# Efficiently Secure Data Privacy on Hybrid Cloud

Xueli Huang and Xiaojiang Du
Dept. of Computer and Information Sciences
Temple University
Philadelphia, PA, 19122 USA
Email: {xueli.huang, dux}@temple.edu

***Abstract*—The growing concerns about the privacy of data stored in public cloud have hindered the widespread adoption of cloud. On one hand, large part of data, such as medical data, has a lot of images, and this kind of data may be private. On the other hand, the cloud service providers have the full access of data, and they may bleach the data for financial or other reasons. The traditional method to protect the privacy of data is to employ cryptographic algorithms, which unavoidably introduces heavy computation. Another way is hybrid cloud consisting of public and private cloud. The sensitive data is separated from non-sensitive data, and only the non-sensitive data is outsourced to public cloud. If we use hybrid cloud method directly, all the private images have to be stored in private cloud, which makes the adoption of cloud computing meaningless. Besides achieving data privacy, we should reduce computation and storage overhead in private cloud, as well as communication overhead between private and public cloud. In this paper, we propose a novel scheme to achieve the above goals. We test our scheme in real network environments (including Amazon EC2). We also propose a novel algorithm to process private image data. Our experimental results show that: (1) Our algorithm achieves data privacy but only takes about 1/1,000 the time of the AES algorithm. (2) The delay of our hybrid cloud approach (including the private and public cloud communications) is only 3% - 5% more compared to the traditional public-cloud-only approach.**

***Index Terms*—Data privacy; hybrid cloud; image**

## I. Introduction

With the rapid development of information and communication technology, the amount of data produced by organizations has grown exponentially, which makes it hard for many organizations to cost-effectively store and manage the data. Cloud computing is a new business model, and it is considered as one of most cost-effective solutions for organizations to improve their IT segment. Cloud computing provides the advantage of reduced cost through sharing of computing and storage resources. It utilizes an on demand provisioning mechanism and a pay-per-use model. Cloud computing has drawn a lot of attentions in recent years [1].

As more and more individuals and organizations stored their data in cloud, there are also concerns about cloud computing, which have affected the wide adoption of cloud [2]. On top of the list are security and privacy concerns. For example, people concern about the storage and processing of sensitive data in remote physical infrastructure that are owned by a third party, i.e., a cloud service provider (CSP). Since a CSP has full control of the data, it is possible that the CSP conduct malicious attacks on users' data for financial or other reasons. For example, a CSP could make money by revealing the data of one client (say C) to C's competitor. Meanwhile, a client's data may be leaked to the public if a CSP does not have good security mechanisms to protect its servers.

Most existing solutions (e.g., [3], [4], [5]) employ encryption/decryption techniques combined with access control and auditing to provide security and privacy for data stored on public cloud. However, in doing so, these solutions inevitably introduce a heavy computational overhead on the data owner for key distribution, data management, data query, and other operations.

In this paper, we consider a different approach: achieving data privacy by utilizing hybrid cloud. A hybrid cloud consists of public cloud (such as Amazon EC2) and private cloud, which is owned and controlled by the data owner. The privacy of data is protected by splitting user data into sensitive data and non-sensitive data, and only outsourcing the non-sensitive data to the public cloud. The sensitive data is stored in user's private cloud.

Many data (such as medical data) stored in cloud have a large number of images, which require a lot of storage and computations. A patient medical image may be private. If we directly take advantage of the approach mentioned above, all the medical images need to be stored in private cloud. This would require a large amount of storage in private cloud, and may cause most data stored (and processed) in private cloud, instead of in public cloud. Typically, one wants to minimize the storage and computation in private cloud. To address the above challenge, an important problem: How to efficiently achieve image data privacy by using hybrid cloud? Compared to using public cloud only, using hybrid cloud would have communication overhead between private and public cloud. Besides achieving data privacy, we want to reduce storage and computation in private cloud, as well as communication overhead between private and public cloud.

