Secure Cloud-based Medical Data Visualization

Manoranjan Mohanty[*]
Dept. of Comp. Sci.
National Univ. of Singapore

Pradeep Atrey
Dept. of Applied Comp. Sci.
Univ. of Winnipeg, Canada

Wei Tsang Ooi
Dept. of Comp. Sci.
National Univ. of Singapore

ABSTRACT

Outsourcing the tasks of medical data visualization to cloud centers presents new security challenges. In this paper, we propose a framework for cloud-based remote medical data visualization that protects the security of data at the cloud centers. To achieve this, we integrate the cryptographic secret sharing with pre-classification volume ray-casting and propose a secure volume ray-casting pipeline that hides the color-coded information of the secret medical data during rendering at the data centers. Results and analysis show the utility of the proposed framework.

Categories and Subject Descriptors: J.3 [Medical Information System]; K.6.5 [Security and Protection]

General Terms: Security

Keywords: Cloud Computing, 3D Medical Data Visualization, Secret Sharing, Ray Casting.

1. INTRODUCTION

Remote data visualization allows distant medical experts to analyze images that are captured by host hospitals. It typically uses a *client-server* architecture, where the host hospital that captures the data acts as the server and the remote display device acts as the client. In this architecture, the 3D medical data rendering is usually performed on the server since it produces better quality images.

Recently, it has been proposed that cloud data centers can be used for rendering [1, 2]. Although cloud-based rendering has many advantages over conventional server-side rendering, security is a major issue when medical data rendering is carried out by the third party cloud providers. An adversary with access to medical data of patients can misuse it in a number of ways. Firstly, for economical benefits, the adversary may illegally sell the disease information of patients to other interested parties such as insurance companies. Secondly, for publicity, both health information and the name of the admitting hospital of a prominent person may be leaked to public and media. Thirdly, a medical image can be purposefully modified to provide misleading information to user [3]. Therefore, medical data visualization techniques involving cloud data centers must prioritize the need for protecting patient data.

In this paper, we address the challenge of keeping color-coded information of medical image secret from the cloud data centers. Color codes are often used to represent the complex structure and function of an entity in the medical image [4]. In addition, abnormality is also color coded in the image. For example, color is used to code the area of shrinkage in MRI images of dementia and the loss of cortical thickness in images of Huntington's diseases [4]. Therefore, withholding of color-coded data from the cloud data centers protects the medical image from any adversary. The opacity value, however, can be disclosed to the cloud data center to accelerate rendering. Although the opacity value of the medical image reveals the shape of an entity, in the absence of color-coded values it does not divulge any confidential information.

This paper proposes a framework for cloud-based rendering that protects color-coded information of the medical data from an adversary with access to cloud data centers. The core idea behind this framework is to integrate a variation of secret sharing method that does not use modular prime operation with the cloud-based rendering. The modified scheme, however, uses the typical property of secret sharing that a secret data can be divided into n shares in a way that the shares themselves are arbitrary and the secret can be reconstructed if and only if at least $k \leq n$ shares are used. The avoidance of modular prime operation keeps in tact the information theoretic security property of the original scheme but with loss of some information that is resulted from the one-to-one relation between the secret value and any of its share value.

In the proposed framework, we create n number of shares of the 3D medical data at the server where the data is captured. These 3D data shares are then distributed among n cloud data centers. Upon receiving rendering request from the client, each of the n data centers performs volume ray-casting on their 3D data shares and produces corresponding share image which is again arbitrary. The share images from any k cloud data centers are received at the client end and the secret image is reconstructed. In this way, cloud data centers can possess only the share data which does not reveal any information about the secret. Hence, patient's data is secured in a group of at most $k - 1$ number of data centers.

Secret sharing method is chosen over other security schemes such as AES since it provides distributed control over the secrecy of the data rather than single point vulnerability and for its homomorphic property that make it compliant with volume ray-casting.

