A Privacy-aware Cloud-assisted Healthcare Monitoring System via Compressive Sensing

Cong Wang*, Bingsheng Zhang†, Kui Ren†, Janet M. Roveda§, Chang Wen Chen†, and Zhen Xu†
*Department of CS, City University of Hong Kong
†Department of CSE, State University of New York at Buffalo
§Department of ECE, University of Arizona at Tucson
Email: *congwang@cityu.edu.hk, †{bzhang26, kuiren, chencw, zxu8}@buffalo.edu, §wml@ece.arizona.edu

***Abstract*—Wireless sensors are being increasingly used to monitor/collect information in healthcare medical systems. For resource-efficient data acquisition, one major trend today is to utilize compressive sensing, for it unifies traditional data sampling and compression. Despite the increasing popularity, how to effectively process the ever-growing healthcare data and simultaneously protect data privacy, while maintaining low overhead at sensors, remains challenging. To address the problem, we propose a privacy-aware cloud-assisted healthcare monitoring system via compressive sensing, which integrates different domain techniques with following benefits. By design, acquired sensitive data samples never leave sensors in unprotected form. Protected samples are later sent to cloud, for storage, processing, and disseminating reconstructed data to receivers. The system is privacy-assured where cloud sees neither the original samples nor underlying data. It handles well sparse and general data, and data tampered with noise. Theoretical and empirical evaluations demonstrate the system achieves privacy-assurance, efficiency, effectiveness, and resource-savings simultaneously.**

I. Introduction

Recent years have witnessed a rapid growth of wireless sensors for monitoring and retrieving information from healthcare medical systems in a versatile manner [1], [2], [3]. Because of sensors' moderate cost and their ability to continuously collect data, health care providers believe that these sensors "carry the promise of drastically improving and expanding the quality of care across a wide variety of settings and segments of the population" [1]. Indeed, a number of clinics and hospitals have already demonstrated the potential of patient monitoring through early prototypes including various forms of systems with imaging sensors [4].

As sensor nodes are known to be resource-constrained, great amount of research efforts have been invested on how to reduce the signal acquisition complexity on these sensing systems and how to enhance the energy efficiency of data communication [1], [2], [3], [5]. For that purpose, a major trend today is to utilize compressive sensing [6], [7], [8], a recent ground-breaking data sampling and reconstruction framework that unifies the traditional sampling and compression process for data acquisition. By leveraging the sparsity of the data, compressive sensing enables sub-Nyquist sampling and low-energy data reduction, making the technique especially attractive in healthcare sensingmonitoring systems.

However, the growing popularity of these systems has also revealed several grand challenges yet to be addressed satisfactorily: how to effectively process the ever-growing healthcare data and simultaneously protect patient's data privacy, while maintaining the low overhead at the sensors? In order to create high quality images or simulate motions and activities, healthcare systems usually require continuous and routine monitoring to capture a deluge of information. For example, a transceiver on a body sensor network monitoring physiological signal of an individual can easily capture and send nearly 2.77 GB of raw data per day. When the sample rate is high, such number can even reach up to 31 GB per day [2]. To effectively manage and process the huge amount of information, a natural choice is to offload the data to cloud for its economic yet abundant computing resources. But security inevitably becomes a major concern. This is not only because the public cloud is an open environment operated by external third-parties [9] but also because many healthcare data, e.g., the medical images with diagnostic results for different patients, are privacy-sensitive by its nature [10]. Thus, it is of critical importance to ensure that security be embedded in the healthcare monitoring system design from the very beginning, so that we can better protect patients' data without sacrificing the usability and accessibility of the information. For a truly practical security design, it is essential to still maintain the low cost on data sensing, as current healthcare monitoring system does with the choice of compressive sensing.

To address these challenges, in this paper, we take the first research attempt and propose a novel cloud-assisted healthcare monitoring system architecture with privacy assurance. Our system leverages techniques from different domains and achieves the following benefits. For efficient data acquisition at sensor, we opt to design our system under the compressive sensing framework. Although it has been exploited recently in a number of biomedical sensors [2], [3], [5], [11], none of these works has ever considered the increasing importance of securing the healthcare data, especially in a cloud-assisted monitoring system. For privacy-protection, in our proposed system, acquired sensitive samples never leave sensor in unprotected form. Such protected samples are later sent from sensors directly to cloud, which functions as a central hub responsible for storage, processing, and disseminating reconstructed data to receivers. The whole process is privacy-assured such that information from neither the samples nor the underlying data content will be revealed.

In our system, one key design consideration among others is to minimise the cost of sensors, in particular the communication cost, when generating and transmitting the protected samples to cloud. This stringent requirement practically ex-

978-1-4799-3360-0/14/$31.00 ©2014 IEEE

cludes the applicability of existing techniques in the context of fully homomorphic encryption [12] due to the hugely enlarged ciphertext and operational cost. Simply adopting ordinary symmetric encryption techniques over samples is also not a viable choice, as it in essence prevents the cloud from performing any meaningful operation of the protected samples, let alone data reconstruction. In our design, we leverage the fact that the compressive sensing data recovery can be achieved by solving a formulated linear programming (LP) problem, and investigate a secure transformation mechanism that randomly maps the original data recovery problem into a random one so as to hide the information of the underlying data. Such a random mapping design has the benefit of maintaining the same bandwidth cost at the sensor as current mechanism does without security consideration. In addition, it brings in considerable computational savings at the receiver side without introducing extra computational complexity at the cloud. We start with the architecture design for the case of sparse data, which is the typical scenario for compressive sensing. Then we show its natural extension to the general data, which allows meaningful tradeoffs between efficiency and accuracy. In order to be truly powerful and robust, we also study how to make the design applicable to real world scenarios where data samples are tampered with noise. Our contributions are as follows:

1. We formulate the first privacy-assured healthcare monitoring system architecture with simplified data acquisition, secure cloud-assisted data reconstruction, and local resource savings.

2. The system supports sparse data, general data, and data tampered with noise, under various application contexts. All cases supports aforementioned design features satisfactorily.

3. Thorough security analysis and experiment results show that the proposed designs can indeed achieve security, efficiency, effectiveness, and resource savings simultaneously.

The rest of the paper is organized as follows. Section II introduces the system and threat model, our design goals, and the preliminaries. Then we provide the detailed mechanism description in Section III, and security and efficiency analysis in Section IV. Section V gives the empirical evaluation, followed by Section VI, which overviews the related work. Finally, Section VII gives the concluding remarks.

