). It has been proved in [7], [10] that if \( A \in \mathbb{R}^{m \times n} \) satisfies Restricted Isometry Property (RIP) with \( m = 2s \) and \( \delta_{2s} < \sqrt{2} - 1 \), then \( f \) can be recovered exactly from \( y \) by solving an \( \ell_1 \) minimization problem

\[
\min \|f\|_1 \quad \text{subject to} \quad y = Af.
\]

According to [9], \( \delta_{2s} \) is the \( 2s \)-th restricted isometry constant of a matrix \( A \), which is defined as the smallest number for which

\[
\forall \|f\|_0 \leq 2s, (1 - \delta_{2s})\|f\|_2^2 \leq \|Af\|_2^2 \leq (1 + \delta_{2s})\|f\|_2^2.
\]

This condition implies that when \( \delta_{2s} \ll 1 \), all pairwise \( \ell_2 \) distances between \( s \)-sparse data must be well-preserved when multiplying with \( A \). We say matrix \( A \) satisfies RIP if it has \( \delta_{2s} \ll 1 \). In practice, we can i.i.d. sample each entry of either \( R \) or \( A \) from normal distribution to make it satisfying RIP [7].

Linear Programming: Linear programming plays an important role in the framework of compressed sensing. In fact, through standard formulation technique [9], to solve the \( \ell_1 \) minimization problem (1) for image reconstruction is equivalent to solve a linear programming (LP) problem as below

\[
\min \quad \|r\|_1 \quad \text{subject to} \quad y = Af, -r \leq f \leq r
\]

where \( r \) is an \( n \times 1 \) vector with positive real variables. To make the problem a more standard form, we can replace \( f \) and \( r \) by letting \( f = u - v \) and \( r = u + v \) respectively in the problem (1), where \( u, v \) are both \( n \times 1 \) vectors with each entry as a real positive variable

\[
\min \quad \|u\|_1 + \|v\|_1 \quad \text{subject to} \quad y = A(u - v), u \geq 0, v \geq 0.
\]

Denote \( g = [u^T, v^T]^T \in \mathbb{R}^{2n} \) and \( F = [A, -A] \in \mathbb{R}^{m \times 2n} \), the above LP problem can be represented as

\[
\min \quad \|u\|_1 + \|v\|_1 \quad \text{subject to} \quad y = F \cdot g, g \geq 0.
\]
with the same size as defined in Eq. (4), it would be difficult for the adversary to tell them apart after the random transformation. Formally, we define the security strength of $\Gamma$ as follows.

**Definition 1:** A transformation scheme $\Gamma = (\text{KeyGen}, \text{ProbTran}, \text{ProbSolv}, \text{DataRec})$ is secure if

$$\forall \Omega_0, \Omega_1 : \Pr\left[K \leftarrow \text{KeyGen}(\Gamma^0) : \text{ProbTran}(K, \Omega_0) = \Omega_k \right] - \Pr\left[K \leftarrow \text{KeyGen}(\Gamma^1) : \text{ProbTran}(K, \Omega_1) = \Omega_k \right] \leq \mu(\kappa)$$

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

Beyond capturing the indistinguishability based intuition, there are a few other design considerations that lead us to the above framework and security definitions. Firstly, for communication efficiency, we are interested in an non-interactive design between data owner/user and the cloud for secure outsourcing image reconstruction. Secondly, for computation efficiency, we want the problem solving algorithm ProbSolv on the cloud side to be as efficient as possible. Therefore, we are particularly interested in some secure transformation ProbTran algorithm which can transform $\Omega$ into $\Omega_k$ but still ensure the $\Omega_k$ is an LP problem. In this way, the problem solving algorithm ProbSolv can be a standard efficient LP solver. And the outsourced image reconstruction design can be naturally ensured as non-interactive. In addition, for practical consideration, we need OIRS to work well over real number and strike a good balance between efficiency and security. All these design requirements practically exclude the applicability of existing cryptographic techniques developed in the context of secure two-party computation and fully homomorphic encryption (See related work at Section II for detailed discussions.)

From the perspective of indistinguishability, such a security formulation is also loosely related to the general formulation of differential privacy [15], [16]. But we note that differential privacy is widely used for privacy-preserving data query in statistical database domain, a different context compared to the above framework and security definitions. Firstly, there are a few other design considerations that lead us to the above framework and security definitions. Secondly, for communication efficiency, we want the problem solving algorithm to be a standard efficient LP solver for efficiency consideration. Therefore, we propose to utilize a series of random linear transformation steps over the objective function, constraints, and feasible region of original problem $\Omega$, so as to preserve that the transformed problem $\Omega_k$ is still an LP problem. Specifically, our transformation procedure is described below.