2. BACKGROUND AND RELATED WORK

Cloud-based Rendering. With the advent of cloud computing, the data rendering step of conventional server-side rendering has

[*]This work was completed when Manoranjan Mohanty was an intern at the University of Winnipeg.

MM'12, October 29–November 2, 2012, Nara, Japan.
Copyright 2012 ACM 978-1-4503-1089-5/12/10 ...$15.00.

been proposed to be outsourced to third party cloud data centers [1, 5]. A few enterprisers like Sinha system [2] have started to use cloud-based rendering for 3D medical data. Although the current cloud-based rendering systems are relatively efficient than the conventional server side rendering ones, they present various security risks that have not been examined yet. *To the best of our knowledge, this work is first attempt to address the security issue in the context of medical data visualization in a cloud environment*.

Volume Ray-casting. The pre-classification volume ray-casting as proposed by Levoy [6] is one of the preferred data rendering technique for cloud-based rendering of 3D medical data. Pre-classification volume ray-casting pipeline is depicted in a number of independent rendering components such as: gradient estimation, classification, shading, ray projection, interpolation, and composition. Among these components, gradient estimation, classification, and shading are performed before a ray is casted; we call these components as parts of *pre ray-projection* step. Similarly, interpolation and composition are performed after the ray is casted to object space. These two components are collectively called *post ray-projection* step. Note that outsourcing only the post ray-projection step suffices, as it affects the client interactivity.

Shamir's Secret Sharing. Shamir's (k, n) secret sharing scheme (also known as threshold scheme) [7] is an algorithm to distribute a secret S between n participants in such a way that at least k participants are required to reconstruct it. This method has been successfully applied to protect patient's medical image and its associated metadata [8] from an adversary. Secret sharing method has also been used for securing 3D objects [9]. *As far as we know, however, we are first to use secret sharing technique in conjunction with volume ray-casting*.

Application of secret sharing to pre-classification volume ray-casting has two hurdles. Firstly, pre-classification volume ray-casting needs to combine RGBA values (value A being the opacity value) of multiple voxels by performing addition and scalar multiplication operations on their share values. This problem, however, can be solved as secret sharing is homomorphic [10]. Secondly, volume ray-casting performs real number operations, which is not suitable for modular prime operation of secret sharing. To overcome this problem, we consider using secret sharing scheme without modular prime operation.

3. CLOUD-BASED SECURED RENDERING

The proposed cloud-based secured rendering framework consists of a modified visualization pipeline which performs data rendering on the cloud data centers and preserves security of medical data in third party cloud data centers by integrating secret sharing technique into the rendering pipeline.

The architecture of the proposed cloud-based rendering pipeline consists of three components: the server (e.g. hospital) that hosts the secret medical data, n cloud data centers, and the client (e.g. doctor) that intends to access the secret data (Figure 1). In this architecture, we assume that: (i) the server and the client are trusted entities, (ii) the cloud data centers are geographically closer to the client, (iii) the cloud data centers do not exchange the confidential data with each other, and (iv) the cloud data centers and the client are connected to each other via a two-way high speed network.

The basic idea of our proposed framework is to securely outsource all of the client's interaction-dependent rendering operations from server to the cloud data centers. Therefore, the conventional visualization pipeline for server-side rendering needs to be modified for our proposed architecture, which is described next.

