Privacy-Preserving Outsourcing of Magnetic Resonance Imaging Reconstruction

by

Zihao Shan

December 14th, 2019

A dissertation submitted to the
faculty of the Graduate School of
the University at Buffalo, The State University of New York
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science and Engineering

Abstract

Since the broad application of Magnetic Resonance Imaging (MRI) in the clinic, faster data acquisition against the physical and physiological constraints has been the subject of intense experimental research and theoretical work. Among these efforts, parallel imaging uses multiple receiver elements to exploit the explicit redundancy of the MR data, which can achieve an aggressive under-sampling rate and hence reduced acquisition time. Unwillingly, it leads to the time-consuming post-processing procedure of image reconstruction instead. In this thesis, we propose to leverage the benefit of cloud computing to address the speeding issue. Based on a pay-per-use manner, a clinic with low computational power can easily outsource the computational-intensive tasks of image reconstruction to Cloud Service Providers (CSPs). However, the privacy concerns with outsourcing patients' private data to public cloud servers are ignited and hinder those practitioners from enjoying the benefits of cloud computing.

Specifically, we explore the problem of privacy-preserving outsourcing the image reconstruction process in a calibration-less parallel imaging reconstruction method, termed simultaneous auto-calibrating and k-space estimation (SAKE). SAKE is one of the state-of-art algorithms in clinical implementation, which structure an under-sampled, multi-channel data-set in the k-space domain into a single data matrix. The reconstruction can then be formulated as a structured low-rank matrix completion problem. This thesis is seeking to enable a clinic to outsource the most computationally-intensive tasks in SAKE of the resource-abundant cloud servers, taking consideration of both the factors of security and efficiency. In particular, two different protocols are put forward in SecSAKE, with extra emphasis on privacy and efficiency, respectively.

The first protocol we propose can enforce a low-complexity matrix transformation over the data in the k-space domain on the clinic end. The clinic can then outsource the transformed data to the cloud and then harness the cloud server to perform iterative computation tasks. The corresponding security analysis shows that the outsourced MRI data are computationally indistinguishable under Chosen Plaintext Attack (CPA) under our assumption.

The second protocol provides an alternative model that pursues efficiency by leveraging the architectures of multiple non-colluding cloud servers instead. This protocol can primarily

reduce the computational complexity on the clinic side since it only needs to perform a one-round data transformation to retrieve the reconstructed MRI data. During the entire functional computation part, the clinic can stay offline with more flexibility. We conduct thorough privacy and efficiency analysis and extensive experiments over real-world image benchmark to evaluate the performance of the proposed designs. Compared with the original SAKE, the experimental results demonstrate that the proposed privacy-preserving mechanism can provide significant reconstruction time savings while achieving comparative performance on the quality of reconstructed images.

In conclusion, we close with a brief discussion of future research directions and then speculate about further approaches to preserving the data privacy in related medical applications, along with the potential obstacles on the road on achieving such objectives that we may encounter.

Acknowledgements

First and foremost, I would like to say thank you to my advisor, Kui Ren, and my co-advisor, Lu Su. It has been an honor to be their Ph.D. student. I sincerely appreciate all their contributions of time, ideas, and funding to make my Ph.D. experience plenteous and exciting. The joy and enthusiasm they have had for their broad areas of research were contagious and motivational for me. I am also thankful for the excellent exemplar they have provided as successful professors. The members of the UbiSec group have contributed immensely to my personal and professional time at the University at Buffalo, the State University at New York. I want to acknowledge honorary senior lab mate, Zhan Qin, who was here a couple of years ago. We worked together on the privacy-preserving computation outsourcing projects, and I very much appreciated his enthusiasm, intensity, and knowledge to provide high-quality advice and collaboration. Other past and present group members that I have had the pleasure to work with or alongside are Si Chen, Chaowen Guan, Sixu Piao, Zhongjie Ba, Tianhang Zheng, and the numerous rotation scholars who have come through the lab.

Regarding this dissertation, I would like to thank other committee members, Marina Blanton and Shambhu Upadhyaya, for their time, interest, and helpful comments. During my Ph. D. study period, they have been my instructors in the courses of (Advanced) Computer Security, which established foundation of my knowledge.

The secure computation outsourcing studies discussed in this dissertation would not have been possible without the expert help from Marina Blanton and Cong Wang, who both provide significantly helpful guidance on developing my writing skills and knowledge comprehension. They perform extraordinary patience throughout the entire process as I strived to improve my work.

In my later work of secure outsourcing MRI reconstruction, I am particularly indebted to Leslie Ying. She kindly introduced the privacy concerns in the field of MRI and gave me much advice on parallel imaging.

Finally, I want to thank my family for all their love and encouragement, without which surely none of this would have been possible.

Contents

Chapter 1

Introduction

Magnetic Resonance Imaging (MRI) is known as a medical imaging technology that uses magnetic fields and radio waves to create detailed images of internal organs and tissues. MRI has been frequently utilized in hospitals and clinics for medical diagnosis, staging of disease and follow-up due to its superiority over other medical imaging techniques such as CT, since the patient can avoid exposing himself to dangerous ionizing radiation. Not until the late 1990s, the data acquisition phase of MRI usually lasts so long that the patient's restlessness and then leads to non-diagnostic images [LKV13]. Instead of deriving at the patient's image directly, MRI initially collects data as an array of numbers representing spatial frequencies of the image, called k-space data. Simply under-sampling the k-space data may cause massive electrical power cost, peripheral nerve stimulation of patients, spatial aliasing of the image, or poor image resolution [DGGS12]. Therefore, the very initial question that motivated researchers to think about was:

How to shorten the lengthy acquisition time of MRI data?

Such a procedure has already been accelerated by a well-established under-sampling technique, called parallel imaging. Multiple receiver coils are concurrently applied in the scanning phase, each collecting under-sampled data in the spatial frequency domain, i.e., k-space domain. This shortens the data acquisition time, which enables the clinic to recover the desired image based on the acquired data and a particular reconstruction algorithm after the patient leaves the scanners. However, the following image reconstruction procedure usually takes quite a long period, which may still impact the prompt medical treatment. A recently proposed reconstruction approach, called SAKE [SLO$^+$14], retains the superiority of removing aliasing artifacts in dynamic MRI over other mainstream approaches, such as SENSE [PWSB99] and SPIRiT [LP10]. Nonetheless, its time cost for reconstruction can still be overdue in the urgent diagnostic case, mainly owing to the 1) expensive computation complexity,

2) large-scale imaging data, and 3) constrained clinical computing capability. This brought out the following question:

Are there any approaches that can further accelerate the MRI reconstruction time?

In the past decade, cloud computing has been widely recognized as an economical and practical solution for the resource-limited data owner to compute computational-intensive tasks. The clinic can take advantage of the available computational power on a pay-per-use basis. However, merely outsourcing the original data to the cloud server will bring about many data privacy challenges. Specifically, exposing the images produced by an MRI scan may reveal the patient's identity and physical condition. Besides, the cloud server may also fail to return a correct solution because of its misbehaviors (to save computational resources) or internal/external attacks, both of which will seriously compromise the diagnostic timeliness. Accordingly, a fine-grained outsourcing protocol addressing all the above security challenges while preserving the computational functionalities is highly demanded in practice. Additionally, the protocol should not impose massive computational overhead to the data owner, or else it is contrary to the original intention of outsourcing.

This topic directly pertains to the primary question we want to solve:

How should we design a privacy-preserving MRI reconstruction so that the clinic can efficiently obtain the reconstructed image?

In recent literature, many secure computation outsourcing schemes have been proposed. Some of the schemes apply cryptographic primitives to encrypt the data to achieve homomorphic computations. However, a general approach of secure computation outsourcing is currently far from practical for engineering applications, even for moderate-scale computations. Meanwhile, another branch of research proposes to targeting specific engineering computations. As a common feature of these schemes, they can achieve practical efficiency in specific engineering computations. Local transformation-based masking over original data will be first done before outsourcing. The parallel imaging reconstruction algorithm we study, unlike other common optimization problems, such as linear programming, involves the problem-specific computation, which cannot be learned from previous solutions in the literature. In this thesis, we propose a secure outsourcing scheme enabling the clinic to efficiently recover the missing entries in the under-sampled k-space based on the specific existing image reconstruction algorithm in the literature. In this chapter, we will go through the background of the proposed problem by first introducing the principle of the parallel imaging and then the commonly used image reconstruction components.

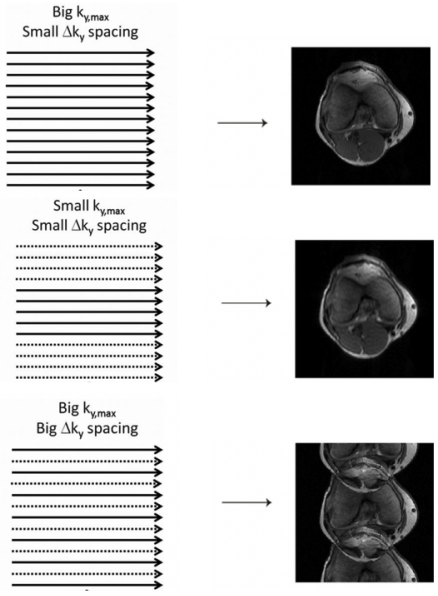

Figure 1.1: (a) Collecting data along closely spaced lines spanning large areas of k-space can result in high-resolution images covering the entire FOV (upper). (b) Reducing the frequency collected in k-space while keeping the spacing between adjacent samples may result in poor image resolution (middle). (c) Increasing the spacing between adjacent samples while keeping the frequency collected in k-space may result in poor image resolution (lower).

1.1 Parallel MR Imaging

Accelerating the data acquisition in MRI is more than a simple under-sampling operation. Parallel Imaging (PI), whose theoretical basis was in the late 1980s, has attracted many researchers from different expertises to explore the conceptual improvement in order to solve the major slow acquisition problem [LN07].

The basic concept of PI is introduced along with how it is implemented in clinical MRI scanner, which also serves as the basis of our following discussions.

1.1.1 K-space

In MRI, the imaging information is collected in the frequency domain, called k-space, instead of in the image domain. Obtained by spatially varying magnetic field gradients, this spatial frequency information is usually converted into the image by using Fourier Transform (FT). PI typically works by leveraging a receiver coil array to obtain a reduced amount of k-space data, which can still end up in successful imaging with acceptable quality.

The visual quality of the MR image is determined by two major terms, the resolution and the field of view (FOV). In the usual case of Cartesian sampling, the k-space data is sampled line-by-line in either 2D or 3D space. The resolution of the transformed image depends on the highest spatial frequency sampled in the frequency encode direction, while the FOV is correlated with the spacing between the samples in the phase encoding direction, as shown in Figure 1.1. Aiming to reduce the acquisition time, either the k-space data must be collected more quickly, i.e., reducing the highest spatial frequency sampled or the amount of k-space data collected must be decreased, i.e., increasing the spacing between the samples.

In the frequency encode direction, the sampling rate is determined by the analog-to-digital converter (ADC) used on the receiver boards of the scanner. In order to lower this sampling rate, the magnetic field gradients have to be significantly faster. This may incur at least three significant drawbacks as a consequence: 1) massive electrical power for adjusting the magnetic field, 2) harmful electrical currents in the patient's body leading to nerve stimulation, 3) unwilling image contrast of the final image.

On the other hand, the sampling rate in the phase encoding direction is determined by the magnitude of the k-space shift applied by the phase encoding gradient lobe. This is a much-preferred method to reduce the data acquisition time compared with less sampling in the frequency encoding direction. However, if one takes as large a step in k-space as possible, the FOV can be so small that the object to be scanned in not fully contained in the final image when it violates the Nyquist criteria [Sha49] and it results in spatial aliasing artifacts in the corresponding image, as shown in Figure 1.1.

The emergence of PI comes in under this scenario. PI technique involves multiple receiver coins, each with distinct sensitivity to the specific tissue, and hence they can provide a partial source of the imaging information in the spatial domain. Besides, a specific image reconstruction algorithm is required to remove the aliasing caused by the missing k-space data.

1.1.2 Image Reconstruction Algorithms

Different reconstruction algorithms may operate the coil data, target data in either image domain or k-space domain. There is also divergence on outputting the exact or approximate solutions [LN07].

Sensitive Encoding (SENSE) method [PWSB99] assigns different weights to each of the signal components according to the coil sensitivities at the locations of the two aliased pixels and builds up the system of equations. The assumption is that if the sensitivities of all the coils are previously known, the intrinsic signal components are solvable. In this method, the coil reference data and the target date to be operated are both in the k-space domain.

The approach of simultaneous acquisition of spatial harmonics (SMASH) [SM97] performs with the target data in the k-space domain, while the coil sensitivities are kept in image space this time instead. In SMASH, a prior estimation of the individual coil sensitivities is required, and the linear combinations of these sensitivities are used to generate k-space shifts like the magnetic field gradients in phase-encoding. With appropriate linear weights, it can generate composite sensitivity profiles fitting the appropriate spatial harmonic with certain accuracy. Similar methods such as Generalized-SMASH (g-SMASH) [BLH02], where the coil sensitivity profiles are represented directly in the k-space domain to provide a general description of the spatial properties of the coils. This enables a more flexible choice and position of the receiver coils.

Like g-SMASH, generalized auto-calibrating partially parallel acquisition (GRAPPA) [GJH+02] is also a pure k-space method but refers to a very different group of information. Instead of determining coil sensitivities to the relevant spatial harmonic, it leverages the full-sampled auto-calibration signal (ACS) as a kernel to solve the reconstruction weights approximately from a least-squares optimized fit to spatial harmonics.

1.1.3 Brief Discussion

The various parallel imaging methods developed to date have differed in using sensitivity information to eliminate aliasing artifacts due to under-sampling, but there are still potential issues regarding efficiency and accuracy. Reconstruction techniques such as SMASH and SENSE require that the user knows the reception profile of each coil element in advance. However, precise coil sensitivity measurements often require a separate calibration scan, which increases the total acquisition time. Besides, any inconsistencies due to small errors in motion or sensitivity estimates will show up as visible visual artifacts in the reconstructed image [SLO+14]. Auto-calibrating methods such as GRAPPA can mitigate the obstacles caused by explicit estimations of sensitivity information from the ACS, but taking the fix-sized ACS from a series of small images may still occupy a considerable portion of the total data reconstruction time.

1.2 Secure Computation Outsourcing: Overview

The exponential growth in the quantity of data generated nowadays has been drawing increasing attention. It is estimated that the amount of usable data created will be over 15 zettabytes by 2020, compared to 0.9 zettabytes in 2013 [Ads14]. This has led to an unavoidable challenge, as data owners and analysts have to figure out a way to properly store and effectively analyze the large-scale data. The storage capacity and computational capabil-

ity of processing large-scale data is still limited by the hardware and available memory of computer systems. Thanks to the advent of the technique of cloud computing, clients like individuals or companies can easily upload their data to a powerful cloud server which can then expedite the time of executing tasks on large-scale data and return the results to the clients or companies. Hence, the clients can take advantage of the available computational power on a pay-per-use basis, even if they cannot get direct access to supercomputers. A server with large computational abilities can share its spare computation resources out of a financial incentive.

Despite the significant benefits offered by the computation outsourcing paradigm, the primary obstacle to its wide adoption is the security issue. Essentially, the data involved in large-scale computation may contain valuable or sensitive information. Because of the physical isolation between the clients and CSPs, the clients are unable to judge whether a given cloud server is trusted or not before outsourcing their computation tasks, which could lead to critical concerns. A curious cloud server may inspect the data and deduce behavioral information of the residents. If the information is captured by a malicious party, it can give rise to many illegitimate consequences, including theft and possibly even terrorism. Therefore, in many cases, clients are unwilling to share the data with others including CSPs, even though the latter claim that their servers are completely secure and can be fully trusted.

Besides, a trusted cloud can also suffer from external threats. An external attacker can exploit technical vulnerabilities of a cloud service to gain further access to the data residing in the cloud. The outsourced data is intrinsically not secure from the point of view of cloud clients, and thus it is highly desired that the servers should not obtain any information about the data used in the computation. Practical privacy and security protection measures should be in place to ensure that cloud computing is appealing to a large range of clients.

Other than the challenges of data confidentiality, another crucial challenge is to ensure computation integrity. In the context of computation outsourcing, this is also called result verifiability (or checkability). In fact, clients should have the ability to check correctness of the results of outsourced computation. The cloud server assigned to a task may not honestly conduct the computation and may simply return an invalid result due to financial or timing reasons. For example, a cloud server can randomly generate the output based on the size and characteristics of the computational task. The computational cost of doing so is often very minor compared to the cost of running the computation itself. Moreover, an honest cloud server may undergo a software or hardware failure during its computation or data transmission. Many articles in the literature have addressed this challenge and carefully designed the verification procedure. It is required that the procedure should be substantially more efficient than the time of executing the original task. Otherwise, it will conflict with the

motivation for computation outsourcing. To achieve efficient verification, certain properties of the function used in the computation are often exploited to verify the returned solution.

Recent research has made steady advances in addressing these concerns, focusing on large-scale engineering and scientific computing problems and computationally intensive applications. One line of research originated from [GGP10] provides a general mechanism for secure computation outsourcing. The solution combined fully homomorphic encryption (FHE) [Gen09b] with the evaluation of Yao's garbled circuits [Yao86], achieving both data confidentiality and result verifiability. However, applying this general mechanism to large-scale computations is currently far from practical.

In contrast, another line of research aims to design solutions of immediate applicability to computation outsourcing. Such publications can be broadly classified into two groups: secure outsourcing of fundamental functions and secure outsourcing of computational tasks for specific applications. Constructions from the first group focus on commonly used mathematical operations or functions such as matrix operations, linear equations, and mathematical optimization problems. They typically exploit properties of the underlying mathematical functions to design specific protocols that can be utilized as building blocks in more complex computations. On the other hand, publications from the second group usually start from an individual application scenario and design an outsourcing solution for the entire task at hand.

1.3 Contributions of This Thesis

We advance our first protocol, named SecSAKE I, to address this challenge. Inspired by the Cadzow's algorithm [Cad88, Gil10], we can alternatively outsource some of the expensive sub-tasks, which are decomposed from the steps of solving this optimization problem. The primary sub-task that we focus on is singular value decomposition (SVD) over the complex field. In particular, the imaging data is masked by random unitary matrices by the data owner before it can securely harness the cloud server to run the iterative computation. Additionally, we consider data consistency projection as another resource-intensive sub-task in the case of non-Cartesian Sampling. Computing this sub-task requires the information of sampling mechanism, which the data owner may not want to reveal in some cases. SecSAKE I enables the data owner to conceal the sampling mechanism according to his preference. By a low-cost transformation, this sub-task can also be securely outsourced to the cloud server. In total, the required operations that the data owner performs in each iteration are of quadratic time complexity. Meanwhile, the data protected by SecSAKE I is computational indistinguishable under chosen plaintext attack under our assumption and verifiable against malicious cloud servers.

A noticeable issue of SecSAKE I is that the clinic has to keep online and participate in each iteration. Aiming to minimize its adverse effect, we propose another protocol SecSAKE II. In particular, we adopt the system architecture of multiple cloud servers. The data owner only needs to execute one round of data transformation in SecSAKE I at first. Then it can allocate the transformed data, key matrices, and different shares of imaging data to different servers. The servers can jointly process the iterations without the intervention of the data owner. Finally, the data owner can transform the obtained data from the cloud server to finalize the reconstruction, while none of the cloud servers can reveal the real imaging data.

Altogether, the contributions in this thesis can be summarized as follows:

- For the first time in the literature, we formalize the problem of outsourcing the undersampled in MRI reconstruction, i.e. low-rank matrix completion, to the cloud. Accordingly, we develop two secure and efficient solutions, SecSAKE I and SecSAKE II, addressing the issue of security and efficiency, respectively.
- In our first protocol, both imaging data and sampling mechanism are well protected as column-wise computational indistinguishable under chosen plaintext attack. The data owner can detect the misbehaviors of the malicious cloud server while only performing computations with quadratic time complexity in total, which is much lower than performing the original computations locally.
- In our second protocol, the data owner only needs to communicate with all the cloud servers by once and then it can receive the reconstructed k-space data. We also prove that the imaging data can be well protected when none of the semi-honest cloud servers can collude with each other.
- We implement and evaluate our design through the experiments over real-world image benchmark. It is shown that both of the protocols can be practically efficient and be well applied to clinical diagnose.

1.4 Thesis Outline

In **Chapter** 2, we first present the basic knowledges of the secure computation outsourcing according to a number of publications in the literature. In **Chapter** 3, we briefly review a specific parallel imaging reconstruction method is introduced, called Simultaneous Autocalibrating and K-space Estimation (SAKE). In **Chapter** 4, a system architecture and threat model of SecSAKE, along with the design goals of our system SecSAKE, is introduced. In **Chapter** 5 and **Chapter** 6, we propose the system design of our single-server and multi-server protocols, respectively. We analyze the privacy and computational complexity of both

protocols additionally. In **Chapter** 7, we demonstrate the conducted extensive experiments to assess the performance of our proposed protocols. In **Chapter** 8 we close with a brief discussion of future research directions.

Chapter 2

Secure Computation Outsourcing

In this chapter, we first describe the system architecture of outsourcing computation applicable to most of the related work. Then, we demonstrate typical security threats and some corresponding requirements. Following that, we briefly discuss the balance between security and efficiency. Lastly, we identify schemes similar but not identical to those using the secure computation outsourcing model at the end of the chapter.