II. Problem Statement

A. System Architecture and Threat Model

The service model envisioned by our proposed cloud-assisted healthcare monitoring design can be illustrated in Figure 1: Wireless sensors are used to continuously monitor and collect raw data under various healthcare contexts. To effectively manage and process the huge amount of sensed data, the sensor in our system will directly offload the acquired data samples to the cloud. Those samples are all in protected form for privacy-protection. Cloud with its abundant resources is responsible to provide various privacy-assured data services for receivers, such as on-demand data recovery, data retrieval, and others. Here the receiver might be a healthcare workstation operated by a physician in a hospital. In the following, we will use medical image reconstruction from compressed samples as a concrete application to demonstrate our system design.

Under such an architecture, cloud is responsible for image recovery from the received samples. The sensor first acquires

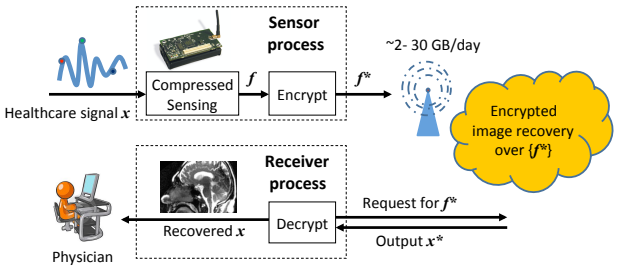

Fig. 1: The proposed system architecture

compressed samples $f \in \mathbb{R}^m$ from some sparse/compressible signal $x \in \mathbb{R}^n$ where $m < n$. Instead of directly sending original samples f, the sensor first encrypts f with a secret key K into $f^* \in \mathbb{R}^m$, and sends f^* to the cloud. For healthcare monitoring, the above process is assumed to be in continuous and routine fashion, and the samples will be accumulated at the cloud. Upon receiving a recovery request with associated metadata from the receiver, the cloud recovers an output $x^* \in \mathbb{R}^n$ over some specified f^* and hands it back to receiver. Finally, the receiver decrypts x^* into x using the secret K. Because f^* and f are with the same size, the security protection does not bring in any extra bandwidth cost at the sensor. Due to the assistance of cloud, the local computation at the receiver is expected to be considerably reduced, and the overall image recovery efficiency improved.

We assume an "honest-but-curious" cloud, which faithfully follows the designated protocol but may voluntarily infer data content for various purposes. Thus, any data leaving the data sensor/receiver have to be encrypted.

B. Design Goals

Our design goals consist of the following: 1) The healthcare monitoring system should be effective and reliable such that it correctly supports the cloud-assisted data recovery, while protecting both the private data samples and the content of the recovered data from the cloud; 2) The system should maintain low local cost, especially the communication cost for acquired sample transmission, on the resource-constrained sensor. It should also keep the local computation at receiver substantially less than the service designs without cloud-assistance; 3) The system should be designed to be compatible with other important data services, like content based data/image retrieval, while providing possible extensible service interfaces.

C. Preliminaries

Sparse Data: Many signals acquired from the physical world are sparse or compressible in the sense that when expressed in the proper orthonormal basis (such as a standard or wavelet basis, etc.), their coefficient entries decay rapidly. Let the signal be $b \in \mathbb{R}^n$ and the basis be $V \in \mathbb{R}^{n \times n}$. The coefficient entries $x \in \mathbb{R}^n$ satisfies that $b = Vx$. The sparsity of x is measured by its number of non-zero entries, denoted as $\|x\|_0$. For sparse data, $\|x\|_0$ is far less than n. We say x is s-sparse when $\|x\|_0 \leq s$. Not all data are strictly sparse. But if s largest coefficients in x contain almost all the information, we can still treat x as sparse, and call it compressible [13].

Compressive Sensing: Compressive sensing combines the traditional sampling and compressing process into a single

non-adaptive linear measurement process by exploiting the data sparsity. Consider an $n \times 1$ compressible data b. The sampling process, i.e., taking compressed samples [13], is done by multiplying an $m \times n$, $m << n$, selecting matrix $R \in \mathbb{R}^{m \times n}$ with full row rank to b that derives an $m \times 1$ sample vector $f = Rb$. Recall that $b = Vx$ and let $A = RV \in \mathbb{R}^{m \times n}$. One can obtain $f = Ax$. Compressive sensing also attempts to recover b, or equivalently x, from f. It is shown in [7], [14] that if A satisfies *Restricted Isometry Property* (RIP), then when x is s-sparse, it can be recovered from f by solving an ℓ_1 minimization (ℓ_1-min) problem,

$$\min \quad \|x\|_1 \quad \text{subject to} \quad f = Ax. \tag{1}$$

In practice, one way to form R or A for RIP is by sampling i.i.d. entries from standard normal distribution [14].

III. The Proposed Design

To instantiate a privacy-aware cloud-assisted healthcare system, we have to ensure acquired samples never leave the sensor in unprotected form. For energy-efficiency, we have to maintain the low cost, especially the bandwidth cost for the sample acquisition and transmission, at the sensor. The protected samples should support processing for healthcare image recovery as needed. Moreover, the images recovered at cloud should still be in a protected form.

A. The Main Idea

To successfully meet all design considerations, we propose to investigate the secure transformation based approaches. Note that the ℓ_1-min problem (1) used in image recovery is essentially a linear programming (LP) problem [15] as below:

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

Here $r \in \mathbb{R}^n$ is an $n \times 1$ vector with real positive variables. As standard, the generalized vector inequality $x \leq r$ means that all scalar entries of x is less than those of r respectively. If we denote $x + r = 2s$ and $x - r = 2t$ for $s, t \in \mathbb{R}^n$, we have $x = s - t$ and $r = s + t$ in the problem (1), which derives

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

We can further rewrite the above LP problem in a more standard form by letting $y = [s^T, t^T]^T \in \mathbb{R}^{2n}$,

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

where Λ is the $m \times 2n$ matrix $[A, -A]$. We denote this problem as $\Phi = (\Lambda, f, I, 1^T)$.

With the above formulation, the problem we need to answer now is how to protect the compressed samples f and the recovered y from the cloud while allowing cloud to efficiently solve the LP problem[1]. Below we propose a random transformation based approach to achieve our design. Such an approach has the benefits of protecting the sensitive samples while maintaining the same bandwidth overhead at the sensor as current mechanisms do without security consideration. Formally, our design consists of 4 probabilistic polynomial time algorithms, $\Gamma = (\mathsf{Gen}, \mathsf{Trans}, \mathsf{Solver}, \mathsf{Recover})$:

- Gen is a key generation algorithm that takes as input the security parameter 1^κ and outputs secret key K.
- Trans is a query transformation algorithm that takes as input the secret key K and the original LP problem Φ and outputs the transformed problem Φ_k.
- Solver is a solving algorithm that takes as input the transformed problem Φ_k and outputs answer z.
- $\mathsf{Recover}$ is the recover algorithm that takes as input the secret key K and the answer z and outputs y as the corresponding answer of original LP problem Φ.