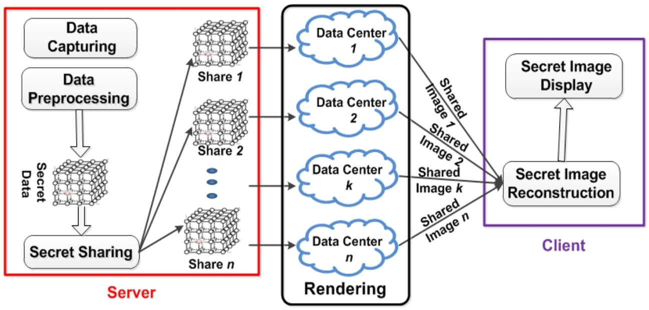


Figure 1: Proposed cloud-based medical data visualization pipeline

3.1 Modified Rendering Pipeline

To protect the patient's 3D medical data, we integrate the secret sharing technique with existing visualization pipeline and propose a modified pipeline for cloud-based secured medical data visualization. As shown in Figure 1, initially, the server performs data capturing and data preprocessing operations. It, then, creates n shares of all the preprocessed information required at the data rendering step and distributes them among n different data centers. In this step, the server also transmits the information about the data centers and the shares they are holding to the client by using an independent secure network channel. Thereafter, instead of communicating with the server, each rendering request from the client is now redirected to all n cloud data centers. Upon receiving the rendering request, each data center performs the rendering operation on its share parallely with other data centers and transmits the corresponding rendered share image to the client. After receiving at least k number of image shares from k different data centers, the client reconstructs the secret image.

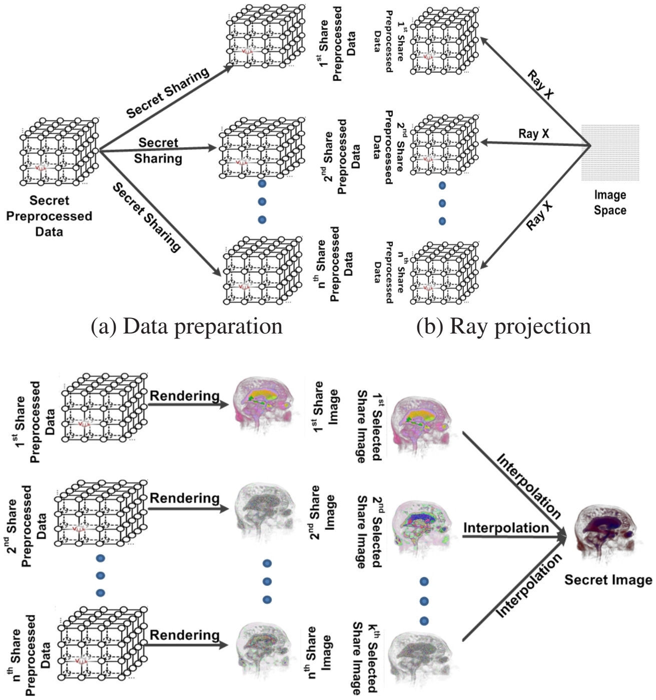
3.2 Secured Volume Ray-casting

The proposed *secured volume ray-casting* integrates secret sharing technique with pre-classification volume rendering by creating n shares of output of pre ray-projection step. As shown in Figure 2, the pipeline for secured pre-classification volume ray-casting consists of the following five main components: *pre ray-projection*, *data preparation*, *ray projection*, *post ray-projection*, and *image reconstruction*. In the following we explain these components and show that the proposed pipeline is mathematically correct, i.e., the reconstructed secret image from any k share images that are rendered via proposed pipeline is equivalent to the image rendered from the corresponding unshared secret voxel.

Pre pay-projection: This step performs classification and shading components of volume ray casting and outputs RGBA value of each data voxel to the data preparation step.

Data preparation: The illuminated RGB components of each data voxel are shared using (k, n) secret sharing method. In order to facilitate early ray termination, opacities of voxels, however, are not shared; rather, they are copied n times. Hence, instead of a single *secret voxel grid* of RGBA components, the preprocessed information is represented as n number of *share voxel grids*. When rendered individually, each share voxel grid can produce the corresponding share image.