1. We use a random generalized permutation matrix $\pi$ with positive entries, i.e., the product of a non-zero positive diagonal and a permutation matrix, to transform the inequality constraints

$$\min \ 1^T \cdot g \ \text{subject to} \ y = F \cdot g, \ \pi \cdot g \geq 0.$$ 

Note that $\pi \cdot g \geq 0$ is equivalent to $g \geq 0$.

2. We randomly pick a $2n \times 2n$ invertible matrix $Q$ and a $2n \times 1$ vector $e$ to protect the solution $g$ via affine mapping $g = Qh - e$

$$\min \ 1^T \cdot (Qh - e) \ \text{subject to} \ F \cdot Q \cdot h = y + F \cdot e, \ \pi \cdot Q \cdot h \geq \pi \cdot e.$$ 

3. We multiply a random $2n \times m$ matrix $M$ to equality constraints and later mix the result together with the inequality constraints

$$\min \ 1^T \cdot (Qh - e) \ \text{subject to} \ F \cdot Q \cdot h = y + F \cdot e, \ (\pi - MF)Qh \geq \pi e - M(y + Fe).$$

This problem is equivalent to the one in Step 2).

4. We multiply a random $m \times m$ matrix $P$ to the both sides of equality constraints

$$\min \ 1^T \cdot (Qh - e) \ \text{subject to} \ PFQ \cdot h = P \cdot (y + F \cdot e), \ (\pi - MF)Qh \geq \pi e - M(y + Fe).$$

The above transformation gives the blueprint, but we can further let the randomly transformed problem sharing the same structure as $\Omega$ in Eq. (4) when necessary. Specifically, we can first make sure $1^TQ$ is equal to $1^T$ in the objective function. This can be done by generating a $Q$ matrix, where for each column we first fix its $(2n - 1)$ random entries and then use $1$ minus the sum of the first $(2n - 1)$ random entries to get the $2n$-th entry. Secondly, we can also make the right hand side of inequality constraints, denoted as $r' = \pi e - M(y + Fe)$, always be zero just as $\Omega$ in Eq. (4). This can be achieved in the similar fashion as generating $Q$. From the construction of $r' = 0$, we observe that for the $i$-th entry of $r'$ to be $0$, $1 \leq i \leq 2n$, only the $i$-th row but no other entries of $Q$ needs to be involved. That is, each row of $Q$ can be generated independently. Thus, for the generation of $Q$, we first randomly pick $\pi$ and $e$. Then for the $i$-th row vector of $Q$, we randomly generate the first row of $Q$.

2For easy presentation, we assume $\pi$ has positive non-zero elements only.
(m − 1) entries, and later use 0 to minus the sum of those pre-generated m − 1 entries to get the m-th entry. By repeating the above process independently for other remaining rows, we can easily generate the random M satisfying r’ = 0, without actually solving a system of linear equations.

In summary, in the above random transformation, we use the secret key as K = (P, Q, e, π, M), where the 2n × 2n matrix Q contains (2n − 1) × 2n random entries, and the 2n × m matrix M contains 2n × (m − 1) random entries. If we ignore the constant term 1T · e in objective function, we have the random LP

\[
\min \ 1^T \cdot h \ \text{subject to} \ y' = F' \cdot h, \ \pi' h \geq 0,
\]

where \( F' = PFQ, \ y' = P \cdot (y + F \cdot e), \ \text{and} \ \pi' = (\pi \cdot Q - MFQ). \) We denote this problem as

\[
\Omega_k = (F', y', \pi', 1^T)
\]  

(5)

Based on above discussion, now we are ready to instantiate the algorithms in the framework \( \Gamma \).

**C. THE SCHEME DETAILS**

We first make two reasonable assumptions about the information shared between data owner and users: 1) a master secret \( sk \) which is used to generate the random sampling matrix \( R \) and the secret \( K \) for each image and 2) an orthonormal basis \( V \), with which the image data \( x \) can be represented as a sparse vector \( f \). Note that for security purpose, we use independent secret key \( K \) and sampling matrices \( R \) for each image. In the following for easy presentation, we omit the index information in the instantiation description.

KeyGen(\( \Gamma, \sigma \)) \( \rightarrow K = (P, Q, e, \pi, M), \) where \( \sigma \) denote random coins. For easy sharing of secret keying materials between users and owners, we will be using a master-keyed pseudo-random function (PRF) with random seeds to generate all the coins to derive random matrices and vectors in \( K \). For each image to be sampled, we use a freshly generated random key \( K \). The instantiation is shown in Algorithm 1.

We defer the discussion on security parameter \( k \) in Section V.

ProbTran(\( K, \Omega \)) \( \rightarrow \Omega_k \). To better present our transformation in a flexible way, we propose to separate the transformation described in Section IV-B into two steps. Namely, we can define ProbTran = (ProbTran1, ProbTran2), where ProbTran1 takes as input the secret key \( K \) and \( y, F \) in original LP \( \Omega \) and outputs a tuple \( y' \) in \( \Omega_k \), while ProbTran2 takes as input \( K \) and \( F \) and outputs tuples \( (F', \pi') \) in \( \Omega_k \). The instantiation is shown in Algorithm 2 and 3.