To let the problem solving algorithm Solver as efficient as possible, we are interested in a secure linear transformation Trans such that the transformed problem Φ_k is still an LP problem. Hence, the problem solving algorithm Solver can be a standard LP solver. The security strength of such scheme Γ depends on the adversary's advantage of guessing Φ given Φ_k. Before giving our security definition, we first list a few design considerations that need to be met for the Γ to function. In order to solve the LP problem non-interactively, we emphasize that some information about $\mathcal{Y}$, the feasible region of Φ, has to be revealed. Specifically, there are three necessary leakage predicates $\{g_i\}_{i=1}^3$ required by the cloud to evaluate the transformed problem correctly. For each leakage predicate g_i over $\mathcal{Y}$, we have the corresponding leakage predicate Leak_i over $\mathcal{Z}$, where $\mathcal{Z}$ is the feasible region of the transformed problem. The predicates are based on problem Φ defined in Eq. (3), where f and Λ of problem Φ are supposed to be protected against the cloud while I and 1^T are public information.

To decide whether objective $1^T \cdot y$ achieves minimum, we need the first leakage predicate to be an order function:

$$g_1(y_1, y_2) = \begin{cases} 0 & 1^T \cdot y_1 \leq 1^T \cdot y_2 \\ 1 & 1^T \cdot y_1 > 1^T \cdot y_2 \end{cases} .$$

We define Leak_1 such that

$$\forall z_1, z_2 \in \mathcal{Z}: \quad \mathsf{Leak}_1(z_1, z_2) = g_1(y_1, y_2), \text{ where } y_i = \mathsf{Recover}(z_i) \text{ for } i = 1, 2 \ .$$

The second leakage predicate is to decide whether $f = \Lambda y$,

$$g_2(y) = \begin{cases} 0 & f = \Lambda y \\ 1 & \text{otherwise} \end{cases} .$$

Similarly, we define the corresponding Leak_2 such that

$$\forall z \in \mathcal{Z}: \quad \mathsf{Leak}_2(z) = g_2(y), \text{ where } y = \mathsf{Recover}(z) \ .$$

The third leakage predicate is to determine whether $y \geq 0$,

$$g_3(y) = \begin{cases} 0 & y \geq 0 \\ 1 & \text{otherwise} \end{cases} .$$

Again, we define the corresponding Leak_3 such that

$$\forall z \in \mathcal{Z}: \quad \mathsf{Leak}_3(z) = g_3(y), \text{ where } y = \mathsf{Recover}(z) \ .$$

Understanding these leakage predicates are very important to guide us to formulate the proper security definition. Ideally, Φ_k should only reveal the minimum necessary information about Φ from the above leakage predicates but nothing else. However, this is challenging as we want the design to be non-interactive and efficient. So far we are not aware of any existing non-interactive cryptographic solutions practically satisfying these stringent requirements. Even for the not-yet-practical fully homomorphic encryption, we are not aware of any FHE scheme that has order preserving property as required by LP constraints. Given our designs are supposed to work well over real number and aim to strike a good balance between efficiency and security, we follow the security definition below.

[1] We will use x and y interchangeably for the original data, and A and Λ for the sampling matrix, according to the context without further notice.

Definition A transform scheme $\Gamma = (\mathsf{Gen}, \mathsf{Trans}, \mathsf{Solver}, \mathsf{Recover})$ is κ-secure if over the random choice $K \leftarrow \mathsf{Gen}(1^\kappa)$:

$$\forall \Phi_0, \Phi_1 \in \mathcal{S} : \mathsf{SD}(\mathsf{Trans}(K, \Phi_0), \mathsf{Trans}(K, \Phi_1)) \leq \mu(\kappa),$$

where $\mu(\cdot)$ is a negligible function, $\mathsf{SD}(\cdot, \cdot)$ stands for the statistical distance, and $\mathcal{S}$ denotes the set of all the LP problems with the same size in the form of Eq. (3).

B. The Case of Sparse Data

Due to different application contexts, the data to be sensed in healthcare systems can be of various types, including sparse data, general data, and data tampered with noise. We will thoroughly study each case of the three. Below we first assume the studied data is sparse, and describe in a step by step manner about the transformation to be instantiated in Trans algorithm in Γ. Then we summarize our complete scheme description.

First of all, we obfuscate the inequality constraints using a generalized permutation matrix D, which is the product of non-singular diagonal and permutation matrices.[2]

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

For our choice of D, we have $Dy \geq 0$ equivalent to $y \geq 0$.

Next, we use affine mapping $y = Mz - r$ to protect the solution y, where M is $2n \times 2n$ invertible matrix and r is a $2n \times 1$ random vector. This gives the transformation as:

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

Thirdly, we randomly mix the equality and inequality constraints, by left-multiplying a random $2n \times m$ matrix λ to equality constraints and then adding them to inequality ones:

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

Finally, we left-multiply a random $m \times m$ invertible matrix Q to the equality constraints.

$$\begin{aligned} \min \quad & 1^T \cdot (Mz - r) \\ \text{s.t.} \quad & Q\Lambda M z = Q \cdot (f + \Lambda r), \\ & (DM - \lambda \Lambda M) z \geq Dr - \lambda \cdot (f + \Lambda r). \end{aligned}$$

With $c' = 1^T M, \Lambda' = Q\Lambda M, f' = Q \cdot (f + \Lambda r), D' = DM - \lambda \Lambda M, r' = Dr - \lambda \cdot (f + \Lambda r)$, we have

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

Here we ignore the constant term $1^T \cdot r$, which will not affect the correctness of the solution. Further, we can let c'^T always equal to 1^T by randomly generating the first $2n-1$ entries in each column of M and then using 1 minus their sum to get the $2n$-th entry. Similarly, we can let r' always equal to 0 by first randomly choosing D and r, then randomly generating the first $(m-1)$ entries in each of the $2n$ rows of λ, and then using $r' = 0$ to fix the m-th entry in that row. Thus, the transformed LP shares the same structure as Φ in Eq. (3):

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

The secret key is $K = (Q, M, r, D, \lambda)$, where the $2n \times 2n$ matrix M contains $(2n-1) \times 2n$ random elements, and the $2n \times m$ matrix λ contains $2n \times (m-1)$ random elements.

[2]We assume D has positive non-zero elements. It won't affect correctness, if D has negative values. But then some elements in z need to be non-positive.

To better present our system design at sensor and receiver, we define $\mathsf{Trans} = (\mathsf{Trans}_1, \mathsf{Trans}_2)$ to separate the overall LP transformation into two steps without affecting correctness. Specifically, Trans_1 takes as input the secret key K and (f, Λ) in original LP Φ and outputs f' in Φ_k, while Trans_2 takes as input K and Λ and outputs tuples (Λ', D') in Φ_k.

