Secure Tensor Decomposition Using Fully Homomorphic Encryption Scheme

Liwei Kuang, Laurence T. Yang, Jun Feng, and Mianxiong Dong

Abstract—As the rapidly growing volume of data are beyond the capabilities of many computing infrastructures, to securely process them on cloud has become a preferred solution which can both utilize the powerful capabilities provided by cloud and protect data privacy. This paper puts forward a new approach to securely decompose tensor, the mathematical model widely used in data-intensive applications, to a core tensor and some truncated orthogonal bases. The structured, semi-structured as well as unstructured data are all transformed to low-order sub-tensors which are then encrypted using the fully homomorphic encryption scheme. A unified high-order cipher tensor model is constructed by collecting all the cipher sub-tensors and embedding them to a base tensor space. The cipher tensor is decomposed through a proposed secure algorithm, in which the square root operations are eliminated during the Lanczos procedure. The paper makes an analysis of the secure algorithm in terms of time consumption, memory usage and decomposition accuracy. Experimental results reveals that this approach can securely decompose tensor models. With the advancement of fully homomorphic encryption scheme, the proposed secure tensor decomposition method is expected to be widely applied on cloud for privacy-preserving data processing.

Index Terms—Tensor Decomposition, Fully Homomorphic Encryption, Lanczos Method, Cloud.

1 Introduction

The size of data in many fields is rapidly increasing towards Terabyte level or even Petabyte level, and the data structures are becoming more varied. The large scale heterogeneous data have posed great challenges on current computing infrastructures, and novel approaches are in urgent need to address them. Cloud Computing [1] is a model that can enable ubiquitous and convenient access to a large pool of configurable computing resources such as platforms, softwares and services. A cloud infrastructure is the collection of hardware and software which can provide capabilities to the consumers on a pay-per-use or charge-per-use basis. It is a quite feasible approach to upload the large scale data to cloud for deeply processing and mining such as dimensionality reduction [2], classification [3], and prediction [4]. However, carrying out such types of tasks on cloud may cause a series of security problems including loss of privacy, disclosure of business information, unauthorized tampering, etc. Therefore, the study of secure data mining and data analyzing on cloud is highly necessary as it is an important method to extract valuable information from the large scale heterogeneous data.

The fully homomorphic encryption scheme, which was suggested in 1978 by Rivest, Adleman, and Dertouzos [5], allows a certain types of operations to be performed on the cyphertext to generate an encrypted result, of which the decryption is identical to the result generated by directly carrying out operations on the plaintext. The ideal lattice based scheme [6] proposed by Gentry in 2009 solves the problem of limited number of operations of fully homomorphic encryption, which paves the way for trusted computing on cloud. The Learning with Errors (LWE) scheme reported in [7] is more practical to be employed in data-intensive applications. Although the previously mentioned schemes provide both additive and multiplicative homomorphisms, they can cause decryption errors when be used by algorithms implemented with non-homomorphic operations such as square root and division, which are frequently carried out during data processing.

Many heterogenous data are modeled as tensor [8, 9], a high-dimension matrix used in a large number of applications. Tensor decomposition is a powerful tool to extract valuable information from large scale raw data. The decomposition is computationally expensive and is strongly suggested to be performed on cloud. Therefore, it is necessary to investigate approaches for secure tensor decomposition on cloud and address the challenges caused by non-homomorphic operations. However, little research has been devoted to this type of method.

This research presents a new computing approach which can securely decompose the tensor model generated from large scale heterogeneous data. The major contributions are summarized as follows.

- We present a holistic framework to address the problem of secure tensor decomposition on cloud.

- *L. Kuang and J. Feng are with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. E-mail: kuanglw@139.com, junfeng989@gmail.com.*
- *L.T. Yang is with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, and Department of Computer Science, St. Francis Xavier University, Antigonish, NS, Canada. Email: ltyang@gmail.com.*
- *M. Dong is with the Department of Information and Electronic Engineering, Muroran Institute of Technology, $27-1$ Mizumoto-cho, Muroran, Hokkaido, $050-8585$, Japan. Email: mx.dong@ieee.org.*

The framework not only allows us to utilize the powerful computational capabilities of the cloud, but also ensures data security during the process of tensor decomposition.

- We introduce a Unified Cipher Tensor (*UCT*) model for heterogeneous data representation. The detailed procedures of how to encrypt the low-order sub-tensors constructed from heterogeneous data as cipher counterparts using the fully encryption scheme, and how to embed them to a base tensor space to generate a unified cipher tensor model are illustrated in this paper.
- We propose to employ the Lanczos method to decompose the generated cipher tensor model to a core tensor and some truncated orthogonal bases. A secure tensor decomposition algorithm is designed in which the non-homomorphic square root operations are removed during the Lanczos procedure. Theoretical analyses of the algorithm in terms of time consumption, memory usage as well as decomposition accuracy are provided.

The rest of this paper is organized as follows. Section 2 recalls the preliminaries. In Section 3, the problem of secure tensor decomposition is formalized and a solution framework is illustrated. Section 4 explores the method to represent the heterogeneous data as a unified cipher tensor. A Lanczos based secure tensor decomposition algorithm is proposed in Section 5. Performance of the proposed approach is evaluated in Section 6. After recalling the related works in Section 7, we offer the conclusion in Section 8.

2 Preliminaries

This section provides the preliminaries on tensor decomposition, fully homomorphic encryption, and Lanczos method. Some symbols frequently used in this paper are demonstrated in Table 1.

2.1 Tensor Decomposition

Tensor model is a type of high-dimension matrix used in a large number of applications [8]. High-Order Singular Value Decomposition (HO-SVD) [10] is an approach that can factorize the tensor model to a core tensor and some truncated orthogonal matrices. Let $\mathrm{T} \in R^{I_1 \times I_2 \times \ldots \times I_N}$ denote an N-th order tensor model, S and $\hat{T}$ refer to the core tensor and approximate tensor respectively, then the HO-SVD method is defined as

$$\begin{aligned} &S = T \times_1 {U_1}^{\mathrm{T}} \times_2 {U_2}^{\mathrm{T}} \ldots \times_N {U_N}^{\mathrm{T}}, \\ &\hat{T} = S \times_1 U_1 \times_2 U_2 \ldots \times_N U_N. \end{aligned} \tag{1}$$

The i-mode product $T \times_i U,\ 1 \leq i \leq N$, of tensor T by matrix U in Eq. (1) is defined as

$$\begin{aligned} &(T \times_i U)_{j_1 j_2 \ldots j_{i-1} k_i j_{i+1} \ldots j_N} \\ &= \sum_{j_i=1}^{I_i} (t_{j_1 j_2 \ldots j_{i-1} j_i j_{i+1} \ldots j_N} \times u_{k_i j_i}), \end{aligned} \tag{2}$$

TABLE 1
Table of symbols.

Symbol	Definition
T	initial tensor
S	core tensor
$\hat{T}$	approximate tensor
$T_{(i)}$	i-mode tensor unfolding
$Sym(T_{(i)})$	symmetric matrix generated with $T_{(i)}$
$D_u,\ D_{semi},\ D_s$	unstructured, semi-structured, structured data
L	tridiagonal matrix
$\alpha,\ \beta$	elements of the tridiagonal matrix
$\times_i$	i-mode product of a tensor by a matrix
$\mathcal{R}$	set of real numbers
$\mathcal{Z}$	set of integers
R ($R[x]$)	ring (polynomial ring)
m	plaintext
c	ciphertext
χ	discrete gauss distribution
e	randomly selected error from χ
$q,\ p$	big prime integers
Enc (Dec)	encryption (decryption) function
Ψ^E	cipher data of Ψ, namely $\Psi^E = Enc(\Psi)$

where $t_{j_1 j_2 \ldots j_{i-1} j_i j_{i+1} \ldots j_i}$ and $u_{k_i j_i}$ refer to the elements of tensor T and matrix U, respectively.

For instance, Fig. 1 demonstrates the generated core tensor model S and three truncated orthogonal bases $U_1,\ U_2,\ U_3$ by decomposing the initial tensor T. The 4 by 4 by 3 tensor is decomposed to a 2 by 2 by 2 core tensor, two matrices of 4 by 2 and a matrix of 3 by 2.