In this paper, we propose a novel algorithm that efficiently achieves data privacy for large data sets, especially images, stored in cloud. In our algorithm, firstly, a random noise is added to image blocks instead of pixels, and the size of block is determined by a balance between the complexity of recovering the image and communication overhead. Then a random shuffle operation is applied on the modified blocks, which makes the image hard to be recognized. To prevent the data from being analyzed, we remove relationship among tables stored in public cloud, using hash functions with different keys.

978-1-4673-3122-7/13/$31.00 ©2013 IEEE

## II. Related Work

### A. Providing Data Privacy via Cryptographic Techniques

[3] proposes a generic framework - SPORC, for building a wide variety of collaborative applications like word processing and calendaring with un-trusted servers. In SPORC, data are encrypted with users' cryptographic keys before being sent to a cloud-hosted server, therefore, the server observes only encrypted data and cannot deviate from the correct execution without being detected. Large computational overhead is introduced due to the use of traditional encryption/decrption.

### B. Access Control in Cloud

[5] proposes a scheme, which utilizes and uniquely combines techniques of attributed-based encryption (ABE), proxy re-encryption and lazy re-encryption. Each data file is associated with a set of attributes and each user is assigned an expressive access structure defined over the attributes of files. Fine-grained access control is received via key-policy ABE (KP-ABE) [6]. However, attributed-based encryption is computational expensive and consumes a lot of time of computing resources.

### C. Achieving Data Storage Security via Third Party Auditor

Users should be able to check the integrity of data placed in cloud, sometimes with the aid of a Third Party Auditor (TPA). [4] proposes an auditing protocol by utilizing the technique of public key based homomorphic authenticator. Even though the TPA in [4] can protect data from being modified or deleted by the cloud providers, the cloud provider can still access the original data and obtain private information from the data. Hence, the scheme in [4] does not provide data privacy.

## III. Achieving Data Privacy via Hybrid Cloud

### A. System and Threat Model

The architecture of a hybrid cloud is illustrated in Fig. 1. The original data come from the private cloud and are sent to a data processing center. For image data containing sensitive information, we divide the image into several blocks with the same size, and add random noise into each color dimension of every block, which breaks the pairwise blocks relationship. Then we shuffle these blocks with a random permutation, which makes the image becomes unreadable, and send it to the public cloud. We store the random noise and the information used to shuffle blocks in the private cloud. The ID of the image is sent to the private cloud and public cloud at the same time. The shuffle order is obtained from the private cloud to reorder the shuffled image downloaded from the public cloud. Then the random noise is obtained from the private cloud to recover the original image.

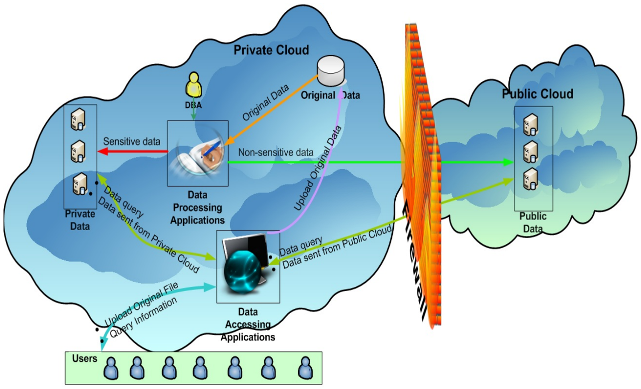

Fig. 1. The architecture of hybrid cloud

We consider an un-trusted public cloud who intends to get sensitive user data and has full control of the hardware, software, and networks in the public cloud.

### B. Data Security on Hybrid Cloud

Since the shuffled and modified image containing sensitive information has become unreadable, it can be put in public cloud.

As shown in Fig. 1, the key information used to recover data is transmitted between data center and web server within the private cloud, a user is not able to obtain the key information. Meanwhile, only data packets between the web server in private cloud and the data center in public cloud can be obtained at the public cloud. The image data sent to public cloud is either blurred or hashed data, and they cannot be used to find connections among tables in public cloud. Therefore, the key information is secured from both the web client and public cloud, which means that our scheme successfully protects data privacy.