2.1 System Architectures for Secure Computation Outsourcing

A common secure computation outsourcing architecture is illustrated in Figure 2.1. A typical asymmetric system involves two main different entities: a client C (or customer, data user, etc.) and a cloud server CS (or server). Due to the inability to carrying out the desired computation, a client C would like to outsource an expensive computational task Φ to a cloud server CS, who possesses massive computational power and significant storage capacity. Because the cloud server CS may not be fully trusted, the client C can locally apply a secret

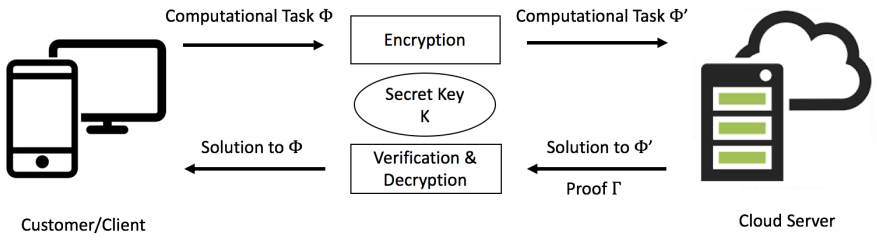

Figure 2.1: A system architecture with sequences of data/message flows.

key K to transform Φ into an encrypted problem Φ' to protect data privacy. Then the client C outsources the problem Φ', in place of Φ, to the cloud server CS and wants to receive the solution of Φ'. The cloud server CS operates on a pay-per-use basis and applies its resources to solve Φ'; it consequently returns the solution of Φ' to the client C together with a proof Γ. Note that the proof Γ is optional and is generated to allow the client to verify that the solution supplied by the CS is correct. After obtaining the returned solution from the cloud server CS, the client C recovers the answer of Φ using the secret key K. Moreover, the client C validates the correctness of solution by checking the solution itself or verifying the proof Γ. Based on the verification result, the client can choose to accept or reject the solution. Also, the cloud server CS may store an encrypted database related to the task uploaded by the client beforehand. The client C and the cloud server CS may engage in several rounds of interaction to solve the computation completely.

Note that some proposed system models may contain another independent party for a specific design objective: a party which may be responsible for results verification, data aggregation, or another function (e.g., see [HL05, QYR+14]). Two or more servers can work independently, e.g., for result verification. The servers can also work collaboratively. However, in either case they are generally assumed to be non-colluding. In some application scenarios, the client's role in the system can be decoupled into two distinct parties—the data owner and data user—to fit the actual situation. This case is mostly present in collaborative outsourced data mining, such as [YLX13]. To be more specific, we briefly introduce three specific variations, which are: 1) a system with three entities including a data owner, a service provider and a data user; 2) a system with two or more cloud servers executing a computation task outsourced by a client with no interaction between the two servers; and 3) a system with two or more cloud servers that collaboratively compute a task outsourced by a client. The cloud servers in 2) and 3) are usually assumed to be non-colluding. Although slightly different, the architectures still align well with the secure computation outsourcing model.

Schemes of the first type are usually presented in publications on outsourcing machine learning or bioinformatics tasks. The interaction often follows the data as a service (DaaS) model, where a data user requests the result of a certain computational task over a certain data set. The data owner is an entity who has complete control over the data and can authorize or deny access to portions of it. The data should be protected from both external service providers and in part from data users. In this system model type, the verification phase sometimes is not even applicable. Although data owner and data user should be considered separately when considering privacy protection in some schemes, these two entities can be viewed as a single entity, or client, in certain outsourcing scenarios such as [LC10].

Solutions of the second type, with two (or more) non-interacting cloud servers are usually found in publications on outsourcing cryptographic tasks. For example, earlier studies on outsourcing modular exponentiations assumed two non-colluding cloud servers to which a client outsources tasks on two correlated inputs without letting each of the servers know the original input. The outputs of the servers are used to verify each other's computation. These schemes are still in the scope of computation outsourcing because a heavy computation task is alleviated on the client side, which is the main difference between secure computation outsourcing and secure multi-party computation.

The third variant of the system model also involves two or more cloud servers which are now required to interact in the protocols. This model is widely used in studies of outsourcing multimedia-related tasks such as image feature extraction and also in outsourcing graph-based computations. A common strategy to outsourcing a task in this case is to split the original data into random matrices locally and distribute them to separate servers. The servers then compute the function on their own private input and interact to communicate their outputs. Additional processing in the form of comparison, aggregation or integration is securely conducted by an additional server or the client.

2.2 Security Threats in Computation Outsourcing

On a positive side, the use of computation outsourcing lowers some of the currently existing security risks for clients. For instance, clients who upload their data to a cloud and no longer store everything locally face a reduced risk of having sensitive information leaked in case their laptop is lost or stolen [Sen13]. However, new challenges and threats to information assets residing in the cloud are introduced because the data stored remotely is out of users' control. In the context of cloud computation, these security threats can be categorized as concerning *data confidentiality* (which refers both to the data used in the computation, i.e., computation input, and to the solution or output of the computation) and *computation integrity*.

From the client's perspective, threats to data confidentiality potentially come from the cloud service provider itself. Thus, it is often required that the cloud server should not gain any knowledge of the possibly sensitive client's data. To model the server's behavior, two types of adversarial models are usually considered: The first one is called the "honest-but-curious," or semi-honest, model [GMW87]. In this model, the server is assumed to faithfully follow the protocol's steps and thus correctly execute the computation and return the correct result to the client. Meanwhile, the server still tries to learn sensitive information about the client's data or the computed results in order to profit from it based on the nature of the computational tasks. The other, stronger security model treats the server as a fully malicious entity that can arbitrarily deviate from the prescribed computation. Then a

malicious server can deviate from the computation in the attempt to learn more information about the client's data than what an honest-but-curious server could or it may want to save its own computational resources (such as energy and time) and intentionally return an incorrect result (such a randomly chosen output) to the client. By returning a seemingly valid but wrong result, the cloud server hopes that the client will not be able to detect the cheating. A server who skips a portion of its computation is also sometimes referred to as a "lazy" server in the literature. Some publications also define a lazy server as one who attempts to lower its work and assumed to not intentionally disrupt the computation by investing more time that what is necessary to complete its computational task.

In addition to internal problems, a cloud server might suffer from external attacks. External threats include remote software or hardware attacks against the cloud infrastructure or application and social engineering. Successful break-ins into the cloud infrastructure both expose the data its servers handle to external parties and can compromise correctness of the returned result if the computation becomes corrupt. From the client's view, any detected problems (such as, e.g., incorrect output returned by the server) will be attributed to the server's misbehavior, regardless of whether they were triggered internally or externally.

Threats in the other direction—originated at the client and intended to harm the cloud server—are not typically discussed in the secure outsourcing literature. Publications that address threats of malicious clients corrupting workloads or attempting to steal information from tasks of other clients sharing the cloud's infrastructure are present in the literature.

2.3 Tradeoffs between Efficiency and Generality

2.3.1 Encryption Techniques for Outsourcing General Functions

When outsourcing computation to a public cloud, it is necessary to ensure confidentiality of the data used in the computation while maintaining relative efficiency of the computation. Data protection can be implemented via different mechanisms, and transformation or encryption techniques adopted in different schemes in the literature reflect the tradeoffs between efficiency and generality. Early homomorphic cryptosystems such as RSA [RSA78], El Gamal [ElG85] and Paillier [Pai99] can only support a single operation on ciphertexts such as addition, multiplication, or XOR and are called partially homomorphic. New cryptographic solutions for computation outsourcing became possible after Gentry's discovery of the first viable fully HE (FHE) which solved a long-standing major problem in cryptography and theoretical computer science [Gen09a, Gen09b]. FHE allows for an unlimited number of additions and multiplications to be performed directly on encrypted data, and hence it allows for evaluation of an arbitrary functionality on encrypted data, represented as an arith-

metic circuit. Notwithstanding, the overhead associated with the currently available FHE techniques makes their use for non-trivial computations infeasible. Aiming to address this performance issue, the area has experienced new research advances.

Yao's garbled circuit (GC) evaluation [Yao82, Yao86], proposed in the 1980s, provides a general approach for secure two-party computation. It is a constant-round protocol for securely evaluating an arbitrary function represented as a Boolean circuit. To preserve data privacy, a circuit has to be garbled anew for each evaluation and the approach is secure in the presence of a semi-honest garbler and malicious evaluator. Capitalizing on the fact that the evaluator is unable to predict (garbled) output without evaluating the function, Gennaro et al. [GGP10] combined the technique with FHE to enable secure and verifiable outsourcing of arbitrary functions. In this solution, the client supplies a garbled circuit once and it can be evaluated by a cloud server in encrypted form on multiple inputs in such a way that cloud's misbehavior is detected by the client. While providing a conceptually elegant and general secure outsourcing solution, the construction suffers from the significant computational burden brought by FHE and the need to decrypt ciphertexts inside FHE. Thus, performance of secure outsourcing of Boolean circuit evaluation would greatly benefit from optimized garbling schemes or improved FHE constructions, which are popular lines of research.

In particular, early research on developing efficient GC protocols focused on making implementions of MPC practical MPC, starting from the work of [MNPS04]. In 2012, Bellare et al. [BHR12] for the first time treated GC as an actual primitive, called garbling scheme, and provided a scheme based on a dual-key cipher (DKC). In this scheme circuit evaluation required one or two calls to a fixed-key blockcipher per gate, which was subsequently improved in [BHKR13] to use a single permutation call per gate and be compatible with other efficiency improvements such as "free" XOR gates and garbled row reduction. Recently, a sequential construction of GC has been proposed in [SHS+15] to achieve compactness and scalability.

The design of different FHE constructions can be divided into three different families based on the underlying mathematics [CCK+13]: (i) schemes based on ideal lattices [Gen09b], (ii) schemes over the integers [CCK+13] that rely on the hardness of finding an approximate greatest common divisor (GCD) of large integers, and (iii) schemes based on the Learning with Errors (LWE) and Ring Learning with Errors (RLWE) assumptions [BV11]. We review recent advances in each category.

The FHE scheme based on ideal lattices proposed by Gentry in [Gen09b] requires that the user first sets up a scheme that supports only a finite number of multiplications. This scheme is referred to as Somewhat Homomorphic Encryption (SwHE). To reduce the noise

accumulated through successive homomorphic multiplications, the encrypted message is to be re-encrypted. This is done by "squashing" the decryption circuit so that it can be handled within the capacity of the SwHE scheme. This operation allows us to obtain a new ciphertext with a reduced amount of noise that encrypts the same plaintext, and this process is called bootstrapping. As a result, the construction can support an unbounded number of homomorphic operations with bootstrapping applied as needed. To improve performance of this scheme, Gentry and Halevi [GH11b] significantly reduce the asymptotic complexity of key generation for the underlying SwHE and introduce a batching technique for encryption. They follow the work of Smart and Vercauteren [SV10] who also built an FHE from a SwHE scheme with small ciphertext and key sizes using a specific type of lattices that can be represented by integers. Meanwhile, a new method for building FHE is given in [GH11a]. The authors propose a hybrid scheme of SwHE and a multiplicatively HE scheme to eliminate the need for the squashing step. Early FHE constructions following Gentry's work were lattice-based. The challenges of schemes in this category are large key and ciphertext sizes. Lattice-based cryptography also contributed to advances in other cryptographic areas, including a new framework for constructing an attribute-based encryption scheme [BGG$^+$14] and a homomorphic signature scheme [GVW15].

The schemes over the integers also follow the construction framework of the initial proposal by Gentry to achieve conceptual simplicity. The first scheme of this kind was proposed by van Dijk et al. [VDGHV10], to which we refer as DGHV. The security of the construction is based on the difficulty of finding an approximate integer GCD and their SwHE scheme uses only integer arithmetics for homomorphic addition and multiplication. A follow-up optimization work by Coron et al. [CMN11] stores key elements in a new form which allows the size of the public key to be reduced, while ensuring semantic security under a stronger approximate GCD assumption. The key size is further reduced in [CNT12] by using a compression technique and a technique called modulus switching is introduced which allows bootstrapping to be eliminated. These two schemes exhibit performance similar to those of existing lattice-based schemes. Lastly, the work of Cheon et al. [CCK$^+$13] extends the DGHV scheme with the capability of using several plaintext bits in a single ciphertext while providing semantic security relying on the hardness of the approximate GCD.

The logic for constructing LWE-based FHE schemes was proposed in [BV11]. After building a SwHE scheme from LWE, one can apply a key-switching technique to control the dimension expansion of a ciphertext that results from homomorphic multiplication of other ciphertexts as well as a modulus-switching technique to manage the noise. The iterations of these two techniques result in an FHE scheme. Construction of RLWE-based FHE schemes follows a similar process. Batched RLWE-based schemes were introduced in

[GHS12b] and [BGV12] to reduce the cost of homomorphic evaluation to be polylogarithmic (in the security parameter). The latter is known as the BGV cryptosystem and is more efficient than the Gentry's initial proposal because of the absence of bootstrapping technique, which is replaced by a technique called modulus switching. However, we note that all of the constructions in this category can only provide evaluations of circuits of a polynomial (in the security parameter) depth, also known as leveled FHE.

As of time of this publication, LWE- and RLWE-based FHE has still been primarily of theoretical interest. Because modulus switching operations in BGV are still not within the reach practicality, Brakerski [Bra12] presented a scale-invariant scheme without switching the modulus. Consequently, Gentry et al. [GW13] proposed a relatively simple FHE scheme based on LWE—known as the GSW cryptosystem—where matrix addition and multiplication are used to represent homomorphism. It was later shown in [BV14] and [ASP14] that GSW achieves polynomial-factor growth in the error rate, which is significantly lower than the quasi-polynomial growth rate in concurrent work [ASP13]. In addition, a variant of multi-key FHE based on a variant of the NTRU computational problem was proposed in [LATV12], called the LTV scheme. This scheme enables computation on values encrypted under multiple and unrelated keys. Lastly, Bos et al. [BLLN13] showed how Brakerki's noise-management technique [Bra12] could be applied to the multi-key LTV FHE scheme [LATV12]. Note that most of the advances in FHE cryptosystems based on LWE and RLWE can be ported to FHE constructions over the integers. For example, instead of using modulus switching for controlling noise in schemes over the integers, Coron et al.[CLT14] apply the technique of [Bra12] to reduce the growth of noise to linear in the multiplicative depth of the evaluation circuit.

Another line of work on improving efficiency of FHE schemes focuses on optimizing SwHE constructions. Naehrig et al. [NLV11] implemented a SwHE scheme based on RLWE which enjoys relative efficiency. By converting between different message encodings in a ciphertext, they were able to further optimize application-specific realizations. In addition, a parallel computing technique called Single Instruction Multiple Data (SIMD) was used in [GHS12a] to improve the speed of FHE operations. It was targeted at the main bottleneck in bootstrapping which requires homomorphically reducing one integer modulo another. Another work [SV14] designed a SwHE scheme which supports SIMD as well as computation in finite fields. It used SIMD operations for parallel re-encryption, which led to a considerable performance improvement. Furthermore, the work [PV15] focused on minimizing the number of bootstrapping operations so as to improve computation time using FHE. A number of software [GHS12c, HS15, HF17] and hardware [PG13, WHC$^+$12, KGV16] implementations of SwHE/FHE schemes have emerged, aiming to achieve a significant performance improvement

Table 2.1: Performance of AES encryption using FHE on the same platform (Source: [DDS14]).

Technique	Time per Block	Technique	Time per Block
GHS (SIMD) [GHS12c]	2400 s	Accelerating NTRU [DDS14]	7.3 s
NTRU [DHS14]	55 s	HElib [HS14]	2 s

of the available constructions.

Using FHE for evaluation of client's outsourced tasks results in representing computation as a circuit, and the approach supports any desired functionality. As seen from Figure 2.2, this type of outsourcing resides at the bottom of the computational hierarchy. Even though a number of breakthroughs have been recently made to improve performance of FHE, secure computation outsourcing based on FHE and Boolean circuits remains to be impractical for most applications (such as those specified at higher levels of the computational hierarchy), especially when input datasets are large.

Note that comprehensive benchmarks for comparing performance of software implementations of FHE schemes are currently not available [BFG+17]. A number of publications, however, report on performance of evaluating AES-128 using FHE. For example, Gentry et al. [GHS12c] was the first to provide a software implementation of FHE based on a modified HE scheme of [BGV12] using the techniques in [SV14], [GHS12b], and new optimizations. A customized implementation of the LTV scheme is available in [LATV12]. Another work [DDS14] builds a library that uses discrete Fourier transform to support FHE evaluation of AES. This implementation brings the evaluation time down to slightly over 7 seconds per block. Shortly after, HElib [HS15]—a software library that implements homomorphic evaluation of AES encryption—brings the amortized per-block time down to about 2 seconds. Table 2.1 gives a brief comparison of performance of different FHE implementations evaluating AES encryption. Note that in the secure outsourcing scheme of [GGP10], homomorphic decryption of an AES circuit (or a similar operation) will need to be performed for each gate of the Boolean circuit representing the outsourced task, effectively multiplying the times in the table by the number of circuit gates.

These general solutions, such as those employing FHE, aim to support any computable function, which can be represented as an arithmetic or Boolean circuit. In such schemes the server will be able to evaluate the circuit in encrypted form and achieve proper data protection. The main conclusion that we can make is that this general approach of secure computation outsourcing is currently far from practical for engineering applications, even for moderate-scale computations.

2.3.2 Encryption Techniques for Outsourcing Specific Functions

The above observation motivates scholars to seek efficient solutions for securely outsourcing specific types of computation, operating at higher levels of the computational hierarchy. Thus, schemes on the other side of spectrum aim to provide an efficient outsourcing strategy for computing a specific task over large-scale data, but the developed solution may not generalize to any other type of computation. Atallah et al. [APRS02] for the first time investigate the random transformation techniques tailored to several specific scientific or engineering computation tasks. Despite the wide scope of discussion over multiple applications, the protocol would lead to privacy leakage. Afterwards, many studies have been focusing on designing schemes on privacy-preserving outsourcing *fundamental mathematical functions*, such as matrix multiplications [ZB14][LLHH14], matrix inversion [LLH$^+$13], matrix factorization [DZL16] [ZL16], linear equations [WRWW13][CHL$^+$15] [SLCL15][SLC$^+$17], etc. However, these schemes are mainly on the computation over real field, which is not applicable to the MRI settings.

Regarding the secure outsourcing of the large-scale mathematical optimization problem, an initial work was done by Wang et al. [WRW11]. In their work, a practically efficient solution of secure outsourcing linear programming is designed while allowing result verification on different conditions. Later on, Nie et al. [NCL$^+$14] apply the sparse matrices to hide the sensitive data in linear programming for faster local preprocessing. Chen et al. [CXY14] leverage the pseudorandom generator to reduce the key size when outsourcing a reformulated form of linear programming. Liao et al. [LDSL16] propose a solution for secure outsourcing convex separable programming. Salinas et al. [SLLL16] provide a secure outsourcing scheme of the quadratic program by solving its Lagrange dual problem. A work with similar objective is [WZRR13], which offers a linear-programming based image recovery service.

We propose to categorize the available solutions in the literature based on the generality of the technique. We organize operations, from elementary to more complex, in a hierarchy of computational levels (first introduced in [WRWW13]) as shown in Figure 2.2 to systematically classify and organize these solutions. This is motivated by the fact that many constructions achieve similar levels of security and cannot be easily categorized based on the security properties they achieve, while using a hierarchy of computational levels allows us to arrive at a well-organized structure for presenting these solutions.

The general classification methodology is based on the following observation: any computational problem can be represented as computations at different computational levels, which can be organized in a hierarchy. A complex task at a top level of the hierarchy can be decomposed into simpler operations at lower levels of the hierarchy. Thus, secure outsourcing solutions for computations at lower levels can be used as building blocks for securely

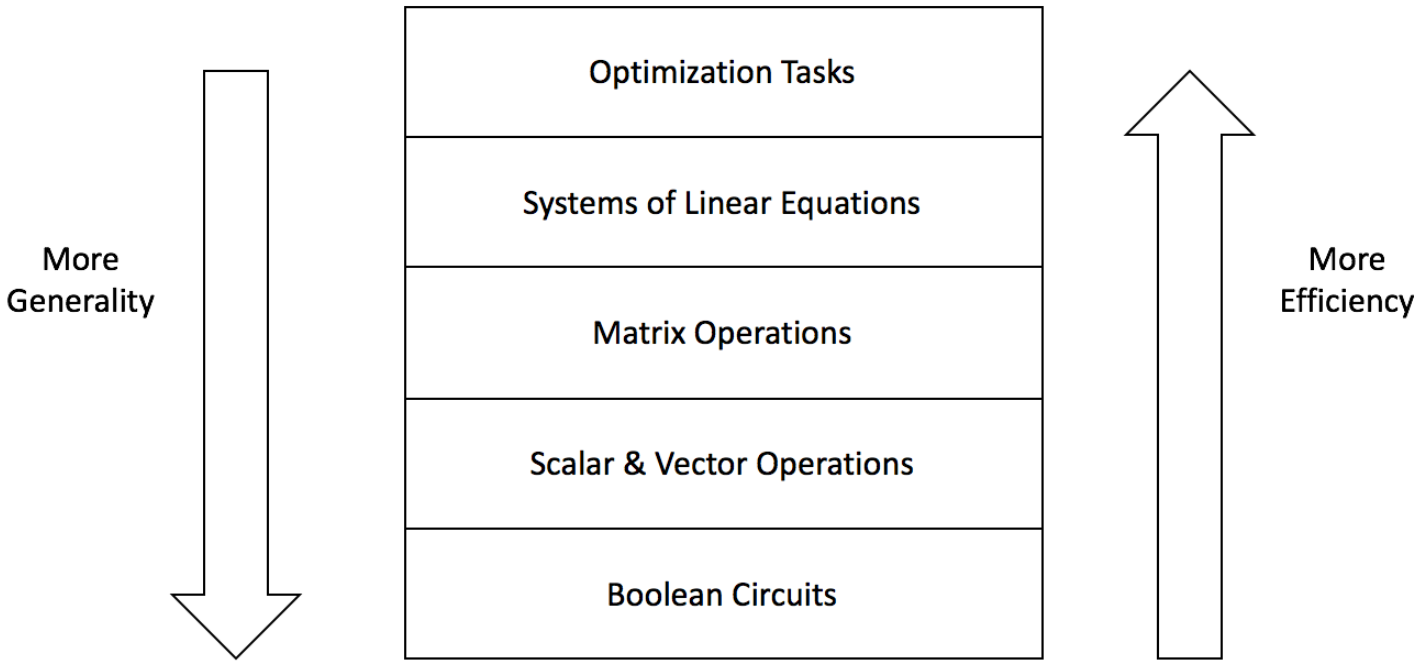

Figure 2.2: A hierarchy of computational levels.

outsourcing a computation at a higher level, potentially resulting in multiple solutions to the task at hand. We demonstrate this on the example of linear programming.

If there exist proper secure outsourcing protocols, the client can decompose its task Φ into a number of sub-tasks at lower levels of the hierarchy, transform each of them into the corresponding protected form, and re-assemble the result from the outputs of the sub-tasks that it receives. For example, instead of directly outsourcing its linear programming task, the client can separately outsource matrix operations or vector operations to a cloud server. This can often increase the client's work due to function decomposition and result assembly or it may also increase the number of interaction rounds.