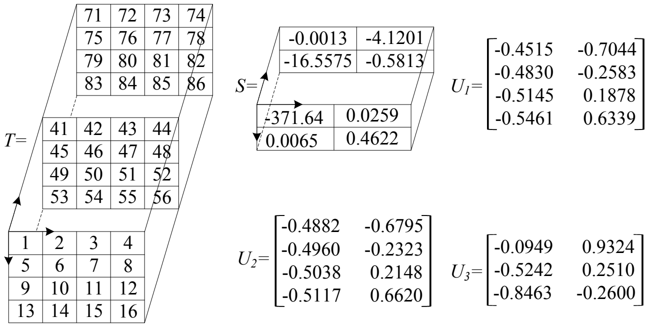


Fig. 1. Decomposing a three-order tensor to a core tensor S and three orthogonal bases U_1, U_2 and U_3.

2.2 Fully Homomorphic Encryption

The homomorphic encryption scheme allows some types of operations to be carried out on the cyphertext to generate the cipher results, on which the decryptions are identical to the results directly computed by performing operations on the plaintext. Two fully homomorphic encryption schemes [6, 11] are proposed using ideal lattice and polynomial ring, respectively. In [12], a General Learning with Errors (GLWE) based scheme is reported, of which the four key steps are as follows:

1) ***E.Setup***$(1^\lambda,\ 1^\mu,\ \boldsymbol{b})$***:*** Choose a μ-bit modulus q, and parameters $d,\ n,\ N,\ \chi$. Suppose R is equal to $Z[x]/(x^d+1)$ and $params$ is equal to $\{q,\ d,\ n,\ N,\ \chi\}$.
2) ***E.SectKeyGen***$(params)$***:*** Set the secret key $sk = s \leftarrow (1,\ s'[1],\ \dots,\ s'[n]) \in R_q^{n+1}$ where s' is from χ^n.
3) ***E.PubKeyGen***$(params,\ sk)$***:*** Choose a vector $e \leftarrow \chi^N$ and generate a matrix $A' \leftarrow R_q^{N\times n}$, then compute $b \leftarrow A's' + 2e$. Generate an $n+1$ column matrix A which consists of vector b and matrix $-A'$. Set the public key $pk = A$.
4) ***E.Enc***$(params,\ m,\ pk)$***:*** Randomly choose a vector $r \leftarrow R_2^N$ and output the ciphertext $c \leftarrow (m,\ 0,\ \dots,\ 0) + A^{\mathrm{T}}r$ where $m \in R_2$ and $c \in R_q^{n+1}$.
5) ***E.Dec***$(params,\ c,\ sk)$***:*** Obtain the plaintext using the equation $m \leftarrow [[< c,\ s >]_q]_2$.

The encryption scheme supports the homomorphism of addition and multiplication, which can be illustrated as Fig. 2. Let m_1 and m_2 be two elements in the plaintext, c_1 and c_2 in the ciphertext, let $c_1 = Enc(m_1)$, $c_2 = Enc(m_2)$, then $m_1 + m_2 = Dec(Enc(m_1) + Enc(m_2))$.

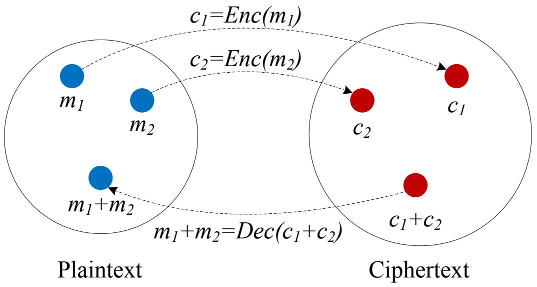


Fig. 2. An illustration of the homomorphic encryption scheme.

2.3 Lanczos Method

The Lanczos method [13, 14] can efficiently compute the eigenvectors as well as eigenvalues of a sparse symmetric matrix. It transforms the matrix M with an orthogonal matrix W, where $W = [w_1,\ \dots,\ w_k]$ and $W^{\mathrm{T}}W = I$, to a tridiagonal matrix as follows

$$L = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_k \\ & & \beta_k & \alpha_k \end{bmatrix}. \tag{3}$$

Equating columns in the expression $MW = WL$, the tridiagonal matrix L can be generated by carrying out the iteration procedures

$$\begin{aligned} &\alpha_j = w_j^{\mathrm{T}} M w_j, \\ &r_j = M w_j - \alpha_j w_j - \beta_j w_{j-1}, \\ &\beta_{j+1} = \|r_j\|_2,\ w_{j+1} = r_j/\beta_{j+1}. \end{aligned} \tag{4}$$

The components of $\alpha,\ \beta,\ r$ can be progressively calculated. Let the eigenvalue decomposition of matrix L be defined as $L = Q\Lambda Q^{\mathrm{T}}$, then the eigenvalues and eigenvectors of matrix M are Λ and WQ, respectively.

3 Problem Description and Solution Overview

This section formalizes the problem of secure tensor decomposition based on the fully homomorphic encryption scheme, and provides an overview of the proposed solution framework.

3.1 Problem Definition

The heterogeneous data generally consist of structured data D_u, semi-structured data D_{semi} as well as unstructured data D_s. Let S denote the core tensor, $U_1,\ U_2,\ \dots,\ U_N$ refer to the truncated orthogonal bases, then the secure tensor decomposition problem can be formalized as

$$\begin{aligned} &f_r : \{Enc(D_u), Enc(D_{semi}), Enc(D_s)\} \to Enc(T), \\ &f_d : Enc(T) \to \{Enc(S), Enc(U_1),\ \dots,\ Enc(U_N)\}. \end{aligned} \tag{5}$$

In Eq. (5), the data representation function f_r integrates all encrypted data as a unified cipher tensor model (*UCT*), on which the decomposition function f_d is carried out to generate the cipher core tensor and truncated orthogonal bases.

The decomposition operations are performed on the encrypted data. Therefore, the user's privacy are protected. In order to guarantee the correctness of the decomposition result, Eq. (5) satisfies $S = T \times_1 U_1^{\mathrm{T}} \times_2 U_2^{\mathrm{T}} \dots \times_N U_N^{\mathrm{T}}$. According to the fully homomorphic encryption scheme, the secure decomposition process satisfies the following equation

$$Dec(sk,\ Eva(pk,\ C_{f_d},\ Enc(T))) = C_{f_d}(T), \tag{6}$$

where Eva, Enc, Dec refer to the evaluation, encryption, and decryption function, pk and sk denote the public key and private key, C_{f_d} refers to the boolean circuits of the tensor decomposition function f_d defined in Eq. (5).

The homomorphism can be guaranteed by performing addition, subtraction, and multiplication operations on the cipher data during the tensor decomposition process. However, new challenges arise when the non-homomorphic operations such as square root and division are adopted in some types of decomposition methods, for example, Lanczos-based algorithm. A secure tensor decomposition algorithm is proposed in this paper to address these challenges.

For convenience, in the following sections this paper adopts the symbol Ψ^E to denote the cipher data according to the plain data Ψ, namely $\Psi^E = Enc(\Psi)$. Therefore, the encrypted tensor $Enc(T)$ is denoted as T^E.

3.2 Overview of the Solution Framework

In order to solve the problem defined above, we propose a secure tensor decomposition approach based on the fully homomorphic encryption scheme. Fig. 3 provides an overview of the framework where the heterogeneous data are first encrypted and represented as a unified tensor model, then securely decomposed to a core tensor and some truncated orthogonal bases. The four representative steps of the solution framework are summarized as follows.