Scheme details: Based on the above transformation, we describe the complete protocol for our system below, which includes the instantiation of the scheme Γ. We first make some reasonable assumptions: 1) The orthonormal basis V is properly shared between the sensor and receiver. 2) A master secret $\mathsf{sk}_1, \mathsf{sk}_2$ used to assist the generation of the random sampling matrix R and the secret transformation keying materials is also shared between the sensor and receiver. 3) All the matrices and vectors to be used in the sampling and secret transformation are generated by using a keyed pseudo-random function (PRF) with random seeds. In this way, the sharing of private matrices and vectors can be made easy by sharing small size seeds. We use different sets of secret transformation key and sampling matrices for different images. Below we denote $F_1(\cdot)$ and $F_2(\cdot)$ as such keyed pseudo-random functions $F_1 : \{0,1\}^k \times \{0,1\}^{\ell_n} \rightarrow \{0,1\}^{\ell_1}$, $F_2 : \{0,1\}^k \times \{0,1\}^{\ell_1} \rightarrow \{0,1\}^{\ell_2}$.

Data Sampling Phase

1) For the i-th sample, the sensor obtains $f_i = R_i b_i$, where b_i denotes source signal to be sampled. It picks a random seed $s_i \leftarrow \mathcal{F}_1(\mathsf{sk}_1, i)$, $i \in [1, 2^{\ell_n}]$, and computes $\sigma_i \leftarrow \mathcal{F}_2(\mathsf{sk}_2, s_i)$.

2) The sensor then uses σ_i as the random coins to sample a random selecting matrix R_i and generates a temporary secret key $K_i = (Q_i, M_i, r_i, D_i, \lambda_i)$ from $\mathsf{Gen}(1^\kappa; \sigma_i)$.

3) With $\Lambda_i = [R_i V, -R_i V]$ and f_i, it calls $\mathsf{Trans}_1(K_i, (f_i, \Lambda_i))$ to encrypt f_i as f'_i, using (Q_i, r_i) in K_i, and sends f'_i to cloud in an authenticated manner.

We use a fresh seed for each image with index i. The index i is attached with the sample f_i when outsourced to cloud. Assume the receiver plans to issue an image recovery request for a specific image sample f'_j to cloud, then:

Image Recovery Phase

1) To recover the j-th image, the receiver computes the related seed s_j from $s_j \leftarrow \mathcal{F}_1(\mathsf{sk}_1, j)$, and uses $\sigma_j \leftarrow \mathcal{F}_2(\mathsf{sk}_2, s_j)$ as the random coins to regenerate the temporary key $K_j = (Q_j, M_j, r_j, D_j, \lambda_j)$ from $\mathsf{Gen}(1^\kappa; \sigma_j)$ and the random selecting matrix R_j.

2) The receiver computes $(\Lambda'_j, D'_j) \leftarrow \mathsf{Trans}_2(K_j, \Lambda_j)$ and sends them to cloud.

3) With Λ'_j, D'_j, and f'_j, the cloud process formulates the transformed LP Φ_k and calls Solver, some general LP solver, to output answer z_j to the receiver.

4) The receiver computes $y_j \leftarrow \mathsf{Recover}(K_j, z_j)$ via $y_j = M_j z_j - r_j$ and recovers the image b_j via basis V and y_j.

We leave the selection of those random secret parameters: Q, M, r, λ, D, etc., to a later section.

Remark. Traditionally for Nyqvist based data acquisition, the sensor first needs to acquire the images and then tries to find ways to compress the image data, i.e., encode the important expansion coefficients under some appropriately

chosen basis [16]. In our system, the acquisition at sensor is simplified, because compressive sensing automatically unifies the sampling and compressing for data acquisition. Besides, sending the transformed samples f' from sensor to cloud for each acquired image introduces no extra bandwidth cost and maintains the privacy-assurance of image recovery, as to be analyzed in Section IV.

Because the secret keying materials and sampling matrices can all be generated by some random seeds and thus easily shared, it allows us to use a fresh set of secret for every image to be captured and recovered. In other words, we can use independent random keys for different images. Recall that we also assume the orthonormal basis V is shared properly between the sensor and the receiver. For different applications, the choice of V can be either data dependent, like Karhunen-Loeve basis (see experiment in Section V), or data independent, like wavelet basis, both of which can be prescribed without much extra overhead.

C. The Case of General Data

In the real world applications, there are many scenarios where healthcare data sources are not exactly sparse. A natural question would be: how to extend the application to those non-sparse data? To answer that, we propose to use sparse data to well approximate general data [13], as long as discarding its small coefficients does not loss much information. Specifically, let x_s be an s-sparse approximation of x obtained by setting all but the largest s entries of x to zero. Here x denotes the coefficients under certain orthonormal basis V for some image b. Let $b_s = Vx_s$. For any orthonormal V, the ℓ_2 norm $\|b - b_s\|_2 = \|Vx - Vx_s\|_2 = \|x - x_s\|_2$. Thus, the difference between original b and the approximated b_s is bounded by that of x and its s-sparse approximation x_s. On the other hand, recent result [14] shows that the solution to problem (1), denoted as x^*, is indeed a good s-sparse approximation of x, bounded by the following condition: $\|x^* - x\|_2 \leq \frac{C_0}{\sqrt{s}} \cdot \|x - x_s\|_1$ for some constant C_0. This error bound applies to any general data [14], indicating that the aforementioned system design can handle general data via good s-sparse approximations.

Remark. Handling general data strikes a balance between efficiency and accuracy. If the general data is nearly sparse, our system will provide good approximation. If the original general data is not quite compressible, our design will still recover the image at its best, by reconstruction from its s-sparse approximation x_s. The efficiency and security of the healthcare monitoring system remains, but the quality of the recovered image might be downgraded.

D. The Case of Data Corrupted with Noise

The data collected from sensors in healthcare systems are not always with high quality. They could be tampered due to the errors in transmission channel, the noise brought by the imperfect sensing devices, etc. As healthcare staffs rely on these systems for accurate diagnosis, it is thus imperative for the system design to robustly handle non-ideal scenarios and still provide acceptable quality of recovered images.

1) The New Problem and Error Model: The modeled problem is: given corrupted samples $f = Ax + e$, for unknown errors e, how to recover x from f? In the literature, there are currently two approaches to recover the underlying image via inaccurate samples. One is via linear programming, and the other is via non-linear optimization, specifically ℓ_1-regularization [8]. They each deal with different error models.

For the former, the error e is assumed to be sparse, but with no limit on the error magnitude. This is suitable to treat the sensing process as linear encoding, where sporadic errors could happen on some positions of the encoded symbols and the error magnitude can be arbitrary. For the latter, the error e is assumed to be a general vector, but its energy level needs to be upper bounded and known at first. This is suitable when the sensing devices may be imperfect and small perturbations with bounded magnitude are added to the measurements.