As an observation related to function decomposition at the client side, we note that in some cases this will allow the client to hide the specifics of the function being outsourced. While throughout this article the task being outsourced is assumed to be known to the cloud and outsourcing of private function evaluation, there is potential for this type of function decomposition to provide a limited form of function privacy. For example, in the case of decomposing a linear programming task into small components, the cloud server might not be aware what exactly the client's task is.

2.4 Requirements for Secure Computation Outsourcing

The original computational problem Φ is to be locally transformed into Φ' by the client. As a result, only Φ' is accessible to the cloud server as the input. Then one of the major security requirements is that the cloud server cannot derive any sensitive or meaningful knowledge about client's data from Φ', which is known as *input privacy*. Furthermore, after evaluating problem Φ' and determining the corresponding output of the function, the cloud server should be unable to learn any information about the result of executing Φ itself, including any intermediate and final results. This is called *output privacy*. In the majority of the constructions that comply with the previously described architecture, input and output privacy is achieved by either data transformation or data encryption. Then to guarantee that the server learns not information about the client's data, the server's view of the outsourced data (in a transformed or encrypted form) should be computationally or statistically indistinguishable from randomly sampled values. Because of slight differences in the system architecture employed by some schemes, the privacy requirement might be formulated differently.

Verifying correctness of the result returned by the cloud server and ensuring that a true answer is obtained has been widely acknowledged as another crucial security requirement of computation outsourcing. This property refers to computation integrity and is called *checkability* [HL05] or *verifiability* [GGP10] in the literature. It requires that an incorrect output to problem Φ' returned by a malicious (or lazy) cloud server should pass the verification process

on the client side with a very small or negligible probability. Note that the checkability is occasionally not a necessary requirement in some publications due to the particularity of the functions or a weaker threat model.

As another major requirement for secure computation outsourcing, *efficiency* commonly refers to the client's ability to reduce its local computation, which serves as the main motivation for utilizing computation outsourcing. The savings are based on the difference between the effort required for executing the computational task Φ locally and the effort involved in using computation outsourcing, typically measured theoretically or in some cases empirically. Computation associated with computation outsourcing involves preparation of Φ' including data encryption, result recovery including decryption, and output verification. The cost of input and output transformation or encryption/decryption depends on the data size and the employed encryption techniques, which are often symmetric.

Lastly, if the server conducts the computation faithfully and sends the correct results to the client, the client should be able to successfully verify and recover the result of the computation. This property is known as *correctness* [Sio08].

2.4.1 Input and Output Privacy Protection

A fundamental security property sought of secure computation outsourcing schemes is that of privacy protection of the data handled by cloud servers. This was formulated as *input privacy*, i.e., inability of a cloud server to derive information about the input data that it receives (in a protected form) and uses in the computation and also *output privacy* that specifically refers to the server's inability to learn information about the result of the computation. To meet this security objective, ideally the server should be unable to learn any information about the data it receives or computes. This is formally modeled as the server's inability to distinguish outsourced data from randomly generated data of the same size using statistical or computational notion of indistinguishability. This general formulation is applicable to constructions with a single server or multiple servers, including the setting where the servers communicate.

Many constructions in the literature meet this definition of data privacy. Examples include [Moh11] and [ZB14]. There are, however, other publications that relax this definition of data privacy and allow some data leakage about private data in order to improve performance of their constructions. We outline common formulations of data privacy and attacks on data privacy in the context of computation outsourcing.

Local data transformation or encryption prior to outsourcing is a necessary step for achieving input/output privacy. Because constructions based on cryptographic techniques and those based on custom transformations to fit the needs of specific problems often have significantly

Table 2.2: Comparison of two different outsourcing strategies.

Technique	Representative work	Advantages	Disadvantages
Transformation-based	[WRW11] [CXLC14]	More efficient	Problem-specific, less security
Transformation-based + partially HE	[WRWW13] [ZB14]	More secure and general	Inefficient

different characteristics, in what follows, we analyze these two categories of techniques separately. Furthermore, because FHE can be used for general computational tasks with strong privacy guarantees, we will only discuss approaches that utilize partially HE.

We start with a rough classification of the publications based on the techniques they employ which is given in Table 2.2. Transformation-based techniques usually use a key in the form of random invertible matrices or random numbers to hide the original dataset. Early publications such as [APRS02] deems it impractical to provide a systematic analysis of data privacy of transformation-based techniques because data disguises are problem-dependent. Furthermore, early attacks against privacy of outsourced data, such as the *statistical attack* of [APRS02] which traverses the output of specific random generators to analyze the transformed data, are considered weak and can be defeated using complex probability distributions or by refreshing the key.

More recently, it became common to see the data privacy property formulated as security under a *ciphertext-only attack* with certain constraints. Note that because of the non-interactive nature of computation outsourcing, it is meaningful to formulate data privacy using encryption terminology. This property requires that, given encrypted input, it is infeasible for the server to obtain or deduce any information about the original data. In works that adopt this definition, it is either shown that the distribution of the transformed data is statistically close to that of a data sampled uniformly at randomly, or the server's advantage in distinguishing the transformed data from random values is negligible. An example of constraints that may come with this formulation of security can be found in the work of [WRWW13]. Its focus is on iterative linear equation outsourcing, and input privacy can be achieved if the number of iterations is limited by a certain value.

Taking the transformation technique of [WRW11] as an example of constructions secure against a ciphertext-only attack, we can see that construction adds randomly chosen vector $\mathbf{r}$ to input $\mathbf{x}$ as a masking layer. This renders different forms of analysis attacks ineffective when only a ciphertext is available to the attacker. It has been later shown in [WRW16]

that if the entries of $\mathbf{r}$ are uniformly chosen from interval $\mathbf{I} = [-2^{\kappa}, 2^{\kappa}]$, where κ is a security parameter, the statistical distance between $\mathbf{x} + \mathbf{r}$ and a vector $\hat{\mathbf{r}}$ randomly and uniformly sampled from $\mathbf{I}$ is a negligible function. In other words, the server's views $\mathbf{x} + \mathbf{r}$ and $\hat{\mathbf{r}}$ are statistically indistinguishable. Hence, protection of individual sensitive matrix and vector values (e.g., $\mathbf{b}$ and $\mathbf{b}' = \mathbf{Q}(\mathbf{b} + \mathbf{xr})$) is achieved.

The above analysis is based on the assumption of that a random transformation is used only once. Later it has been observed that using the same random transformation multiple times may allow the attacker to accumulate specific plaintext/ciphertext pairs until the original matrix could be recovered by solving a linear equation or certain information about original matrices can be learned. For example, several schemes lead to data leakage if users use a single matrix $\mathbf{M}$ to hide their data vectors $\mathbf{x}$ as $\mathbf{Mx}$. If a client applies the same transformation matrix $\mathbf{M}$ as the secret key to hide another data vector $\mathbf{y}$, the server can recover information about the vector in the form of $\mathbf{y}^{-1}\mathbf{x}$ by multiplying $(\mathbf{My})^{-1}$ with $\mathbf{Mx}$. Therefore, under the assumption that the same transformation key can be used to transform different inputs, it becomes meaningful to consider security under a *known-plaintext attack*. Given knowledge of the plaintext, the objective of the attacker then becomes to recover the secret key. It can be easily shown that many schemes from the literature become vulnerable in this security model. This is often because they apply addition and/or multiplication of random matrices to protect the original data matrix and these operations are distributive.

The above definitions of ciphertext-only and known-plaintext attacks capture the notion of no information about the input being revealed to the server. A number of publications relax these definitions and allow the server to learn certain information about the data. For example, transformations that multiply input by random matrices can reveal the number of zero elements in the original data or even the location of zero elements in the input matrix, which is proposed in [YLW+16]. There are also solutions that preserve certain properties of the input matrix after the transformation, such as retaining the matrix sign or rank, which also amounts to information leakage. Note that the property of rank-preserving is often necessary when encrypting the coefficient matrices in linear systems and linear programming. These constructions would not satisfy the requirements of security of indistinguishability but is still secure against ciphertext-only attack.

To better preserve input privacy, several studies apply encryption schemes with limited homomorphic properties. For example, [Moh11] design computation over ciphertexts encrypted using different HE techniques to outsource matrix multiplication. Because the server receives multiple transmissions from the client, data privacy is defined with respect to all the messages the server sees during the protocol, as we originally define using the notion of indistinguishability. Other publications such as privacy-preserving sequence comparisons in

[AL05] and ridge regression in [NWI+13] utilize multiple servers and would fall under the same formulation of data privacy as above. Similar levels of privacy protection can be found in [ZB14], and many others. The properties of constructions for outsourcing fundamental functions and their detailed comparison can be found in Table 2.3. Table 2.5 explains some abbreviations and notation used in Table 2.3.

Most of the outsourcing schemes we have discussed also aim to protect output privacy. Because the transformation process is often symmetric, output privacy in these schemes is usually achieved at the same security strength as that of input privacy. In contrast, there are also a number of existing schemes that only protect input privacy or provide a limited form of output protection. For example, information about output may be exposed to a cloud server in the form of revealing records that matched the client's query. This is present in outsourcing biometric-related and data mining related tasks, such as searching a set of iris codes for a match [BA12] and searching the k-nearest neighbors over the outsourced database [ESJ14].

As a brief summary, we observe the following trends: 1) For transformation-based schemes to achieve the security guarantees put forward in the respective publications, the masking material in the form of vectors and matrices should be randomly chosen anew for each input. 2) As encryption aids privacy protection, it is easier to achieve input privacy if semantically secure encryption is used. 3) When multiple servers are utilized, input privacy cannot be guaranteed if they collude.

2.4.2 Ensuring Correctness through Verification

Checkability, or verifiability, is one of the key requirements of secure computation outsourcing. As previously described, it refers to the user's ability to detect server misbehavior. Current designs can be divided into two cases: the ones with deterministic verification and with non-deterministic verification.

The designs with *deterministic verification* can always provide deterministic checkability. One example can be found in [WRW11] which deals with linear programming computation. Compared to many outsourced tasks that correspond to fundamental functions whose output falls in a single case, the LP problem may not have an optimal solution. This raises the challenge of designing the verification procedure. Besides verifying the returned optimal solution, the user has to ensure that the 'infeasible' or 'unbounded' output is honestly reported by the cloud server. Hence, the server is instructed to return a proof Γ that includes different options for different cases. In the case of feasible LP, a dual optimal solution should be included in the proof. Then the user can validate correctness of the solution if both the primary and dual objective values are equal. In the case that the server wants to show

Table 2.3: Protocol design choices in outsourcing of fundamental functions.

Task	Scheme	Threat model	Technique	Interaction	Verification dependence	Security strength
Matrix multiplication	[BA08]	Malicious	TF + HE	Once	p	COA(NS)
	[AF10]	Malicious	SS	Vrf	t	IND
	[LLHH14]	Malicious	TF	No	l	COA(NS)
	[Moh11]	Malicious	TF + HE	No	Deterministic	IND
	[ZB14]	Malicious	TF + PRF + HE	No	Deterministic	IND
		Semi-honest	TF + PRF + HE	No	Deterministic	IND
	[FG12]	Malicious	PRFC	No	Deterministic	IND
Matrix inversion	[LLH+13]	Malicious	TRF	No	l	COA(NS)
Nonnegative matrix factorization	[DZL16]	Malicious	TF	Vrf	Deterministic	COA(NS)
Matrix eigen-decomposition	[ZL16]	Malicious	TF	No	l	COA(NS)
Matrix determinant	[LLHL15]	Malicious	TF	No	l	COA(NS)
System of Linear Equations	[WRWW13]	Semi-honest	TF + HE	Solv, Vrf	Deterministic	COA
		Malicious	TF + HE	Solv	l	COA
	[CXY14]	Malicious	TF	No	Deterministic	COA(NS)
	[CHL+15]	Malicious	TF	No	Deterministic	COA(NS)
Linear programming	[WRW11]	Malicious	TF	No	Deterministic	COA
	[NCL+14]	Malicious	TF	No	Deterministic	COA(NS)
Quadratic programming	[SLLL16]	Malicious	TF	Yes	Deterministic	IND
	[ZL15]	Malicious	TF	No	Deterministic	COA(NS)

infeasibility of the original task, it has to solve the auxiliary LP problem. Then both the optimal solution of the auxiliary problem and its proof of optimality should be included in the proof Γ. The unbounded case is treated similarly.

The design with *non-deterministic verification* can guarantee correctness with certain probability. The probability of a client accepting a wrong result can be adjusted using security parameters to balance performance and the probability of error. One typical non-deterministic verification example is the design of [WRWW13], where computation of the solution to a linear equation proceeds in iterations until the convergence criterion is satisfied. In its verification phase, to check correctness of $\mathcal{L}$ iterations in a batch, the client randomly selects $\mathcal{L}$ l-bit numbers to be used as the coefficients in a linear combination. Correctness of the received answer in $\mathcal{L}$ iterations can then be tested by checking whether the linear combination of the returned vectors with the specified random coefficients equals to the value that the client expects. The equality always holds when the output in each iteration is correctly computed. It can also be proved that a wrong result even with only one incorrect value in the received vector will be undetected with probability less than 2^{-l}.

The design of a verification process is highly related to efficiency. A poor verification mechanism can be costly to the client and result in computation comparable to executing the task itself. Hence, a careful design of verification is quite needed and was developed in many outsourcing schemes as we discussed. Although non-deterministic verification shows its weakness on the accuracy of misbehavior detection compared to deterministic approaches, it can be more flexible in practice when efficiency is the most important metric. Furthermore, non-deterministic verification often provides sufficient guarantees in the long run when the dataset is large enough.

2.5 Performance Evaluation

For a given computational task, client's performance can be evaluated by comparing the encryption/decryption or transformation overhead as well as task verification cost to computation overhead of performing the original task. Because of the original motivation of outsourcing large-scale computation, performance speed-up is a necessary requirement for outsourcing schemes. Thus, Table 2.4 summarizes performance speed-up for different schemes, except those that use FHE, whose encryption overhead can be very high (notation can be found in Table 2.5). For example, in [LLH+13] and related schemes, the transformation uses a random matrix for hiding the original data. Client's computation is dominated by several matrix additions and matrix-vector multiplications, which take $O(n^2)$ time. HE is another common data protection technique used in secure computation outsourcing constructions. For example, it can be found in the design of [WRWW13], where a client performs a matrix-

Table 2.4: Efficiency comparison of schemes for outsourcing fundamental functions.

Task	Scheme	Enc/Dec time	Task time	Verification time
Matrix multiplication	[BA08]	$O(n^2)$	$O(n^\rho)$	$O(n^2)$
	[AF10]	$O(t^2n^2)$	$O(tn^\rho)$	$O(n^2)$
	[LLHH14]	$O(mn+ns+ms)$	$O(msn)$	$O(msl)$
	[Moh11]	$O(n^2)$	$O(n^\rho)$	$O(n^2)$
	[ZB14]	$O(mn+ns+ms)$	$O(msn)$	$O(ms)$
		$O(mn+ns+ms)$	$O(msn)$	$O(1)$
	[FG12]	$O(\max(m,n)s)$	$O(mns)$	$O(\max(m,n)s)$
Matrix inversion	[LLH+13]	$O(n^2)$	$O(n^\rho)$	$O(ln^2)$
Nonnegative matrix factorization	[DZL16]	$O(\max(m,n)^2)$	$O(imnr)$	$O(mnr)$
Matrix eigen-decomposition	[ZL16]	$O(n^2)$	$O(ln^2)$	$\Omega(n^3)$
Matrix determinant	[LLHL15]	$O(n^2)$	$O(n^\rho)$	$O(ln^2)$
System of linear equations	[WRWW13]	$O(in+n^2)$	$O(n^\rho)$	$O(n^2)$
		$O(in+n^2)$		$O(ln^2)$
	[CXY14]	$O(n^2)$		$O(n^2)$
	[CHL+15]	$O(\lambda n^2)$		$O(n^2)$
Linear programming	[WRW11]	$O(n^\rho)$	$\Omega(n^3)$	$O(n^2)$
	[NCL+14]	$O(\lambda n^2)$		$O(n^2)$
Quadratic programming	[SLLL16]	$O(\max(mn,n^2))$	$\Omega(n^3)$	$O(\max(mn,n^2))$
	[ZL15]	$O(n^2)$		$O(n^2)$

vector multiplication before encrypting the elements of the original matrix. If the number of computation iterations is less than the data size, the client's work is $O(n^2)$ including $O(n^2)$ encryptions.

To make a fair comparison, another issue that should be taken into consideration is the external memory I/O operations when the data is large and cannot reside in local memory, which was first discussed in [SLCL15]. In [LLH+13] and similar schemes, the cost of encryption is $4n^2$ I/O memory operations, decryption requires another $2n^2$ memory I/O operations, while the verification procedure needs another n^2 memory I/O operations. Hence, the memory I/O usage of this design is around $7n^2$ in total. In the scheme of [WRWW13], if all interactive operations occur within memory, the memory I/O usage is also around $7n^2$. For comparison, the scheme in [SLCL15] for outsourcing the same task uses vector operations on the client side instead of matrix-vector operations, and as a result achieves better memory I/O usage requiring $4n^2$ memory I/O operations, without considering verification.

Table 2.5: Abbreviations and notation used in Tables 2.3 and 2.4.

Abbreviations		Notation	
CPA	chosen-plaintext attack	n	dimensions of a square input matrix or the number of columns in the first non-square matrix
COA	ciphertext-only attack		
IND	computationally indistinguishable	m	the number of rows in the first non-square matrix
		s	the number of rows in the second non-square matrix
Vrf	verification	t	secret sharing threshold
TF	transformation-based	ρ	the power in the asymptotic complexity of matrix multiplication
HE	homomorphic encryption	l	the number of iterations in the verification process
PRF	pseudorandom function	p	modulus size used in homomorphic encryption
PRFC	PRF with closed form efficiency	r	dimension parameter for matrix factorization
SS	Shamir's secret sharing	i	the number of iterations needed in the computation
Solv	solving computational task	λ	the upper bound on the number of non-zero elements in each matrix row
NS	not provably secure	p	the number of rows in the constraint matrix for optimization problems

2.6 Overview of Related Topics

According to the system architecture and requirements described in this chapter, two classes of related work are identified, on which we comment below.

Secure multi-party computation (MPC) allows for cooperative evaluation of an arbitrary function by multiple parties, taking each party's private data as the input and preserving data confidentiality throughout the computation. The result of the computation is returned to an agreed-upon set of participants according to the specified functionality. The earliest general solutions for secure function evaluation were given by Yao for the two-party setting secure in the presence of semi-honest participants [Yao82] and by [GMW87] for the multi-party setting secure against malicious participants. The general architecture of secure MPC typically assigns symmetric workloads to the computational parties, and the data contributed by every party resides in the system in a protected form. This setup is typically not suitable for achieving the goals of outsourcing large-scale computations, where a client wishes to improve the speed of running computational tasks.

Delegating computation with cheating detection assumes a network composed of several computational devices of different computational capabilities that interact with each other. It allows weak devices to delegate their computational tasks to more powerful devices, which is similar to the architecture of securely outsourcing large-scale computations. Several early

results include provisions for detecting server's misbehavior. However, they permit the server to have access to the data used in the delegated tasks in the clear, which violates one of the main security requirements of secure computation outsourcing. These publications include [GKR08, GM01, DMJ10], and others. In addition to these early studies, several more recent results on delegating specific computations such as those in [CLM+14, XAG17] treat both computation verification and data privacy so that they can be classified as secure outsourcing schemes. For more detailed readings, please refer to [SRBW18].

2.7 Summary

In this chapter, we give a systematic overview of existing solutions for securely outsourcing large-scale computations. Efficiency of client's computation and proper data confidentiality protection from the cloud server conducting the task are the two most important goals that prominent schemes from the literature aim to achieve. Additionally, verifiability of the computed result becomes an essential property for state-of-the-art secure computation outsourcing solutions in the presence of servers who are not fully trusted. We also identified tradeoffs between security and efficiency among different application domains.

Chapter 3

Simultaneous Auto-calibrating and K-space Estimation

In this chapter, a specific parallel imaging reconstruction method is introduced, called Simultaneous Auto-calibrating and K-space Estimation (SAKE) [SLO+14]. According to the introduction we present in Chapter 1, the traditional image reconstruction method suffers from several drawbacks. SAKE essentially works on pure k-space data without leveraging the information of the coil sensitivities, which can reduce the data acquisition time from the separated scans and hence avoid the significant imaging alias caused by the inconsistency due to motion or small errors in the sensitivity estimation. Furthermore, SAKE, as a calibrationless method, also shows its superiority over other pure k-space data methods using sufficient ACSs for accurate calibration, such as GRAPPA [GJH+02], SPIRiT [LP10], ESPIRiT (an eigenvalue approach to autocalibrating parallel MRI) [ULM+14].

3.1 Theory in SAKE

In general, SAKE jointly considers the data of the multiple channels by arranging the acquired data into a single, structured matrix, called data matrix. This structured matrix also is known as a low rank matrix due to the linear dependency of data. The image reconstruction problem is then transformed into a structured low-rank matrix completion problem and formulated as a constraint optimization problem. Below we present the brief summary of the projection-onto-sets type algorithm the paper [SLO+14] adopted to iteratively solve the problem.

3.1.1 Generating Data Matrix

In this phase, the data acquired from the multiple coils is integrated into the single data matrix **A**. Initially, all the vacant entries in coils are filled with zeros. After that, a square

sliding window with the chosen size of $w \times w$ moves across the data within an $N_x \times N_y$ coil. Each value in the window is then projected to a row vector, which is later attached to the data matrix. The above operations are repeated over each of the N_c coil. Hence, the constructed data matrix will contain $w^2 N_c \times (N_x - w + 1)(N_y - w + 1)$ entries. Specifically, the linear operator to generate the data matrix $\mathbf{A}$ is defined as H_w:

$$H_w : \mathbb{C}^{N_x \times N_y \cdot N_c} \rightarrow \mathbb{C}^{(N_x - w + 1)(N_y - w + 1) \times w^2 N_c} \tag{3.1}$$

As shown in Figure 3.1., the generated data matrix contains many duplicate entries, i.e. the same entry in k-space dataset is anti-diagonally projected to a set of indexes in $\mathbf{A}$. In particular, the data matrix is structured as a Hankel structure block by block. It has been shown in [SLO+14] that the data matrix, created with an appropriate window size, should have the property of *low rankness* if generated from a fully sampled k-space dataset. This is because, under the reasonable assumption that coil sensitivities have compact k-space support, a data matrix in block Hankel form can be structured to become a rank-deficient matrix for an appropriately chosen window size.