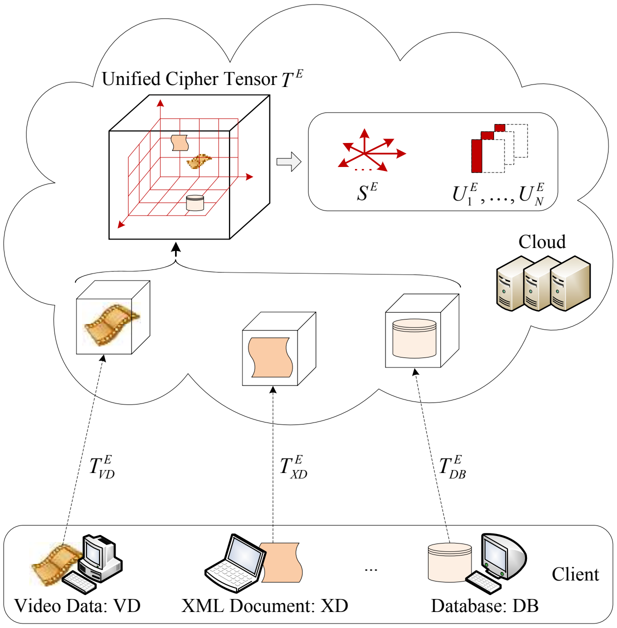


Fig. 3. Framework overview of the secure tensor decomposition approach.

1) ***Data Representation, Encryption and Submission:*** The heterogeneous data collected in the clients are represented as low-order sub-tensors using the method proposed in our previous work [9]. Then the sub-tensors are encrypted using the fully homomorphic encryption scheme and the generated cipher results are submitted to the cloud for unification and decomposition. In Fig. 3, the unstructured video data VD, semi-structured XML document XD, and structured database DB are transformed to cipher low-order sub-tensors T_{VD}^E, T_{XD}^E, T_{DB}^E, respectively.
2) ***Construction of Cipher Tensor:*** The generated sub-tensors T_{VD}^E, T_{XD}^E, and T_{DB}^E are then embedded to a base tensor model $T_{base} \in R^{I_{tim} \times I_{spa} \times I_{clt}}$ to generate a unified cipher tensor model T^E using the tensor extension operation [9] $T^E = T_{base} \vec{\times} T_{VD}^E \vec{\times} T_{XD}^E \vec{\times} T_{DB}^E$. The three orders I_{tim}, I_{spa}, I_{clt} of the base tensor model denote time, space and client.
3) ***Secure Tensor Decomposition:*** After unfolding the N-order unified cipher tensor T^E to matrices $T_{(1)}^E, \ldots, T_{(N)}^E$, the symmetrization transformation is performed on each tensor unfolding to generate the symmetric matrix $sym(T_{(i)}^E) = T_{(i)}^E (T_{(i)}^E)^{\mathrm{T}}$, $(1 \leq i \leq N)$. The eigen vectors of the symmetric matrix $sym(T_{(i)}^E)$ are corresponding to the left singular vectors of matrix $T_{(i)}^E$. The Lanczos method is employed to perform the eigen value decomposition, namely, $sym(T_{(i)}^E) = U_i^E \Lambda^E (U_i^E)^{\mathrm{T}}$. The cipher core tensor S^E can be computed by applying Eq. (1) to the truncated bases $U_1^E, \ldots, U_N^E$ and the unified cipher tensor T^E.
4) ***Obtain the Plain Core Tensor and Bases:*** By decrypting the cipher core tensor and cipher truncated bases generated in Step 3, the plain core tensor S and plain truncated orthogonal bases $U_1, \ldots, U_N$ can be computed. As the homomorphism are supported during the secure tensor decomposition, the generated results are correct and are identical to that directly computed using the plain data.

This paper focuses on Step 2 and Step 3, which correspond to the secure representation function f_r and secure tensor decomposition function f_d.

4 Construction of Cipher Tensor On Cloud Using Fully Homomorphic Encryption Scheme

This section illustrates the process of representing the heterogeneous data as a unified cipher tensor model using the fully homomorphic encryption scheme. New concepts and operations closely related to the cipher tensor model are introduced.

4.1 Cipher Tensor and Nil Element

In order to clearly describe the process of representing the structured, semi-structured as well as unstructured data as a unified cipher tensor model, this paper introduces some definitions as follows.

Definition 1: Cipher Tensor. A cipher tensor model T^E is generated by encrypting the elements in the plain tensor T using the fully homomorphic encryption scheme. The construction process is defined as $T^E = \{Enc(t) | t \in T\}$.

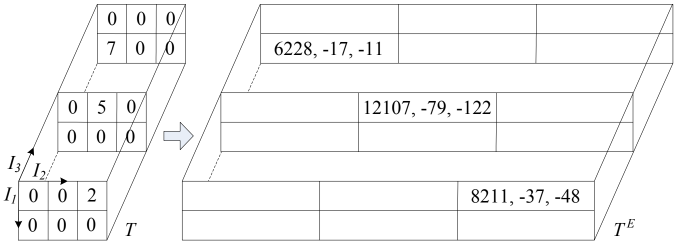


Fig. 4. A plain tensor and the corresponding cipher tensor.

Fig. 4 demonstrates a plain tensor and the corresponding cipher tensor. In this instance, the private key is $sk = (1,\ 7,\ 8)$ and the public key is

$$pk = A = \begin{pmatrix} 134 & -3 & -2 \\ 218 & 0 & -3 \\ 507 & -2 & -1 \end{pmatrix}. \tag{7}$$

The encryption function is formalized as $t_{ijk}^E = t_{ijk} + (pk)^{\mathrm{T}} r$ where r is a randomly selected three-dimensional vector. The decryption function is $t_{ijk} = \left[\left[t_{ijk}^E,\ s\right]_q\right]_p$

where the parameter q and p are 65537 and 97 respectively. The two tensors in Fig. 4 satisfy the equation $T^E = Enc(pk,\ T)$. This is a simple example to illustrate the construction of a cipher tensor. In practice, the fully homomorphic encryption scheme chooses large integers for these parameters.

Definition 2: Nil Cipher Element. The element generated by encrypting the plain element 0 is called the Nil cipher element. This paper adopts symbol 0^E to denote the Nil cipher element, namely $0^E = Enc(0)$.

In the fully homomorphic encryption scheme, the plain element 0 may be encrypted to different Nil cipher elements.

Definition 3: Sparse Cipher Tensor. A cipher tensor containing a large portion of Nil elements is called a sparse tensor. In this paper, a sparse tensor is assumed to contain more than 60% Nil elements.

In Fig. 4, the cipher tensor T^E consists of 18 elements. There are 15 Nil elements and 3 nonzero elements. Therefore, T^E is a sparse cipher tensor.

Definition 4: Reduced Cipher Tensor. A reduced cipher tensor is generated by removing all the Nil cipher elements from the cipher tensor model.

As the zero element in the plain data may be encrypted to different Nil cipher elements in the cipher tensor, special methods are needed to remove the Nil cipher elements to obtain the reduced cipher tensor. In the proposed solution framework demonstrated in Fig. 3, the clients are responsible for removing the zero elements from the plain tensor models before encryption. This method can reduce the communication traffic as well as the encryption time.

4.2 Constructing a Unified Cipher Tensor Model on Cloud

In this paper, the heterogenous data are first represented and encrypted as cipher low-order sub-tensors on the clients, then they are submitted to the cloud for unification. To integrate all the cipher sub-tensors, a base tensor model is proposed, which is defined as $T_{base} \in R^{I_{tim} \times I_{spa} \times I_{clt}}$, where I_{tim}, I_{spa}, I_{clt} refer to time, space and client. The three orders serve as bases to which the encrypted sub-tensors can be embedded for generation of a unified cipher tensor model.

For example, the unstructured video data VD can be represented as a four-order tensor model $T_{VD} \in R^{I_f \times I_h \times I_w \times I_{cs}}$ [9], where the tensor orders I_f, I_h, I_w, I_{cs} denote frame, height, width, and color space. The semi-structured XML document XD can be transformed to a three-order tensor model $T_{XD} \in R^{I_{ia} \times I_{ib} \times I_r}$ [9], where the orders I_{ia}, I_{ib}, I_r denote the XML elements and relationships. In Fig. 5, the two sub-tensors T_{VD} and T_{XD} are encrypted to cipher counterparts T^E_{VD}, T^E_{XD}, respectively, which are then embedded to the base tensor model T_{base}. The unified cipher tensor T^E is as follows

$$T^E \in R^{I_{tim} \times I_{spa} \times I_{clt} \times I_h \times I_w \times I_{cs} \times I_{ia} \times I_{ib} \times I_r}. \quad (8)$$

The frame order of the unstructured video data is integrated to the order I_{tim}. The nine-order tensor in Eq. (8) contains all data characteristics of the video, XML document and base tensor. All elements in the cipher tensor T^E get involved in the secure tensor decomposition.

