PRIVACY-PRESERVING DATA COLLECTION AND RECOVERY OF COMPRESSIVE SENSING

Tsung-Hsuan Hung, Sung-Hsien Hsieh and Chun-Shien Lu

Institute of Information Science, Academia Sinica, Taipei, Taiwan

ABSTRACT

Energy-efficient data collection and privacy-preserving data recovery have received much attention recently. We propose the first encryption framework for the computation-intensive basis pursuit problem to be securely solved in the cloud with the data being efficiently collected using compressive sensing. We provide security and efficiency analyses to show the effectiveness of our method. Simulations and comparison with state-of-the-art are also conducted.

***Index Terms*—** Basis pursuit, compressive sensing, convex optimization, encryption, security

1. INTRODUCTION

Wireless sensors have been popularly used to monitor critical environments for data collection and retrieval. Nevertheless, due to the limited energy and storage of sensors, the collected data usually should be transmitted to the cloud side for storage and further processing. After data collection, users have two choices for utilizing the data stored in the cloud. First, users can explore all non-processed data directly downloaded from the cloud. This, however, will create computational overhead due to data have exploded in the last few years. The second framework [1, 2, 3, 4] needs the cloud to process the data and what the users need to do is to receive the outcomes from the cloud side. Though this framework can save computational power for sensors and users, it cannot guarantee the users' privacy and security when the cloud is untrusted.

To further reduce the energy consumption of sensors, [1] formulates the first healthcare monitoring system architecture with compressed sensing [5, 6], a ground-breaking sub-Nyquist sampling for sensor side and reconstruction for cloud side. In [1], the reconstruction process in compressed sensing, which needs to be protected, is solved in the cloud such that it can reduce the users' computational cost. Moreover, the skill for protecting the recovery process is to transform the basis pursuit (BP) problem to linear programming (LP) problem such that off-the-shelf encryption systems [7, 8] can be employed to solve the secure LP problem while achieving data privacy in the cloud. Nevertheless, transforming BP to LP and encrypting the resulting LP problem will increase the communication cost and computational complexity. Also, [9, 10] discuss the security issue for solving optimization problems in the cloud.

In this paper, we follow the same architecture of [1] but propose an encryption system for directly solving BP instead of LP in the cloud. Below, we summarize the contributions of this paper as follows.

1. To our knowledge, we first propose an encryption framework based on the BP problem, which has not been found up to now, such that it can be securely solved in the untrusted cloud.

2. Due to saving the step of transforming BP into LP, our scheme can reduce communication cost and computational complexity.

3. As for security analysis, we prove the proposed encryption system can resist the $ciphertext\text{-}only\ attack$ (COA), which is also the focus in [1].

2. PROPOSED SCHEME

We first briefly introduce compressive sensing in Sec. 2.1 as it is the foundation of our method. Second, we illustrate the main idea and security definition in Sec. 2.2. Finally, we describe the whole structure and design ideas in Sec. 2.3.

2.1. Preliminary on Compressive Sensing

In the context of compressive sensing (CS) [5, 6], let $v \in \mathbb{R}^{N\times 1}$ denote an original signal, let $\Phi \in \mathbb{R}^{m\times N}$ represent a sampling matrix, and let $y \in \mathbb{R}^{m\times 1}$ be the measurement vector with $m < N$ and m/N being the measurement rate. At the encoder, the measurement vector y in CS is obtained from sensing the original signal v via random projection defined as:

$$y = \Phi v. \tag{1}$$

v can be expressed in terms of sparse representation as:

$$v = \Psi x, \tag{2}$$

where $\Psi \in \mathbb{R}^{N\times N}$ is an orthonormal basis and $x \in \mathbb{R}^{N\times 1}$ is a sparse vector. Specifically, we say that x is $S\text{-}sparse$ if there are S nonzero entries in x. Substututing Eq. (2) in Eq. (1), we have y=Ax, where $A = \Phi\Psi$.