ProbSolv(\( \Omega_k \)) \( \rightarrow h \). Because our transformation based design outputs \( \Omega_k \) as a standard LP problem, this algorithm on cloud side can be a general LP solver and thus its description is omitted.

DataRec(\( K, h \)) \( \rightarrow g \). The user uses the secret key \( K \) to recover the original answer \( g \) for problem \( \Omega \) from protected answer \( h \) of \( \Omega_k \) returned by cloud. The instantiation is shown in Algorithm 4.

**Scheme details:** Based on the above instantiation, we describe the complete protocol for the OIRS framework \( \Gamma \) below, which includes both the image sampling phase and the data recovery phase. Let \( F = [0, 1]^k \times [0, 1]^{e1} \rightarrow [0, 1]^{e2} \) denote a keyed pseudo-random function for random coins generation.

1) **DATA SAMPLING PHASE**

1) Data owner picks a fresh seed \( s \leftarrow [0, 1]^{e1} \) and computes \( \sigma \leftarrow F(sk, s) \). He then uses \( \sigma \) as coins to sample a random sensing matrix \( R \) and generates a secret key \( K = (P, Q, e, \pi, M) \) from KeyGen(\( \Gamma, \sigma \)).

2) He acquires the sample \( y = Rx = RVf = Af \) (see Section III-C). With \( F = [RV, −RV] \) and \( y \), he calls ProbTran1(\( K, (y, F) \)) to encrypt \( y \) as \( y' \) by using \( (P, e) \) from \( K \), and sends \( (y', s) \) to cloud.

We assume the samples and the related seeds \( \{y', s\} \) are all stored in an authenticated manner at cloud. Assume that data

<table>
<thead>
<tr>
<th>Algorithm 1</th>
<th>Key Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data:</strong> security parameter ( \Gamma ), random coins ( \sigma )</td>
<td></td>
</tr>
<tr>
<td><strong>Result:</strong> ( K = (P, Q, e, \pi, M) )</td>
<td></td>
</tr>
<tr>
<td><strong>% discussion on choice of ( k ) deferred in Section V:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>begin</strong></td>
<td></td>
</tr>
<tr>
<td>1) uses ( \sigma ) to generate random ( P, e, \pi )</td>
<td></td>
</tr>
<tr>
<td>2) uses ( \sigma ) to generate random ( Q ) and ( M )</td>
<td></td>
</tr>
<tr>
<td>3) % satisfying the structure of ( \Omega_k ) in Prob 5</td>
<td></td>
</tr>
<tr>
<td><strong>return</strong> secret key ( K = (P, Q, e, \pi, M) )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm 2</th>
<th>Problem Transformation Step 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data:</strong> transformation key ( K ) and original LP ( \Omega )</td>
<td></td>
</tr>
<tr>
<td><strong>Result:</strong> protected sample ( y' ) in ( \Omega_k )</td>
<td></td>
</tr>
<tr>
<td><strong>begin</strong></td>
<td></td>
</tr>
<tr>
<td>1) picks ( P, e ) from ( K ) and ( F ) from ( \Omega )</td>
<td></td>
</tr>
<tr>
<td>2) return ( y' = P \cdot (y + F \cdot e) )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm 3</th>
<th>Problem Transformation Step 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data:</strong> transformation key ( K ) and original LP ( \Omega )</td>
<td></td>
</tr>
<tr>
<td><strong>Result:</strong> protected coefficient matrices ( F', \pi' ) in ( \Omega_k )</td>
<td></td>
</tr>
<tr>
<td><strong>begin</strong></td>
<td></td>
</tr>
<tr>
<td>1) picks ( P, Q, \pi, M ) in ( K ) and ( F ) in ( \Omega )</td>
<td></td>
</tr>
<tr>
<td>2) computes ( F' = PFQ ) and ( \pi' = (\pi - MFQ) )</td>
<td></td>
</tr>
<tr>
<td>3) return transformed ( F', \pi' )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm 4</th>
<th>Original Answer Recovery</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data:</strong> transformation key ( K ) and protected answer ( h ) of ( \Omega_k )</td>
<td></td>
</tr>
<tr>
<td><strong>Result:</strong> answer ( g ) of original problem ( \Omega )</td>
<td></td>
</tr>
<tr>
<td><strong>begin</strong></td>
<td></td>
</tr>
<tr>
<td>1) picks ( Q, e ) from ( K )</td>
<td></td>
</tr>
<tr>
<td>2) return ( g = Qh - e )</td>
<td></td>
</tr>
</tbody>
</table>
user issues an image recovery request for an image sample \( y' \) to data owner, then:

**2) IMAGE RECOVERY PHASE**

1) Data owner downloads the seed \( s \) from cloud, computes \( \sigma \leftarrow \mathcal{F}(sk, s) \), and uses \( \sigma \) to regenerate the matrix \( R \) and the key \( K = (P, Q, e, \pi, M) \) from KeyGen\((I^k, \sigma)\). He calls ProbTran\(_2(K, F)\) to get \((F', \pi')\) and sends them to cloud.