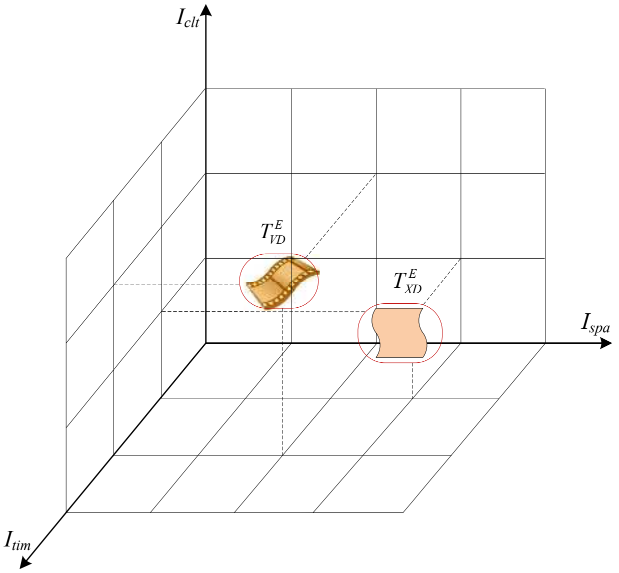
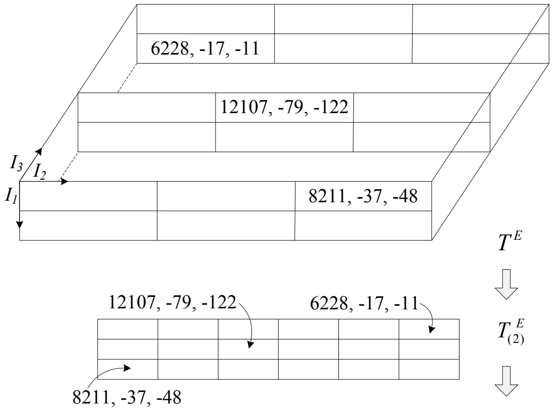


Fig. 5. Embedding two encrypted sub-tensors to the base tensor model on cloud.

4.3 Tensor Unfolding and Memory Storage Scheme

When the unified cipher tensor is generated, the next critical step is to obtain the unfolded matrices, which are used to construct the symmetric matrices. This paper proposes to use the Compressed Row Storage (CRS) [15] scheme to store the sparse unfolded matrices. Additionally, in order to decrease execution time of the secure tensor decomposition algorithm, the data-intensive application can employ $T^E_{(i)}((T^E_{(i)})^T v)$ to perform the matrix-vector operation on the symmetric matrix of the i-mode tensor unfolding. Namely, the vector v is first left multiplied with matrix $(T^E_{(i)})^T$ to obtain a temporary vector, which is then left multiplied with matrix $T^E_{(i)}$.

In order to unfold a cipher unified tensor, the non-Nil elements in the cipher tensor are rearranged along the rows of the corresponding unfolded matrices. Fig. 6 demonstrates the 2-mode tensor unfolding of a three-order cipher tensor. The three none-Nil elements in tensor T^E are rearranged to the unfolded matrix $T^E_{(2)}$. This unfolded cipher matrix is used to perform the matrix-vector product operation during the Lanczos algorithm. The CRS scheme of the unfolded matrix is demonstrated at the bottom table in Fig. 6. The array *val* consists of the three non-Nil elements, array *col-ind* includes the column indices, *row-ptr* stores the four locations that start new rows. For example, the element 3 in array *row-ptr* indicates that the cipher element $(8211,\ -37,\ -48)$ in the unfolded matrix $T^E_{(2)}$ starts a new row.



Storage Scheme

val	8211, -37, -48	12107, -79, -122	6228, -17, -11	
col-ind	1	3	6	
row-ptr	3	2	1	4

Fig. 6. The 2-mode unfolded matrix of a three-order cipher tensor and the corresponding storage scheme.

4.4 A Cipher Tensor Representation Algorithm on Cloud

Based on the previously mentioned methods, this paper proposes Algorithm 1 to represent the heterogeneous data as a unified cipher tensor (*UCT*) model on cloud.

```
Algorithm 1 Cipher Tensor Representation. $T^E = f_r(D_s,\ D_{semi},\ D_u)$
Input:
    The structured, semi-structured and unstructured data ($D_s$, $D_{semi}$, data $D_u$).
Output:
    The unified cipher tensor model $T^E$.
1: Represent the various types of local data as low-order sub-tensors, and encrypt them to cipher low-order sub-tensors on clients.
2: Upload the generated cipher sub-tensors to cloud.
3: Embed all the cipher sub-tensors to the base tensor model $T_{base} \in R^{I_{tim} \times I_{spa} \times I_{clt}}$, and obtain the unified cipher tensor model $T^E$.
4: Unfold the cipher tensor to matrices and generate the symmetric matrices for tensor decomposition.
```

In Line 1 of the proposed Algorithm 1, the structured, semi-structured as well as unstructured data are transformed to low-order sub-tensors, which are then encrypted using the fully homomorphic encryption scheme on clients. All the cipher sub-tensors are uploaded to cloud for unified representation. In this paper, the zero elements of the plain data are removed during the encryption procedure. The cloud embeds all the cipher sub-tensors to the base tensor model in Line 3 to generate the unified cipher tensor model T^E. Line 4 generates the symmetric matrices of each cipher tensor unfolding for secure tensor decomposition.

5 Secure Tensor Decomposition on Cloud and Client

This section presents a secure tensor decomposition algorithm and makes an analysis of it in terms of time consumption, memory usage and decomposition accuracy.

5.1 Non-Homomorphic Operations During Lanczos-based Decomposition

Table 2 shows the five types of operations utilized in the Lanczos iteration, namely, addition $+$, subtraction $-$, multiplication $\times$, division $\div$, and square root $\sqrt{x}$. To guarantee the correctness of the decomposition results of the cipher tensor, new methods need to be developed to address the challenges of non-homomorphic operations performed on the cipher tensor, which can be described as follows:

Challenge 1: Non-Homomorphic Square Root Operation on Cipher Data. This challenge is to perform the operation $\beta_{j+1} = \|r_j\|_2$, which is responsible for computing the second norm of the vector r_j.

Challenge 2: Non-Homomorphic Division Operation on Cipher Data. The division operation is used to generate the normalized orthogonal vectors with the following equation $\omega_{j+1} = {}^{r_j}/_{\beta_{j+1}}$.

TABLE 2
Operations in the Lanczos Method.

Operation	Homomorphic	Step
$+$	yes	$\alpha_j = w_j^{\mathrm{T}} M w_j$
$-$	yes	$r_j = M w_j - \alpha_j w_j - \beta_j w_{j-1}$
$\times$	yes	$\alpha_j = w_j^{\mathrm{T}} M w_j$
$\div$	no	$\omega_{j+1} = {}^{r_j}/_{\beta_{j+1}}$
$\sqrt{x}$	no	$\beta_{j+1} = \|r_j\|_2$

5.2 Removing the Non-homomorphic Square Root Operation in Lanczos Procedure

Theorem 1: Lanczos-based Cipher Tensor Decomposition without Square Root Operation. Let T^E denote an N-order cipher tensor, S^E refer to the cipher core tensor, $U_1^E,\ \ldots,\ U_N^E$ be the truncated orthogonal bases. Then with the Lanczos method, the core cipher data $core^E = \{S^E,\ U_i^E\}$ can be generated without performing the non-homomorphic square root operations in the Lanczos procedure.