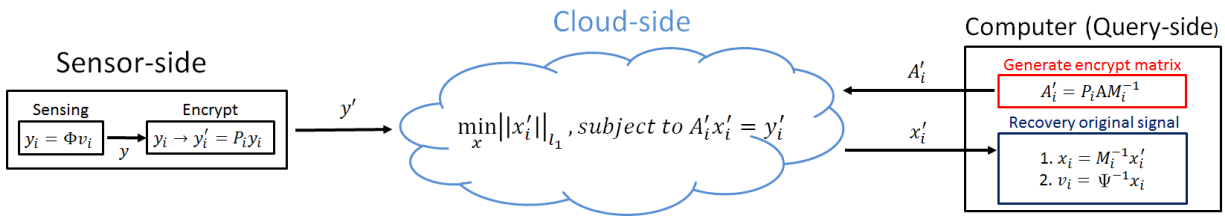

Fig. 1. Block diagrams of proposed method, including the encryption task on both the sensor and query side, and the recovery task on the server side.

At the decoder of CS, [11] shows that if A satisfies $Restricted\ Isometry\ Property$ (RIP), then we can recover x (S-$sparse$) given y and A by solving a Basis Pursuit (BP) problem as follow:

$$\min_x \|x\|_1, \text{ subject to } Ax = y, \tag{3}$$

where $\|\cdot\|_1$ is l_1-$norm$. In addition to BP that is a kind of convex optimization algorithms, greedy algorithms like OMP have also been popularly used in CS. However, the reason for choosing BP here is that it exhibits higher reconstruction capability, which implies the need of smaller measurement rates than greedy algorithms [12]. Smaller measurement rates also imply less sampling and communication cost in the sensor side of our scheme.

2.2. Main idea and security definition

Our goal is to propose a scheme so that the server can solve the BP problem in Eq. (3) and keep the original signal v secret. We denote BP program as a task $\Omega = (y, A)$ and our scheme consists of four probabilistic polynomial time algorithms, $\Gamma = (Gen, Tra, Sol, Rec)$:

- $Gen(1^\kappa)$ is a key generation algorithm, which takes security parameter 1^κ as input to yield the secret key K.
- $Tra(K, \Omega)$ is a transformation algorithm, which takes secret key K and task Ω as input and generates the transformed task $\Omega^K = (y', A')$, where y' and A' are the ciphertexts of y and A, respectively.
- $Sol(\Omega_K)$ is a solver algorithm, which takes the transformed task Ω^K as input to obtain x' that is a ciphertext of x by solving the optimization problem Ω^K.
- $Rec(x', K)$ is a recovery algorithm, which takes the ciphertext x' and secret key K as input to yield x for subsequent recovery of the original signal v.

Since the algorithm Sol is fixed to solve BP problems, it is the key and challenging for us to design a secure transformation Tra such that Ω^K is still a BP problem. As mentioned previously, Sol works in the cloud, so the security strength of such scheme Γ relies on x', y', and A' of Ω^K. Moreover, if x', y', and A' can be designed to be secret, then Γ will be a secure scheme. In the following, we define the security definitions for these considerations.

Definition 1. If $\Gamma = (Gen, Tra, Sol, Rec)$ satisfies the following conditions:

1. There is no polynomial time algorithm to decrypt x' without $K \leftarrow Gen(1^\kappa)$.
2. Given any two distinct tasks Ω_0 and Ω_1, and a random choice $K \leftarrow Gen(1^\kappa)$, the distributions of y_0' from $Tra(\Omega_0, K)$ and y_1' from $Tra(\Omega_1, K)$ are indistinguishable, and the distributions of A_0' from $Tra(\Omega_0, K)$ and A_1' from $Tra(\Omega_1, K)$ are indistinguishable.

then Γ is κ-secure.

Remark. The most important thing here is to keep x secret, so our method has to satisfy condition 1. In addition, the transformed task must protect y and A, so we need the distributions regarding pairs of y_0' and y_1', and A_0' and A_1' of the two transformed tasks to be indistinguishable, respectively.

2.3. The scheme process and design ideas

In this section, we will describe in a step by step manner about the scheme details to be instantiated in Γ. The key generation Gen is divided into two parts, $Gen^\alpha(1^\kappa, i)$ and $Gen^\beta(1^\kappa, i)$, where Gen^α is the key generation of sensor side, Gen^β is the key generation of query side, and i is the index for i-th sampling. The scheme process is illustrated in Fig. 1 and is described as follows.