2) With \( \Omega_k = (F', \pi', y', \mathbb{I}^T) \), the cloud calls ProbSolv\((\Omega_k)\) to output answer \( h \) to user, together with seed \( s \).

3) The user computes \( \sigma \leftarrow \mathcal{F}(sk, s) \), and uses \( \sigma \) to generate the key \( K \) from KeyGen\((I^k)\). He then calls DataRec\((K, h)\) to get \( g = Qh - e \) and recovers the image \( x = Vf \), where \( f \) is derived from \( g \).

The separation of problem transformation into two steps ProbTran = (ProbTran\(_1, \text{ProbTran}_2\)) makes our protocol implementation more flexible. Such flexibility allows us to allocate different computation burdens to either data owners or data users, depending on the application contexts. Note that in OIRS architecture, we assume the data user is only with lightweight computing power, such as large-screen mobile devices. Thus, to reduce the computation burden on data users to the minimum, we intentionally require data owner in the above scheme description to call ProbTran\(_2\) for the problem transformation in step (1) of image recovery phase. That is, the data owner conducts all the problem transformation. But in practice the call of ProbTran\(_2\) can be done by data user as well, which increases the extra overhead to data user but can completely eliminate the involvement of data owner in the image recovery phase. Such a choice is desirable when data owner is with weak computing devices, such as wireless sensors, while data user is with more powerful local workstations.

**Remark:** For efficiency, the protected samples in OIRS share the same size of original ones without outsourcing due to the random transformation design. This is desirable as it introduces no extra storage cost on cloud during outsourcing. Because cloud helps solve the optimisation problem for image reconstruction task, a considerable amount of computation savings can be provided to data users with weak computational devices. For security, OIRS uses a fresh \( K \) for each image to be captured/recovered. This can be made possible by our choices of using random seeds/coins to generate \( K \), which can easily shared between owner and users. We will thoroughly analyse the efficiency and security strength in Section V.

Also note that the choice of orthonormal basis \( V \) in OIRS can be either data independent, like wavelet basis, or data dependent, like Karhunen-Loeve basis [13]. But no matter which case it is, we can always easily prescribe it for data owner and users. In other words, the assumption of pre-sharing the orthonormal basis \( V \) between owner and users can be considered a reasonable one.

**D. THE EXTENSION TO NON-SPARSE DATA**

So far, we have been assuming that OIRS operates over sparse data only. That is, \( f \) is exactly sparse. However, there are many cases where physical data sources are not exactly sparse. To further broaden the application spectrum of OIRS design, we now show how to extend from the case of sparse data to the non-sparse general data. For easy presentation purposes in this section, we will be using \( f \) and \( A \) to denote the original image data and the original sensing matrix respectively. In fact, by comparing prob. 2 and prob. 4, it is not hard to see that \( f \) and \( A \) is equivalent to \( g \) and \( F \) we have been using in the previous Sections.

Here we propose to leverage the idea of approximation. That is under certain conditions, we can always use sparse data to well approximate the large coefficients in the non-sparse vector data, as long as the small coefficients in those non-sparse data do not contain too much information [11]. Specifically, assuming under orthonormal basis \( V \), the image data \( x \)'s coefficient vector \( f \) is non-sparse. We denote \( f_s \) as an \( s \)-sparse approximation of \( f \), which can be derived by setting all but the largest \( s \) entries of \( f \) to zero. Let \( x_s = Vf_s \). Because \( V \) is orthonormal, then

\[
\|x - x_s\|_2 = \|Vf - Vf_s\|_2 = \|f - f_s\|_2.
\]

This equation implies that the difference between \( f \) and its \( s \)-sparse approximation \( f_s \) is exactly equal to the difference between the original image \( x \) and the approximated image \( x_s \). On the other hand, the current advancement in compressed sensing [7] has shown that for any non-sparse general data \( f \), the related solution to the Prob. 1, denoted as \( f^* \), will always be a sparse data. And its difference compared to the actual \( s \)-sparse approximation \( f_s \) satisfies the following bound,

\[
\|f^* - f_s\|_2 \leq \frac{C_0}{\sqrt{s}} \cdot \|f - f_s\|_1,
\]

where \( C_0 \) is some constant.