Proof. During the decomposition process, the square root operation is employed to compute the second norm of a vector, namely, $\beta_{j+1} = \|r_j\|_2$. Inspired by [16], we extend the orthogonal unitary matrix W to an orthogonal but non-unitary matrix to remove the square root operation. Let W be a matrix consisting of orthogonal vectors, $W^{\mathrm{T}} W = D$, and $D = diag(\delta_1,\ \delta_2,\ \ldots)$. Then by multiplying the symmetric matrix of the i-mode tensor unfolding with matrices W^{T} and W, we obtain

$$W^{\mathrm{T}}T^{E}_{(i)}(T^{E}_{(i)})^{T}W=\begin{bmatrix}\delta_1 & & & \\ & \delta_2 & & \\ & & \delta_3 & \\ & & & \ddots\end{bmatrix}\begin{bmatrix}\alpha_1 & \beta_2 & & \\ 1 & \alpha_2 & \beta_3 & \\ & 1 & \alpha_3 & \ddots \\ & & \ddots & \ddots\end{bmatrix}. \tag{9}$$

Let $\beta_j = {}^{\delta_j}/_{\delta_{j-1}}$, Eq. (9) can be transformed to

$$W^{\mathrm{T}}T^{E}_{(i)}(T^{E}_{(i)})^{T}W=\begin{bmatrix}\alpha_1\delta_1 & \delta_2 & & \\ \delta_2 & \alpha_2\delta_2 & \delta_3 & \\ & \delta_3 & \alpha_3\delta_3 & \ddots \\ & & \ddots & \ddots\end{bmatrix}. \tag{10}$$

Selecting the j-th vector of the result matrix, we obtain the following equation

$$T^{E}_{(i)}(T^{E}_{(i)})^{T}w_j=\delta_j w_{j-1}+\alpha_j\delta_j w_j+\delta_{j+1}w_{j+1}. \tag{11}$$

As the vectors of matrix W are orthogonal, according to Eq. (11), all the parameters in the tridaigonal matrix can be computed as

$$\begin{aligned}&\alpha_j = {w_j^{\mathrm{T}}T^{E}_{(i)}(T^{E}_{(i)})^{T}w_j}\Big/{\delta_j},\\ &w_{j+1}=v_j-\alpha_j w_j,\\ &\delta_{j+1}=w_{j+1}^{\mathrm{T}}w_{j+1},\\ &\beta_{j+1}={}^{\delta_{j+1}}/_{\delta_j},\\ &v_{j+1}=T^{E}_{(i)}(T^{E}_{(i)})^{T}w_{j+1}-\beta_{j+1}w_j.\end{aligned} \tag{12}$$

In the above procedures, parameter i refers to the i-mode unfolded matrix of the cipher tensor T^E, and j denotes the j-th iteration of the Lanczos procedure. δ_j is a non-zero element prematurely. Based on the tridiagonalized matrix, we can compute the left singular matrix U_i^E of the cipher tensor unfolding $T^{E}_{(i)}$. Therefore, the core tensor S^E can be computed with equation $S^E = T^E\times_1(U_1^E)^{\mathrm{T}}\times_2(U_2^E)^{\mathrm{T}}\ldots\times_N(U_N^E)^{\mathrm{T}}$. In Eq. (12), the non-homomorphic square root operations are removed.

Note that the parameters $\alpha,\ \beta,\ \delta,\ w,\ v$ are all in ciphertext format. For convenience, the superscripts are omitted during the proof procedure in this paper. The division operations are transferred to the client. In each Lanczos iteration, the cloud send $w_j^{\mathrm{T}}T^{E}_{(i)}(T^{E}_{(i)})^{T}w_j$, δ_{j+1}, δ_j to the client, where the division operations are performed and the results are passed back to cloud in ciphertext format.

5.3 Secure Tensor Decomposition Algorithm on Cloud and Client

In this paper, Algorithm 2 is presented for secure tensor decomposition. The numbers defined in real fields for the raw data are multiplied with 10^d [17] to obtain the corresponding integers. Hence, all operations are defined in the integer field. The non-homomorphic operation, namely square root, is removed from the Lanczos procedure.

In Line 1 of Algorithm 2, the unified cipher tensor model T^E is unfolded to N matrices which are then transformed to symmetric matrices $T^{E}_{(i)}(T^{E}_{(i)})^{\mathrm{T}}$. A random integer vector is selected in Line 3. The left singular vector matrix of $T^{E}_{(i)}$ is equal to the eigen vector matrix of $T^{E}_{(i)}(T^{E}_{(i)})^{\mathrm{T}}$. From Line 2 to Line 13, Algorithm 2 computes the truncated orthogonal bases using the Lanczos method. The non-homomorphic operation, square root, is removed during the iterations, and the division challenge is addressed by transferring the operations to client in Line 5 and Line 9. The tridiagonal matrix L generated in Line 12 is used for eigen value decomposition. This paper employs the symmetric QR algorithm [18] to compute the eigen values and eigen vectors of the tridiagonal matrix L on clients. The truncated orthogonal bases are computed in Line 13, and the cipher core tensor is generated in Line 14.

Algorithm 2 Secure Tensor Decomposition on Cloud and Client $\{S^E,\ U_1^E,\ \ldots,\ U_N^E\} = f_d(T^E)$.

```
Input:
    The reduced cipher tensor $T^E$.
Output:
    The cipher core tensor $S^E$ and cipher truncated orthogonal bases $U_1^E,\ \ldots,\ U_N^E$.
1:  Unfold the cipher tensor to matrices and obtain the corresponding symmetric matrices $T^{E}_{(i)}(T^{E}_{(i)})^{\mathrm{T}}$.
2:  for each matrix $T^{E}_{(i)}(T^{E}_{(i)})^{\mathrm{T}}$, $1 \le i \le N$, do
3:      Initialize the parameters by setting $j = 1$, $w_j = random\ integer\ vector$, $\delta_j = w_j^{\mathrm{T}}w_j$, $\beta_1 = 1$, $v_j = T^{E}_{(i)}(T^{E}_{(i)})^{\mathrm{T}}w_j$.
4:      while $\delta_j \neq 0$ do
5:          Compute $w_j^{\mathrm{T}}T^{E}_{(i)}(T^{E}_{(i)})^{\mathrm{T}}w_j$ and obtain the parameter $\alpha_j$ by receiving the division result computed on the client.
6:          Compute vector $w_{j+1} = v_j - \alpha_j w_j$.
7:          Increase $j$ by 1, namely, $j = j + 1$.
8:          Replace $\delta_j$ with $w_j^{\mathrm{T}}w_j$.
9:          Send $\delta_j$ and $\delta_{j-1}$ to the client and receive the division result $\beta_j$.
10:         Compute vector $v_j = T^{E}_{(i)}(T^{E}_{(i)})^{\mathrm{T}}w_j - \beta_j w_{j-1}$.
11:     end while
12:     Construct the tridiagonal matrix $L$ using the generated elements $\alpha_j,\ \beta_j$, and compute the eigen values and eigen vectors on client.
13:     Generate the left singular vector matrices and truncated orthogonal bases.
14:     Compute the cipher core tensor $S^E$ using equation $S^E = T^E\times_1(U_1^E)^{\mathrm{T}}\times_2(U_2^E)^{\mathrm{T}}\ldots\times_N(U_N^E)^{\mathrm{T}}$.
15:     Return tensor $S^E$ and the bases $U_1^E,\ \ldots,\ U_N^E$.
16: end for
```

5.4 Algorithm Analysis

The performance of the proposed secure tensor decomposition algorithm is theoretically analyzed in this paper in terms of time consumption, memory usage and decomposition accuracy.

Time Consumption: Execution time of the proposed secure decomposition algorithm consists of matrix unfolding, Lanczos-based singular value decomposition of each tensor unfolding as well as product of a tensor by the truncated matrices. Let $Time_{unf}$, $Time_{lan}$ and $Time_{prod}$ denote the time used by the above processes, respectively, then the total time consumption $Time$ may be defined as

$$Time = Time_{unf} + Time_{lan} + Time_{prod}. \tag{13}$$

Tensor unfolding is a simple transformation with $O(1)$ time complexity. $Time_{lan}$ is equal to $Time_{lan1} + Time_{lan2} + ... + Time_{lanN} = \sum_{i=1}^{N} Time_{lani}$, where $Time_{lani}$ refers to the decomposition time consumed by unfolded matrix $T^{E}_{(i)}$. The time complexity of generating a tridiagonal matrix for a tensor unfolding using the Lanczos method is $O(kn)$. In this paper, the time complexity of eigen value decomposition of the tridiagonal matrix on client is not considered. For a truncated orthogonal basis U with k column vectors, time complexity of the product of a tensor by a matrix is $O(kn^2)$, where k is the number of vectors in the truncated basis U^{E}_{i}. Hence, to decompose an N-order cipher tensor T^{E} with N unfolded matrices, the time complexity of the proposed secure tensor decomposition algorithm is $O(1) + O(Nkn) + O(Nkn^2)$, namely $O(Nkn^2)$, where N refers to the number of tensor orders, n denotes the dimensionality of the tensor unfolding.