Given any function, $F(x, a_0) = a_0 + \sum_{i=1}^{k-1} a_i x^i$, a variation of Shamir's (k, n) secret sharing method is used to divide the illuminated RGB component of each $\alpha\beta\gamma^{th}$ data voxel into n shares. Mathematically, p^{th} share of R, G, B components of $\alpha\beta\gamma^{th}$ voxel



(a) Data preparation (b) Ray projection

(c) Post ray projection rendering (d) Image reconstruction

Figure 2: Proposed secured volume ray-casting.

is calculated as:

$$C'_{\alpha,\beta,\gamma}(p) = F(p, C_{\alpha,\beta,\gamma}) = C_{\alpha,\beta,\gamma} + f(p) \qquad (1)$$

where, $f(p) = \sum_{i=1}^{k-1} a_i p^i$ and C can be replaced with each of the color components, R, G, and B. Note that, for simplicity, we have used the same function F to share the R, G, B values of each voxel; however, it does not necessarily have to be the same.

Ray projection: Rays from the image space are projected to each share voxel grid, i.e, from each projecting pixel X on the image space, n replicated copies of the ray $\vec{X}$ are projected onto all n share voxel grids.

Post ray-projection: It performs sampling, interpolation, and composition components of pre-classification volume ray-casting on each share voxel grid parallely. As the processing is same for all the shared voxel grids, we will focus our further discussion on p^{th} share voxel grid.

Step 1 (Sampling): If a projected ray $\vec{X}$ is sampled at $s_1, s_2, ..., s_t$ points of the secret voxel grid V, then, it must be sampled at the same t points of the p^{th} share voxel grid. Tri-linear interpolation is used to find the RGBA value of all these sampled points from the RGBA value of data voxels of p^{th} share voxel grid.

Step 2 (Interpolation): Given, sample points $s_1, s_2, ..., s_t$ and projected ray $\vec{X}$, without any loss of generality, for $u \leq t$, assume that color-component (C) and opacity (A) of the sample point s_{u+1} is interpolated from $C'_{u,v,w}(p), C'_{u+1,v,w}(p), \ldots, C'_{u+1,v+1,w+1}(p)$ and $A_{u,v,w}(p), A_{u+1,v,w}(p), \ldots, A_{u+1,v+1,w+1}(p)$ respectively. Hence, the interpolated color component of s_{u+1} is given by:

$$\begin{aligned} I(p, C', s_{u+1}) \quad &= \quad I(V, C, s_{u+1}) + f(p)(D_{u,v,w} \\ &\quad + D_{u+1,v,w} + ... + D_{u+1,v+1,w+1}) \end{aligned} \qquad (2)$$

where $I(V, C, s_{u+1})$ is the interpolated color component of s_{u+1} when $\vec{X}$ is projected on V. Each $D_{u,v,w}$ is a function of 3D coordinates of the sample point and the voxels required to interpolate it. Therefore, it is a constant for all share voxel grids as well as the secret voxel grid. Therefore, by replacing $D_{u,v,w} + D_{u+1,v,w} + ... + D_{u+1,v+1,w+1}$ with another symbol $D(u, u+1)$, we rewrite Equation (2) as:

$$I(p, C', s_{u+1}) = I(V, C, s_{u+1}) + f(p)D(u, u+1) \qquad (3)$$

As we do not create shares of the opacity value, the interpolated opacity of s_{u+1} of the p^{th} share voxel grid is given by:

$$I(p, A, s_{u+1}) = I(V, A, s_{u+1}), \qquad (4)$$

where, $I(V, A, s_{u+1})$ is the interpolated opacity of s_{u+1} when $\vec{X}$ is projected on V.

Step 3 (Composition): For any projected ray $\vec{X}$ along p^{th} share grid of voxels, the composition step accumulates color and opacity of all sample points along that ray. Note that, the composition of color components of a sample point is independent from the color component of other sample points and the composition of opacity is free from the color component.

For p^{th} share voxel grid, suppose $A(X, p)$ is the accumulated opacity of all sample points $(s_1, s_2, \ldots, s_t)$ along $\vec{X}$. Then, by Equation (4), $A(X, p)$ is equal to the accumulated opacity of V along ray $\vec{X}$. Mathematically,

$$A(X, p) = A(X, V) \qquad (5)$$

where, $A(X, V)$ is the accumulated opacity of V along ray $\vec{X}$.