- In the sensor side (Φ is fixed):

1. For the i-th sampling, the sensor gets $y_i = \Phi v_i$, where Φ is the sampling matrix and v_i is the original signal to be sampled.
2. Use $Gen^\alpha(1^\kappa; i)$ to generate a key $K_i^\alpha = \{P_i\}$ for i-th sampling, where $P_i \in \mathbb{R}^{m \times m}$ is a random matrix and is different in every sampling.
3. Calculate and upload $y_i' = P_i y_i$ with index i to the cloud.

- In the query side:

4. Use $Gen^{\beta}(1^{\kappa}; i)$ to generate a key $K_i^{\beta} = \{P_i, M_i\}$ for i-th sampling, where $M_i \in \mathbb{R}^{N \times N}$ is a random permutation matrix and is different in every sampling.
5. Set $A = \Phi\Psi$, where Ψ is an orthonormal basis, and upload $A_i' = P_i A M_i^{-1}$ with index i to the cloud.

- In the cloud side:

6. Combine y_i' and A_i' to obtain the task $\Omega_i^{K_i} = (y_i', A_i')$. Then, use Sol and $\Omega_i^{K_i}$ to yield x_i'.

- Back to the query side:

7. Download x_i' from the cloud, use Rec and x_i' to obtain $x_i = M_i^{-1} x_i'$, and recover $v_i = \Psi x_i$ to find the original signal for i-th sampling.

Note that Step 3 and Step 5 belong to the Tra algorithm. Since Tra is the key related to security of our whole scheme, we focus on designing it as follows.

Design ideas: In the Tra algorithm, we need to ensure x and Ω are secret. We start with keeping x secret; an intuitive idea for encrypting x is to multiply a random matrix $M \in \mathbb{R}^{N \times N}$ with x. Under the circumstance, Eq. (3) can be transformed to

$$\min_{x'} \|M^{-1}x'\|_1, \ subject\ to\ AM^{-1}x' = y, \tag{4}$$

where $x' = Mx$. This new task (Eq. (4)), however, cannot fit our considerations in Sec. 2.2 since it cannot keep x secret when solving the problem in the cloud and it does not maintain the framework of BP. To conquer this problem, we let M be a random permutation matrix such that the term $\|M^{-1}x'\|_1$ is equivalent to $\|x'\|_1$ according to the nature of l_1-$norm$. Thus, Eq. (4) can be rewritten as:

$$\min_{x'} \|x'\|_1, \ subject\ to\ AM^{-1}x' = y. \tag{5}$$

As we can see from Eq. (5), we can solve it without sending M^{-1} to the cloud and the formulation is still a BP problem. However, Eq. (5) will reveal y and A to the cloud. So, we adopt another random matrix $P \in \mathbb{R}^{m \times m}$ and multiply it with both sides of the constraint in Eq. (5) to yield:

$$\min_{x'} \|x'\|_1, \ subject\ to\ A'x' = y', \tag{6}$$

where $A' = PAM^{-1}$ and $y' = Py$. In sum, the Tra algorithm in our scheme Γ is a transformation and needs two matrices M and P to protect x, y, and A.

In the sensor side, it only conducts sampling on v to obtain y, uses P to encrypt y, and sends y' to the cloud. On the other hand, in the query side, it needs to use P and M to encrypt A, sends A' to the cloud side, downloads the solution x' from the cloud, and recovers the original signal v. Obviously, the characteristic of our method is that both the query and cloud sides bear most computational and communication cost of the whole framework while the sensor side operates in an energy conservative way.

3. ANALYSIS

This section provides the analysis of security and efficiency of our scheme. In the part of security, we assume the adversary to be COA and derive the security guaranteed under this situation. As for efficiency analysis, we show that our scheme is more efficient than [1].