Memory Usage. The consumed memory of the secure decomposition algorithm is related to the number of non-Nil elements of the ciper tensor model T^{E}. Assume m^{i}_{nz} denote the number of rows containing non-zero elements in matrix $Sym(T^{E}_{(i)})$, then the memory usage can be computed using equation $2N \times nnz(T^{E}) + \sum\limits_{i=1}^{N} m^{i}_{nz}$. According to the CRS scheme described in previous section, the proposed secure tensor decomposition algorithm can significantly save computer memory.

Decomposition Accuracy. The reconstruction error between the initial tensor and the generated approximate tensor can be computed using the Frobenius Norm [19] which is defined as

$$\left\|T - \hat{T}\right\|_{F} = (\sum_{i_1=1}^{I_1}, ..., \sum_{i_p=1}^{I_P} (a_{i_1,...,i_p} - \hat{a}_{i_1,...,i_p})^2)^{\frac{1}{2}}. \tag{14}$$

For the unfolded matrix $T_{(i)}$ of initial tensor T, the approximate matrix is $\hat{T}_{(i)} = U_i \Sigma_i V_i^{\mathrm{T}}$. The reconstruction error is caused by approximation of all unfolded matrices. To clearly analyze tensor dimensionality reduction degree and tensor approximation degree, this paper employs two ratios. The reduction ratio is defined as $\rho = \frac{nnz(S) + \sum\limits_{i=1}^{N} nnz(U_i)}{nnz(T)}$, where S denotes the core tensor, and U_i is the i-mode truncated orthogonal basis. As only nonzero elements of the core data set are stored, ratio ρ can accurately reflect the dimensionality reduction degree. The reconstruction error ratio is $\vartheta = \frac{\left\|T - \hat{T}\right\|_F}{\left\|T\right\|_F}$, which reflects the degree of reconstruction error with tensor Frobenius Norm. In this paper, the pair (ρ, ϑ) is used to describe the dimensionality reduction degree and reconstruction error degree. Obviously, the ratio ρ is inversely proportional to ratio ϑ.

6 Performance Evaluation

This section illustrates some very preliminary evaluation results of the proposed secure tensor decomposition approach. We performed the experiments on commodity computers, each of them is of Intel(R) Core(TM) $i5-3470$ CPU @3.20 GHZ, 8 GB RAM, and is running CentOS 6.4 Operating System. We adopted the software library HElib [1] which implements the BGV fully homomorphic encryption scheme. The NTL-6.2.1 mathematical library was compiled and installed in the experimental machines. The experimental data are from our university including the unstructured video data collected with fixed cameras, semi-structured XML documents about staffs and students in our university, and structured trajectory data collected by mobile phones. These data are encrypted and integrated as a unified cipher tensor model for secure tensor decomposition. We implemented a number of secure algorithms on cipher data including singular value decomposition, eigen value decomposition, tensor construction. But due to space constraints, we only present some representative results.

To evaluate the effects of dimensionality reduction of secure tensor decomposition, we utilized a three-order tensor formed by gray video clips, which is of MPEG4 format, 15 frames per second. The tensor was unfolded to three matrices, which were transformed to symmetric matrices and then factorized using the Lanczos method. We adopted different truncation ratios to preserve the left singular vector matrices which contain the unitary orthogonal vectors of the tensor unfolding. This section demonstrates some experimental results of the singular values, orthogonal bases, core tensor, dimensionality reduction ratios and tensor approximation ratios.

The results demonstrated in this section need to be viewed with a limitation in mind, that is more extensive experiments should be carried out on cloud and client to evaluate the performance of the secure tensor decomposition approach.

6.1 Singular Values of Unfolded Matrices

Fig. 7 demonstrates the singular values of the three unfolded matrices $T_{(1)}$, $T_{(2)}$, and $T_{(3)}$. We also drew the super-diagonal values of the core tensor in the figure for comparison. The graph shows that the first singular values of the three tensor unfolding are generally greater than the others. In our experiments, the first singular

[1] https://github.com/shaih/HElib

values are 6.559, 5.856, and 6.652 times of the second singular values of the tensor unfolding, respectively. In addition, there is an obvious declining trend from the second singular value to the eighth singular value. From the ninth singular value, the rate of decrease slows down. The scaling ratios of the first singular values to the thirtieth singular values are 48.00, 51.73, and 103.35, respectively.

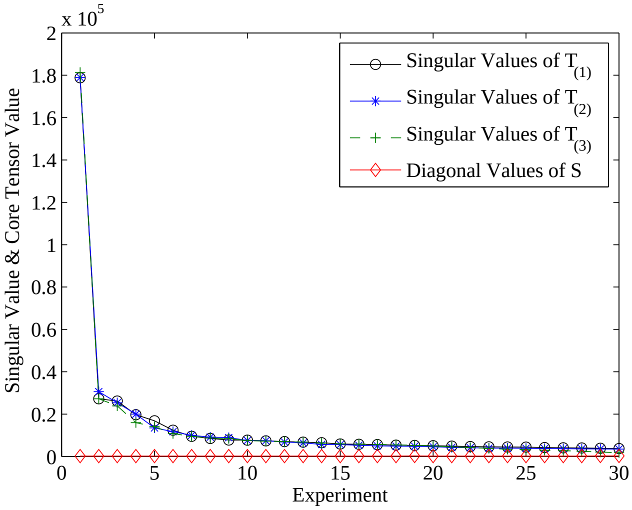


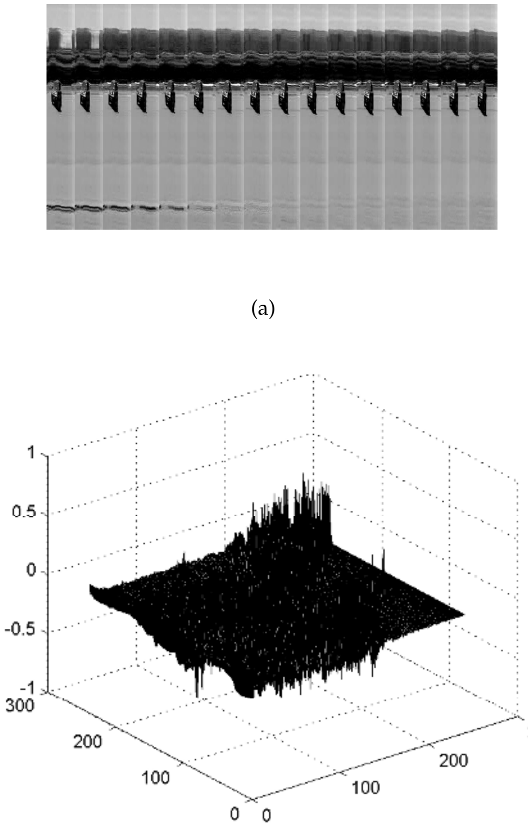
Fig. 7. The singular values of unfolded matrices and the diagonal values of the core tensor.

6.2 Unfolded Matrices and Truncated Orthogonal Bases

Fig. 8 shows an example of the 1-mode tensor unfolding $T_{(1)}$ and the truncated left singular vector matrix U_1. The number of rows in matrix $T_{(1)}$ is equal to the dimensionality of order I_1, and the number of columns is equal to $I_2 \times I_3$. The 1-mode unfolded matrix in Fig. 8(a) depicts the elements of the initial tensor model along the first order. The singular vector matrices are composed of unitary orthogonal vectors, the elements of the orthogonal vectors are normalized which are between -1 and 1. In Fig. 8(b), the maximum elements of matrix U_1 is 0.53, and the minimum value is -0.73. The elements between $(-0.2,\ 0.2)$ account for 98.54 percent. About 33.96 percent of the elements range from -0.01 to 0.01.