The above elaboration suggests that the aforementioned OIRS design can be still applied to the non-sparse general data. The quality of recovered data from compressed sensing shall be well approximated by the reconstruction from its \( s \) largest coefficients, i.e., \( f_s \). All the design steps shall follow. Therefore, exploiting the \( s \)-sparse approximation in case of non-sparse general data provides us a flexible way to adjust between efficiency and accuracy. If \( f \) is nearly sparse, OIRS will provide good approximation. If \( f \) is not sparse, then OIRS will recover the image at its best, by reconstruction from its \( s \)-sparse approximation \( f_s \). The quality of the recovered image might not be as good as the previous case, but the simplicity of compressed sensing remains.

**V. THEORETICAL ANALYSIS**

**A. SECURITY ANALYSIS**

Following the security definition in Section IV-A, we now show that given \( \Omega_0, \Omega_1 \), the transformed problems are indistinguishable. Here we assume OIRS works over finite precision floating numbers, and each entry \( g_i \) of the original
solution $g$ is in range $(-L, L)$, where $L = \text{poly}(\kappa)$ and $\kappa$ is our security parameter. Let the system input $n = \theta(\kappa)$.

First, we replace the random coins $\sigma$ generated by $F(8k, s)$ with real random coins. This modification is indistinguishable to a probabilistic polynomial time adversary $A$ if the PRF $F$ is secure.

Next, we show that $h$ does not reveal $g$ via $g = Qh - e$, as long as each entry $e_i$ of $e$ is randomly picked from a relatively big interval $[-2^k, 2^k]$ with fixed precision. We first prove the following theorem.

**Theorem 5.1:** Given a random vector $\chi$, where each entry in $\chi$ is sampled from the uniform distribution over $[-2^k, 2^k]$ with fixed precision, denoted as $U(-2^k, 2^k)$, then the statistical distance

$$\text{SD}(g + e, \chi) \leq \mu(\kappa)$$

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

**Proof:** First, we show for each individual entry, $\text{SD}(g_i + e_i, \chi_i), i \in \{1, \ldots, 2n\}$, is negligible. We denote two hypothesises, $H_0$ with ranges $[-2^k - L, 2^k + L]$ and $H_1$ with range $[-2^k, 2^k]$, respectively. It is obvious to see that the best distinguishing strategy is to output 0 if the input is from $[-2^k - L, -2^k)$ and $(2^k, 2^k + L)$, and output a random guess $b \leftarrow \{0, 1\}$, otherwise. This success probability of such a distinguisher is

$$p = \frac{1}{2} + \Pr[g_i + e_i \in [-2^k - L, -2^k)] + \Pr[g_i + e_i \in (2^k, 2^k + L)]$$

$$\leq \frac{1}{2} + \frac{2L}{2^k} = \frac{1}{2} + \mu(\kappa)$$

where $\mu'$ is a negligible function. Then, by applying union bound, we have $\text{SD}(g + e, \chi) \leq \mu(\kappa)$ where $\mu(\kappa) = 2n \ast \mu'(\kappa)$ as claimed. $\blacksquare$

Given the theorem, the cloud’s view of $g + e$ and $\chi$ is statistically indistinguishable. Equivalently, the cloud’s view of $h = Q^{-1}(g + e)$ and $h^* = Q^{-1}\chi$ is statistically indistinguishable. Thus we can switch $h$ with $h^*$, which implies that $h$ statistically hides $g$. Note that the above argument doesn’t require any stringent condition on $\chi$. But for correctness of our transformation, the random $Q$ needs to be invertible and satisfy $Q^TQ = I^T$ for the consistency of problem structure as in $\Omega$.

Next we show that $y' = P(y + Fe)$ statistically hides $y$. Since $y = Fg$, we have $y' = PF(g + e)$. Again, according to Theorem 5.1, we can replace $g + e$ with $e^*$, as each entry of $e$ is sampled from $U(-2^k, 2^k)$. The cloud’s views of $y' = PF(g + e)$ and $y^* = PFe$ are statistically indistinguishable. Hence, for all $y_0, y_1$, the statistical distance between $y_0', y_1'$ are at most $\mu(\kappa)$.

Now we are about to show our OIRS design satisfies our security definition. As claimed above, the cloud’s views are indistinguishable if we replace $y'$ with $y^*$. Thus, given $y_0$, we only need to show the distribution of $y^*_b = P_bF_be_b$. $F_b = P_bF_bQ_b$ and $\pi_b' = (\pi_b - M_bF_b)Q_b$ are indistinguishable for $b \in \{0, 1\}$. Recall that $F = [RV, -RV]$, where $R$ is randomly sampled for each problem and $V$ is an orthonormal basis. It is easy to see that all the components $P_b, e_b, F_b, Q_b, \pi_b, M_b$ that are used to generate $y^*_b, F'_b, \pi'_b$ are randomly sampled for each problem. Hence, the distribution of $y^*_b, F'_b, \pi'_b$ is indistinguishable for different $y_b$.