In this work, we refine ourselves to the linear programming recovery and continue to explore the tradeoffs between robustness and efficiency of our privacy-aware system design. We leave the secure recovery via non-linear optimization as important future work. As noted by [15], [17], it is not possible to recover successfully via LP if e corrupts large fraction of the components of f. In order for success, two assumptions are required. First, e needs to be sparse, i.e., $\|e\|_{\ell_0} \leq \rho \cdot m$, for ρ being state-of-the-art theoretical limit 0.239 [17]. Second, over-sampling is a must, where the number of measurements must be larger than the dimension of the original signal, i.e., $m > n$. In the following, we make the same assumptions.

2) The New Problem Transformation: Our design is inspired by the work of [15], [17]. In particular, since the number of measurements $m > n$, the key to recover x is to get an accurate estimation of corruption e. With both $f = Ax + e$ and e, one can directly solve x from the over-determined equation $Ax = f - e$. The approach is then as follows: When regenerating the $m \times n$ sampling matrix A, the receiver also constructs an $n \times m$ matrix G. Here G and A satisfy that the null space of G subsumes the column subspace of A, i.e., $G \cdot A = 0$. It is obvious that $G \cdot f = G \cdot (Ax + e) = G \cdot e$. Now if we view the sparse error vector e as the $m \times 1$ signal, then $G \cdot f$ is the corresponding $n \times 1$ measurement vector where $n < m$. Therefore, by the standard ℓ_1-min optimization [15],

$$\min \quad \|e\|_1 \quad \text{s.t.} \quad G \cdot f = Ge, \tag{5}$$

one can directly solve e quite accurately. By exactly following our aforementioned transformation design, the recovery of sparse error e can be securely outsourced to cloud. Once receiving and decrypting the error e, the receiver can solve the following linear equation locally to derive the original x:

$$f - e = Ax. \tag{6}$$

Remark. We use over-samples, i.e., $m > n$, to compensate the error corruption in the measurements, and this is the price we have to pay even in the plaintext scenario without security consideration. Because we rely on the cloud to recover e but not $f = Ax + e$, the data flow will be slightly different than the case of sparse data and general data. Specifically, for each sampling, the sensor first acquires the over-sampled linear measurements f, encrypts it as f' via some standard symmetric key encryption, e.g., AES, and then sends the encrypted f' to cloud. The encryption key can be generated and shared in the same fashion as the previous cases by using random coins. The same bandwidth overhead at the sensor side is still ensured as before while the samples get protected. For the data recovery, the receiver first requests the protected f' from the cloud,

decrypts f, generates $n \times m$ matrix G, and outsources a random LP problem according to Eq. (5) using Gf, G, and the random key K as input for the LP transformation. Cloud recovers the transformed noise vector, in the form of $e^* = M^{-1}(e + r)$ where M, r are from the random K generated as in the case of sparse data. Upon receiving e^*, the receiver first recovers noise $e = Me^* - r$ and solves original signal x accordingly via the linear equation as in Eq. (6). We note an extra round between receiver and cloud is required in the above protocol, compared to previous cases, but local resource savings can still be expected as shown later.

IV. Theoretical Analysis

A. Parameters Setting and Security

We assume our system uses finite precision floating numbers, and each entry y_i of the original solution y should be in range $(-L, L)$, where $L = \mathsf{poly}(\kappa)$ and κ is our security parameter. Let the system input $n = \mathsf{poly}(\kappa)$. Following our security definition in Section III-A, we show that our transform scheme is κ-secure. Namely, for all $\Phi_0, \Phi_1 \in \mathcal{S}$, the randomly transformed problems are statistically indistinguishable.

First, we argue that the transformed optimal solution z does not reveal y. Recall that $y = Mz - r$ and $z = M^{-1}(y + r)$, where M is a randomly chosen invertible matrix. We don't have stringent requirement on the random choice of M, as long as it is invertible and satisfies $1^T M = 1^T$ to keep the problem structure consistent with Φ. As for the random r, we uniformly pick each entry r_i of r from a relatively big interval $[-2^\kappa, 2^\kappa]$ with fixed precision. Denote uniform distribution from $[-2^\kappa, 2^\kappa]$ with fixed precision as $\mathcal{U}(-2^\kappa, 2^\kappa)$. If we pick a random vector r^*, where each entry in r^* is sampled from the uniform distribution $\mathcal{U}(-2^\kappa, 2^\kappa)$, we show that the distribution of $y + r$ is statistically close to r^*.

Theorem 4.1: The statistical distance $\mathsf{SD}(y + r, r^*) \leq \mu(\kappa)$, where $\mu(\kappa)$ is a negligible function.

Proof: We first show that for each individual entry, $\mathsf{SD}(y_i + r_i, r_i^*)$, $i \in \{1, \ldots, 2n\}$, is negligible. We have two hypothesises $\mathcal{H}_0$ and $\mathcal{H}_1$, whose ranges are $[-2^\kappa - L, 2^\kappa + L]$ and $[-2^\kappa, 2^\kappa]$, respectively. It is easy to see the optimal distinguishing strategy is to output 0 if the input is from $[-2^\kappa - L, -2^\kappa)$ and $(2^\kappa, 2^\kappa + L]$, or otherwise a random guess $b \leftarrow \{0, 1\}$. This distinguisher's success 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 μ' is a negligible function. Then by applying union bound, we have $\mathsf{SD}(y + r, r^*) \leq \mu(\kappa)$ where $\mu(\kappa) = 2n * \mu'(\kappa)$ as claimed. ■

Therefore, from the cloud's view, $y + r$ and r^* are statistically indistinguishable. Equivalently, the cloud's view of $z = M^{-1}(y + r)$ and $z^* = M^{-1}r^*$ is statistically indistinguishable. Since the cloud's views are indistinguishable if we switch z with z^*, then z does note reveal y.

Next we show that $f' = Q(f + \Lambda r)$ statistically hides f. Since $f = \Lambda y$, we have $f' = Q\Lambda(y + r)$. Again, since each entry of r is sampled from the uniform distribution $\mathcal{U}(-2^\kappa, 2^\kappa)$, according to Theorem 4.1, we can replace $y + r$ with r^*. The cloud's views of $f' = Q\Lambda(y + r)$ and $f^* = Q\Lambda r^*$ are statistically indistinguishable. Hence, for all f_0, f_1, the statistical distance between f_0', f_1' are at most $\mu(\kappa)$.