The composited color component $C(X, p)$ of p^{th} share voxel grid can be derived as:

$$C(X, p) = \sum_{j=1}^{t} k_j I(p, C', s_j) \qquad (6)$$

Each k_j is a function of opacities of sample points along ray $\vec{X}$. Therefore, by Equation (5), it is independent of secret sharing. By substituting Equation (3) in Equation (6), we get:

$$C(X, p) = \sum_{j=1}^{t} k_j I(V, C, s_j) + f(p) \sum_{j=1}^{t} k_j D(j-1, j) \qquad (7)$$

where, each $D(j-1, j)$ is also independent of secret sharing. Therefore, for a constant K we can rewrite Equation 7 as:

$$C(X, p) = C(X, V) + f(p)K \qquad (8)$$

where, $C(X, V)$ is the composited color of V along ray $\vec{X}$.

Image reconstruction: In this step, we show that if any k shares of the RGBA values are combined, we get back the secret RGBA value. Mathematical derivation of the color components and opacity of X^{th} pixel of p^{th} share image is provided above. As shown in Equation (5), the opacity of X^{th} pixel has already been found from any of the selected shares. Therefore, interpolation is not necessary. The color components of X^{th} pixel of p^{th} share as given in Equation (8), however, represent the p^{th} share of the secret $C(X, V)$ with function $F(x, a_0) = a_0 + K\sum_{i=1}^{k-1} a_i x^i$. Therefore, by the secret sharing rule, if any k shares out of these n shares are interpolated, then, $C(X, V)$ is reconstructed.

4. RESULTS AND ANALYSIS

We implemented cloud-based secure rendering by simulating the operation of server, cloud data centers, and the client in a notebook. We assume that all 3D test data have been preprocessed by

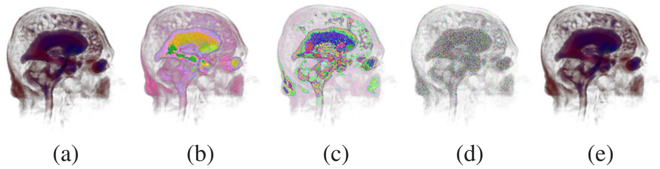


Figure 3: Secured volume ray-casting on Head volume data: (a) secret, (b, c, d) 1^{st} , 2^{nd} and 5^{th} shares, and (e) reconstructed.

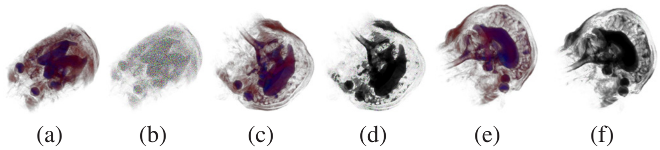


Figure 4: Images from different viewpoints: (b, d, f) are the 10^{th}, 30^{th} and 50^{th} shares of the (a, c, e) secret images respectively.

the server. Therefore, for validating our proposed work, we implemented pre-classification secured volume ray-casting by using an open source visualization package VTK and integrating a (3,5) secret sharing method into it. In the following, we discuss the results and analyze the security of proposed framework against any perceptual or computational attacks.

Figure 3 and Figure 4 show the result of the pre-classification secured volume ray-casting on head MRI data in single and multiple viewpoints respectively. As can be verified from these figures, the color information of the secret image has been arbitrated in their respective share images. Therefore, any adversary having access to individual share image cannot perceptually infer the color code of the secret image. We, however, observed that the arbitrariness of the color code in share images increases with an increase in share number (Figure 4). Such property is anticipated as the difference of share RGB values between two pixels increases with an increase in share number. Moreover, for a sufficiently higher share number, color of most of the pixels of share image may be black as value of its all three RGB components may exceed the upper limit 255.

By trial and error method, it is computationally infeasible for an adversary (say an *individual* cloud data center) to determine the secret color code from a share image. For each pixel point, there are 256^3 possibilities in determining the secret RGB value. Hence, for an image having 515×512 pixels, there are $(256)^{(3\times512\times512)}$ number of permutations to find out color information of the secret image. Moreover, even if an adversary has computational power to perform such huge number of permutations, the probability of inferring the secret image is $\frac{1}{(256)^{(3\times512\times512)}} \approx 0$.