## IV. Security of Image Data

### A. Modifying Image

*1) Dividing Image into Blocks:* Different from text, image has a larger size. It is inefficient to perform operations based on pixels, no matter what kind of encryption is taken. To speed up the operation on images, we divide a large image (size of $N \times N$) into $n$ number of blocks, where each block has the same size $k \times k$. Take the lena image for example, which has a size of $256 \times 256$, the size of each block is set to $32 \times 32$. The image is divided into $n = 64$ pieces.

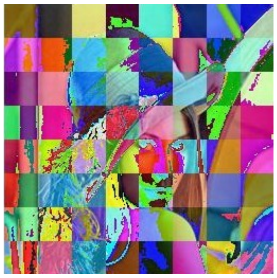

Fig. 2. Modified image

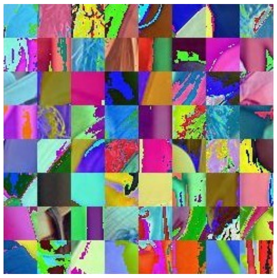

Fig. 3. Shuffled image

*2) Adding Noises to Each Block:* For each color dimension of a block, whose size is $k \times k$, each pixel value is subtracted from a random value $rnd$ (between 0 and 255). After this modification, features along two adjacent blocks are made unrelated because different random values are used on each color dimension of the two blocks.

Fig. 2 is the lena image after it is blurred on the block level. Because of human visual capability, the blurred image can still be identified by people who have seen the original one. In addition, people may still be able to gain information from a blurred image even though they have never seen the original image. For example, one can tell that Fig. 2 is a photo of girl wearing a hat.

### *B. Random Shuffle of Blocks*

To make the modified image un-recognizable, we shuffle the blocks of modified image according to our image shuffle algorithm. We cluster the $n$ blocks into a number of groups with randomly chosen strides, and each cluster has the same size $m$. The steps are given below:

1) For each cluster of modified image.
   a) Randomly choose value $stride$ from $1, 2, ..., n$.
   b) Randomly choose value $start$ from $0, 1, ..., n-1$.
   c) Keep increasing one $start$ until we find a block that has not been chosen before, and copy the block to shuffled image in order.
   d) Add $stride$ to $start$, and goto step c) until $m$ number of block have been chosen.
2) Goto step 1) until all blocks of modified image have been chosen.

After modification and shuffle operations are applied on the lena image, we obtain an image shown in Fig. 3, which has become very hard to be identified. This shows that our modification and shuffle approach can protect the privacy of images, which makes image could be stored to public cloud. In subsection IV.D, we will mathematically prove that our approach is secure.

### *C. Recovering Images*

When an image is queried, the request is sent to both the private cloud and public cloud at the same time. As we mentioned above, the information used to recover the image is obtained from the private cloud. We obtain the shuffle order $permut$ via Algorithm 1, and we re-order the blocks of shuffled image from the public cloud, which make us get the modified image. Then we obtain the random values from the private cloud, and use them to recover the original image from the modified image.

## V. Performance Evaluations

### *A. Security Analysis*

After we divide an image into blocks and shuffle it, we convert the problem to the ”jigsaw puzzle problem”, which has been proven to be NP-Complete if the pairwise affinity among jigsaw pieces is unreliable [7].

Some recent literature propose to use the features of adjacent block's edge to recover an image such that humans can identify the image content in polynomial time. [8] finds that the dissimilarity-based compatibility, which is exploited to measure the color difference along the adjacent boundary, is more discriminative. However our modifications on an image prevent such features being used to recover a shuffled image. This is illustrated in Fig. 4 and Fig. 5. With the pixel modification in each color dimension of every block, the features of adjacent block edges are also removed. According to [7], the jigsaw puzzle problem that we have is NP-complete.