6.3 Matrices of Core Tensor

In order to illustrate the structure of the core tensor, we extracted four slices and demonstrated them in Fig. 9. The projection coordinates are contained in the core tensor which has the same number of orders as the initial tensor. The matrix $S(:,:,3)$ has more larger elements than matrix $S(:,:,30)$. The maximum element of matrix $S(:,:,3)$ and $S(:,:,30)$ are 6286.07 and 156.27, respectively. The elements in matrix $S(:,:,12)$ are between -936.40 and 893.84, while the elements in matrix $S(:,:,21)$ are



(a)

(b)

Fig. 8. (a) The 1-mode unfolded matrix of a tensor; (b) the corresponding left singular matrix of the tensor unfolding.

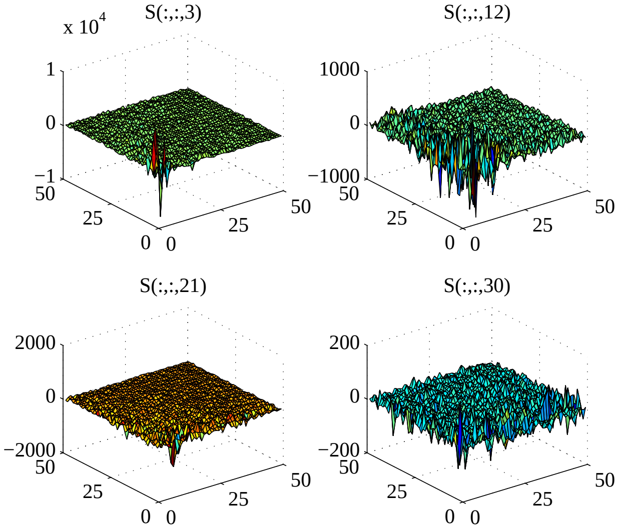


Fig. 9. Illustration of the four matrices of a core tensor.

between -1085.67 and 539.93. In our experiment, the average values of the four matrices are -0.87, 0.41, -0.69, and 0.65, respectively.

6.4 Reduction Ratio and Approximation Ratio

We decomposed the unified tensor model to a core tensor multiplied with some truncated orthogonal bases. The dimensionality reduction ratio and approximation ratio which is equal to the subtraction of the reconstruction error ratio from 100%, are utilized for evaluation. Fig. 10 demonstrates that the dimensionality reduction ratio increases from 0.28% to 78.19% during the experiments, while the tensor approximation ratio increases slowly from 79.21% to 98.56%. In the fourteenth experiment, 14.72% core data accounted for 92.66% approximation accuracy. In the eighteenth experiment, 23.41% core data accounted for 94.20% approximation accuracy. The line graph of dimensionality reduction ratio in Fig. 10 increases sharply than the tensor approximation ratio. Averagely, about 21% core data can guarantee 94% approximation accuracy during tensor decomposition.

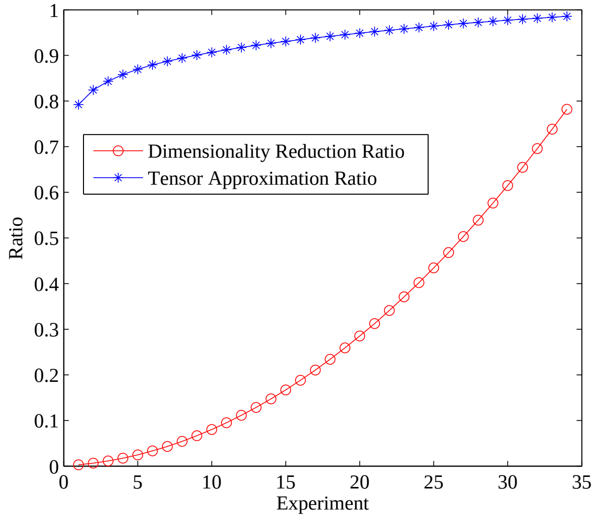


Fig. 10. The relationship between the dimensionality reduction ratio and tensor approximation ratio.

6.5 Approximation Comparison Between the Cipher and Plain Tensor Decomposition

We carried out experiments of tensor decomposition on small cipher tensor and plain tensor models. In order to preserve the desired precision, we multiplied the elements of the tensor models with some coefficients. The elements are scaled up and the decimal parts are dropped. In these three experiments, the coefficients were 10, 100 and 1000, respectively. During the Lanczos procedure, the division results were rounded to the nearest integers for the following iterations.

Fig. 11 demonstrates the experimental results of tensor decomposition on the plain and cipher tensors. In the first experiment, the approximation ratio on the cipher tensor is 81.72%, while the ratio on the plain tensor increases to 86.60%. In the third experiment, the approximation ratios of the cipher and plain tensors are 95.24% and 95.45%, respectively. The column chart reveals that with large coefficients, there is not a great deal of differences between the approximation ratios of the cipher tensor and plain tensor during tensor decomposition.

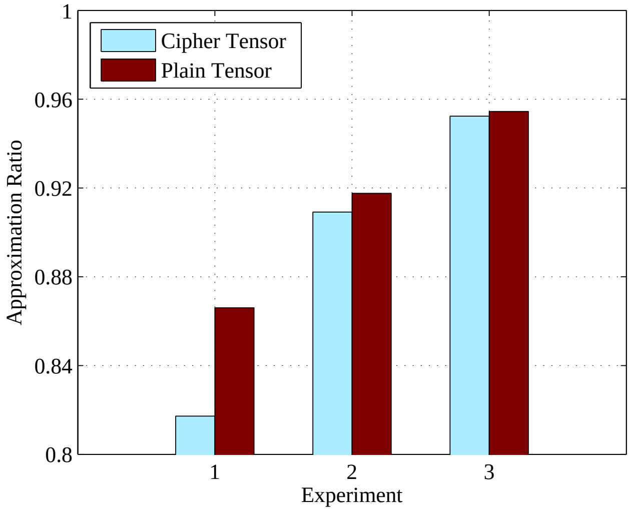


Fig. 11. Approximation comparison of the cipher tensor and plain tensor.

In practice, according to the requirements of the data-intensive applications, the larger coefficients can be carefully selected to preserve computational accuracy.

7 Related Work

This section reviews some previous studies on tensor decomposition, fully homomorphic encryption scheme, and Lanczos method.

7.1 Tensor and High-Order Singular Value Decomposition

A tensor model is used to illustrate the linear relations between the scalars, vectors and other tensors. Tensor [8, 20], usually called multidimensional array, is a generalization of the matrix model. It can effectively represent the heterogeneous data as a concise model and extract high-quality core data using High Order Singular Value Decomposition (HO-SVD) [10] approach. HO-SVD is a special case of the widely used TUCKER [21] decomposition approach.

7.2 Fully Homomorphic Encryption Scheme

The fully homomorphic encryption concept was first reported in 1978 [5]. The encryption schemes reported in [22–26] support either addition homomorphism or multiplication homomorphism. However, none of them can support both operations in a single scheme. A new approach is presented in [27] which constructs a scheme capable of carrying out both addition and multiplication operations. In 1999, Gentry [6] constructed a fully homomorphic encryption scheme. From then on many studies [7, 11, 12, 28, 29] have been performed in order to present new efficient fully homomorphic encryption schemes.

7.3 Lanczos Method

The Lanczos method [13], an adaptation of the power method, is employed to compute several extreme eigenvalues and eigenvectors of a sparse symmetric matrix. A block Lanczos type algorithm is presented in [30] to construct a tridiagonal matrix. A new algorithm [16] is reported that can remove the square root operation from the Lanczos iteration. An implicitly restarted method is explored in [31] for obtaining the smallest singular triplets. In [32], a new error bound for Lanczos method is introduced.