Finally, we show that given $y^*_b, F'_b$ and $\pi'_b$, it is infeasible to solve the key components $P_b, e_b, F_b, Q_b, \pi_b, M_b$. Indeed, it is well known that solving the system of non-linear equation is NP hard, and the problem size is $O(n^3 \alpha) = \theta(\kappa)$. Thus solving this underdetermined system of non-linear equation takes at least $\exp(\kappa)$ running time. Hence, polynomial running time adversary has negligible chance to succeed [22], [31]. Later in empirical evaluations, we show that the chosen image blocks are usually with size of at least $32 \times 32$. This means $n$ is at least 1024, which further corroborates above analysis on the security strength of OIRS.

### B. Efficiency Analysis

The most time-consuming operations in the proposed transformation is the matrix-matrix operations, which cost asymptotically $O(n^3)$ for some $2 \leq \alpha \leq 3$ due to $m \leq 2n$. On the other hand, solving the LP problem $\Omega_k$ usually requires more than $O(n^3)$ time, e.g. [23]. Clearly, outsourcing image recovery service to cloud provides data owners/users considerable computational savings in theory. Moreover, with our proposed transformation, the cloud process can utilize any existing solvers for the LP problem $\Omega_k$, which ensures the cloud side efficiency.

As for the storage overhead, we note the fact that the size of compressed samples $y$ and the randomly transformed one $y'$ that is outsourced to cloud in OIRS are with the same size. Therefore, OIRS incurs almost the same storage cost as the current practice, e.g. this work in [13], based on compressed sensing without security considerations. This study in [13] has shown that using compressive sensing can reduce storage overhead up to 50%, compared to storing the original data or images in uncompressed format.

### VI. Further Discussions

Enabling secure image outsourcing services will significantly boost the wide application spectrum of secure computing outsourcing. For example, the proposed OIRS can be adopted by image service applications like MRI in health care system, remote sensing in geographical system, and even military image sensing in various mission critical contexts. In the following, we give some further discussions on how the proposed OIRS can serve as a stepping stone and discuss the possible performance speedup through hardware built-in design.

### A. Speedup with Hardware Built-in Design

In order to make these promising image services in OIRS truly efficient and practically deployable, it is pivotal to further explore how to embed the security and efficiency guarantee from the start through a hardware built-in system design. Compared to software based approaches, an effective
hardware design can significantly boost the performance of functionalities that are to be implemented in the proposed service architecture. Among others, we are particularly interested in the hardware based acceleration for the image recovery performance on the cloud. For that purpose, we propose to explore a recently developed iterative recovery algorithm, CoSaMP [26], in the compressed sensing literature. Our key observations are: 1) the CoSaMP algorithm only involves matrix/vector multiplications in each iteration of its greedy pursuit, which is much cheaper to implement from the hardware perspective compared to existing general optimization based solutions and 2) because each iterative approximation only interacts with one matrix through its actions on multiple vectors, it further allows parallelization, which thus makes faster image recovery, i.e., the speedup, possible. Note that in such a hardware built-in design, the aforementioned security guarantee still holds, since we can always treat it as an image recovery black box and apply the design rationale of random transformation. For example, by giving the hardware design the transformed image samples \( P(y + Ae) \) and the sensing matrix \( PAQ \) satisfying \( (PAQ) \cdot (Q^{-1}(f + e)) = P(f + Ae) \) as in Eq. (1), it would still give us a randomly transformed output \( Q^{-1}(f + e) \) as the encrypted result. While further investigation are needed for this proofs of concept idea, we believe a hardware built-in design offers great benefits in achieving the secure OIRS with best possible service performance and user experience. This task is one of our important future works.

VII. EMPIRICAL EVALUATIONS

A. EXPERIMENT SETTINGS

We now show the experiment results of the proposed OIRS. We implement both the data owner/user and the cloud side processes in MATLAB and use the MOSEK optimization toolbox (http://www.mosek.com/) as the LP solver. All experiments are done on the same workstation with an Intel Core i5 CPU running at 2.90 GHz and 6 GB RAM. We ignore the communication latency between the data owners, users, and the cloud for this preliminary study, and only focus on the computational evaluations. Also note that the sensing process on data owner is implemented via software simulation, where existing image data is considered as signal source to be compressed sensed. All test images are with size \( 384 \times 256 \). To avoid expensive large matrices handling overhead, we decompose each image into smaller size image blocks, with the size of \( 32 \times 32 \) and \( 48 \times 48 \), and process the sampling and recovering of each image block independently. Correspondingly, each \( 32 \times 32 \) will be treated as \( 1024 \times 1 \) column vector, while \( 48 \times 48 \) as \( 2304 \times 1 \) column vector. Once all small image blocks are reconstructed, we then put them together to reconstruct the original “big” image. We generate the compressed sensing matrix \( R \) by sampling i.i.d. entries from the Gaussian distribution \( N(0, 1) \) and derive \( A = R \cdot V \). As for the orthonormal basis \( V \), we follow the similar settings in [13]. Specifically, we adopt the data-dependent Karhunen-Loeve (KL) basis in our experiment, derived by diagonalising the covariance matrix of a sample of image blocks [13]. We have conducted experiments over images from different datasets, but for space interest we show below only a few representative ones.