Recall that $\Lambda = [RV, -RV]$, where R is randomly sampled for each problem and V is an orthonormal basis. As claimed above, the cloud's views are indistinguishable if we replace f' with f^*. Given f_b we only need to show the distribution of $f_b^* = Q_b\Lambda_b r_b$, $\Lambda_b' = Q_b\Lambda_b M_b$ and $D_b' = (D_b - \lambda_b\Lambda_b)M_b$ are indistinguishable for $b \in \{0, 1\}$. Indeed, since all the components $Q_b, r_b, \Lambda_b, M_b, D_b, \lambda_b$ that are used to generate f_b^*, Λ_b' and D_b' are randomly sampled for each problem, the distribution of f_b^*, Λ_b' and D_b' is indistinguishable for different f_b. Therefore, we have

$$\mathsf{SD}(\mathsf{Trans}(K, \Phi_0), \mathsf{Trans}(K, \Phi_1)) \leq \mu(\kappa) \ .$$

Now we show that the proposed system is secure if the underlying PRF's $\mathcal{F}_1, \mathcal{F}_2$ are secure. Note that our system uses $\mathcal{F}_2(\mathsf{sk}_2, \mathcal{F}_1(\mathsf{sk}_1, i))$ instead of real random coins. As a modified system, we replace $\mathcal{F}_1(\mathsf{sk}_1, i)$ with uniformly random $s_i \leftarrow \{0, 1\}^{\ell_1}$. This modification is indistinguishable to any probabilistic polynomial time adversary $\mathcal{A}$ since the PRF $\mathcal{F}_1(\mathsf{sk}_1, \cdot)$ is indistinguishable with a truly random function. Next, we replace the random coins σ generated by $\mathcal{F}_2(\mathsf{sk}_2, s_i)$ with real random coins. Again, this modification is indistinguishable to any probabilistic polynomial time adversary $\mathcal{A}$ since the PRF $\mathcal{F}_2(\mathsf{sk}_2, \cdot)$ is indistinguishable with a truly random function. As our transform scheme is statistically secure, the proposed system is secure as claimed.

TABLE I: Asymptotic efficiency and cost comparison

	Existing System	Our System
Storage cost	$\{f\}$ at receiver	$\{f'\}$ at cloud
Sensor bandwidth	$\|f\|$ per sample	$\|f\|$ per sample
Local computation	over $O(n^3)$	$O(n^\theta), 2 < \theta < 3$
Privacy protection	No	Yes

B. Efficiency Analysis

On the sensor side, the most critical cost from the energy-efficiency perspective is the communication cost [3]. We note the fact that the size of compressed samples f and the randomly transformed one f' that is sent to cloud are with the same size. Therefore, our healthcare monitoring system incurs the same bandwidth cost on the sensor as the current practice based on compressive sensing does without security consideration. For the same reason, our system incurs the same storage cost on the cloud side as the current practice does without security consideration. Compared to storing the original data in whole or images in uncompressed format, a recent study in [18] has shown that using compressive sensing can reduce storage cost for up to 50%.

Our system also provides local computational savings on the receiver. The most expensive operations in the transformation design is the matrix-matrix multiplications, which cost asymptotically $O(n^\theta)$ for some $2 < \theta < 3$ due to $m < 2n$. If performing everything locally without outsourcing, solving the LP problem Φ usually costs more than $O(n^3)$ time, e.g. [19]. Note that though compared to existing systems without security consideration, the sensor in our system needs to do extra

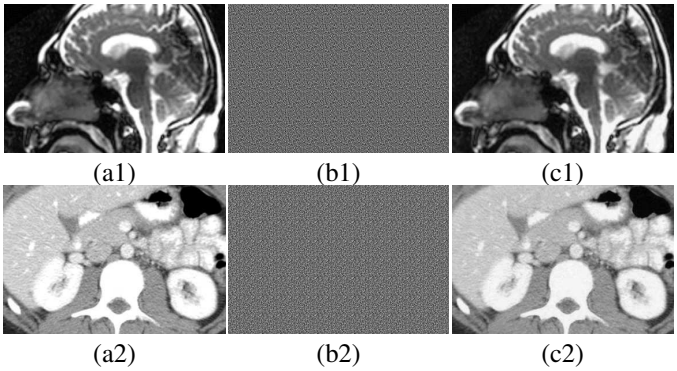

Fig. 2: Demonstration of privacy-assurance and effectiveness for the case of compressible data. (a1)(a2): simulated signal source; (b1)(b2) reconstruction via encrypted data; (c1)(c2) reconstruction at receiver after decryption.

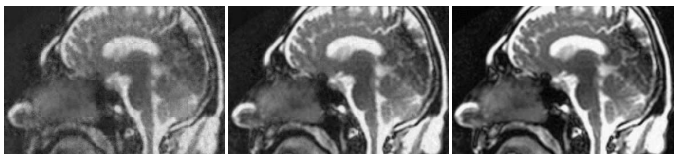

Fig. 3: Recovered images via different measurements m.

computation in the sensing process for the sample protection, solving the LP dominates the overall local computation cost. Thus, using cloud-assistance with privacy-awareness provides sensor/receivers considerable computational savings in theory. Besides, with our proposed transformation, the cloud process can utilize any existing solvers for the LP problem Φ_K, which further ensures the cloud side efficiency. Detailed comparison between our proposed cloud-assisted privacy-aware healthcare monitoring system and existing compressive sensing based systems is summarised in Table I.

V. Experiment

In this preliminary study, we use software implementation to simulate data sensing, and treat selected image data as the signal source to be sensed. Empirical evaluation is conducted via MATLAB and MOSEK optimisation toolbox, on a workstation with an Intel Core i5 CPU running at 2.90 GHz and 6 GB RAM. In our settings, all selected images are with size 384×256. To avoid the handling of large matrices in the experiment, we divide each image into 96 image blocks with size 32×32 and process the sampling and recovering of each image block independently. The selecting matrix R in $A = RV$ is populated by sampling i.i.d. entries from the normal distribution $N(0, 1)$. The orthonormal basis V we use is the Karhunen-Loeve (KL) basis, generated via diagonalising the covariance matrix from a set of sampled image blocks, as suggested in [18]. Using the KL-basis V, we view each 32×32 image block as a $n \times 1$ sparse or compressible vector, where $n = 1024$. For space interest, we report below only a few representative images for cases studied.

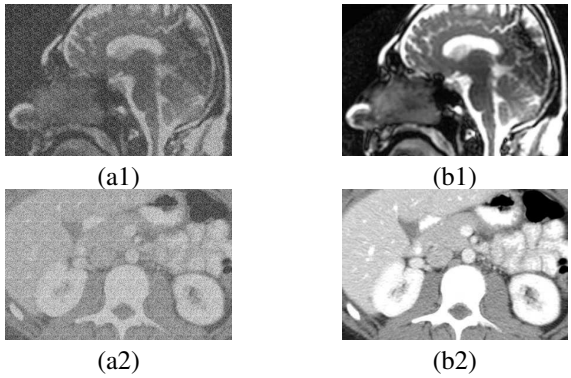

Fig. 4: Demonstration of effectiveness for the case of noise data. (a1)(a2): reconstruction without oversampling; (b1)(b2) reconstruction at receiver after decryption

A. Effectiveness Evaluation

Our aim for effectiveness evaluation is to show the empirical results on correctness and privacy-assurance of the design.