**Algorithm 1** Obtaining shuffle order

```
Input: $n$: The number of blocks,$shuffle$:a string composed of $start$ value and $stride$ value.
Output: Shuffle order $perm$.
$start \leftarrow$ the start value of $shuffle$;
$stride \leftarrow$ the stride value of $shuffle$;
$cluster\_n \leftarrow$ the number of start;
$m \leftarrow \frac{n}{cluser\_n}$; //cluster size
$left \leftarrow n$; $cluster\_id \leftarrow 0$; $i1 \leftarrow 0$;
for $i = 0 \rightarrow n - 1$ do
    $orig[i] \leftarrow i$;
while $left > 0$ do
    $i \leftarrow start[cluster\_id]$;
    if $m < left$ then
        $m \leftarrow left$;
    for $j = 0 \rightarrow m - 1$ do
        while $orig[i]$ has been chosen do
            $i \leftarrow i + 1$;
            if $i \geq n$ then
                $i \leftarrow i - n$;
        $permut[i1] \leftarrow orig[i]$;
        $i1 \leftarrow i1 + 1$; $left \leftarrow left - 1$;
    $i \leftarrow i + stride[cluser\_id]$;
    $cluster\_id \leftarrow cluster\_id + 1$;
```

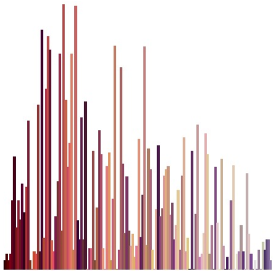

Fig. 4. Original image histogram

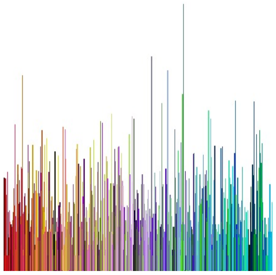

Fig. 5. Modified image histogram

We conduct experiments on several color-standard-test images of size $512 \times 512$, and each experiment is run $100$ times for each setting. The pairwise affinity is judged by the dissimilarity-based compatibility measurement of the sum of block color difference along adjacent boundaries. Take two blocks $blk_i$ and $blk_j$ for example, the Left-Right dissimilarity between them is calculated via Equation 1:

$$D(blk_i, blk_j) = \sum_{k=1}^{K}\sum_{l=1}^{d3}((blk_i(k,u,l) - random(i,l)) - ((blk_j(k,v,l) - random(j,l)))^2 \tag{1}$$

where $d3$ is the number of image color dimensions and each block is a $K \times K \times d3$ matrix, $u$ indexes the last column of $blk_i$, $v$ indexes the first column of $blk_j$, and $random(i,k)$ is the array of random values used to modify the original image. The number of blocks $n$ is set to $\lceil \frac{N \times N}{K \times K} \rceil$ is determined.

The color difference square D is assumed conform to an exponential distribution, and the probability density function is given in Equation 2.

$$P_{i,j}(blk_j | blk_i) = \lambda e^{-\lambda D(blk_i, blk_j)} \tag{2}$$

where $\lambda$ is the variance of $D(blk_i, blk_j)$ among all $blk_j$ and $blk_j \neq blk_j$. The sample space $\Omega$ is set to $\{blk_i | blk_i\ has\ a\ right\ adjacent\ block\}$.

Define Event A = "the block $blk_i^{'}$s right adjacent block has the highest compatibility score among all blocks". For both the original image and the modified image, the probability that the right adjacent block has the highest compatibility is:

$$P(A) = \frac{|A|}{|\Omega|} \tag{3}$$

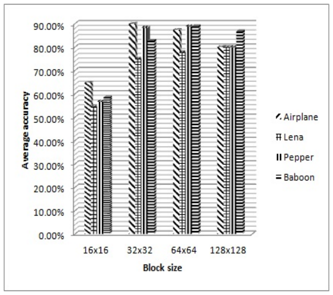

Fig. 6. Original image accuracy

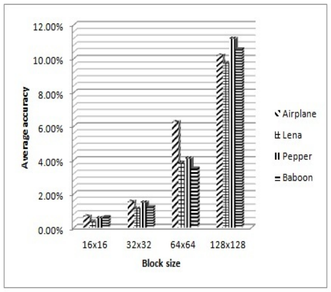

Fig. 7. Modified image accuracy

Fig. 6 presents the accuracy probability of identifying block's adjacent block correctly from the shuffled original image according to Equation 3, and the pairwise affinity method is reliable. However, after we modify the original image, it can been seen from Fig. 7 that the method has become unreliable.