3.1.2 Low-rankness Projection

The objective in this stage is to separate the desired data and noise by projecting the data matrix onto a more compact subspace. The Cadzow's algorithm [Cad88] adopted in SAKE proposes to apply the technique based on Singular Value Decomposition (SVD) in order to enforce the low-rankness of data matrix. The data originated from the k-space can be separated into the signal and noise subspaces. As observed in [SLO+14], the rank of the data matrix $\mathbf{A}$ is highly dependent on the type of scanned object and coil configuration. This indirectly implies that the $rank(\mathbf{A})$ can be approximately estimated before the image reconstruction. In this thesis, we all suppose $N_c, w \ll N_x, N_y$. Denote $p = (N_x - w + 1)(N_y - w + 1)$, $q = w^2 N_c$ and $*$ as the conjugate transpose. First, the data matrix can be uniquely decomposed as

$$\mathbf{A} = \sum_{i=1}^{n} A_i = \sum_{i=1}^{n} \sigma_i \mathbf{u_i} \mathbf{v_i}^* \tag{3.2}$$

where $\mathbf{u_i} \in \mathbb{C}^{p \times 1}$ and $\mathbf{v_i} \in \mathbb{C}^{q \times 1}$ are left and right singular vectors of $\mathbf{A}$ which correspond to the singular value σ_i. If the singular values are given in a descending order, the least squares estimate of the desired signal with rank $\hat{r}$ is given in a truncated version:

$$\hat{\mathbf{A}} = \sum_{i=1}^{\hat{r}} A_i = \sum_{i=1}^{\hat{r}} \sigma_i \mathbf{u_i} \mathbf{v_i}^* \tag{3.3}$$

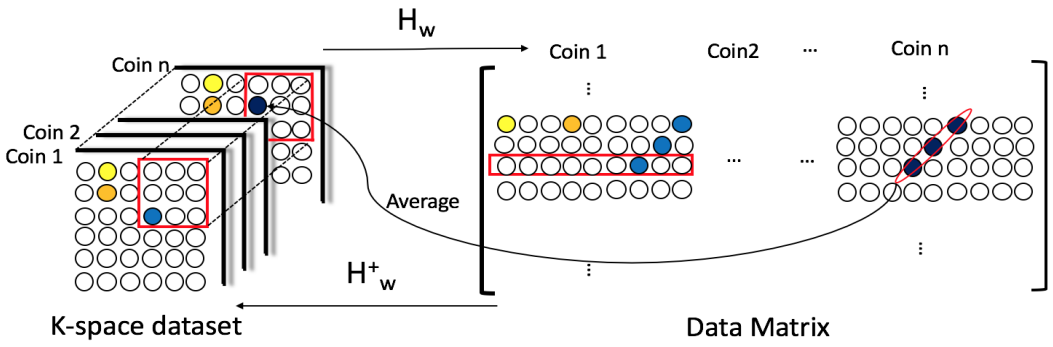

Figure 3.1: The procedure of generating data matrix and enforcing structural consistency projection in [SLO+14]. The white dots represent non-sampled points.

3.1.3 Structural Consistency Projection

Correspondingly, a reverse operator of H_w which projects the data matrix back to multi-coil k-space data is also defined as $H_w^\dagger$:

$$H_w^\dagger : \mathbb{C}^{(N_x-w+1)(N_y-w+1)\times w^2 N_c} \to \mathbb{C}^{N_x \times N_y \cdot N_c} \tag{3.4}$$

This process is shown in Figure 1. Note that the inputting data matrix, like $\hat{\mathbf{A}}$, may not be structured as a block-wise Hankel matrix. Alternatively, $H_w^\dagger$ obtains the values of the entries on the anti-diagonal direction and then projects their average values onto a particular entry of the k-space dataset.

3.1.4 Data Consistency Projection

Let $\mathbf{x}_n$ be the current estimate of the k-space data given by $\mathbf{x}_n = H_w^\dagger(\hat{\mathbf{A}})$. The entries at the sampled locations now contain different values from the original ones. More precisely, denote $\mathbf{y}$ as the chained raw k-space data that is acquired from the multiple coils, with the size of $N_x \times N_y \cdot N_c$. Given the linear operator $\mathbf{D}$ which only selects the sampled locations, the data consistency can be achieved by a least square solution to $\mathbf{D}\mathbf{x_n} = \mathbf{y}$:

$$\mathbf{x_{n+1}} = (\mathbf{I} - \mathbf{D}\mathbf{D}^\dagger)\mathbf{x_n} + \mathbf{D}^\dagger\mathbf{y} \tag{3.5}$$

where the pseudo-inverse of $\mathbf{D}$, represented as $\mathbf{D}^\dagger$, serves for vectorizing the chained data and filling vacant entries with zeros.

According to the above definitions and notations, the low-rank matrix completion problem formulated in SAKE [SLO+14] can be expressed as follows:

$$arg\min_{\mathbf{x}} \|\mathbf{D}\mathbf{x} - \mathbf{y}\|^2$$
$$\text{subject to } rank(\mathbf{A}) = \hat{r},\ \mathbf{x} = H^\dagger(\mathbf{A})$$

where $\mathbf{y}$ and $\mathbf{x}$ are the k-space data acquired from coils and to be reconstruction, respectively.

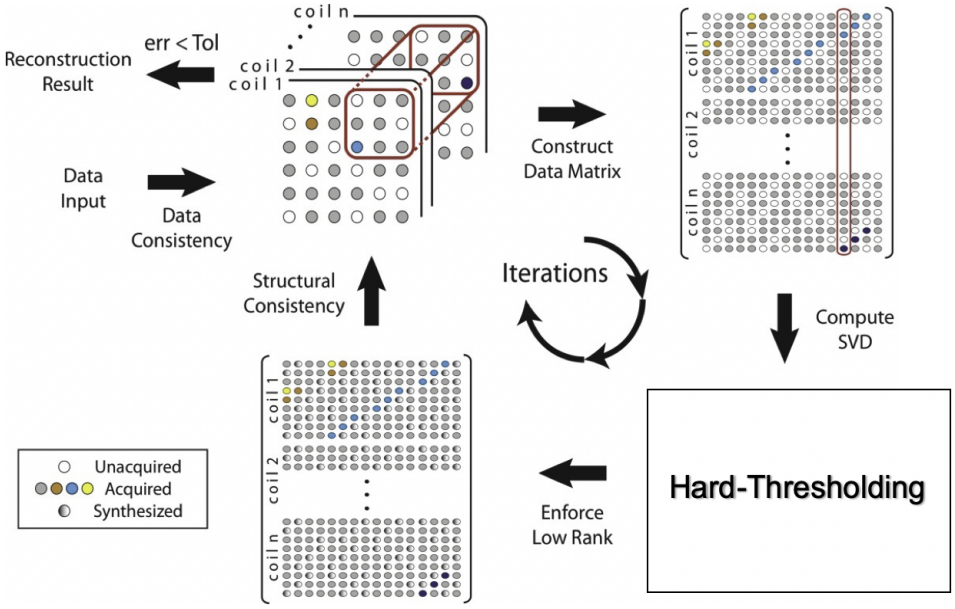

Figure 3.2: The procedure of enforcing all the projections to solve the low-rank completion problem in [SLO+14].

3.1.5 Iterations

It is noted that each of the above projections is defined to satisfy the constraints given in this optimization problem. To solve this problem, an initial estimate on k-space data is first given to the problem solver, which is usually selected as $\mathbf{D}^{\dagger}\mathbf{y}$. In the iterative process, the data matrix generated from the current estimate is successively enforced by low-rankness projection, structural consistency projection, and data consistency projection. The iteration ends when the consecutive estimates are within a predefined bound. Finally, the image can be recovered by computing the 2D inverse Fourier transform over the reconstructed k-space data. The diagram of iterative reconstruction in SAKE is illustrated in Figure 3.2.

3.2 Cartesian and Non-Cartesian Sampling

In modern MRI, there roughly exist two categories of traversal strategies for k-space data sampling, known as Cartesian Sampling and non-Cartesian Sampling. In the Cartesian Sampling case, the data is sampled with regular intervals in the k-space, which enables convenient implementation. The non-Cartesian Sampling approaches, including spiral, zig-zag, radial, etc, are more popular in recent years [WHG+14]. These approaches acquire data in a non-uniform manner, which cause fewer motion artifacts and then can reconstruct high-resolution

images.

The sampling operator $\mathbf{D}$ in Equation (3.5) has different implementations given the sampling strategy. In Cartesian Sampling, the $\mathcal{DO}$ only needs to substitute the k-space data $\hat{\mathbf{x}}_n$ at the sampled locations with the original data acquired from the multiple coils. In this way, the estimate of the k-space data is easily updated to $\mathbf{x}_{n+1}$. Differently, the operator $\mathbf{D}$ in non-Cartesian Sampling refers to an interpolation operator which transforms the Cartesian reconstructed data to non-Cartesian locations and remains fixed according to the sampling approach. This step can be approximately implemented by Equation (3.5) and requires to be repeated by a few times before getting good results. Despite the various sampling approaches, the MRI scanning machine usually holds only one piece of sampling code in clinical settings.

In the following chapters, we will follow the definitions provided in this chapter and introduce our proposed models.

Chapter 4

Problem Statement

In this chapter, we will briefly take a glance of two of our proposed models. Our assumptions are introduced, including how the user and server are set in the system architecture, what are the possible attacks that the adversaries can execute, and the designing goals under these assumptions.

4.1 System Architecture

In this thesis, we focus on securely outsourcing the computations of solving low-rank matrix completion problem to the cloud server. In SecSAKE, two system architectures are considered corresponding to two of our different designs, both of which are composed of two group of entities, the Data Owner $\mathcal{DO}$ and Cloud Servers $\mathcal{CS}$. Under the circumstance of performing MRI in the clinic, the collected data from scanning patient's body is supposed to be securely kept by the clinic. As the $\mathcal{DO}$ in our system, however, the clinic may not be able to reconstruct the diagnosable image by itself efficiently, due to the large-scale data samples and its limited computation resources. Then the $\mathcal{DO}$ is inclined to outsource the most computational expensive parts to the $\mathcal{CS}$, while presercing both the data confidentiality and computation functionality. Based on the capability of the computation and the security requirement of the $\mathcal{DO}$, the settings within the $\mathcal{CS}$ of two designs are differentiated as shown in Figure 4.1.

In our first design, only one server is included in the $\mathcal{CS}$. In each iteration, the $\mathcal{DO}$ will locally generate the data matrix from the acquired data. By carefully encrypting the data matrix, the $\mathcal{DO}$ can outsource the computation task of SVD to the $\mathcal{CS}$ in a privacy-preserving manner. The solution is then returned and decrypted by the $\mathcal{DO}$ before it performs the structural consistency projection. To enforce the data consistency projection, the $\mathcal{DO}$ pre-stores an encrypted form of the matrix related to the sampling mechanism on the $\mathcal{CS}$. Then the $\mathcal{DO}$ can securely outsource the current estimate to the $\mathcal{CS}$ for updating. Finally,

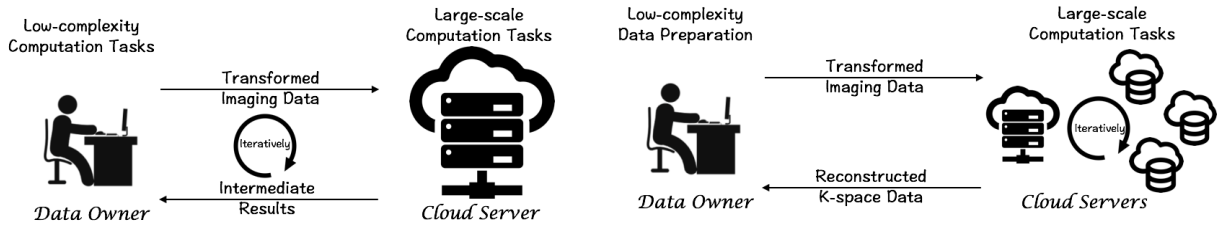

Figure 4.1: Left: System Architecture of SecSAKE I. The $\mathcal{DO}$ keeps online to communicate with the $\mathcal{CS}$ in each iteration;
Right: System Architecture of SecSAKE II. The data owner keeps offline while the servers communicate with each other.

the $\mathcal{DO}$ can stop the iteration when the requirement of the convergence is met.

In our second design, four independent cloud servers are included: one *encryptor* and three *shareholders*. At first, the scanned imaging data is randomly split into three shares. In the initialization phase, the $\mathcal{DO}$ distributes the first-round key matrices along with the shares to the *shareholders*. Besides, the data owner encrypts the imaging data and outsources it to the *executor*. The executor computes SVD, applies an additional mask to hide the temporary results and then securely passes different units of results to different *shareholders*. With the encrypted results, the *shareholders* can jointly compute their own share of the estimate but none of them can reveal the imaging data individually. These shares along with the stored imaging data shares will then be securely aggregated and sent to the *executor*, as the input of the next iteration. At last, the $\mathcal{DO}$ can reveal the results by transformation of imaging data after it manually stops the iteration.

4.2 Threat Model

In reality, the exposure of imaging data, including both the intermediate and the final reconstruction estimate in the k-space domain, may incurs an increased risk of patient's identity and physical condition. Meanwhile, we observe that the clinics seldom updates their sampling method. The adversary may take advantage of the fixed sampling operator to track a specific patient. Hence, in this thesis, we address that two parts of the data involved in the computation may potentially lead to privacy leakage: the imaging data $\mathbf{y}$ (and $\mathbf{x}$) and the sampling operator $\mathbf{D}$.

In addition, in our first design, we assume the $\mathcal{CS}$ to be malicious. The $\mathcal{CS}$ tries to extract information from the patient's data and from the results of its own computations. Also, $\mathcal{CS}$ can intentionally deviate from the proposed protocols or undermine the integrity of

the computation results, in the hope of not being detected. Also, $\mathcal{CS}$ may also suffer from external attacks that could change or leak the data used in the process of computation, which may cause erroneous results or potential data privacy issues.

In the second design, we assume all the servers are semi-honest. They will follow the algorithm to execute the communication and computation and may also attempt to derive meaningful information from the data. A semi-trusted cloud is consistent with the users trying to maintain the security of a sufficient portion of cloud resources, but some parts of the cloud may be under the control of untrusted parties. To undermine the confidentiality or integrity of data or calculations, the undermined party appears to be the major source of the threat, the purpose of which is to gather more information by combining the observations of malicious adversaries. We also assume any two of the servers are non-colluding. More specifically, none of the servers included in the system can share the part of the data that is not scheduled in the protocol to other servers. Such model is reasonable and can be guaranteed by the co-statement between cloud service providers in practice [CAF13].

To enable the user to securely outsource computing tasks to the $\mathcal{CS}$, the data that the user shares with the $\mathcal{CS}$ should appear random. This notion of privacy is known as computational indistinguishability. To define computational indistinguishable with a random matrix, we follow the definition and method used in [SLCL15]:

Definition 4.1. *Given any polynomial time distinguisher D, there exists a negligible function $\mu(\cdot)$ such that*

$$|Pr[D(r_{i,j}) = 1] - Pr[D(s_{i,j}) = 1]| < \mu \tag{4.1}$$

where the distinguisher D outputs 1 when its input is identified as from a uniform distribution.

Formally, we use the definition of column-wise *Computational Indistinguishability* as follows:

Definition 4.2. *A matrix transformation is column-wise computational indistinguishable under a chosen-plaintext attack (CPA) if: For any adversary $\mathcal{A}$ modeled by a probabilistic polynomial time Turing machine, the advantage of distinguishing from two matrix transformations at the same column can be bounded by a negligible function in a CPA indistinguishability experiment.*

4.3 Design Goals

We aim to provide a secure and efficient outsourcing scheme for the MRI reconstruction. Several key design objectives are listed as follows:

Imaging Data Privacy: None of the entities in the $\mathcal{CS}$ should get access to the exact value of the imaging data, including the both the sampled value and the the estimates during the iteration, or can reconstruct the final diagnose image based on its accessible data. Besides, in our first design, the locations where the data are sampled should also be protected according to the $\mathcal{DO}$'s preference.

Efficiency Gain: In SecSAKE, the time cost for computation on the $\mathcal{DO}$ side should be lower than solving the computational problem by himself. Moreover, any entity in the $\mathcal{CS}$ should be able to compute the problem within the comparable time complexity with an existing efficient algorithm.

Verifiablity: In our first design, the $\mathcal{DO}$ should be capable to validate the correctness of the returned results from the $\mathcal{CS}$ in each loop of the iteration.

4.4 Summary

In this chapter, we briefly review the core algorithm used to solve the low-rank matrix completion problem. Within each iteration of the algorithm, we identified the most computation expensive parts, where the clinic can outsource the only these parts to the server as a compact design mode. In the next two chapters, we illustrate the detailed design of these two protocols.

Chapter 5

SecSAKE I

In this chapter, we focus on our first design of SecSAKE. We follow the sequence of operations within one iteration in SAKE. Among all the related computation tasks in SAKE, we identify two of them are computational intensive – SVD in the low rankness projection and matrix multiplication in the data consistency projection. Then the approach to verify both of the results is given afterwards.

5.1 Secure Singular Value Decomposition

At the beginning of one iteration of SAKE, the $\mathcal{DO}$ aims to securely outsource the computation of SVD over the complex field to the $\mathcal{CS}$. Before proceeding to this step, the $\mathcal{DO}$ has either 1) just initialized the first estimate as $\mathbf{x_0} = \mathbf{D}^{\dagger}\mathbf{y}$, where $\mathbf{y}$ is the acquired data or 2) received the estimate of k-space data x_n from the previous iteration in the $(n-1)$th loop. With no loss of generality, we only consider the second case throughout this chapter.

Firstly, the k-space data x_n can be trivially mapped into a data matrix $\mathbf{A}_n$ with block-wise Hankel structure by enforcing H_w. The $\mathcal{DO}$ then aims to encrypt the data matrix $\mathbf{A}_n \in \mathbb{C}^{p\times q}$ before outsourcing the computation of SVD to the cloud. In order to to preserve the value privacy and the structure of data matrix, two random diagonal unitary matrices and random permutations are applied in this step, respectively. Suppose the diagonal unitary matrices we use are named $\mathbf{\Lambda}_1 \in \mathbb{C}^{p\times p}$ and $\mathbf{\Lambda}_2 \in \mathbb{C}^{q\times q}$, respectively. To construct $\mathbf{\Lambda}_1 = diag(\lambda_1, \lambda_2, ..., \lambda_p)$ and $\mathbf{\Lambda}_2 = diag(\gamma_1, \gamma_2, ..., \gamma_q)$, on input the security parameter κ, the $\mathcal{DO}$ applies a pseudorandom function F_κ to generate the ratio of real to the imaginary part of each non-zero entry.

More precisely, taking $\mathbf{\Lambda}_1$ as an example, if F_κ is defined over $\{0,1\}^\kappa \times \{0,1\}^\kappa \to \{0,1\}^\kappa$,

the sequence of numbers are given as follows:

$$\begin{gathered}\Re(\lambda_l) = \sqrt{1/(1+\eta_l^2)}, \Im(\lambda_l) = \eta_l\sqrt{1/(1+\eta_l^2)}, \forall l \in [1,p] \\ \eta_l = F_\kappa(r_l, c), \forall l \in [0,p]\end{gathered} \tag{5.1}$$

where r_l is a random string and c is the constant. Suppose $\mathbf{\Lambda}_2$ has the same construction and denote the outputs of the pseudorandom function as $\eta'_k, \forall k \in [0,q]$. After that, the $\mathcal{DO}$ can hide the data matrix $\mathbf{A}_n$ by

$$\tilde{\mathbf{A}}_n = \eta_0\eta'_0\mathbf{\Lambda}_1\mathbf{A}_n\mathbf{\Lambda}_2 \tag{5.2}$$

Here we provide a sketch of proof before reaching the conclusion that the transformation given by Equation (5.2) is *computational indistinguishable*. Firstly and trivially, according to property of diagonal matrix, given any two matrices $\mathbf{A}_n^1$ and $\mathbf{A}_n^2$ over complex field with same structure, i.e. have same positions for non-zero entries, the $\tilde{\mathbf{A}}_n^1$ and $\tilde{\mathbf{A}}_n^2$ have the same structures. Take a more in-depth observation, each data entry in $\tilde{\mathbf{A}}_n$ is given by $\tilde{a}_{i,j}^n = \eta_0\eta'_0\lambda_i a_{i,j}^n\gamma_j$, where the real part is given by:

$$\begin{aligned}\Re(\tilde{a}_{i,j}^n) &= \eta_0\eta'_0[\Re(\lambda_i)\Re(\gamma_j) - \Im(\lambda_i)\Im(\gamma_j)]\Re(a_{i,j}^n) + \\ &\quad [\Re(\gamma_j)\Im(\lambda_i) + \Re(\lambda_i)\Im(\gamma_j)]\Im(a_{i,j}^n) \\ &= \theta[(1-\eta_i\eta'_j)\Re(a_{i,j}^n) + (\eta_i+\eta'_j)\Im(a_{i,j}^n)]\end{aligned} \tag{5.3}$$

Similarly, the imaginary part is given by:

$$\Im(\tilde{a}_{i,j}^n) = \theta[(1-\eta_i\eta'_j)\Re(a_{i,j}^n) - (\eta_i+\eta'_j)\Im(a_{i,j}^n)] \tag{5.4}$$

where $\theta = \eta_0\eta'_0\sqrt{1/(1+\eta_i^2)(1+\eta_j'^2)}$. The best strategy for the adversary $\mathcal{A}$ to deduce the real part of $a_{i,j}^n$ is to compute

$$\Re(\tilde{a}_{i,j}^n) + \Im(\tilde{a}_{i,j}^n) = \eta_0\eta'_0\sqrt{1+(\eta_i+\eta_j)^2/(1+\eta_i^2)(1+\eta_j'^2)}\Re(a_{i,j}^n) \tag{5.5}$$

Due to the pseudo-randomness of η_l, it can be easily concluded that any value derived from both the real part and the imaginary part cannot reveal more information than $\eta_0\eta'_0 a_{i,j}^n$. According to the CPA indistinguishability experiment and corresponding conclusion proposed in [SLCL15], the transformation in Equation (5.2) can achieve column-wise computational indistinguishability under a CPA for non-zero elements under the Definition 4.1.