1) The Case of Sparse and General Data: Because compared to sparse data, the case for general data in essence achieves the tradeoff between efficiency and recovery accuracy (see Section III-C), thus we do not intentionally differentiate the two cases in this preliminary evaluation. Figure 2-(a1) and (a2) give two examples of the simulated signal source. The recovered image after the transformation and receiver decryption $y = Mz - r$ is shown in Figure 2-(c1) and (c2). For good experiment results, we use relatively large $m = 256$ linear measurements for each block recovery, whereas the sparsity level of many 32×32 blocks is around a few dozen. This follows the "four-to-one" rule suggested by [13], i.e., the number of samples is roughly $4\times$ the sparsity level of the targeted signals. For comparison purposes, recovered images using $m = 128, 192, 256$ measurements are shown in Figure 3.

Figure 2-(b1) and (b2) shows the recovered image before receiver decryption, based on the blinded vector $z = M^{-1}(y + r)$, is indistinguishable from a randomly generated image with only noise background, because of the random affine mapping by M and r. This is true even if the adversary knows the basis V, as fresh secret keys are generated for each image block reconstruction.

2) The Case of Samples Corrupted with Noise: In this case, stable recovery of data is achieved at the of over-sampling (See Section III-D). Here we follow a similar experiment setting as in [15]. For each 32×32 image block, we over-sample it by a factor of 2, which yields $m = 2048$ measurements. For each $m \times 1$ measurement vector, we corrupt it with an $m \times 1$ sparse error e with $10\% \times m$ non-zero entries. The magnitude of each non-zero entry in e is randomly chosen and comparable to the average non-zero entries of original image[3].

The experiment is still based on the simulated signal sources as shown in Figure 2. To give a perceptual observation, Figure 4-(a1) and (a2) represent the recovered images using corrupted measurements without oversampling. We can see

[3]As noted in [15], [17], for successful recovery, the magnitude of non-zero entries in e can be arbitrary, as long as its sparsity level is upper bounded.

TABLE II: Local savings for the sparse/general data (Benchmark I), and data tampered with noise (Benchmark II).

Benchmark I		Original Recovery	Secure Image Recovery		Asymmetric Speedup
#	image block size	$t_{original}$ (sec)	t_{sensor} (sec)	$t_{receiver}$ (sec)	$\frac{t_{original}}{t_{sensor}+t_{receiver}}$
1	32×32	1.88	0.025	0.45	4.0 ×
2	48×48	15.37	0.193	4.38	3.4 ×
Benchmark II		Original Recovery	Secure Image Recovery		Asymmetric Speedup
#	image block size	$t_{original}$ (sec)	t_{sensor} (sec)	$t_{receiver}$ (sec)	$\frac{t_{original}}{t_{sensor}+t_{receiver}}$
1	32×32	38.65	0.195	5.35	7.0 ×
2	48×48	393.57	1.53	54.99	7.0 ×

that even if the added noise e is sparse, the recovered image quality is still very poor. But by following the transformation and oversampling method in Section III-D, we can leverage the cloud to securely recover an encrypted version of error e, in the form of $M^{-1}(e+r)$, decrypt it, and then solve for the original x for each image block, shown in Figure 4-(b1) and (b2). Compared to the simulated signal sources in Figure 2, again the difference is hardly noticeable. Throughout the recovering process, cloud only sees the randomly transformed LP problem and an encrypted version of the error e.

B. Efficiency Evaluation

we first focus on the computational cost of privacy-assurance done by the sensor in our system as compared to existing sensing systems without security consideration. In addition, we also measure the overall local computation savings to the sensor and receiver as a whole, by using cloud-assisted image reconstruction. The benchmark is based on the simulated signal sources as before. For this preliminary study, we only choose two different image block sizes, 32×32, and 48×48, to avoid the handling of large matrix. All the results represent the mean of 10 trials and each trial focuses on one randomly selected image block recovery.

1) Evaluation on Sensing Overhead : Compared to existing systems, the sensor in our system needs to do extra computation for the sample protection. For each sample $f = Rb$, the sensor needs to further transform f to $f' = Q(f + \Lambda r)$. For the 32×32 block size, the time to compute f' and f is 0.025 sec and 0.0045 sec, respectively. For the 48×48 block size, it is 0.1933 sec and 0.0205 sec, respectively. Therefore, for security we inevitably increase the computation cost on sensor by a factor of 5.6 × and 9.4 ×, respectively. Considering the overall local computation savings (shown next) our system brings to the sensor and receiver as a whole, we believe such overhead is acceptable from a practical point of view. Also, since compressive sensing is non-adaptive and sampling each image block is independent, we can always increase the system performance by adding more sensors operating in parallel.

2) Evaluation on Local Computational Saving: Table II benchmark I corresponds to the evaluations for the cases of sparse and general data. The first two columns report benchmark size. For 256 measurements from the 1024×1 image data, we create a ℓ_1-min problem with 1024 variables and 256 constraints. The time for image recovery without security consideration, $t_{original}$, is reported in the third column, which relates to solving LP. The time to do the local transformation is reported in the fourth and fifth columns, separated into the time for the sensor t_{sensor} to do the sample protection, and the time for the receiver $t_{receiver}$ to do the problem transformation and data decryption. For fair efficiency gain calculation, we do not include the final image recovery cost (0.009 sec and 0.021 sec for the 32×32 and 48×48 image block sizes respectively), as it needs to be performed by receiver in both the existing and proposed cloud-assisted systems. We assess the local computational savings by *Asymmetric Speedup*, calculated as $\frac{t_{original}}{t_{sensor}+t_{receiver}}$. Here $t_{sensor} + t_{receiver}$ represents the total local computation cost. The table shows we can always achieve more than $3.4\times$ savings for the selected image block size. Because each image may involve many such blocks, our results suggest substantial computational cost can be shifted from local to cloud. The overall local savings also justify our previous argument that the increased sensing overhead for sample protection is practically acceptable. Note that our system can directly utilise any LP solver at cloud, suggesting its easy adoption in practice.

The benchmark II evaluates the case of data tampered with noise. Recall that we need to oversample for successful recovery, which creates a larger ℓ_1-min problem. Take the 32×32 image block for example. With $m = 2048$ measurements, we have a ℓ_1-min with 2048 variables and 1024 constraints to solve for e (see Section III-D). The larger problem size explains why it takes more time for image recovery, compared to benchmark I. Again, for fair efficiency gain calculation, we do not include the final image recovery cost (0.938 sec and 9.521 sec for solving the linear equations for the 32×32 and 48×48 image block sizes respectively), which must be done by receiver in both existing and proposed cloud-assisted systems. Larger than the case of sparse data and general data, our system provides at least $7.0\times$ *Asymmetric Speedup* for the selected sizes of image block reconstruction. This is because we use oversampling to compensate the noise, resulting in a larger LP for the same image block size. When the LP size increases, it becomes harder to solve than performing large matrix multiplications for local transformation.