We can choose value of $K$ such that the probability in Equation 3 is less than a predefined maximum threshold $maxi\_probability$. In Fig. 7, we choose $K \leq 32$ for Airplane image and $K \leq 64$ for Lena, Pepper and Baboon images, and the average accuracy is low. We can choose other values of $K$ to make the accuracy even lower, which will raise the overhead on both storage and communication.

For a given shuffled image and 200 years is considered as a long enough time to obtain the modified image from the shuffled image on a 1000MIPS machine, suppose it is divided into $n$ pieces and then randomly clustered with size of $m$. We get inequality as follows:

$$\begin{aligned} &\frac{n \times n \times (n-m) \times n \times (n-2m) \times n \times \cdots \times 1 \times n}{1000 \times 10^6 \times 60 \times 60 \times 24 \times 365} \\ &= \frac{n^{\frac{n}{m}} \times m^{\frac{n}{m}} \times (\frac{n}{m})!}{1000 \times 10^6 \times 60 \times 60 \times 24 \times 365} > 200 \end{aligned} \tag{4}$$

### B. Overhead Analysis

Achieving data privacy on public cloud is not our only goal, our another goal is to minimize communication overhead introduced by our scheme. Given an image divided into $n$ pieces and randomly clustered with size $m$, the communication overhead on private cloud is computed in Equation 5:

$$f(m,n) = (n \times n \times 3) + (\frac{n}{m} \times 3) + c \tag{5}$$

where $c$ is related with the size of TCP/IP header and the image file header. We should minimize the overhead on private cloud, which is formulated as the following optimization problem:

$$\begin{aligned} &Minimize\ f(m,n) = (n \times n \times 3) + (\frac{n}{m} \times 3) + c \\ &With\ constraints\ on: \\ &\frac{n^{\frac{n}{m}} \times m^{\frac{n}{m}} \times (\frac{n}{m})!}{1000 \times 10^6 \times 60 \times 60 \times 24 \times 365} > 200 \\ &and \\ &P(A) = \frac{|A|}{|\Omega|} < maxi\_probability \end{aligned} \tag{6}$$

As discussed above, the minimum value of $n$, $min\_n$, is determined via inequality $P(A) = \frac{|A|}{|\Omega|} < maxi\_probability$, and the maximum value of $m$, $max\_m$, is determined by Inquality 4. We can get all the pairs of $(m,n)$ that satisfy the above constraints, then we can find the minimum $f(m,n)$.

$$\begin{aligned} &min\_overhead = minimum(f(m,n)), \\ &where\ 1\ \leq m \leq max\_m\ and \\ &n \geq min\_n \end{aligned} \tag{7}$$

### C. Evaluation of Efficiency

To evaluate the efficiency of our privacy preserving method, we compare our algorithm with the Advanced Encryption Standard (AES) algorithm, which is a standard cryptographic algorithm based on permutations and substitutions.Both our algorithm and AES (128-bit key) are run on the Matlab platform installed in the same computer. For each size of image, we run 100 times modification and recovery operations and get the average time. The results are given in Table I, where the time unit is second.

TABLE I
RUNNING TIME OF OUR ALGORITHM AND AES

| Image Size | Our algorithm | AES | Time ratio |
|---|---|---|---|
| $128 \times 128$ | 0.1317 | 122.6 | 931.1 |
| $256 \times 256$ | 0.4593 | 490.1 | 1067 |
| $512 \times 512$ | 1.858 | 1957 | 1054 |
| $1024 \times 1024$ | 7.010 | 7824 | 1116 |

From Table I, we can see that the time for processing image increases when the image size becomes larger. For all the block sizes, our algorithm is about 1,000 times faster than the AES algorithm. The reason is that our algorithm has no iteration, while AES consists of four stages with many rounds. To sum up, our algorithm provides image data privacy and it is much more efficient than AES.

### D. Experiments using Amazon EC2

Our private cloud is set up in a server located at CIS department of Temple University, and public cloud is built on Amazon EC2 Cloud. The Microsoft SQL server 2005 is installed in the local server, which is used to store private and sensitive data. Amazon Relational Database Service (Amazon RDS) SQL server 2008 is installed in Amazon EC2 and is used to store non-sensitive data. The Microsoft Visual Studio 2010 software is utilized to create web-sites that provide services through webpages, which are developed using ASP.NET and C♯. Internet Information Services (IIS) is chosen as the web server, which supports both Data Processing Applications and Data Accessing Applications.