Proposed cloud-based secured rendering technique is secure even if at most $k-1$ share images are available to an adversary (or a *group* of $k-1$ cloud centers exchanges their share information) as they cannot reconstruct the secret image unless they have information about all k share images. The only possibility of getting the missing share image is to estimate it using trial and error method. Estimation of such a share image, however, is computationally harder than the estimation of the secret image as the upper limit of range of share RGB values is greater than 255.

Cloud-based secured rendering using (k, n) secret sharing method ensures data integrity and provides protection against tampering of medical data that can occur at any of the cloud data center. The $k < n$ condition results $\binom{n}{k}$ different ways of reconstructing the secret image. Therefore, if any adversary changes the color values of one or more share image, then the reconstructed images from the tampered share images will differ to each other. Hence, by comparing two or more reconstructed images client can detect any tampering to share medical data. In addition, within a certain condition, client can also infer the uncorrupted image out of all available reconstructed images. Suppose, share images of n_1 ($n_1 < n$ and $(n - n_1) > k$) cloud data centers have been corrupted, then, there exists $\binom{n-n_1}{k}$ number of other ways to reconstruct the uncorrupted images. Unlike the corrupted images that are different than each other, all pixels of the uncorrupted images have same color values. Therefore, for $\binom{n-n_1}{k} > 1$, client can infer that any image among the $\binom{n-n_1}{k}$ number of alike images is the secret image.

5. CONCLUSION

Security of patient's data is of utmost importance in cloud-based remote medical data visualization systems. We proposed a secured cloud-based rendering framework that integrates a cryptographic secret sharing method with pre-classification volume ray-casting. Experiments and analysis show that the proposed framework is highly secure; however, it may have computational and data overhead, which requires further investigation.

Acknowledgement

We thank Ee-Chien Chang and the anonymous reviewers for their valuable comments on this research; and the Canadian Bureau for International Education for supporting the first author's internship at University of Winnipeg.

6. REFERENCES

[1] Karlheinz Dorn, Vladyslav Ukis, and Thomas Friese. A cloud-deployed 3D medical imaging system with dynamically optimized scalability and cloud costs. *Soft. Eng. and Adv. Appl., Euromicro Conf.*, 0:155–158, 2011.

[2] 3Di-cloud based medical image management and visualization platform. `http://www.shina-sys.com/assets/brochures/3Di.pdf`.

[3] Nasir A. Memon and S. A. M. Gilani. Watermarking of chest CT scan medical images for content authentication. *Intl. J. of Computer Mathematics*, 88:265 – 280, 2011.

[4] Geoffrey D. Schott. Colored illustrations of the brain: some conceptual and contextual issues. *The Neuroscientist*, 16:508–518, 2010.

[5] Denis Vazhenin. Cloud-based Web-service for health 2.0. In *Proc. of the 2012 Joint Intl. Conf. on Human-Centered Comp. Env.*, pages 240–243, 2012.

[6] Marc Levoy. Display of surfaces from volume data. *IEEE Comput. Graph. Appl.*, 8:29–37, 1988.

[7] Adi Shamir. How to share a secret. *Commun. ACM*, 22, 1979.

[8] Mustafa Ulutas, Güzin Ulutas, and Vasif V. Nabiyev. Medical image security and EPR hiding using Shamir's secret sharing scheme. *J. Syst. Softw.*, 84:341–353, 2011.

[9] Esam Elsheh and A. Ben Hamza. Secret sharing approaches for 3D object encryption. *Expert Syst. Appl.*, 38:13906–13911, 2011.

[10] Josh Cohen Benaloh. Secret sharing homomorphisms: keeping shares of a secret secret. In *Proc. on Adv. in Cryptology*, pages 251–260, 1987.