8 Conclusion

Aiming to propose an efficient approach that can securely process large scale heterogeneous data, this paper formalizes the secure tensor decomposition problem, and proposes a holistic solution framework to address it. A unified cipher tensor model is presented to integrate all the encrypted low-order sub-tensors as a unified model. Concise examples are provided for illustrating the process of cipher tensor construction and unfolding. A Lanczos-based secure tensor decomposition algorithm is introduced, in which the non-homomorphic square root operations in Lanczos procedure are removed. Some very preliminary experiments are carried out to evaluate the performance of the methods. The results support that the proposed approach is feasible and can pave a way for secure data processing on cloud.

9 Acknowledgment

This work is partially supported by JSPS KAKENHI Grant Number 26730056, JSPS A3 Foresight Program.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, and I. Stoica, “A View of Cloud Computing,” *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] L. J. van der Maaten, E. O. Postma, and H. J. van den Herik, “Dimensionality Reduction: A Comparative Review,” *J. Mach. Learning Research*, vol. 10, no. 1-41, pp. 66–71, 2009.

[3] J. Han and M. Kamber, *Data Mining, Southeast Asia Edition: Concepts and Techniques*. Morgan kaufmann, 2006.

[4] M. Kantardzic, *Data Mining: Concepts, Models, Methods, and Algorithms*. John Wiley & Sons, 2011.

[5] R. L. Rivest, L. Adleman, and M. L. Dertouzos, “On Data Banks and Privacy Homomorphisms,” *Found. Secure Computation*, vol. 4, no. 11, pp. 169–180, 1978.

[6] C. Gentry, “Fully Homomorphic Encryption Using Ideal Lattices,” in *Proc. 41st Ann. ACM Symp. Theory of Comput.*, vol. 9, 2009, pp. 169–178.

[7] M. Naehrig, K. Lauter, and V. Vaikuntanathan, “Can Homomorphic Encryption be Practical?” *Proc. ACM CCSW*, pp. 113–124, 2011.

[8] T. G. Kolda and B. W. Bader, “Tensor Decompositions and Applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[9] L. Kuang, F. Hao, L. T. Yang, M. Lin, C. Luo, and G. Min, “A Tensor-Based Approach for Big Data Representation and Dimensionality Reduction,” *IEEE Trans. Emerging Topics in Comput.*, vol. 2, no. 3, pp. 280–291, 2014.

[10] L. De Lathauwer, B. De Moor, and J. Vandewalle, “A Multilinear Singular Value Decomposition,” *SIAM J. Matrix Anal. Applicat.*, vol. 21, no. 4, pp. 1253–1278, 2000.

[11] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, “Fully Homomorphic Encryption Over the Integers,” in *Advances in Cryptology–EUROCRYPT 2010*. Springer, 2010, pp. 24–43.

[12] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, “Fully Homomorphic Encryption Without Bootstrapping,” in *Proc. 3rd Innovations in Theoretical Comput. Sci. Conf.* ACM, 2012, pp. 309–325.

[13] J. K. Cullum and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Vol. 1: Theory*. SIAM, 2002, vol. 41.

[14] M. Grüning, A. Marini, and X. Gonze, “Implementation and Testing of Lanczos-based Algorithms for Random-Phase Approximation Eigenproblems,” *Comput. Mater. Sci.*, vol. 50, no. 7, pp. 2148–2156, 2011.

[15] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, 2000, vol. 11.

[16] R. J. Lambert, *Computational Aspects of Discrete Logarithms*. Ph.D thesis, Electrical Engineering, University of Waterloo, Canada, 1997.

[17] J. Powers and K. Chen, “Secure Computation of Top-K Eigenvectors for Shared Matrices in the Cloud,” in *IEEE Sixth Int. Conf. on Cloud Comput. (CLOUD)*. IEEE, 2013, pp. 155–162.

[18] G. H. Golub and C. F. Van Loan, *Matrix computations*. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 2012, vol. 3.

[19] C. Meyer, *Matrix Analysis and Applied Linear Algebra Book and Solutions Manual*. SIAM, 2000, vol. 2.

[20] C. M. Martin, “Tensor Decompositions Workshop Discussion Notes,” in *Proc. AIM Workshop Tensor Decompositions, Palo Alto, CA, USA*, Jul. 2004, pp. 1–27.

[21] L. R. Tucker, "Some Mathematical Notes on Three-mode Factor Analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[22] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[23] S. Goldwasser and S. Micali, "Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information," in *Proc. 14th Ann. ACM Symp. Theory of Comput.* ACM, 1982, pp. 365–377.

[24] T. ElGamal, "A Public Key Cryptosystem and A Signature Scheme Based on Discrete Logarithms," in *Advances in Cryptology*. Springer, 1985, pp. 10–18.

[25] J. D. Cohen and M. J. Fischer, "A Robust and Verifiable Cryptographically Secure Election Scheme," in *Proc. 26th IEEE Symp. Found. of Comput. Sci.*, vol. 85, 1985, pp. 372–382.

[26] P. Paillier, "Public-key Cryptosystems Based on Composite Degree Residuosity Classes," in *Advances in Cryptology, Ann. Eurocypt Conf.* Springer, 1999, pp. 223–238.

[27] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF Formulas on Ciphertexts," in *Theory of Cryptography*. Springer, 2005, pp. 325–341.

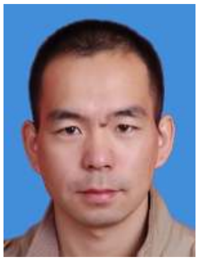
[28] C. Gentry and S. Halevi, "Fully Homomorphic Encryption Without Squashing Using Depth-3 Arithmetic Circuits," in *Proc. 52nd IEEE Symp. Found. of Comput. Sci.*, 2011, pp. 107–109.

[29] N. P. Smart and F. Vercauteren, "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes," in *Public Key Cryptography–PKC 2010*. Springer, 2010, pp. 420–443.

[30] I. Popovyan, "Efficient Parallelization of Lanczos Type Algorithms," *Cryptology ePrint Archive, Rep. 2011/416, http://eprint.iacr.org/*, pp. 1–9, 2011.

[31] Z. Jia and D. Niu, "A Refined Harmonic Lanczos Bidiagonalization Method and An Implicitly Restarted Algorithm for Computing the Smallest Singular Triplets of Large Matrices," *SIAM J. Sci. Comput.*, vol. 32, no. 2, pp. 714–744, 2010.

[32] Q. Ye, "Error Bounds for the Lanczos Methods for Approximating Matrix Exponentials," *SIAM J. Numer. Anal.*, vol. 51, no. 1, pp. 68–87, 2013.



Liwei Kuang is currently studying for the Ph.D degree in School of Computer Science and Technology at Huazhong University of Science and Technology, Wuhan, China. He received the master's degree in School of Computer Science from Hubei University of Technology, Wuhan, China, in 2004. From 2004 to 2012, he was a Research Engineer with FiberHome Technologies Group, Wuhan, China. His research interests include big data, pervasive computing and cloud computing.

Laurence T. Yang received the B.E. degree in Computer Science and Technology from Tsinghua University, China and the Ph.D degree in Computer Science from University of Victoria, Canada. He is a professor in the School of Computer Science and Technology at Huazhong University of Science and Technology, China, and in the Department of Computer Science, St. Francis Xavier University, Canada. His research interests include parallel and distributed computing, embedded and ubiquitous/pervasive computing, and big data. His research has been supported by the National Sciences and Engineering Research Council, and the Canada Foundation for Innovation.

Jun Feng is currently studying for the Ph.D degree in School of Computer Science and Technology at Huazhong University of Science and Technology, Wuhan, China. He received the master's degree from the School of Computer Science and Information, Guizhou University, Guiyang, China, in 2013. His research interests include big data, pervasive computing and information security.

Mianxiong Dong received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan. He is currently an Assistant Professor with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. Before join Muroran-IT, he was a Researcher with National Institute of Information and Communications Technology (NICT), Japan. He was a JSPS Research Fellow with School of Computer Science and Engineering, The University of Aizu, Japan and was a visiting scholar with BBCR group at University of Waterloo, Canada supported by JSPS Excellent Young Researcher Overseas Visit Program from April 2010 to August 2011. Dr. Dongs research interests include Wireless Networks, Big Data and Cloud Computing.