Besides the matrices $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$, the $\mathcal{DO}$ needs to apply two random permutation matrices to hide the locations zero elements. In brief, two pseudorandom permutations, π_1 and π_2 are applied to $\tilde{\mathbf{A}}_n$ through multiplications of the matrices $\mathbf{P} \in \mathbb{R}^{p\times p}, \mathbf{Q} \in \mathbb{R}^{q\times q}$, i.e.

$$\tilde{\mathbf{A}}'_n = \mathbf{P}\tilde{\mathbf{A}}_n\mathbf{Q} = \eta_0\eta'_0\mathbf{P}\mathbf{\Lambda}_1\mathbf{A}_n\mathbf{\Lambda}_2\mathbf{Q} \tag{5.6}$$

where all the non-zero entries in $\mathbf{P} = (p_{i,j})$ and $\mathbf{Q} = (q_{i,j})$ are filled with 1's, whose positions can be denoted by:

$$\begin{aligned} &p_{\pi_1(i),j}, \forall i \in [1,p], i = j \\ &q_{\pi_2(i),j}, \forall i \in [1,q], i = j \end{aligned} \tag{5.7}$$

Algorithm 1: Secure SVD and Low-rankness Projection

Input: A block-wise Hankel structured data matrix $\mathbf{A}_n$

Output: The best rank-$\hat{r}$ approximation of $\mathbf{A}_n$

1: Generate η_0, η_0', $\mathbf{\Lambda}_1$ and $\mathbf{\Lambda}_2$ by Equation (5.1)
2: Generate $\mathbf{P}$ and $\mathbf{Q}$ by Equation (5.7)
3: Compute $\tilde{\mathbf{A}}_n' = \eta_0\eta_0'\mathbf{P}\mathbf{\Lambda}_1\mathbf{A}_n\mathbf{\Lambda}_2\mathbf{Q}$ and send $\tilde{\mathbf{A}}_n'$ to the $\mathcal{CS}$
4: Receive $\mathbf{U}'$, $\mathbf{\Sigma}'$, $\mathbf{V}'$
5: Compute $\mathbf{U}'' = (\mathbf{P}\mathbf{\Lambda}_1)^*\mathbf{U}'$, $\mathbf{\Sigma}'' = (1/|\eta_0\eta_0^{'}|)\mathbf{\Sigma}'$, $\mathbf{V}'' = \mathbf{\Lambda}_2\mathbf{Q}\mathbf{V}'$
6: Retain the first $\hat{r}$ columns of $\mathbf{U}''$, $\mathbf{\Sigma}''$ and $\mathbf{V}''$
7: Compute $\hat{\mathbf{A}}_n$ via Equation (5.11)

Theorem 5.1. *The transformation of matrix over complex field given in Equation (5.6) is computational indistinguishable transformations under a chosen-plaintext attack.*

Next, we show that both the matrices $\mathbf{P_1} = \mathbf{P}\mathbf{\Lambda}_1$ and $\mathbf{Q_1} = (\mathbf{\Lambda}_2\mathbf{Q})^* = \mathbf{Q}^T\mathbf{\Lambda}_2^*$ are unitary matrices. Recall the construction of $\mathbf{\Lambda}_1$, the complex conjugate of the lth diagonal element is given by $\overline{\lambda_l} = \sqrt{1/1+\eta_l^2} - \eta_j\sqrt{1/(1+\eta_l^2)}i$. Hence, $\mathbf{\Lambda}_1\mathbf{\Lambda}_1^* = I$. According to the definition of $\mathbf{P}$, each column is mutually perpendicular and normalized. Thus $\mathbf{P}_1\mathbf{P}_1^* = \mathbf{P}\mathbf{\Lambda}_1(\mathbf{P}\mathbf{\Lambda}_1)^* = \mathbf{P}\mathbf{\Lambda}_1\mathbf{\Lambda}_1^*\mathbf{P}^T = I$. In this way, we can also find $\mathbf{Q}_1$ unitary.

The property of the unitary enabls the $\mathcal{DO}$ to recover the correct solution of SVD by reversing the matrix transformation. More precisely, the $\mathcal{DO}$ outsources the transformed matrix $\tilde{\mathbf{A}}_n'$ to the $\mathcal{CS}$, who later finds the eigenvalues and eigenvectors of $\tilde{\mathbf{A}}_n'\tilde{\mathbf{A}}_n'^*$ denoted as σ' and u', respectively. With the descending order of eigenvalues found on the diagonal entries of diagonal matrix $\mathbf{\Sigma}'$, the eigenvectors uniquely form the columns of the left-singular matrix $\mathbf{U}'$. This procedure can be represented by

$$\tilde{\mathbf{A}}_n'\tilde{\mathbf{A}}_n'^*\mathbf{U}' = \eta_0^2\eta_0^{'2}(\mathbf{P}_1\mathbf{A}_n\mathbf{Q}_1^*)(\mathbf{Q}_1\mathbf{A}_n^*\mathbf{P}_1^*)\mathbf{U}' = \mathbf{\Sigma}'^2\mathbf{U}' \tag{5.8}$$

Then the equation can be rewritten as

$$\mathbf{A}_n\mathbf{A}_n^*\mathbf{P}_1^*\mathbf{U}' = (1/\eta_0^2\eta_0^{'2})\mathbf{\Sigma}'^2\mathbf{P}_1^*\mathbf{U}' \tag{5.9}$$

Let $\mathbf{U}'' = \mathbf{P}_1^*\mathbf{U}'$ and $\mathbf{\Sigma}'' = (1/|\eta_0\eta_0^{'}|)\mathbf{\Sigma}'$. It is obvious that $\mathbf{U}''$ and $\mathbf{\Sigma}''$ are the left-singular matrix and diagonal matrix containing the non-increasing singular values of $\mathbf{A}_n$.

Symmetrically, the right-singular matrix can be computed by the $\mathcal{CS}$ as follows:

$$\mathbf{A}_n^* \mathbf{A}_n \mathbf{Q}_1^* \mathbf{V}' = (1/\eta_0^2 \eta_0^{'2}) \mathbf{\Sigma}'^2 \mathbf{Q}_1^* \mathbf{V}' \tag{5.10}$$

let $\mathbf{V}'' = \mathbf{Q}_1^* \mathbf{V}'$. Then $\mathbf{V}''$ becomes the right singular matrix of $\mathbf{A}_n$. The $\mathcal{CS}$, however, has access to neither $\mathbf{P}_1$ nor $\mathbf{Q}_1$. Hence, the $\mathcal{CS}$ will send the $\mathbf{U}'$, $\mathbf{V}'$ and $\mathbf{\Sigma}'$ to the $\mathcal{DO}$. The $\mathcal{DO}$ can apply the reverse transformation to these matrices to get the singular vectors and values of $\mathbf{A}_n$.

Algorithm 2: Privacy-preserving Data Consistency Projection (Non-Cartesian sampling & Sampling Operator Hiding)

Input: The current estimate of k-space data $\hat{\mathbf{x}}_n$, the sampling matrix D, its pseudo-inverse $\mathbf{D}^\dagger$ and precomputed $\mathbf{D}\mathbf{D}^\dagger$.

Output: $\mathbf{x}_{n+1}$: Projection onto the solution set for $\mathbf{D}\mathbf{x} = \mathbf{y}$.

If $n = 0$:

1: Generate random matrices $\mathbf{R}_1 = \mathbf{u_1}\mathbf{v_1^T}$ and $\mathbf{R}_2 = \mathbf{u}_2\mathbf{v}_2^*$ via Equation (5.15).

2: Compute $\hat{\mathbf{N}} = (\mathbf{I} - \mathbf{D}\mathbf{D}^\dagger) + \mathbf{R}_1$ and send $\hat{\mathbf{N}}$ to the $\mathcal{CS}$.

3: Compute $\mathbf{s}_0 = (\mathbf{N}\mathbf{u}_2)\mathbf{v}_2^* + \mathbf{u_1}\mathbf{v_1^T}\mathbf{u}_2\mathbf{v}_2^* - \mathbf{x}_0$ and store $\mathbf{s}_0$ locally.

4: Compute $\hat{\mathbf{x}}'_n = \hat{\mathbf{x}}_n + \mathbf{R}_2$ and send $\hat{\mathbf{x}}'_n$ to the $\mathcal{CS}$.

5: Compute $\mathbf{s}_1 = \mathbf{u_1}(\mathbf{v_1^T}\hat{\mathbf{x}}'_n)$

6: Receive $\hat{\mathbf{N}}\hat{\mathbf{x}}'_n$ from the $\mathcal{CS}$.

7: Compute $\mathbf{x}_{n+1} = \hat{\mathbf{N}}\hat{\mathbf{x}}'_\mathbf{n} - \mathbf{s}_0 - \mathbf{s}_1$.

5.2 Low-rankness and Structural Consistency Projection

After computing the SVD, the $\mathcal{DO}$ needs to enforce the low-rankness to $\mathbf{U}''$, $\mathbf{V}''$ and $\mathbf{\Sigma}''$. Suppose the pre-estimated rank value is $\hat{r}$. Then the projection is simply done by deleting columns $\hat{r}+1$, $\hat{r}+2$, ..., p of $\mathbf{U}''$, and the columns $\hat{r}+1$, $\hat{r}+2$, ..., q of $\mathbf{V}''$ and $\mathbf{\Sigma}''$. Let $T_{\hat{r}}$ be the operator to retain the first $\hat{r}$ columns of the matrix. Denote $\mathbf{U}''_{||} = T_{\hat{r}}(\mathbf{U}'')$, $\mathbf{V}''_{||} = T_{\hat{r}}(\mathbf{V}'')$ and $\mathbf{\Sigma}''_{\hat{r}} = T_{\hat{r}}(\mathbf{\Sigma}'')$. Then above low-rankness projection isformulated as a hard-thresholding procedure and can be formed as

$$\hat{\mathbf{A}}_n = \mathbf{U}''_{||}\mathbf{\Sigma}''_{\hat{r}}\mathbf{V}''^*_{||} \tag{5.11}$$

By doing so, $\hat{\mathbf{A}}_n$ ceases to be a block-wise Hankel matrix but becomes one with the rank $\hat{r}$. As we assumed, the estimated rank $\hat{r}$ is small and satisfies $\hat{r} \ll q < p$. In this way, the matrices $\mathbf{U}''_{||}$ and $\mathbf{V}''_{||}$ can be sparse and the procedure of computing $\hat{\mathbf{A}}_n$ by matrix multiplication can be cheap. A summary of the operations on enforcing low-rankness projection is presented in

Algorithm 1. Then the values of matrix $\hat{\mathbf{A}}_n$ on the anti-diagonal direction will be averaged and mapped back to the k-space data $\hat{\mathbf{x}}_n$ by the $\mathcal{DO}$, i.e.,

$$H_w^{\dagger}(\hat{\mathbf{A}}_n) = \hat{\mathbf{x}}_n \tag{5.12}$$

5.3 Privacy-preserving Data Consistency Projection

Taking the $\hat{\mathbf{x}}_n$ from the previous step as the input, the $\mathcal{DO}$ enforces the data consistency by projecting the k-space data onto the solution set of $\mathbf{Dx} = \mathbf{y}$. Then the operations in Equation (3.5) can be simplified into the following form:

$$\mathbf{x_{n+1}} = (\mathbf{I} - \mathbf{DD}^{\dagger})\hat{\mathbf{x}}_\mathbf{n} + \mathbf{D}^{\dagger}\mathbf{y} = \mathbf{N}\hat{\mathbf{x}}_\mathbf{n} + \mathbf{x}_0 \tag{5.13}$$

In case of Cartesian Sampling, the $\mathcal{DO}$ only needs to replace the updated entries in the sampled locations with their original values, which involves very little computational overhead.

Algorithm 3: Results Verification of SVD for each loop

Input: $\mathbf{U}'$, $\mathbf{\Sigma}'$, $\mathbf{V}'$ and $\tilde{\mathbf{A}}'_n$.

Output: "Accept" or "Reject".

For all $j = 1$ to l

1: Uniformly generate a random vector $r_j \in \{0, 1\}^q$

2: Test $\mathbf{U}'\mathbf{\Sigma}'(\mathbf{V}'r_j) \stackrel{?}{=} \tilde{\mathbf{A}}'_n r_j$

If False:

3. Return "Reject"

4: Return "Accept".

Regarding the non-Cartesian Sampling, we suppose the operators related to this step are defined in a finite-dimensional space. Besides the initial estimate $\mathbf{x}_0$, we also assume the sampling-related matrices have been re-computed by the $\mathcal{DO}$, such as $\mathbf{D}$, $\mathbf{D}^{\dagger}$ and $\mathbf{DD}^{\dagger}$ [LP10]. Conducting the update can still be very computational expensive due to the matrix multiplication $\mathbf{N}\hat{\mathbf{x}}_\mathbf{n}$. To efficiently compute the $\mathbf{N}\hat{\mathbf{x}}_\mathbf{n}$, the $\mathcal{DO}$ can either choose to 1) only protect the imaging data $\hat{x}_n$ or 2) protect both the imaging data $\hat{x}_n$ and sampling operator $\mathbf{N}$. Without loss of generality, we focus on the second case, which is more general. Inspired by the idea of proof given in [SLCL15], we supplement a matrix transformation scheme for the field of complex numbers as follows:

Suppose the matrix to be transformed is denoted as $\mathbf{M}$, where $\mathbf{M} \in \mathbb{C}^{n\times l}$. Assuming the values of $\mathbf{M}$ can be represented by a combination of the real part $\mathfrak{R}$ and the imaginary part $\mathfrak{I}$, i.e $\mathbf{M} = \mathfrak{R}(\mathbf{M}) + \mathfrak{I}(\mathbf{M})i$. A closest range of the values is given by $\mathfrak{R}(\mathbf{M}) \in [-K_a, K_a]$ and $\mathfrak{I}(\mathbf{M}) = [-K_b, K_b]$, where $K_a = 2^{\alpha}$ and $K_b = 2^{\beta}$. To hide the values in $\mathbf{M}$, a random matrix $\mathbf{R}$ can be applied as follows:

$$\hat{\mathbf{M}} = (\hat{m}_{i,j})_{n\times l} = \mathbf{M} + \mathbf{R} \tag{5.14}$$

In order to minimize the overhead for further usage of $\hat{\mathbf{M}}$ (e.g. matrix multiplication), the matrix $\mathbf{Z}$ can be constructed by the outer product of two vectors $\mathbf{u}$, $\mathbf{v} \in \mathbb{R}^{n\times 1}$ and $\mathbb{C}^{l\times 1}$, respectively i.e:

$$\mathbf{R} = (r_{i,j})_{n\times l} = \mathbf{u} \otimes \mathbf{v} \tag{5.15}$$

Since $\mathbf{M}$ is in its complex field, we let the entries of $\mathbf{u}$ be randomly selected from [-c, c], where $c = 2^d$ and d is a positive integer. Correspondingly, the values in the vector $\mathbf{v}$ are construction separately for the real part and imaginary part, whose range are both in $[2^{\max\{n,l\}}, 2^{\max\{n,l\}+\iota}]$, where $\iota > 0$.

The $\mathcal{DO}$ can firstly generate two random matrices $\mathbf{R}_1 \in \mathbb{R}^{s\times s}$ and $\mathbf{R}_2 \in \mathbb{C}^{s\times t}$ according to the construction of Equation (5.15), where $s = N_x$ and $t = N_y N_c$. Then the $\mathcal{DO}$ can mask the matrix $\mathbf{N}$ by $\hat{\mathbf{N}} = \mathbf{N} + \mathbf{R_1} = \mathbf{N} + \mathbf{u_1}\mathbf{v_1^T}$ and pre-store the $\hat{\mathbf{N}}$ on the $\mathcal{CS}$. Similarly, once there comes an estimate $\hat{\mathbf{x}}_n$, the $\mathcal{DO}$ can mask it as $\hat{\mathbf{x}}'_n = \hat{\mathbf{x}}_n + \mathbf{R}_2 = \hat{\mathbf{x}}_n + \mathbf{u}_2\mathbf{v}_2^*$ and outsources $\hat{\mathbf{x}}'_n$ to the $\mathcal{CS}$. The $\mathcal{CS}$ will execute the computation of $\hat{\mathbf{N}}\hat{\mathbf{x}}'_n$. Meanwhile, the $\mathcal{DO}$ can compute $\mathbf{s}_0 = (\mathbf{N}\mathbf{u}_2)\mathbf{v}_2^* + \mathbf{u_1}(\mathbf{v_1^T}\mathbf{u}_2)\mathbf{v}_2^* - \mathbf{x}_0$ and $\mathbf{s}_1 = \mathbf{u_1}(\mathbf{v_1^T}\hat{\mathbf{x}}'_n)$. The most expensive operation in both of the term is matrix-vector multiplication, which saves much local computational overhead. Furthermore, the matrix $\mathbf{s}_0$ can be precomputed if the $\mathbf{R}_1$ and $\mathbf{R}_2$ is acquired before the iteration.

After receiving $\mathbf{M}_n = \hat{\mathbf{N}}\hat{\mathbf{x}}'_n$ from the $\mathcal{CS}$, the $\mathcal{DO}$ can compute:

$$\mathbf{x}_{n+1} = \mathbf{M}_n - \mathbf{s}_0 - \mathbf{s}_1 \tag{5.16}$$

Algorithm 4: Batched Validation of Data Consistency Projection

Input: τ, τ_0; $\mathbf{M}_k$, $\hat{\mathbf{N}}$ and $\hat{\mathbf{x}}'_k$, $\forall k$.

Output: "Accept" or "Reject".

1: $\tau = \min(\tau_0, \tau)$

2: Generate a vector with random numbers: $t = (t_k)_{k=1,2,\ldots\tau}$

3: Compute $\psi = \sum_{k=1}^{\tau} t_k \hat{\mathbf{x}}'_k$, $\zeta = \sum_{k=1}^{\tau} t_k \mathbf{M}_k$

4: Test $\hat{\mathbf{N}}\psi \stackrel{?}{=} \zeta$

If False

5. Return "Reject"

6: Return "Accept".

The $\mathcal{DO}$ is in control of the current estimate $\mathbf{x}_{n+1}$ and previous estimate $\mathbf{x}_n$, it can decide whether to stop the iterations by checking whether they are close enough. Note that the matrix multiplication of $\mathbf{D}^\dagger\mathbf{y}$ can be done in the same way at very beginning, i.e. additively transform $\mathbf{D}^\dagger$ and $\mathbf{y}$ and outsource it to the $\mathcal{CS}$. A summary of this method is given in Algorithm 2.

5.4 Correctness Verification

According to the threat model, we consider the case that the $\mathcal{CS}$ can maliciously deviate from the protocol. During several consecutive rounds, the $\mathcal{CS}$ should return a set of matrices which may contain fake ones. This may prevent the computation from convergence in τ times of iterations, where τ is the parameter indicating the maximum allowable number of iterations. Accordingly, the $\mathcal{DO}$ needs to verify the correctness of the results in each round of the iteration. In 5.1, the intuitive method is to compute the $\mathbf{U}''\mathbf{\Sigma}''\mathbf{V}''^{*}$ and compare the results with $\mathbf{A}_n$. However, the matrix multiplication could incur undesirable time cost. Alternatively, the $\mathcal{DO}$ can use the well-known Freivalds' algorithm [Fre77, MR10] by running l rounds of tests in each iteration (See Algorithm 3). Meanwhile, we notice that the matrix multiplication in 5.3 always involve the same matrix $\mathbf{N}$ if the masking matrix $\mathbf{R}_2$ is fixed. The $\mathcal{DO}$ can alternatively validate the results in 5.3 by batch tests (See Algorithm 4). We assume that main iteration may converge in τ_0 rounds or exceed τ before entering Algorithm 4. In Theorem 5.2, we gives the effectiveness of Algorithm 3 and Algorithm 4.

Theorem 5.2. *If the $\mathcal{CS}$ returns a false result, the probabilities that the result passes the Algorithm 3 and 4 are $\frac{1}{2^l}$ and $\frac{1}{2^\tau}$, respectively.*

Theorem 5.3. *If the $\mathcal{CS}$ returns a false result, the probability that the result passes the Algorithm 4 is $\frac{1}{2^{l'}}$.*

The detailed proof can be found in the next chapter.

5.5 Privacy Analysis

We prove the matrix transformation scheme proposed in 5.1 (i.e. Equation (5.2)) and 5.3 (i.e. Equation (5.14)) can achieve column-wise computationally indistinguishable under CPA.

Firstly, we provide a sketch of proof before reaching the conclusion that the transformation given by Equation (5.2) is *computationally indistinguishable*. According to property of diagonal matrix, given any two matrices $\mathbf{A}_n^1$ and $\mathbf{A}_n^2$ over complex field with same structure, i.e. have same positions for non-zero entries, the $\tilde{\mathbf{A}}_n^1$ and $\tilde{\mathbf{A}}_n^2$ have the same structures. Take a more in-depth observation, each data entry in $\tilde{\mathbf{A}}_n$ is given by $\tilde{a}_{i,j}^n = \eta_0\eta_0'\lambda_i a_{i,j}^n\gamma_j$, where the real part is given by:

$$\begin{aligned}\Re(\tilde{a}_{i,j}^n) &= \eta_0\eta_0'[\Re(\lambda_i)\Re(\gamma_j) - \Im(\lambda_i)\Im(\gamma_j)]\Re(a_{i,j}^n) + \\ &\quad [\Re(\gamma_j)\Im(\lambda_i) + \Re(\lambda_i)\Im(\gamma_j)]\Im(a_{i,j}^n) \\ &= \theta[(1-\eta_i\eta_j')\Re(a_{i,j}^n) + (\eta_i + \eta_j')\Im(a_{i,j}^n)]\end{aligned} \tag{5.17}$$

Similarly, the imaginary part is given by:

$$\Im(\tilde{a}^n_{i,j}) = \theta[(1 - \eta_i\eta'_j)\Re(a^n_{i,j}) - (\eta_i + \eta'_j)\Im(a^n_{i,j})] \tag{5.18}$$

where $\theta = \eta_0\eta'_0\sqrt{1/(1+\eta_i^2)(1+\eta_j'^2)}$. The best strategy for the adversary $\mathcal{A}$ to deduce the real part of $a^n_{i,j}$ is to compute

$$\Re(\tilde{a}^n_{i,j}) + \Im(\tilde{a}^n_{i,j}) = \eta_0\eta'_0\sqrt{1 + (\eta_i + \eta_j)^2/(1+\eta_i^2)(1+\eta_j'^2)}\Re(a^n_{i,j}) \tag{5.19}$$

Due to the pseudo-randomness of η_l, it can be easily concluded that any value derived from both the real part and the imaginary part cannot reveal more information than $\eta_0\eta'_0 a^n_{i,j}$. According to the CPA indistinguishability experiment and corresponding conclusion proposed in [SLLL16], the transformation in Equation (5.2) can achieve computational indistinguishability under the column-wise CPA for non-zero elements. More discussion on the details can be found in the next chapter.