B. EFFICIENCY EVALUATION

We first measure the efficiency of the proposed OIRS. Specifically we focus on the computational cost of privacy-assurance done by the data owner and data users, i.e., the local side, and the cost done by the cloud side. Table 1 gives our experimental results, where each entry in the table represents the mean of 10 trials. Each trial focuses on one randomly selected image block recovery, and Fig. 2 shows some of the tested images.

The first two columns report the size of image blocks in the test. Recall in OIRS, data owner computes the randomly transformed optimization problem (in two steps) to the cloud. The cloud solves it for the data user, who then performs a decryption process to get the original image data vector and then recover the image. Thus, the third and the fourth columns in Table 1 reports the local computational cost by the data owner and data users, specifically including the time cost for overall problem transformation done by the data owner \( t_{owner} \), and the time cost of image data decryption done by the data user \( t_{user} \). The fifth column reports original image reconstruction cost without outsourcing, \( t_{original} \), which measures the time of solving the original LP problem locally. In order to better report the practical efficiency, we evaluate how much computational savings OIRS can provide to data owner/users via outsourcing. This can be measured via the total time of image reconstruction without outsourcing (i.e., original local cost) divided by the total local time cost of problem transformation and image decryption in OIRS (i.e., current local cost), denoted as \( \text{asymmetric speedup} = \frac{t_{asymmetric}}{t_{owner} + t_{user}} \). From the table, we can see that OIRS can bring more than \( 3.4 \times \) savings for the selected size of image blocks. Because reconstructing large image may require computation over many small image blocks, it is suggested that OIRS can shift a considerable amount of computational overhead from local to cloud. Note that OIRS does not require any specific solver algorithm, and

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Secure Image Recovery in OIRS</th>
<th>Original Recovery</th>
<th>Asymmetric Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td># image block size</td>
<td>( t_{owner} ) (sec)</td>
<td>( t_{user} ) (sec)</td>
<td>( t_{original} ) (sec)</td>
</tr>
<tr>
<td>1</td>
<td>32 × 32</td>
<td>0.44</td>
<td>0.01</td>
</tr>
<tr>
<td>2</td>
<td>48 × 48</td>
<td>4.36</td>
<td>0.24</td>
</tr>
</tbody>
</table>
the cloud side process can utilise any existing LP solver for image reconstruction.

For asymmetric speedup measurement, we do not include the final cost on the data user, who performs one matrix-vector multiplication using the orthonormal basis and the decrypted sparse vector to recover the actual image block. This is because of the consideration that such a step anyway needs to be performed by data user in both the original recovery without outsourcing and the proposed OIRS. For completeness, we report the time cost here. For $32 \times 32$ image block it is 0.009 sec on average, while for $48 \times 48$ image block size it is 0.021 sec on average.

C. EFFECTIVENESS EVALUATION

We next assess the effectiveness of OIRS design. Our goal is to show the correctness of the design and also the empirical results on the privacy assurance.

1) CORRECTNESS EVALUATION

For correctness of the design, we show that all the images, after transformation and later recovered on the data user side, still preserves the same level of visual quality as the original images. Here we want to point out that the reconstructed image quality increases along with the number of measurements, and the more the better. In our experiments, we follow the “four-to-one” rule according to [11]. It is suggested that if the number of captured compressed samples is roughly $4 \times$ the sparsity level of the targeted signals, then the samples would be sufficient for successful image reconstruction. Because for most of the $32 \times 32$ sparse image blocks tested in the experiment, their sparsity level under the selected KL-basis is around a few dozens at most, we use $m = 256$ linear measurements to ensure good enough quality of the reconstructed image blocks.

As we mentioned previously, besides sparse data, OIRS also naturally supports the case of general data. Therefore, our experiment covers both two cases. Fig. 2-(a) denote the sparse data, while Fig. 2-(a1)–(a3) denote the non-sparse general data. The recovered image after the transformation and user decryption is shown in Fig. 2-(c1)–(c3). For either sparse or general data case, the image block recovery is based on the number of measurements $m = 256$. Because of the relatively sufficient size of $m$, we can hardly see the visual difference when comparing the reconstructed images and the original images. For completeness, we also tested the recovered images using different number of measurements.
Fig. 3 gives a particular image reconstruction example by setting $m$ as 128, 192, and 256 respectively. As we can see, the quality downgrade is quite obvious when $m$ decreases. Note that all recovered images have been through three steps of random sample mapping at data owner, reconstruction over randomly transformed problem at cloud, and decryption at the user.