VI. Related Work

Compressive sensing [6], [7], [8] has received lots of attention in signal processing community recently. Because it unifies the sampling and compression for data acquisition, compressive sensing has found of great interest in a number of recent biomedical sensing applications [2], [3], [5], [11]. But those works consider neither the cloud assistance to handle large amount of data nor the importance of securing the healthcare data. These are indispensable design requirements in our system design, and we maintain the same sensing simplicity and low bandwidth cost at the sensor. A line of

research loosely related to the proposed work is the study on the security and robustness of encryption via compressive sensing, by Orsdemir et al. [20], Rachlin et al. [21], and others therein. They investigate the secrecy of the linear measurement by assuming that the adversary has no knowledge of the sensing matrix. Their results suggest that by keeping the sensing matrix as secret, approaches of brute-force searching that try to recover original data can be computationally infeasible. Note that in our system, the privacy-assurance on the samples and recovered data is based on the random transformation (see Section IV). A recent work [18] by Divekar et al. shows how to leverage compressive sensing to compress the storage of existing correlated image datasets. The cloud-assisted image recovery over encrypted samples in our system is also akin to secure computation outsourcing [12], [22], [23] (to list a few), which aims to protect both input and output privacy of the outsourced workloads. With garbled circuits [24] and fully homomorphic encryption (FHE), a general solution is shown feasible in theory by Gennaro et al. [12], where the computation is represented by an encrypted combinational boolean circuit that allows to be evaluated with encrypted private inputs. But it is not practical, for the extremely large circuit size and huge computation cost of FHE operations. Besides general solutions, researchers have been working on customized mechanisms for securely outsourcing specialized computations, such as matrix multiplications [23], modular exponentiations [22], etc. Though elegant, they are not directly applicable in our system as they handle different computations.

VII. Conclusion

In this paper, we have proposed a privacy-aware cloud-assisted healthcare monitoring system via compressive sensing. The design exploits techniques from different domains, and achieves the following novel benefits. In our architecture, the sensor can utilise the framework of compressive sensing to consolidate the sampling and compression via only linear measurements. The random mapping based protection ensures no sensitive samples would leave the sensor in unprotected form. Such a security approach also minimises the communication cost for sensor data acquisition and transmission. On the receiver side, the cloud-assisted image recovery over encrypted samples provides great computational savings, yet without revealing either the received compressed samples, or the content of the recovered underlying image. We showed that our proposed design is able to achieve robustness and effectiveness in handling image recovery in cases of sparse data, general data, and even data samples corrupted with noise. Extensive security analysis and empirical experiments have been provided to demonstrate the privacy-assurance, efficiency, and the effectiveness of our healthcare monitoring system.

As future work, we will conduct thorough evaluations of the system on sensor and cloud testbed. We will investigate the possible extension on content based image retrieval, and even performance speedup via hardware built-in system design.

Acknowledgement

This work is supported in part by Research Grants Council of Hong Kong under the ECS grant CityU 138513, and by US National Science Foundation under grants CNS-1262277, CNS-1262275, and IIA-0853371.

References

[1] J. Ko, C. Lu, M. Srivastava, J. Stankovic, A. Terzis, and M. Welsh, "Wireless sensor networks for healthcare," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1947–1960, 2010.

[2] M. Shoaib and H. Garudadri, "Digital pacer detection in diagnostic grade ecg," in *Proc. of IEEE Conf. E-health, Networking, Appl. and Services*, 2011, pp. 326–331.

[3] M. Shoaib and N. Jha and N. Verma, "A compressed-domain processor for seizure detection to simultaneously reduce computation and communication energy," in *Proc. of IEEE Custom Integrated Circuits Conf.*, 2012, pp. 1–4.

[4] J. Ko, J. Lim, Y. Chen, R. Musaloiu-E., A. Terzis, G. Masson, T. Gao, W. Destler, L. Selavo, and R. Dutton, "MEDiSN: Medical emergency detection in sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 10, no. 1, pp. 11:1–11:29, 2010.

[5] W. Xu and M. Zhang and A. Sawchuk and M. Sarrafzadeh, "Co-recognition of human activity and sensor location via compressed sensing in wearable body sensor networks," in *Proc. of IEEE Conf. Body Sensor Networks*, 2012, pp. 124–129.

[6] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[7] E. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[8] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[9] European Network and Information Security Agency, "Cloud computing risk assessment," Online at http://www.enisa.europa.eu/act/rm/files/deliverables/cloud-computing-risk-assessment, November 2009.

[10] 104th United States Congress, "Health Insurance Portability and Accountability Act of 1996 (HIPPA)," online at http://www.hhs.gov/ocr/privacy/hipaa/understanding/index.html, 1996.

[11] F. Chen and A. Chandrakasan and V. Stojanovic, "A signal-agnostic compressed sensing acquisition system for wireless and implantable sensors," in *Proc. of IEEE Custom Integrated Circuits Conf.*, 2010.

[12] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Proc. of CRYPTO*, 2010, pp. 465–482.

[13] E. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Proccess. Mag.*, vol. 25, no. 2, pp. 21–30, 2008.

[14] E. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9-10, pp. 589–592, 2008.

[15] E. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[16] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 14–20, 2008.

[17] C. Dwork, F. McSherry, and K. Talwar, "The price of privacy and the limits of LP decoding," in *Proc. of STOC*, 2007, pp. 85–94.

[18] A. Divekar and O. Ersoy, "Compact storage of correlated data for content based retrieval," in *Proc. of Asilomar Conf. on Signals, Systems and Computers*, 2009, pp. 109–112.

[19] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373–396, 1984.

[20] A. Orsdemir, H. Altun, G. Sharma, and M. Bocko, "On the security and robustness of encryption via compressed sensing," in *Proc. of IEEE MILCOM*, 2008, pp. 1–7.

[21] Y. Rachlin and D. Baron, "The secrecy of compressed sensing measurements," in *Proc. of Allerton Conf. on Communication, Control, and Computing*, 2008, pp. 813–817.

[22] S. Hohenberger and A. Lysyanskaya, "How to securely outsource cryptographic computations," in *Proc. of TCC*, 2005, pp. 264–282.

[23] M. Atallah and K. Frikken, "Securely outsourcing linear algebra computations," in *Proc. of ASIACCS*, 2010, pp. 48–59.

[24] A. Yao, "Protocols for secure computations (extended abstract)," in *Proc. of FOCS*, 1982, pp. 160–164.