Accordingly, the following theorem can be derived:

Theorem 5.4. *The transformation of matrix over complex field given in Equation (5.6) is computationally indistinguishable transformations under a chosen-plaintext attack.*

Next, to show the matrix transformation in Equation (5.14) is computational indistinguishable with a random matrix, we follow the definition and method used in Equation (4.1). Suppose there exists a random matrix $\mathbf{R}' = (r_{i,j})_{n\times l}$ with its real part of entries ranging from $[-c\Re(v_j), c\Re(v_j)]$, $\forall j \in [1, l]$. Given the security parameter $\kappa = \iota + \max\{n, l\} + d + 1$, it can be seen that both $\Re(r_{i,j})$ and $\Re(\hat{m}_{i,j})$ fall within $[-2^\kappa, 2^\kappa]$. Then for the values in real field,

$$|Pr[D(\Re(\hat{m}_{i,j})) = 1] - Pr[D(\Re(r_{i,j})) = 1]| \leq K_a/2c\Re(v_j) \tag{5.20}$$

Similarly, for the values in imaginary field, we can derive:

$$|Pr[D(\Im(\hat{m}_{i,j})) = 1] - Pr[D(\Im(r_{i,j})) = 1]| \leq K_b/2c\Im(v_j) \tag{5.21}$$

Thus, the chance that a distinguisher can determine whether $\hat{m}_{i,j}$ is randomly originated from a uniform distribution is at most at:

$$\mu(\kappa) \leq 2^{\kappa - \max\{n,l\} - q + 1} \tag{5.22}$$

For more details on the Equations 5.20 and 5.22, please refer to the next chapter.

Theorem 5.5. *The matrix $\hat{\mathbf{M}}$ in Equation (5.14) is computationally indistinguishable with the random matrix $\mathbf{R}'$, the entries of whose jth columns are uniformly sampled from $[-cv_j, cv_j]$.*

In SecSAKE I, the $\mathcal{CS}$ can only get access to three matrices in each iteration, the encrypted data matrix $\hat{\mathbf{A}}_n$, the stored encrypted sampling operator $\hat{\mathbf{N}}$ and the encrypted estimate in k-space $\hat{\mathbf{x}}_n$. Note that the keys for masking $\hat{\mathbf{A}}_n$ and $\hat{\mathbf{x}}_n$ are generated loop by loop. Theorem 5.4 indicates that the $\hat{\mathbf{A}}_n$ is computationally indistinguishable with a random matrix over complex field. Meanwhile, Theorem 5.5 ensures that the $\hat{\mathbf{x}}_n$ is also column-wise computationally indistinguishable under CPA. Hence, the $\mathcal{CS}$ cannot reveal any sensitive information of the imaging data during the computation process in one loop, even if it may want to construct a data matrix from $\hat{\mathbf{x}}_n$ in order to correlate it with $\hat{\mathbf{A}}_n$. The $\mathcal{CS}$ is not capable to determine the value of $\hat{\mathbf{N}}$ if R_1 is kept secret according to Theorem 5.5.

SecSAKE II involves four independent parties in the $\mathcal{CS}$, we analyze the privacy issues by inspecting the accessible matrices in each party. The *executor* in this protocol is assigned the same computation task but different input. The encrypted data matrix is provided by three *shareholders* in three shares. Since the *executor* does not have the updated key pair in the new round, each share of the data matrix can be protected the same as SecSAKE I. The *shareholders I & II* are supposed to conduct the matrix transformation given the re-encrypted results of SVD. Even though they know the round key, they cannot determine the results which have been masked by Z_1 and Z_2. In addition, the shares of initial imaging data are well protected by the randomly splitting. As long as none of the *shareholders* can collude with another party, there is no chance for any of the *shareholders* to learn any information of the image.

5.6 Efficiency Analysis

We theoretically deduce the time complexity of the proposed protocols by looking into the dominated steps within the proposed Algorithms. We follow the notations in previous chapters, i.e. the size of data matrix is $p \times q$ and the size of k-space dataset is $s \times t$. A quick relationship between these values is $s < t < q < p$ according to our previous assumption. By inspecting one loop in SAKE, we find that the SVD computation occupies the most of the computation time in image reconstruction with the time complexity as $O(\min\{pq^2, p^2q\}) = O(pq^2)$[HGI07]. Another computational intensive task in SAKE is the matrix multiplication during data consistency projection when non-Cartesian sampling is adopted. If the matrix $\mathbf{N} = \mathbf{I} - \mathbf{D}\mathbf{D}^\dagger$ has been pre-computed as we assume, the computation complexity during this process is dominated by $\mathbf{N}\hat{\mathbf{x}}_\mathbf{n}$ as $O(s^2t)$. Hence, the total time complexity to perform one loop of SAKE by the $\mathcal{DO}$ can be represented by $O(\max\{pq^2, s^2t\}) = O(pq^2)$. To show that our protocols can achieve the efficiency gains, we mainly analyze the time complexity on the $\mathcal{DO}$ side in SecSAKE I.

The main computational overhead of the $\mathcal{DO}$ is related to the procedures of encryption,

decryption (Algorithm 1, 2) and verification (Algorithm 3, 4) for outsourcing the two computation tasks above. In Algorithm 1 *Line 3*, the $\mathcal{DO}$ multiplies the data matrix by two diagonal matrices and then permutes its rows and columns. These operations take $O(pq)$ time. The *Line 5* corresponds to the reverse operations, which also take $O(pq)$ time. As discussed in Chapter 5.2, the computation in *Line 7* will only cost $O(\hat{r}pq)$ time, where $\hat{r}$ can be seen as a small constant compared with p.

Regarding the Algorithm 3, the equality test on *Line 2* essentially involves 4 matrix-vector multiplications and is repeated by l times. The required time for each verification is of the complexity of $O(lpq)$. Thus, in Cartesian sampling case, the total time cost on the $\mathcal{DO}$ side is $O(\max\{\hat{r}, l\}pq)$.

As for non-Cartesian sampling scheme, the dominant computation occurs in Algorithm 2 *Line 3* and *Line 5*. In *Line 3*, the $\mathcal{DO}$ needs to conduct one multiplication between matrix and vector and then several multiplications between vectors. Besides, *Line 5* requires the $\mathcal{DO}$ to conduct one multiplication between matrix and vector and one multiplication between vectors. Note that all of the terms in *Line 3* can be precomputed. Hence, the total runtime for Algorithm 2 should be $O(\max\{s^2, st\})$. Additionally, the batch test in Algorithm 4 takes $O(\tau st)$ time due to the τ vector-matrix multiplications in *Line 4*. The total time cost in this case is $O(\max\{\tau st, s^2\})$.

According to Theorem 5.2, the l and τ are both parameters controlling the efficiency-security tradeoff. A small value of τ or l can already guarantee negligible chance of missing the misbehaviors of the $\mathcal{CS}$. This fact can imply that $l \ll \min\{p, q\}$ and $\tau \ll \min\{s, t\}$. Thus, SecSAKE I can achieve a quadratic time complexity in each iteration on the $\mathcal{DO}$ side instead of the cubic time complexity of running SAKE, which implies large amount of efficiency gains.

5.7 Summary

In this chapter, we talk about the first protocol of our design and show its strength of security protection along with the efficiency gain. SecSAKE I only requires one single server to achieve the goals of security, its efficiency is highly dependent on the number of iterations. The $\mathcal{DO}$ has to keep online, while managing to encrypt, decrypt and interact with the $\mathcal{CS}$ in each iteration until the convergence is reached. In the next chapter, we will introduce our second protocol.

Chapter 6

SecSAKE II

In SecSAKE II, we propose to involve several more independent cloud servers, called *executor* and *shareholders*, who can communicate with each other and separately fulfill the computation tasks without leaking the information of imaging data. The $\mathcal{DO}$ only participates in the first iteration and distributes the imaging data in a privacy-preserving manner, which can save much computational overhead. A concise system overview has been given previously. In this chapter, we directly describe the protocol at each party's view and show the correctness of protocol afterwards.

6.1 Data Owner: Preparation

In this step, the $\mathcal{DO}$ generates and allocates all the required input to the $\mathcal{CS}$. We assume the sampling operator $\mathbf{D}$, the sliding window size w and the estimated rank $\hat{r}$ have all been determined before processing the computation. The first estimate of data in the k-space domain from the acquired data is still set as $\mathbf{x_0} = \mathbf{D}^\dagger \mathbf{y}$. An illustration of the required operations is given in Figure 3.

Early Encryption: After projecting the data onto the block-wise Hankel matrix $\mathbf{A_0} = H_w(\mathbf{x_0})$, the $\mathcal{DO}$ will manage to encrypt the $\mathbf{A_0}$ in exactly the same way as in Chapter 5 to obtain $\tilde{\mathbf{A}}_0'$. Then $\tilde{\mathbf{A}}_0'$ is sent to the *executor*.

Key Distribution: Denote the collection of involved key value and matrices as $K_1 = (\mathbf{P_1}, \mathbf{Q_1})$. Both the first round key K_1, the sampling matrix $\mathbf{D}$ and the estimated rank $\hat{r}$ are sent to every *shareholders*. Note that the factors η_0 and η_0' are kept secret by the $\mathcal{DO}$.

Image Splitting: Similar with the splitting procedure, the $\mathcal{DO}$ also additively splits the $\mathbf{x_0}$ in k-space domain into three shares of matrices. Specifically, given the construction of Equation (5.15), two random matrices $\mathbf{G_1}$ and $\mathbf{G_2}$ can be efficiently generated to hide $\mathbf{x_0}$. To create the convenience of the protocol design, the factor $\delta = |\eta_0 \eta_0'|$ will be multiplied

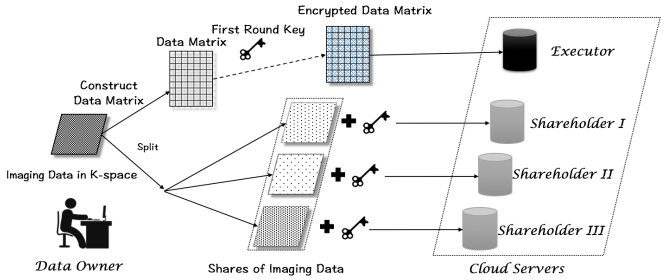

Figure 6.1: SecSAKE II: Preparation Procedures on the Data Owner Side

to the shares. Finally, the three shares of matrices $\mathbf{x_0}$ can be represented by $\mathbf{C_1} = \delta\mathbf{G_1}$, $\mathbf{C_2} = \delta\mathbf{G_2}$ and $\mathbf{C_3} = \delta(\mathbf{x_0} - \mathbf{G_1} - \mathbf{G_2})$, which are then distributed to the three *shareholders*, respectively.

6.2 Executor: SVD Computation and Results Distribution

As shown in Figure 4, the *executor* can be considered as a hub among all the cloud servers and it is supposed to shoulder the following responsibilities:

Singular Value Decomposition: After obtaining the encrypted data matrix $\tilde{\mathbf{A}}'_n$ from the $\mathcal{DO}$, the *executor* will compute the SVD by any method and get $\mathbf{U}'$, $\mathbf{\Sigma}'$ and $\mathbf{V}'$. Without the access to any part of K_1, the *executor* cannot gain any advantages over what it has in the protocol proposed in Chapter 5 so far. It is suggestive that the *executor* cannot proceed to further computation tasks.

Re-encryption: The *executor* then needs to seek help from the *shareholders*, named *shareholdes I*, *shareholder II* and *shareholder III*, respectively. In light of the fact that each of them possesses K_1, it is necessary to transform results of SVD before distributing them. Particularly, the *executor* initializes the random matrices $\mathbf{Z_1} \in \mathbb{C}^{s\times s}$ and $\mathbf{Z_2} \in \mathbb{C}^{s\times t}$ and hides the left-singular matrix $\mathbf{U}'$ and right-singular matrix $\mathbf{V}'$, respectively. More precisely, we denote:

$$\begin{aligned} \mathbf{U'_1} &= \mathbf{U'} + \mathbf{Z_1}, \mathbf{U'_2} = \mathbf{U'} - \mathbf{Z_1} \\ \mathbf{V'_1} &= \mathbf{V'} + \mathbf{Z_2}, \mathbf{V'_2} = \mathbf{V'} - \mathbf{Z_2} \end{aligned} \tag{6.1}$$

Results Distribution: The tuple ($\mathbf{U'_1}$, $\mathbf{V'_1}$, $\mathbf{\Sigma}'$) is sent to *shareholder I* and another tuple ($\mathbf{U'_2}$, $\mathbf{V'_2}$, $\mathbf{\Sigma}'$) is sent to *shareholder II*. Since the random matrices $\mathbf{Z_1}$ and $\mathbf{Z_2}$ are not correlated with K_1, this step cannot disclose any information of $\mathbf{U}''$ and $\mathbf{V}''$ to either *shareholder I* or *shareholder II* according to Equation (5.15). Meanwhile, the *executor* sends the matrices $\mathbf{Z_1}$ and $\mathbf{Z_2}$ to *shareholder III*.

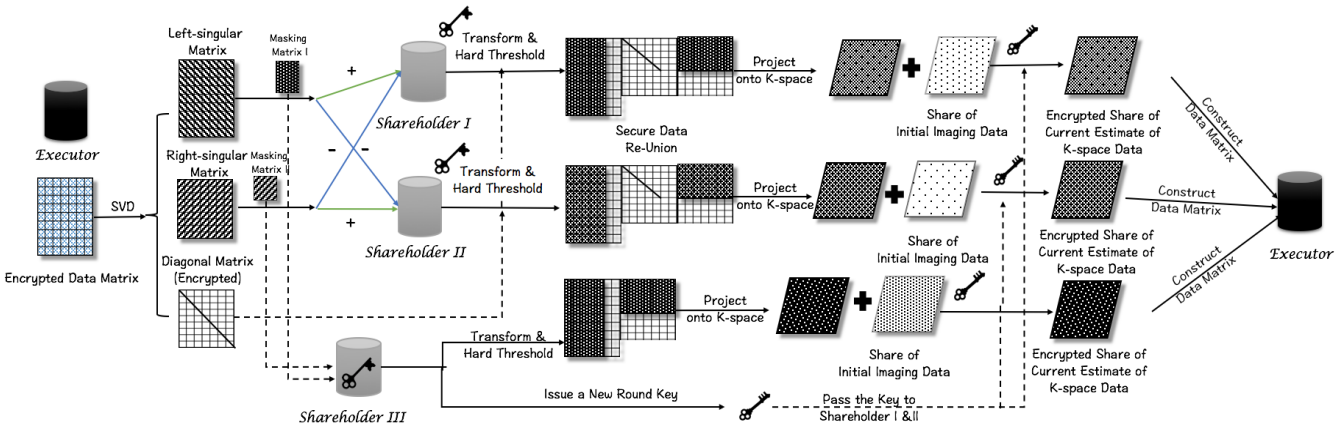

Figure 6.2: SecSAKE II: The Procedures of Computation in the $\mathcal{CS}$ within One Iteration

6.2.1 Shareholders: Parallel Processing and Cooperative Encryption

Upon receiving the input from the *executor* and $\mathcal{DO}$, the three independent *shareholders* will securely conduct the rest computation tasks in a loop. In addition, they should bear the encryption of the data before releasing it to the *executor*. According to the involved operations, this stage can be organized into three consecutive steps in a sequence. All the involved operations can be operated in a parallel manner among the three *shareholders*. We firstly describe these operations and leave the correctness of the algorithm behind. An illustration to this step is also presented in Figure 4.

K-space data Generation: In this step, both *shareholder I* and *shareholder II* are supposed to securely execute the low-rankness projection for left and right singular matrices at first. Note that the $\mathbf{U}'_\mathbf{1}$ and $\mathbf{V}'_\mathbf{1}$ can be represented by $\mathbf{P}_1\mathbf{U}'' - \mathbf{Z}_1$ and $\mathbf{Q}_1\mathbf{V}'' + \mathbf{Z}_2$. Given the estimated rank $\hat{r}$, we denote the operator of hard-thresholding as $T_{\hat{r}}$. Obviously, $T_{\hat{r}}$ is a linear operator. After receiving K_1 and the tuple ($\mathbf{U}'_\mathbf{1}$, $\mathbf{V}'_\mathbf{1}$, $\mathbf{\Sigma}'$), the *shareholder I* performs the hard-thresholding step as follows:

$$\begin{aligned}\mathbf{U}''_{1||} &= T_{\hat{r}}(\mathbf{P}_1^*\mathbf{U}'_\mathbf{1}) = T_{\hat{r}}(\mathbf{U}'' - \mathbf{P}_1^*\mathbf{Z}_1) = \mathbf{U}''_{||} - T_{\hat{r}}(\mathbf{P}_1^*\mathbf{Z}_1)\\ \mathbf{V}''_{1||} &= T_{\hat{r}}(\mathbf{Q}_1^*\mathbf{V}'_\mathbf{1}) = T_{\hat{r}}(\mathbf{V}'' + \mathbf{Q}_1^*\mathbf{Z}_2) = \mathbf{V}''_{||} + T_{\hat{r}}(\mathbf{Q}_1^*\mathbf{Z}_2)\end{aligned} \tag{6.2}$$

In this way, the non-zero values in the truncated matrix $\mathbf{U}''_{1||}$ and $\mathbf{V}''_{1||}$ can be protected by $T_{\hat{r}}(\mathbf{P}_1^*\mathbf{Z}_1)$ and $T_{\hat{r}}(\mathbf{Q}_1^*\mathbf{Z}_2)$ against the *shareholders I*, respectively. After retaining the $\hat{r}$ non-zero values in $\mathbf{\Sigma}'$, denoted as $\mathbf{\Sigma}'_{\hat{r}}$, the *shareholder I* reunites these derived matrices can be performed by matrix multiplication in the first iteration:

$$\hat{\mathbf{A}}_0^1 = \frac{1}{2}\mathbf{U}''_{1||}\mathbf{\Sigma}'_{\hat{r}}\mathbf{V}''^*_{1||} \tag{6.3}$$

In the same way, the *shareholder II* can obtain the truncated matrices $\mathbf{U}''_{2||} = \mathbf{U}''_{||} + T_{\hat{r}}(\mathbf{P}_1^*\mathbf{Z}_1)$, $\mathbf{V}''_{2||} = \mathbf{V}''_{||} - T_{\hat{r}}(\mathbf{Q}_1^*\mathbf{Z}_2)$ and then will compute:

$$\hat{\mathbf{A}}_0^2 = \frac{1}{2}\mathbf{U}''_{2||}\mathbf{\Sigma}'_{\hat{r}}\mathbf{V}''^*_{2||} \tag{6.4}$$

Correspondingly, the *shareholders III* is responsible for computing a different task:

$$\hat{\mathbf{A}}_0^3 = T_{\hat{r}}(\mathbf{P}_1^*\mathbf{Z}_1)[T_{\hat{r}}(\mathbf{Q}_1^*\mathbf{Z}_2)]^* \tag{6.5}$$

With the uniform window size w, each of the *shareholder* locally enforces the structural consistency projection to generate the k-space data by averaging the values on the anti-diagonal direction, i.e. $\hat{\mathbf{x}}_0^l = H_w^\dagger(\hat{\mathbf{A}}_0^l)$ for $l = 1, 2, 3$.

Data Matrix Regeneration: Suppose the matrix $\mathbf{N}$ has been pre-computed by each *shareholder*, where $\mathbf{N} = \mathbf{I} - \mathbf{D}\mathbf{D}^\dagger$. Before entering into the stage of data consistency projection, the l-th *shareholder* also possesses a share of the split image $\mathbf{C}_l$ and the k-space data $\hat{\mathbf{x}}_0^l$ obtained from the previous step. To gain the share of the current estimation of k-space data, the l-th *shareholder* still performs the matrix-matrix operations as follows:

$$\hat{\mathbf{x}}_1^l = \mathbf{N}\hat{\mathbf{x}}_0^l + \mathbf{C}_l \tag{6.6}$$

To enforce the block-wise Hankel matrix for the next loop, the l-th *shareholder* projects its own share $\hat{\mathbf{x}}_1^l$ onto the data matrix $\overline{\mathbf{A}}_1^l$ by H_w, i.e. $\overline{\mathbf{A}}_1^l = H_w(\hat{\mathbf{x}}_1^l)$.

Cooperative Encryption: By bringing $\overline{\mathbf{A}}_1^l$ into the second iteration, the *shareholders* needs to pass their shares to the *executor*. Before doing so, one of the *shareholders* is required to generate a new pair of keys $K_2 = (\mathbf{P_2}, \mathbf{Q_2})$ (with the same generating process with K_0) and share the it with other *shareholders*. Then, each of them individually encrypts their own share of $\overline{\mathbf{A}}_1$ as:

$$\tilde{\mathbf{A}}_1'^l = \mathbf{P}_2\overline{\mathbf{A}}_1^l\mathbf{Q}_2 \tag{6.7}$$

Finally, they all send their own share of $\tilde{\mathbf{A}}_1'^l$ to the *executor*.