3.1. Security analysis

In our framework, the adversary is honest-but-curious and is only allowed to corrupt the server passively. In addition, our scheme achieves κ-secure under the assumption that the adversary model is COA. The security of our scheme depends on the security parameter 1^{κ} since the key $K = (P, M)$ is generated by $Gen(1^{\kappa})$ and is different in every sampling. Moreover, we assume the key is secure for the appropriate parameter 1^{κ} and pseudorandom generation. The following theorem proves the security strength of our scheme described in Sec. 2.3.

Theorem 2. *The scheme $\Gamma = \{Gen, Tra, Sol, Rec\}$ we proposed is κ-secure under the situation of COA, which runs in polynomial time.*

Proof. Since the adversary is assumed to be COA, what we can leak to it are $\Omega^K = \{y', A'\}$ and x'. Under the circumstance, there are two conditions need to be satisfied for Γ to be κ-secure.

1) Since the ciphertext $x' = Mx$ in our scheme is permuted from the original x and x' has been exposed, the adversary knows the number of non-zero entries and their actual values in x, but does not know the correct positions. The adversary, however, does not own any prior knowledge for x, so it can only use x' to guess the positions of entries with the hope of getting correct x. The issue of guessing the positions is a subclass selection problem, which is NP-hard. Here, we will give some arguments for it, though [13] mentioned it is a NP-hard problem. Now, if the adversary wants to arrange x' to get x, the computation may cost

$$\frac{N!}{(N-k)!} = N(N-1)...(N-k+1), \tag{7}$$

where N is the length of x' and k is the number of non-zero entries in x'. Clearly, the lower bound of Eq. (7) is:

$$(N-k+1)^k \leq N(N-1)...(N-k+1). \tag{8}$$

Specifically, Eq. (8) shows the computational cost to be $O(N^k)$ for arranging x' to become x that is not a polynomial time mission.

2) We want the distributions of y_b' and A_b' from $Tra(\Omega_b, K)$ to be indistinguishable for $b \in \{0, 1\}$, where $y_b' = P_b y_b$

and $A'_b = P_b A M_b^{-1}$. Trivially, y'_b and A'_b are generated by the random components P_b and M_b for each sampling, so the distributions for a pair of y'_0 and y'_1 and the distributions for a pair of A'_0 and A'_1 are both indistinguishable. Therefore, Condition 2 holds.

We conclude that our method satisfies these two conditions and is κ-secure under COA, which runs in polynomial time. □

3.2. Efficiency analysis

The argument of efficiency is divided into communication cost and computational complexity. We compare our scheme with [1] with respect to the sensor side (SS) and query side (QS) in Table 1.

Table 1. Efficiency comparison between our scheme and [1].

	Our scheme	[1]
SS's communication cost	$O(m)$	$O(m)$
SS's computational cost	$O(mN)$	$O(mN)$
QS's communication cost	$O(mN)$	$O(N^2)$
QS's computational cost	$O(m^2N)$	$O(mN^2)$

We can observe from Table 1 that, in the sensor side, the communication cost and computational complexity of our scheme are the same with those of [1] due to the use of compressive sensing. Nevertheless, we do not have to change the sensing matrix Φ in every sampling, which is more practical in sensor side. On the other hand, for communication cost in the query side, our scheme just sends an $m \times N$ matrix to the cloud but [1] needs to send two matrices with different sizes of $m \times 2N$ and $2N \times 2N$. Moreover, the computational cost of our scheme is $O(m^2N)$, which is less than $O(mN^2)$ of [1], since we just permute the columns of matrix in Step 5 (query side) for multiplying M^{-1} with A.

It should be noted that the base for our method to be reasonable is that the cost of needing the cloud to solve a hard problem must be definitely larger than that of our encryption tasks. According to [14], the cost of solving BP problem is $O(m^2N^{\frac{3}{2}})$, which is larger than the computational cost of our encryption task that is $O(m^2) + O(m^2N)$. This gives an evidence that our scheme is actually efficient.

4. SIMULATIONS

Our framework can be adopted in many application scenarios that require fast encoding (data acquisition) with slow but secure decoding/recovery, where slow recovery can be conquered by means of strong capability of the cloud server. In this section, we take image sensing and encryption as an example to illustrate the effectiveness of our method.