2) PRIVACY-ASSURANCE EVALUATION

Recall that OIRS provides the privacy-assurance that users can harness the cloud to securely recover the image without revealing the underlying image content. This can be achieved because what cloud really recovers, $h$, protects the original sparse vector $h$ via a general affine mapping $g = Qh - e$ with a random choices of $Q$ and $e$. To give the empirical results on privacy-assurance, we show in Fig. 2-(b)–(b3) the recovered image before user decryption, i.e., recovering using the blinded vector $h = Q^{-1}(g + e)$. Again, Fig. 2-(b) corresponds to the sparse data while Fig. 2-(b1)–(b3) relate to the case of non-sparse data.

In both cases, the random affine mapping enabled by $Q$ and $e$ over $g$ provides good enough privacy-assurance on image content protection. This demonstrates what adversary can see given the basis $V$ and the recovered encrypted vector $h$ only consists of obfuscated image blocks. Compared to random noises, these image blocks are perceptually indistinguishable. By comparing those protected images in Fig. 2-(b)–(b3) with either the original images or the reconstructed images, it is safe to say that OIRS provides satisfactory privacy-assurance. That is, without knowing the secret key, the actual content of the protected underlying image cannot be perceived.

VIII. CONCLUSION

In this paper, we have proposed OIRS, an outsourced image recovery service from compressed sensing with privacy assurance. OIRS exploits techniques from different domains, and aims to take security, design complexity, and efficiency into consideration from the very beginning of the service flow. With OIRS, data owners can utilize the benefit of compressed sensing to consolidate the sampling and image compression via only linear measurements. Data users, on the other hand, can leverage cloud’s abundant resources to outsource the image recovery related $\ell_1$ optimization computation, without revealing either the received compressed samples, or the content of the recovered underlying image. Besides its simplicity and efficiency, we show OIRS is able to achieve robustness and effectiveness in handling image reconstruction in cases of sparse data as well as non-sparse general data via proper approximation. Both extensive security analysis and empirical experiments have been provided to demonstrate the privacy-assurance, efficiency, and the effectiveness of OIRS. On top of the current architecture, we also demonstrate a proof-of-concept of possible performance speedup through hardware built-in system design, which we believe is our important future work to be pursued.

REFERENCES

BINGSHENG ZHANG is a Post-Doctoral Researcher with the Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY, USA. He received the B.Eng. degree in computer science from the Zhejiang University of Technology, Zhejiang, China, in 2007, the M.Sc. degree in information security from University College London, London, U.K., in 2008, and the Ph.D. degree in computer science from the University of Tartu, in 2011. He was a part-time Research Associate with University College London and a full-time Researcher with Cybernetica AS.

KUI REN is an Associate Professor with the Computer Science and Engineering Department, State University of New York at Buffalo, Buffalo, NY, USA. He was an Associate Professor with the Electrical and Computer Engineering Department, Illinois Institute of Technology, Chicago, IL, USA. He received the Ph.D. degree in electrical and computer engineering from Worcester Polytechnic Institute, in 2007. His current research interests include security and privacy in cloud computing, wireless security, smart grid security, and sensor network security. His research is supported by NSF, DoE, AFRL, and Amazon. He is a co-recipient of the Best Paper Award from the IEEE ECNP in 2011. He is a recipient of NSF Faculty Early Career Development (CAREER) Award in 2011 and Sigma Xi/IIT Research Excellence Award in 2012. He is a member of ACM.

JANET M. ROVEDA received the B.S. degree in computer science from The East China Institute, in 1991, the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, CA, USA, in 1998 and 2000, respectively. She was a recipient of the National Science Foundation Career Award and the PEACASE Award in 2005 and 2006, respectively. She received the Best Paper Award at ISQED in 2010 as well as best paper nominations at ASPDAC in 2010, ICCAD in 2007, and ISQED in 2005. She is a recipient of the 2008 R. Newton Graduate Research Project Award from DAC and the 2007 UoS University of Arizona Outstanding Achievement Award. Her current research interests include robust VLSI circuit design, biomedical instrument design, smart grid, VLSI circuit modeling and analysis, and low power multi-core system design.

CONG WANG is an Assistant Professor with the Computer Science Department, City University of Hong Kong, Hong Kong. He received the B.E and M.E degrees from Wuhan University, Wuhan, China, in 2004 and 2007, respectively, and the Ph.D. degree from the Illinois Institute of Technology, Chicago, IL, USA, in 2012, all in electrical and computer engineering. He was a Research Intern at Palo Alto Research Center, Palo Alto, CA, USA, in the summer of 2011. His current research interests include cloud computing and security, with current focus on secure data services in cloud computing, and secure computation outsourcing. He is a member of ACM.