Correctness: We aim to briefly show the current shares can be used in further loops. Here we show the addition of the above shares multiplied by the constant factor δ is equivalent with encrypted data matrix originated from the current estimate in the k-space domain, i.e.

$$\tilde{\mathbf{A}}_1' = \sum_{l=1}^{3}\tilde{\mathbf{A}}_1'^l = \delta\mathbf{P}_2\mathbf{A}_1\mathbf{Q}_2 \tag{6.8}$$

Proof. We firstly show that the aggregated shares after the low-rankness projection is δ times

the reconstructed data matrix, i.e. $\hat{\mathbf{A}}_0 = (1/\delta)\sum_{l=1}^{3}\hat{\mathbf{A}}_0^l$. Simply,

$$\begin{aligned}\sum_{l=1}^{3}\hat{\mathbf{A}}_0^l &= \frac{1}{2}[\mathbf{U}''_{||} - T_{\hat{r}}(\mathbf{P}_1^*\mathbf{Z}_1)]\mathbf{\Sigma}'_{\hat{r}}[\mathbf{V}''_{||} + T_{\hat{r}}(\mathbf{Q}_1^*\mathbf{Z}_2)]^* \\ &+ \frac{1}{2}[\mathbf{U}''_{||} + T_{\hat{r}}(\mathbf{P}_1^*\mathbf{Z}_1)]\mathbf{\Sigma}'_{\hat{r}}[\mathbf{V}''_{||} - T_{\hat{r}}(\mathbf{Q}_1^*\mathbf{Z}_2)]^* + T_{\hat{r}}(\mathbf{P}_1^*\mathbf{Z}_1)[T_{\hat{r}}(\mathbf{Q}_1^*\mathbf{Z}_2)]^* \\ &= \mathbf{U}''_{||}\mathbf{\Sigma}'_{\hat{r}}\mathbf{V}''^*_{||} = \delta\mathbf{U}''_{||}\mathbf{\Sigma}''_{\hat{r}}\mathbf{V}''^*_{||} = \delta\hat{\mathbf{A}}_0\end{aligned} \tag{6.9}$$

According to the definition of the $H^\dagger$, the projection onto the k-space domain relates to the average values on the anti-diagonal directions. Hence, $H^\dagger$ is an linear operator and it also satisfies that $\hat{\mathbf{x}}'_0 = (1/\delta)\sum_{l=1}^{3}\hat{\mathbf{x}}_0^l$. Thus,

$$\mathbf{x_1} = \mathbf{N}\hat{\mathbf{x}}_\mathbf{0} + \mathbf{D}^\dagger\mathbf{y} = (1/\delta)\mathbf{N}\sum_{l=1}^{3}\hat{\mathbf{x}}_0^l + \sum_{l=1}^{3}\mathbf{C}_l = \sum_{l=1}^{3}\hat{\mathbf{x}}_1^l \tag{6.10}$$

The linear operator H_w ensures $\sum_{l=1}^{3}\overline{\mathbf{A}}_1^l = \delta\mathbf{A}_1$ and the associative principle of matrix multiplication then guarantees that $\tilde{\mathbf{A}}'_1 = \sum_{l=1}^{3}\tilde{\mathbf{A}}'^l_1$. □

The Equation (6.8) allows the *executor* to get through the SVD computation and proceed to the rest of the second loop.

6.3 Data Owner: Finalization

One challenging issue of this protocol is that none of the cloud servers can determine when to stop the iterations. The $\mathcal{DO}$ can stop the iteration after a preset number of rounds. The $\mathcal{DO}$ can also alternatively download, transform and compare any two shares of estimates from one of the *shareholders* in two consecutive iterations. Afterwards, the $\mathcal{DO}$ can request all the shares of estimates from all the *shareholders* and locally aggregate the transformed shares.

6.4 Privacy Analysis

SecSAKE II involves four independent parties in the $\mathcal{CS}$, we analyze the privacy issues by inspecting the accessible matrices in each party. The *executor* in this protocol is assigned the same computation task but different input. The encrypted data matrix is provided by three *shareholders* in three shares. Since the *executor* does not have the updated key pair in the new round, each share of the data matrix can be protected the same as SecSAKE I. The *shareholders I & II* are supposed to conduct the matrix transformation given the re-encrypted results of SVD. Even though they know the round key, they cannot determine the results which have been masked by Z_1 and Z_2. In addition, the shares of initial imaging data

are well protected by the randomly splitting. As long as none of the *shareholders* can collude with another party, there is no chance for any of the *shareholder* to learn any information of the image.

6.4.1 Proof of Theorem 5.2

Proof. We separately show the probability of getting a false positive error and a false negative error in the cheating detection phase.

1) Consider the case that $\mathbf{U}'\mathbf{\Sigma}'\mathbf{V}' = \tilde{\mathbf{A}}'_n$:

$$\begin{aligned} \mathbf{U}'\mathbf{\Sigma}'(\mathbf{V}'r_j) - \tilde{\mathbf{A}}'_n r_j = \\ (\mathbf{U}'\mathbf{\Sigma}'\mathbf{V}' - \tilde{\mathbf{A}}'_n)r_j = \mathbf{0} \end{aligned} \tag{6.11}$$

Hence, the probability of returning 'false' for a correct $\tilde{\mathbf{A}}'$ is always 0.

2) Consider the other case, i.e. $\mathbf{U}'\mathbf{\Sigma}'\mathbf{V}' \neq \tilde{\mathbf{A}}'_n$:

Let $\mathbf{Z} = \mathbf{U}'\mathbf{\Sigma}'\mathbf{V}' - \tilde{\mathbf{A}}'_n$ and $\mathbf{D} = \mathbf{Z}r_j$. Then there exist at least one non-zero entry z_{vw} in $\mathbf{Z}$. Note that the vth element of $\mathbf{D}$ can be represented by:

$$d_v = z_{v1}r_1 + z_{v2}r_2 + ... + z_{vw}r_w + ... + z_{vp}r_p \tag{6.12}$$

Let $m = d_v - z_{vw}r_w$, then the chance the corresponding entry is zero can be described as:

$$\begin{aligned} P(d_v = 0) &= P(d_v = 0|m = 0)P(m = 0) + P(d_v = 0|m \neq 0)P(m \neq 0) \\ &= P(r_w = 0)P(m = 0) + P(r_w = 1, z_{vw}r_w = -m)P(m \neq 0) \\ &\leq \frac{1}{2}P(m = 0) + \frac{1}{2}(1 - P(m = 0)) = \frac{1}{2} \end{aligned} \tag{6.13}$$

Hence, $P(\mathbf{D} = \mathbf{0}) \leq \frac{1}{2}$. After repeating the above process by l times, the probability that a false result passes the Algorithm 3 is less than $\frac{1}{2^l}$. □

6.4.2 Proof of Theorem 5.3

Proof. Similarly, we show the probability of getting a false positive error and a false negative error in the cheating detection phase, respectively.

1) Consider the case that $\hat{\mathbf{N}}\hat{\mathbf{x}}'_{\mathbf{k}} = \mathbf{M_k}$:

$$\begin{aligned} \hat{\mathbf{N}}\psi - \zeta &= \hat{\mathbf{N}}\sum_{k=1}^{\tau} t_k\hat{\mathbf{x}}'_k - \sum_{k=1}^{\tau} t_k\mathbf{M}_k \\ &= \sum_{k=1}^{\tau} t_k(\hat{\mathbf{N}}\hat{\mathbf{x}}'_{\mathbf{k}} - \mathbf{M_k}) \end{aligned} \tag{6.14}$$

Hence, the probability of returning 'false' is always 0, if every included $\mathbf{M}$ is correct.

2) Consider the other case, i.e. $\hat{\mathbf{N}}\hat{\mathbf{x}}'_{\mathbf{k}} \neq \mathbf{M_k}$:
Without the loss of generality, we assume that the very first entry of $\hat{\mathbf{N}}\psi$ and ζ is different. Suppose $\mathbf{z}_k = \hat{\mathbf{N}}\hat{\mathbf{x}}'_{\mathbf{k}}[\mathbf{1}][\mathbf{1}] - \mathbf{M_k}[\mathbf{1}][\mathbf{1}]$. If it still holds that $\hat{\mathbf{N}}\psi = \zeta$, this equation can be represented by the followings:

$$\hat{\mathbf{N}}\psi_{11} - \zeta_{11} = 0 = \sum_{k=1}^{\tau} t_k \mathbf{z}_k = t_1\mathbf{z}_1 + \sum_{k=2}^{\tau} t_k \mathbf{z}_k \tag{6.15}$$

Hence,

$$t_1 = -\mathbf{z}_1^{-1}(\sum_{k=2}^{\tau} t_k \mathbf{z}_k) \tag{6.16}$$

If every t_k has been fixed except for t_1, the chance that $\hat{\mathbf{N}}\psi_{11} - \zeta_{11}$ would be equivalent with a random selection over the field $\{0,1\}^{l'}$, i.e. $\frac{1}{2^{l'}}$. Note that if there are less than $\tau - 1$ of t_k are fixed, the probability of $\hat{\mathbf{N}}\psi_{11} - \zeta_{11}$ holds will be less than $\frac{1}{2^{l'}}$. □

6.4.3 Proof of Theorem 5.4

Proof. Note that the encryption represented by Equation (5.2) can be translated into $\tilde{a}_{ij} = \eta_0\eta'_0\lambda_i a_{ij}\gamma_j$.

Suppose the probabilistic polynomial-time adversary A holds a pair of arbitrary numbers with the same length n, denoted as a^0_{ij} and a^1_{ij}. Then A outputs these two numbers to the oracle, who later choses a random bit b from $\{0,1\}$ and computes $\eta_0\eta'_0\lambda_i a^b_{ij}\gamma_j$. A then guesses the chosen bit as b' by obtaining the result from oracle. The experiment outputs 1 if $b' = b$, or 0 otherwise.

Let the possibility that A can output 1 in the above experiment to be $\frac{1}{2} + p(n)$. If the encryption function is truly random, the probability A outputs 1 is at most $\frac{1}{2} + \frac{q(n)}{2}$, where $q(n)$ is the number of oracle queries made by A. There exists a Distinguisher D who has the access to the oracle. D makes guesses on whether the encryption function is pseudorandom or truly random based on the output of the previous experiment. Then D's distinguishing probability can be described as:

$$|\frac{1}{2} + \frac{q(n)}{2} - (\frac{1}{2} + p(n))| = \frac{q(n)}{2} - p(n) \tag{6.17}$$

which is known as negligible. Hence, $p(n)$ is negligible and the advantage that A can make a correct guess is negligible over $\frac{1}{2}$. □

6.4.4 More details on Equation (5.20) and (5.22)

$$\begin{aligned}
&|Pr[D(\Re(\hat{m}_{i,j})) = 1] - Pr[D(\Re(r_{i,j})) = 1]| \\
&\quad = |Pr[\Re(\hat{m}_{i,j}) > c\Re(v_j)] \\
&\quad + Pr[\Re(\hat{m}_{i,j}) < -\Re(v_j)] - Pr[D(\Re(r_{i,j})) = 1]| \\
&\quad = |Pr[\Re(z_{i,j}) > \Re(cv_j - m_{i,j})] \\
&\quad + Pr[\Re(z_{i,j}) < -\Re(cv_j + m_{i,j})] - \frac{1}{2}]| \\
&\quad \leq K_a/2c\Re(v_j)
\end{aligned} \tag{6.18}$$

$$\begin{aligned}
\mu(\kappa) &= 1 - (1 - K_a/2c\Re(v_j))(1 - K_b/2c\Im(v_j)) \\
&\leq 1 - (1 - 2^n/2^{d+\max\{n,l\}})(1 - 2^l/2^{d+\max\{n,l\}}) \\
&\leq 2^{1-d} + 2^{-2d} \leq 2^{2-d} \leq 2^{\kappa-\max\{n,l\}-q+1}
\end{aligned} \tag{6.19}$$

6.5 Summary

We propose our second protocol of privacy-preserving SAKE. As analyzed, this protocol can achieve obvious better efficiency gain compared with our first protocol and we will show more experimental details in the next chapter. But as a tradeoff, it has to assume the non-colluding property within a group of servers. This gives the clinic more flexible choice for adopting an approach with more efficiency than the privacy guarantee.

Chapter 7

Experimental Evaluation

In this chapter, we demonstrate the conducted extensive experiments to assess the performance of our proposed protocols. The computation tasks for the $\mathcal{DO}$ is implemented by Matlab 2014b and the procedures are executed on a laptop with 2.7 GHz CPU, 8GB RAM memory and a 256GB solid drive. Our primary focus is on the evaluating the efficiency gain on the $\mathcal{DO}$ side. We test two of the real-world image benchmarks (undersampled knee and abdomens scans), which is originated from the public web page [Lus17]. We compare our protocol with original SAKE based on the the library offered by [Lus16]. To provide a more thorough study, we implement both the Cartesian and non-Cartesian sampling schemes, along with different choices on the imaging data size and sliding window size. Throughout the experiment, we take the fixed threshold of window-normalized number as 1.4 for the knee data, i.e. $\hat{r} = \lfloor 1.4 \times w^2 \rfloor$ and 1.6 for the abdomens data. The k-space data is all $3\times$ undersampled by eight coils in parallel. Since the computation process occupies most of the time cost, we ignore the communication cost between the $\mathcal{DO}$ and $\mathcal{CS}$ as suggested in [WRWW13].

$s \times t$	$w \times w$	t_s	t_{l1}	θ_1	t_{l2}	θ_2
200×200	4×4	383.26	100.62	3.81	5.83	65.74
200×200	5×5	419.34	105.39	3.98	6.91	60.69
288×288	5×5	538.10	130.42	4.13	8.44	63.74
288×288	6×6	736.75	160.90	4.58	10.03	73.45
320×288	5×5	892.43	180.74	4.94	11.08	80.54
320×288	6×6	990.14	204.55	4.84	11.98	82.64
320×320	6×6	1112.32	218.39	5.09	12.33	90.21
320×320	8×8	1384.39	250.23	5.53	11.31	122.40

Table 7.1: The Local Processing Time in Cartesian Image Reconstruction. The measured data (t_s, t_{l1} & t_{l2}) are in seconds.

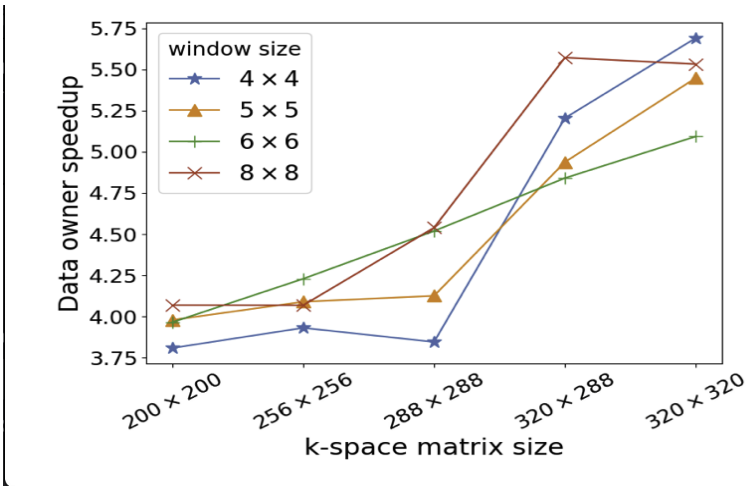

Figure 7.1: The ratio of time cost on the $\mathcal{DO}$ side between SAKE and SecSAKE I in the case of Cartesian Sampling.

7.1 Cartesian Sampling

In the Cartesian Sampling, our primary focus is on the time cost of outsourcing algorithm at the $\mathcal{DO}$ side. Some representative results are given in Table 1. The benchmark is the time cost for the iterations in SAKE only run by the $\mathcal{DO}$ and is denoted as t_s. Meanwhile, we evaluate the time spent for the local processing (encryption, decryption and verification) plus the time for performing other local operations, which is denoted as t_{l1} and t_{l2} for SecSAKE I and SecSAKE II, respectively. We assume that the t_{l1} terminate the iterations in SecSAKE II following the number of rounds in SAKE. Various size of imaging data in k-space domain $(s \times t)$ with 8 slices and size of square window $(w \times w)$ are tested. Two pairs of comparisons between t_{l1}, t_{l2} and t_s have been made by exploring the ratios $\theta_1 = t_s/t_{l1}$ and $\theta_2 = t_s/t_{l2}$. For instance, the local computation for reconstructing the MRI image with the k-space size $(320 \times 320) \times 8$ and sliding window size 6×6 in total costs the $\mathcal{DO}$ around 218.39 seconds to encrypt, decrypt the data matrix and perform the other operations, e.g. structural and data consistency projection, etc. Note that the size of constructed data matrix in each iteration is correspondingly as large as 288×97969. The $\mathcal{DO}$ enjoys a high efficiency gain when outsourcing the task of SVD to the $\mathcal{CS}$. As shown in Figure 5, the efficiency gain can be better achieved when the size of imaging data or sliding window goes larger. Moreover, most of the computations in SecSAKE II on the $\mathcal{DO}$ side in within one loop of SecSAKE I. The speedup depends on the extra time on randomly splitting the k-space data and the stopping criteria (e.g. the fixed iteration times herein). As shown in Figure 6,the SecSAKE II can realize a speedup of more than 100× compared with the original SAKE. The image

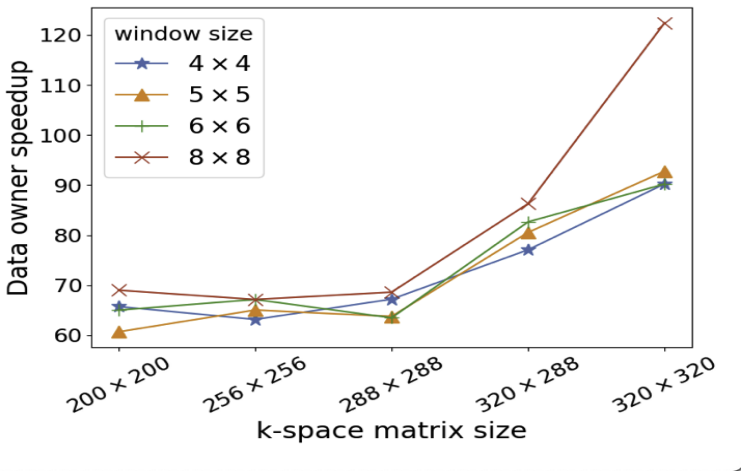

Figure 7.2: The ratio of time cost on the $\mathcal{DO}$ side between SAKE and SecSAKE II in the case of Cartesian Sampling.

reconstructed from the experiment is shown in Figure 7 (after the inverse Fourier Transform from k-space domain).

7.2 Non-Cartesian Sampling

We also conduct the experiment on dataset characterized as non-Cartesian sampling. The $\mathcal{DO}$ is expected to execute an extra matrix multiplication for data consistency projection in SAKE. One primary issue that we care about is whether the protocol offers the comparative speedup with the case of Cartesian Sampling. We also consider the practical case that the data consistency projection is conducted twice to achieve a better reconstruction result in spiral non-Cartesian sampling. Considering the size of sliding window doesn't directly affect the speed of data consistency projection, we fix the size of sliding window as 6. In the experiment, we compare the local processing time t'_{l1} with the time cost for performing SAKE only when altering the size of imaging data in k-space, and calculate the ratio $\theta_3 = t'_s/t'_{l1}$ to represent the speed-up. As shown in Figure 8, the SecSAKE II can unsurprisingly achieve higher speedup on the $\mathcal{DO}$ side because the $\mathcal{DO}$ doesn't have to participate in the data consistency projection. In contrast, the speedup of SecSAKE I falls down, which may mainly due to the lower efficiency gain in Data Consistency Projection compared with Secure Singular Value Decomposition.

7.2.1 Effectiveness

Our experiment also shows the effectiveness of the protocols, i.e. the $\mathcal{DO}$ can reconstruct clear diagnostic images by running SecSAKE. We select SecSAKE I and pick up the sagittal

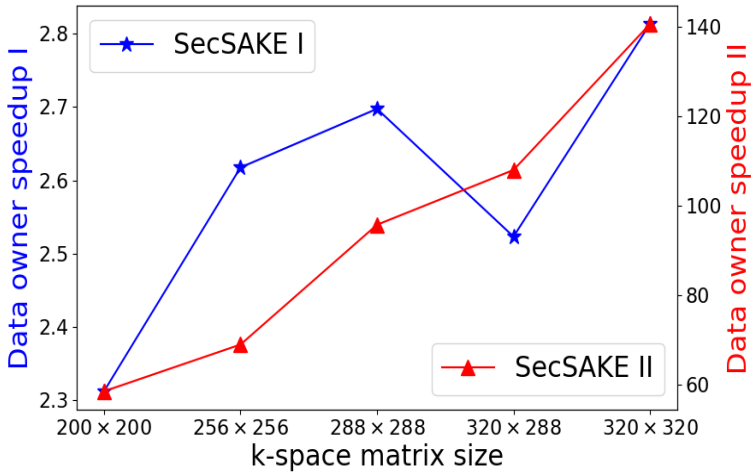

Figure 7.3: The ratio of time cost on the $\mathcal{DO}$ side between SAKE and SecSAKE I (Blue Lines), & between SAKE and SecSAKE II (Red Lines) in the case of Non-Cartesian Sampling.

Figure 7.4: A 320 × 320 sagittal image on of knee (lateral side). Unreconstructed image (Left); Reconstructed image after 15 iterations (Middle) and 30 iterations (Right) of SecSAKE I.

image on the lateral side of the knee as the example to illustrate. As can be seen in Figure 7, the scanned object is visually unrecognizable in the unreconstructed image on the left. Some of the aliasing artifacts have been eliminated during the iteration and the anterior cruciate ligament comes into view in the middle image (marked by red circle). However, it still remains difficult to observe the articular cartilage in the gray area around the bone (marked by the yellow circle) until more iterations are done. As presented in the last image, the details of the key area are clear enough for further clinical diagnosis.

7.3 Summary

In this chapter, we show both the efficiency gain and the effectiveness of both of the proposed protocols by showing the results of comprehensive experiments. Note that the environment and power of computing in the clinic may depend on many factors, the actual speed-up ratio may vary in the practical case.

Chapter 8

Conclusion and Future Research

In this thesis, we for the first time explore the problem of privacy-preserving outsourcing the image reconstruction process in SAKE, which is one of the state-of-art algorithms in clinical implementation. Addressing its most expensive computation task, we propose SecSAKE, which includes two protocols with extra emphasis on security and efficiency, respectively. Our first protocol can achieve column-wise computationally indistinguishable under CPA for both the imaging data and sampling operator. In each round, the computation overhead of the clinic can be reduced to quadratic complexity. Our second protocol further reduce the computation time on the clinic side by utilizing multi-server architecture of the cloud. The privacy of imaging data can also be well protected if none of the cloud servers colludes with the other. A theoretical analysis is then illustrated to prove that both of the protocols are secure and efficient. We also conduct extensive experiments to demonstrate the practical efficiency and effectiveness of SecSAKE. Here are some possible future research problems:

1) *Can SecSAKE be further accelerated while maintaining the security?*
   In our protocol, we actually decompose the computation of SAKE into several sub-problems. If there would be a chance to transform the computation of the optimization problem entirely, we believe that SAKE can be processed more securely and efficiently.