Four different sizes of lena image are chosen to evaluate the security, efficiency and overhead of our scheme. We record the delay between the time (t1) when a user sends the request and the time (t2) when the web server has the data ready, and the delay when our scheme is not used. The average of the 100 runs are reported in Table II, where the time unit is millisecond.

TABLE II
COMPARISON OF DELAY

| Image Size | Using our scheme | Without our scheme | increase |
|---|---|---|---|
| 128 × 128 | 23.3123 | 22.5023 | 3.60% |
| 256 × 256 | 76.1976 | 73.4673 | 3.72% |
| 512 × 512 | 242.4657 | 230.6346 | 5.13% |
| 1024 × 1024 | 976.4206 | 927.3578 | 5.29% |

Table II displays that the delay increases as the image size becomes larger, which is easy to understand. Table II also exhibits that our scheme only increases the delay a little bit, between $3.60\% - 5.29\%$. This proves the efficiency of our scheme. Meantime, if we compare the data of Table I with that of Table II, we discover that the execution time of our scheme implemented in C♯ (Table II) is much less than that implemented in Matlab (Table I, where the time unit is second). The reason is that we adopt the LockBitmap class that converts bitmaps to byte-arrays in the C♯ implementation, which greatly accelerates the image processing.

The communications between the private cloud and the public cloud cause bandwidth overhead.

We also run experiments to measure the communication overhead introduced by our scheme. From Table III, the communication overhead on private cloud becomes larger with the increase of cluster size when image is divided into 64 blocks. The reason is that the number of random $start$ value and $stride$ value increases even though the image size is same.

TABLE III
OVERHEAD ON PRIVATE CLOUD

| Cluster size | Number of cluster | Overhead (byte) |
|---|---|---|
| 1 | 64 | 571 |
| 2 | 32 | 389 |
| 4 | 16 | 298 |
| 8 | 8 | 245 |

From Table III, our scheme introduces little overhead compared with the image data.

## VI. CONCLUSION

To address the increasing concern of data privacy in cloud, we proposed a novel scheme that can provide data privacy, especially for image data. In our scheme, an image is divided into into blocks and the blocks are shuffled with random start position and random stride. Our scheme operates at the block level instead of the pixel level, which greatly speeds up the computation. We converted the image privacy problem into the jigsaw puzzle problem. To make the jigsaw puzzle problem NP-complete, we modified the image data based on blocks by subtracting a random value for each pixel within the same block and same color dimension. These operations make the pairwise affinity un-reliable and make the shuffled image un-recognizable. We formulated an optimization problem to minimize the overhead. By carefully selecting the number of blocks and the cluster size, the communication overhead of our scheme on private cloud can be greatly reduced. We implemented our scheme in real network environments (including the Amazon EC2) and tested the security, efficiency and communication overhead. Both our analysis and experimental results showed that our scheme is secure, efficient and has little overhead.

## ACKNOWLEDGMENT

This work was supported in part by the US National Science Foundation under grants CNS-0963578, CNS-1022552 and CNS-1065444.

## REFERENCES

[1] P. Mell and T. Grance, "Draft nist working definition of cloud computing," *Referenced on June. 3rd*, 2009.
[2] M. D. Ryan, "Cloud computing privacy concerns on our doorstep," *Communications of the ACM*, 2011.
[3] A. Feldman, W. Zeller, M. Freedman, and E. Felten, "Sporc: Group collaboration using untrusted cloud resources," *OSDI, Oct*, 2010.
[4] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *INFOCOM, 2010 Proceedings IEEE*, 2010.
[5] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *INFOCOM, 2010 Proceedings IEEE*, 2010.
[6] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on CCS*, 2006.
[7] E. Demaine and M. Demaine, "Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity," *Graphs and Combinatorics*, vol. 23, 2007.
[8] T. Cho, S. Avidan, and W. Freeman, "A probabilistic image jigsaw puzzle solver," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010.