4.1. Experimental Setup

All simulations were conducted on PC equipped with Windows 7 with 3.40GHz Intel Core i7 CPU and 4 GB RAM. We programmed the proposed method using MATLAB and popularly used CVX toolbox for CS recovery.

Here we divided an image into 256 blocks of size 32×32, which can be regarded as many information collected from different sensors. Thus, the original signal v corresponds to a 1024-dimensional vector, and the sampling and recovery for each image block are independent. The sensing matrix Φ was selected from the normal distribution $\mathbb{N}(0, 1)$ and the orthonormal basis Ψ was the Karhunen-Loeve(KL) basis in order to satisfy RIP. For encryption matrices, P was an $m \times m$ matrix generated by $\mathbb{N}(0, 1)$ and M was an $N \times N$ random permutation matrix. Note that we currently focus on security aspect of this work and do not consider measurement noises.

4.2. Results

In our simulations, the goal is to examine if the signal recovered from the original task in Eq. (3) is the same with the signal decrypted from the ciphertext x' that is obtained by solving the task in Eq. (6). Although we only show the results under $m/N = 0.25$ for the Lena image below but we actually observe similar results under different measurement rates.

According to our simulation results, we see that x' will not make any leakage to the cloud, and the same recovery performance (see PSNR values in Table 2) can be obtained from the image recovered from the original task and that from our method. Table 2 further shows that the original task is more costly than the whole encryption system for each block; *i.e.*, $7.146s > 0.377s + 10^{-4}s$.

Table 2. Comparison between the original task (Eq. (3)) and our encryption task for each block, where s means seconds.

Block size	Original task	Our encryptions	
	$t_{original}$	t_{sensor}	t_{query}
32×32	$7.146(s)$	$6 \times 10^{-4}(s)$	$0.377(s)$

Speedup	PSNR (dB)	
$\frac{t_{original}}{t_{sensor}+t_{query}}$	Original	Ours
18.92	32.41	32.41

5. CONCLUSION

In this paper, we propose the first encryption framework for the computation-intensive BP problem to be securely solved in the cloud with the data being efficiently collected using compressive sensing. The computational and communication costs in both the sensor side and query side of our scheme are analyzed to be lower than that of [1]. Moreover, we have proved the security of our scheme to resist COA. In the future, we will consider sensing noises in the context of compressive sensing and consider adversaries with more powerful capability beyond COA.

6. ACKNOWLEDGMENT

This work was supported by Ministry of Science and Technology, Taiwan, ROC, under grants MOST 102-2221-E-001-002-MY2 and 102-2221-E-001-022-MY2.

7. REFERENCES

[1] C. Wang, B. Zhangy, K. Reny, J. M. Rovedax, C. Wen Cheny, and Z. Xuy, "A privacy-aware cloud-assisted healthcare monitoring system via compressive sensing," *in IEEE INFOCOM*, pp. 2130–2138, 2014.

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

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

[4] M.Shoaib, 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.*, pp. 1–4, 2012.

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

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

[7] C. Wang, K. Ren, and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," *in Proc. 31st IEEE Intl Conf. Computer Communications (INFOCOM 11), IEEE Press*, pp. 820–828, 2011.

[8] J. Dreier and F. Kerschbaum, "Practical privacy-preserving multiparty linear programming based on problem transformation," *in Proc. IEEE Int. Conf. on Info. Privacy, Secu., Risk and Trust, Boston, USA*, pp. 916–924, 2011.

[9] C. Weeraddana, G. Athanasiou, C. Fischione, and J. S. Baras, "Per-se privacy preserving solution methods based on optimization.," *In IEEE Int. Conf. on Decision and Control*, pp. 206–211, 2013.

[10] Z. Xu, C. Wang, and K. Ren, "Proof-carrying cloud computation: the case of convex optimization.," *In IEEE Transactions on Information Forensics and Security*, vol. 9, no. 11, pp. 1790–1803, 2014.

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

[12] T. Blumensath and M. Davies, "Normalized iterative hard thresholding: Guaranteed stability and performance," *in IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 298–309, 2010.

[13] B. K. Natarajan, "Sparse approximate solutions to linear systems," *in SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, 1995.

[14] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit.," *in IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.