2) *Is there any other chance that MRI reconstruction can be securely accelerated?*
   Yes. As we illustrated in the previous chapters, there are many other parallel imaging techniques that leverage different form of the imaging data. Other than the parallel imaging methods, there are many signal processing-based methods. They focuse on the prior information of magnetic resonance images, and use them as canonical items to constrain the reconstruction process, which have the advantage of being unrestricted by physical, physiological and hardware conditions. Finally, the magnetic resonance images are reconstructed from the under-sampled k-space data through a

series of optimization algorithms. At present, there are many researches on the use of signal processing in magnetic resonance image reconstruction. The most representative research results are embodied in Compressed Sensing (CS) [D$^+$06], which uses non-coherent under-sampling and image sparsity for fast magnetic Resonance imaging, such as wavelet [CPBBC11], total variation [BUF07], joint total variation [CLH13], non-local total variation [LWCY11], and dictionary learning [Wel16] are used to improve the sparseness of the magnetic resonance image to be reconstructed, thus achieving a higher acceleration factor.

3) *What will be the main obstacles when designing the secure outsourcing solutions?*

Adaptive and flexible encryption design . Currently available outsourcing solutions mostly focus on one or two encryption (or transformation) approaches for a given computational task. The security parameter is often not explicitly given and the level of security is not tunable, but rather design choices were made to provide a tradeoff between efficiency and security. Engineering tasks, however, may have diverse requirements with some of them emphasizing fast computation over security, while others favoring stronger security guarantees. Thus, designs that allow for adaptive selection of security parameters and models with formalized analysis may be an interesting direction to explore.

Parallel computations In some of the existing schemes, encryption of different portions of the original data is not correlated. Hence, the encryption can be carried out in parallel to reduce client's computation time. This topic is rarely discussed in existing literature. Additionally, designing a parallel computing algorithm may open new interesting optimization possibilities and computational savings on the cloud side. Future solutions should take this design factor into consideration, with the goal of achieving larger time savings for the client.

Secure computation outsourcing with dynamic data Dynamic data analysis is becoming more and more popular in modern data mining research. Besides transforming or encrypting a computational task in its entirety, there is a need to be able to handle streaming or quickly changing data, where the speed of the response may be prioritized over its precision. Outsourcing schemes are often necessary when a fast real-time response is required, such as for traffic monitoring or route planning. Security and efficiency properties may need to be revisited for such dynamic environments.

Bibliography

[Ads14] A. Adshead. Data set to grow 10-fold by 2020 as internet of things takes off, April 2014.

[AF10] M. Atallah and K. Frikken. Securely outsourcing linear algebra computations. In *ASIACCS*. ACM, 2010.

[AL05] M. Atallah and J. Li. Secure outsourcing of sequence comparisons. *IJISP*, 4(4):277–287, 2005.

[APRS02] M. Atallah, K. Pantazopoulos, J. Rice, and E. Spafford. Secure outsourcing of scientific computations. *Advances in Computers*, 54:215–272, 2002.

[ASP13] Jacob Alperin-Sheriff and Chris Peikert. Practical bootstrapping in quasilinear time. In *Advances in Cryptology–CRYPTO 2013*, pages 1–20. Springer, 2013.

[ASP14] Jacob Alperin-Sheriff and Chris Peikert. Faster bootstrapping with polynomial error. In *CRYPTO*, pages 297–314. Springer, 2014.

[BA08] D. Benjamin and M. Atallah. Private and cheating-free outsourcing of algebraic computations. In *Privacy, Security and Trust, 2008.*, pages 240–245. IEEE, 2008.

[BA12] M. Blanton and M. Aliasgari. Secure outsourced computation of iris matching. *JCS*, 20(2-3), 2012.

[BFG+17] Guillaume Bonnoron, Caroline Fontaine, Guy Gogniat, Vincent Herbert, Vianney Lapôtre, Vincent Migliore, and Adeline Roux-Langlois. Somewhat/fully homomorphic encryption: Implementation progresses and challenges. In *C2SI*, pages 68–82. Springer, 2017.

[BGG+14] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryp-

tion, arithmetic circuit abe and compact garbled circuits. In *EUROCRYPT*, pages 533–556. Springer, 2014.

[BGV12] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *The 3rd Innovations in Theoretical Computer Science Conference*, pages 309–325. ACM, 2012.

[BHKR13] M. Bellare, V. T. Hoang, S. Keelveedhi, and P. Rogaway. Efficient garbling from a fixed-key blockcipher. In *IEEE Symposium on Security and Privacy (SP)*, pages 478–492. IEEE, 2013.

[BHR12] M. Bellare, V. T. Hoang, and P. Rogaway. Foundations of garbled circuits. In *ACM Conference on Computer and Communications Security*, pages 784–796. ACM, 2012.

[BLH02] Mark Bydder, David J Larkman, and Joseph V Hajnal. Generalized smash imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 47(1):160–170, 2002.

[BLLN13] J. Bos, K. Lauter, J. Loftus, and M. Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In *IMA Int. Conf.*, pages 45–64. Springer, 2013.

[Bra12] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *CRYPTO*. Springer, 2012.

[BUF07] Kai Tobias Block, Martin Uecker, and Jens Frahm. Undersampled radial mri with multiple coils. iterative image reconstruction using a total variation constraint. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 57(6):1086–1098, 2007.

[BV11] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. In *FOCS 2011, IEEE*, 2011.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-based fhe as secure as pke. In *ITCS*, pages 1–12. ACM, 2014.

[Cad88] James A Cadzow. Signal enhancement-a composite property mapping algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(1):49–62, 1988.

[CAF13] Ruichuan Chen, Istemi Ekin Akkus, and Paul Francis. Splitx: High-performance private analytics. *ACM SIGCOMM Computer Communication Review*, 43(4):315–326, 2013.

[CCK+13] J. Cheon, J. Coron, J. Kim, M. Lee, T. Lepoint, M. Tibouchi, and A. Yun. Batch fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 315–335. Springer, 2013.

[CHL+15] X. Chen, X. Huang, J. Li, J. Ma, W. Lou, and D. Wong. New algorithms for secure outsourcing of large-scale systems of linear equations. *Transactions on Information Forensics and Security*, 10(1), 2015.

[CLH13] Chen Chen, Yeqing Li, and Junzhou Huang. Calibrationless parallel mri with joint total variation regularization. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 106–114. Springer, 2013.

[CLM+14] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou. New algorithms for secure outsourcing of modular exponentiations. *IEEE Transactions on Parallel and Distributed Systems*, 25(9):2386–2396, 2014.

[CLT14] J. Coron, T. Lepoint, and M. Tibouchi. Scale-invariant fully homomorphic encryption over the integers. In *International Workshop on Public Key Cryptography*, pages 311–328. Springer, 2014.

[CMN11] J. Coron, A. Mandal, and M. Naccache, D.and Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In *Annual Cryptology Conference*, pages 487–504. Springer, 2011.

[CNT12] J. Coron, D. Naccache, and M. Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 446–464. Springer, 2012.

[CPBBC11] Lotfi Chaâri, Jean-Christophe Pesquet, Amel Benazza-Benyahia, and Philippe Ciuciu. A wavelet-based regularized reconstruction algorithm for sense parallel mri with applications to neuroimaging. *Medical image analysis*, 15(2):185–201, 2011.

[CXLC14] F. Chen, T. Xiang, X. Lei, and J. Chen. Highly efficient linear regression outsourcing to a cloud. *IEEE Transactions on Cloud Computing*, 2(4):499–508, 2014.

[CXY14] F. Chen, T. Xiang, and Y. Yang. Privacy-preserving and verifiable protocols for scientific computation outsourcing to the cloud. *Journal of Parallel and Distributed Computing*, 74(3):2141–2151, 2014.

[D+06] David L Donoho et al. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[DDS14] W. Dai, Y. Doröz, and B. Sunar. Accelerating ntru based homomorphic encryption using gpus. In *HPEC*. IEEE, 2014.

[DGGS12] Anagha Deshmane, Vikas Gulani, Mark A Griswold, and Nicole Seiberlich. Parallel mr imaging. *Journal of Magnetic Resonance Imaging*, 36(1):55–72, 2012.

[DHS14] Y. Doröz, Y. Hu, and B. Sunar. Homomorphic aes evaluation using ntru. *IACR*, 2014.

[DMJ10] W. Du, M. Murugesan, and J. Jia. Uncheatable grid computing. In *Algorithms and theory of computation handbook*, pages 30–30. Chapman & Hall/CRC, 2010.

[DZL16] J. Duan, J. Zhou, and Y. Li. Secure and verifiable outsourcing of nonnegative matrix factorization (nmf). In *The 4th ACM Workshop on Information Hiding and Multimedia Security*, pages 63–68. ACM, 2016.

[ElG85] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[ESJ14] Y. Elmehdwi, B. Samanthula, and W. Jiang. Secure k-nearest neighbor query over encrypted data in outsourced environments. In *ICDE*, pages 664–675. IEEE, 2014.

[FG12] D. Fiore and R. Gennaro. Publicly verifiable delegation of large polynomials and matrix computations, with applications. In *CCS*, pages 501–512. ACM, 2012.

[Fre77] Rusins Freivalds. Probabilistic machines can use less running time. In *IFIP congress*, volume 839, page 842, 1977.

[Gen09a] C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

[Gen09b] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC, 2009*, volume 9, pages 169–178, 2009.

[GGP10] R. Gennaro, C. Gentry, and B. Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *CRYPTO*, pages 465–482. Springer, 2010.

[GH11a] C. Gentry and S. Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In *FOCS*, pages 107–109. IEEE, 2011.

[GH11b] C. Gentry and S. Halevi. Implementing gentry's fully-homomorphic encryption scheme. In *EUROCRYPT*, pages 129–148. Springer, 2011.

[GHS12a] C. Gentry, S. Halevi, and N. Smart. Better bootstrapping in fully homomorphic encryption. In *International Workshop on Public Key Cryptography*, pages 1–16. Springer, 2012.

[GHS12b] C. Gentry, S. Halevi, and N. Smart. Fully homomorphic encryption with polylog overhead. In *EUROCRYPT*. Springer, 2012.

[GHS12c] C. Gentry, S. Halevi, and N. Smart. Homomorphic evaluation of the aes circuit. In *CRYPTO 2012*. Springer, 2012.

[Gil10] Jonathan Gillard. Cadzow's basic algorithm, alternating projections and singular spectrum analysis. *Statistics and its interface*, 3(3):335–343, 2010.

[GJH+02] Mark A Griswold, Peter M Jakob, Robin M Heidemann, Mathias Nittka, Vladimir Jellus, Jianmin Wang, Berthold Kiefer, and Axel Haase. Generalized autocalibrating partially parallel acquisitions (grappa). *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 47(6):1202–1210, 2002.

[GKR08] S. Goldwasser, Y. Kalai, and G. Rothblum. Delegating computation: interactive proofs for muggles. In *STOC*, pages 113–122. ACM, 2008.

[GM01] P. Golle and I. Mironov. Uncheatable distributed computations. In *RSA*, pages 425–440. Springer, 2001.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *STOC*. ACM, 1987.

[GVW15] S. Gorbunov, V. Vaikuntanathan, and D. Wichs. Leveled fully homomorphic signatures from standard lattices. In *STOC*, pages 469–477. ACM, 2015.

[GW13] A. Gentry, C.and Sahai and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO 2013*, pages 75–92. Springer, 2013.

[HF17] Vincent Herbert and Caroline Fontaine. Software implementation of 2-depth pairing-based homomorphic encryption scheme. *IACR Cryptology ePrint Archive*, 2017:91, 2017.

[HGI07] Michael Holmes, Alexander Gray, and Charles Isbell. Fast svd for large-scale matrices. In *Workshop on Efficient Machine Learning at NIPS*, volume 58, pages 249–252, 2007.

[HL05] S. Hohenberger and A. Lysyanskaya. How to securely outsource cryptographic computations. In *TCC*, pages 264–282. Springer, 2005.

[HS14] S. Halevi and V. Shoup. Helib-an implementation of homomorphic encryption, 2014.

[HS15] S. Halevi and V. Shoup. Bootstrapping for helib. In *EUROCRYPT*, pages 641–670. Springer, 2015.

[KGV16] Alhassan Khedr, Glenn Gulak, and Vinod Vaikuntanathan. Shield: scalable homomorphic implementation of encrypted data-classifiers. *IEEE Transactions on Computers*, 65(9):2848–2858, 2016.

[LATV12] A. López-Alt, E. Tromer, and V. Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *STOC*, pages 1219–1234. ACM, 2012.

[LC10] K. Lin and M. Chen. Privacy-preserving outsourcing support vector machines with random transformation. In *The 16th SIGKDD*, pages 363–372. ACM, 2010.

[LDSL16] Weixian Liao, Wei Du, Sergio Salinas, and Pan Li. Efficient privacy-preserving outsourcing of large-scale convex separable programming for smart cities. In *High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016 IEEE 18th International Conference on*, pages 1349–1356. IEEE, 2016.

[LKV13] Michael Lustig, Kurt Keutzer, and Shreyas Vasanawala. Introduction to parallelizing compressed sensing magnetic resonance imaging. *The Berkeley par lab:*

progress in the parallel computing landscape. Redmond, WA: Microsoft Corporation, pages 105–139, 2013.

[LLH+13] X. Lei, X. Liao, T. Huang, H. Li, and C. Hu. Outsourcing large matrix inversion computation to a public cloud. *IEEE Transactions on Cloud Computing*, 1(1):1–1, 2013.

[LLHH14] X. Lei, X. Liao, T. Huang, and F. Heriniaina. Achieving security, robust cheating resistance, and high-efficiency for outsourcing large matrix multiplication computation to a malicious cloud. *Information Sciences*, 280:205–217, 2014.

[LLHL15] X. Lei, X. Liao, T. Huang, and H. Li. Cloud computing service: The case of large matrix determinant computation. *IEEE Transactions on Services Computing*, 8(5):688–700, 2015.

[LN07] David J Larkman and Rita G Nunes. Parallel magnetic resonance imaging. *Physics in Medicine & Biology*, 52(7):R15, 2007.

[LP10] Michael Lustig and John M Pauly. Spirit: iterative self-consistent parallel imaging reconstruction from arbitrary k-space. *Magnetic resonance in medicine*, 64(2):457–471, 2010.

[Lus16] Michael Lustig. Espirit: Reference implementation of compressed sensing and parallel imaging in matlab, March 2016.

[Lus17] Michael Lustig. Mr datasets for compressed sensing, December 2017.

[LWCY11] Dong Liang, Haifeng Wang, Yuchou Chang, and Leslie Ying. Sensitivity encoding reconstruction with nonlocal total variation regularization. *Magnetic resonance in medicine*, 65(5):1384–1392, 2011.

[MNPS04] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella. Fairplay – a secure two-party computation system. In *USENIX Security Symposium*, 2004.

[Moh11] P. Mohassel. Efficient and secure delegation of linear algebra. *IACR*, 2011:605, 2011.

[MR10] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Chapman & Hall/CRC, 2010.

[NCL+14] H. Nie, X. Chen, J. Li, J. Liu, and W. Lou. Efficient and verifiable algorithm for secure outsourcing of large-scale linear programming. In *AINA*, pages 591–596. IEEE, 2014.

[NLV11] M. Naehrig, K. Lauter, and V. Vaikuntanathan. Can homomorphic encryption be practical? In *Cloud computing security workshop*, pages 113–124. ACM, 2011.

[NWI+13] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft. Privacy-preserving ridge regression on hundreds of millions of records. In *SP*, pages 334–348. IEEE, 2013.

[Pai99] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, pages 223–238. Springer, 1999.

[PG13] T. Pöppelmann and T. Güneysu. Towards practical lattice-based public-key encryption on reconfigurable hardware. In *SAC*, pages 68–85. Springer, 2013.

[PV15] Marie Paindavoine and Bastien Vialla. Minimizing the number of bootstrappings in fully homomorphic encryption. In *SAC*, pages 25–43. Springer, 2015.

[PWSB99] Klaas P Pruessmann, Markus Weiger, Markus B Scheidegger, and Peter Boesiger. Sense: sensitivity encoding for fast mri. *Magnetic resonance in medicine*, 42(5):952–962, 1999.

[QYR+14] Z. Qin, J. Yan, K. Ren, C. Chen, and C. Wang. Towards efficient privacy-preserving image feature extraction in cloud computing. In *ACMMM*, pages 497–506. ACM, 2014.

[RSA78] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[Sen13] J. Sen. Security and privacy issues in cloud computing. *Architectures and Protocols for Secure Information Technology Infrastructures*, pages 1–45, 2013.

[Sha49] Claude Elwood Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[SHS+15] E. Songhori, S. Hussain, A.-R. Sadeghi, T. Schneider, and F. Koushanfar. Tinygarble: Highly compressed and scalable sequential garbled circuits. In *IEEE Symposium on Security and Privacy (SP)*, pages 411–428. IEEE, 2015.

[Sio08] R. Sion. Towards secure data outsourcing. In *Handbook of Database Security*, pages 137–161. Springer, 2008.

[SLC+17] Sergio Salinas, Changqing Luo, Xuhui Chen, Weixian Liao, and Pan Li. Efficient secure outsourcing of large-scale sparse linear systems of equations. *IEEE Transactions on Big Data*, 2017.

[SLCL15] S. Salinas, C. Luo, X. Chen, and P. Li. Efficient secure outsourcing of large-scale linear systems of equations. In *INFOCOM*, pages 1035–1043. IEEE, 2015.

[SLLL16] S. Salinas, C. Luo, W. Liao, and P. Li. Efficient secure outsourcing of large-scale quadratic programs. In *ACM Asia Conf. Comput. Commun. Secur.*, pages 281–292. ACM, 2016.

[SLO+14] Peter J Shin, Peder EZ Larson, Michael A Ohliger, Michael Elad, John M Pauly, Daniel B Vigneron, and Michael Lustig. Calibrationless parallel imaging reconstruction based on structured low-rank matrix completion. *Magnetic resonance in medicine*, 72(4):959–970, 2014.

[SM97] Daniel K Sodickson and Warren J Manning. Simultaneous acquisition of spatial harmonics (smash): fast imaging with radiofrequency coil arrays. *Magnetic resonance in medicine*, 38(4):591–603, 1997.

[SRBW18] Zihao Shan, Kui Ren, Marina Blanton, and Cong Wang. Practical secure computation outsourcing: a survey. *ACM Computing Surveys (CSUR)*, 51(2):31, 2018.

[SV10] N. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *International Workshop on Public Key Cryptography*, pages 420–443. Springer, 2010.

[SV14] N. Smart and F. Vercauteren. Fully homomorphic simd operations. *Designs, codes and cryptography*, pages 1–25, 2014.

[ULM+14] Martin Uecker, Peng Lai, Mark J Murphy, Patrick Virtue, Michael Elad, John M Pauly, Shreyas S Vasanawala, and Michael Lustig. Espirit?an eigenvalue approach to autocalibrating parallel mri: where sense meets grappa. *Magnetic resonance in medicine*, 71(3):990–1001, 2014.

[VDGHV10] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 24–43. Springer, 2010.

[Wel16] Daniel Weller. Reconstruction with dictionary learning for accelerated parallel magnetic resonance imaging. In *2016 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, pages 105–108. IEEE, 2016.

[WHC+12] W. Wang, Y. Hu, L. Chen, X. Huang, and B. Sunar. Accelerating fully homomorphic encryption using gpu. In *HPEC*, pages 1–5. IEEE, 2012.

[WHG+14] Katherine L Wright, Jesse I Hamilton, Mark A Griswold, Vikas Gulani, and Nicole Seiberlich. Non-cartesian parallel imaging reconstruction. *Journal of Magnetic Resonance Imaging*, 40(5):1022–1040, 2014.

[WRW11] C. Wang, K. Ren, and J. Wang. Secure and practical outsourcing of linear programming in cloud computing. In *INFOCOM*, pages 820–828. IEEE, 2011.

[WRW16] C. Wang, K. Ren, and J. Wang. Secure optimization computation outsourcing in cloud computing: A case study of linear programming. *IEEE Transactions on Computers*, 65(1):216–229, 2016.

[WRWW13] C. Wang, K. Ren, J. Wang, and Q. Wang. Harnessing the cloud for securely outsourcing large-scale systems of linear equations. *IEEE Transactions on Parallel and Distributed Systems*, 24(6), 2013.

[WZRR13] C. Wang, B. Zhang, K. Ren, and J. Roveda. Privacy-assured outsourcing of image reconstruction service in cloud. *IEEE Transactions on Emerging Topics in Computing*, 1(1):166–177, 2013.

[XAG17] G. Xu, G. Amariucai, and Y. Guan. Delegation of computation with verification outsourcing: Curious verifiers. *IEEE Transactions on Parallel and Distributed Systems*, 28(3):717–730, 2017.

[Yao82] A. Yao. Protocols for secure computations. In *FOCS*, pages 160–164. IEEE, 1982.

[Yao86] A. Yao. How to generate and exchange secrets. In *FOCS*, pages 162–167. IEEE, 1986.

[YLW+16] Y. Yu, Y. Luo, D. Wang, S. Fu, and M. Xu. Efficient, secure and non-iterative outsourcing of large-scale systems of linear equations. In *ICC*, pages 1–6. IEEE, 2016.

[YLX13] B. Yao, F. Li, and X. Xiao. Secure nearest neighbor revisited. In *ICDE*, pages 733–744. IEEE, 2013.

[ZB14] Y. Zhang and M. Blanton. Efficient secure and verifiable outsourcing of matrix multiplications. In *International Conference on Information Security*, pages 158–178, 2014.

[ZL15] L. Zhou and C. Li. Outsourcing large-scale quadratic programming to a public cloud. *IEEE Access*, 3, 2015.

[ZL16] L. Zhou and C. Li. Outsourcing eigen-decomposition and singular value decomposition of large matrix to a public cloud. *IEEE Access*, 4:869–879, 2016.