Integrity and Privacy Sustenance of Shared Large Scale Images in the Cloud by Ring Signature

Annamalai R
Assistant Professor
Information Technology
Jeppiaar Institute of Technology,
Sriperumbudur, Chennai

Srikanth J
Assistant Professor
Information Technology
Jeppiaar Institute of Technology,
Sriperumbudur, Chennai

M Prakash, Ph.D.
Associate Professor
Information Technology
Jeppiaar Institute of Technology,
Sriperumbudur, Chennai

ABSTRACT
Data integrity is the protection of information from damage or deliberate manipulation. Large Scale image datasets are being shared exponentially. When these are outsourced in the cloud, compression and decompression of the images is required. Though images stored in the cloud can be shared across multiple users, the integrity of this shared data is prone to hardware or software failures and human errors. Previously designed techniques enable both data owners and public verifiers to efficiently audit the integrity without retrieving the entire data, butrevealing confidential information such as privacy identity. Auto Image compression and decompression from the cloud server eases the work of users. This paper proposes public auditing by exploiting ring signatures to compute verification metadata for preserving integrity. The signer's identity on each block is kept private from public verifiers who verify shared data integrity without requiring compressing and decompressing large image files which is automated through BCIF framework. The ring layout also enables simultaneous multiple auditing.

General Terms
Cloud, Compression and Decompression, Images, Public Auditing

Keywords
Large Scale Image Integrity, Ring Signature, system Owner-Identity Confidentiality for Large Scale Image Public Verification

1. INTRODUCTION
The internet is growing with cloud providers which facilitate the sharing of large scale images generated exponentially. This torrential sharing of images requires manual compression and decompression to save storage. Also there is a high possibility of the images being corrupted due to any hardware or software failure. Since cloud service providers are independent administrative entities, the fate of the images deployed is handed over to the user's ultimate control. It is critical to ensure that security must be embedded in the image service outsourcing design so that the owners' data privacy can be protected without sacrificing the usability and accessibility of the information. Hence it is essential to check for the correctness of these images by a third party auditor.

Auditing of the images ensures their integrity and bandwidth. The traditional approach for checking the data integrity is to retrieve the complete large scale image from the cloud. Then the image integrity is verified by signatures of the individual owner of the image. Hence a hint of the identity of the image sharer is liable to be exposed. Thus these previously introduced integrity techniques concentrate on public auditing and data privacy but ignore the identity privacy of the image owner in a cloud (table 1). This becomes an issue since the existing mechanisms could lead to the leakage of identity privacy to public verifiers.

Table 1: Various Auditing Mechanisms

	PDP	WWRL	Ring Structure
Public Auditing	✓	✓	✓
Data Privacy	•	✓	✓
Identity Privacy	•	•	✓

Due to the high dimension of the large scale images it is essential that the images are outsourced consuming less bandwidth on the cloud. A compression and decompression framework ensures reduced consumption of storage by the client, thus availing additional space. The auto compression and decompression of images carried out in the server does not require the client to decompress the compressed files manually.

An instance of identity leakage: Alice and Bob work together as a group and share a file in the cloud. This file is divided into blocks, which are independently signed by the two users with existing RSA oriented public auditing solutions. If this shared file is modified by a user, he or she has sign with his or her private key. Every time a modification is done the private key of the modifier is used in the block. To verify the integrity of the entire data an auditor has to choose the corresponding public key of each modifier. Thus the frequent usage of the identity is learned by the third party auditor and can easily match it to the signer on each block due to the unique binding of the public and private key through the digital certificates of public key infrastructure. This will lead to reveal the confidential information and hence it can be compromised.

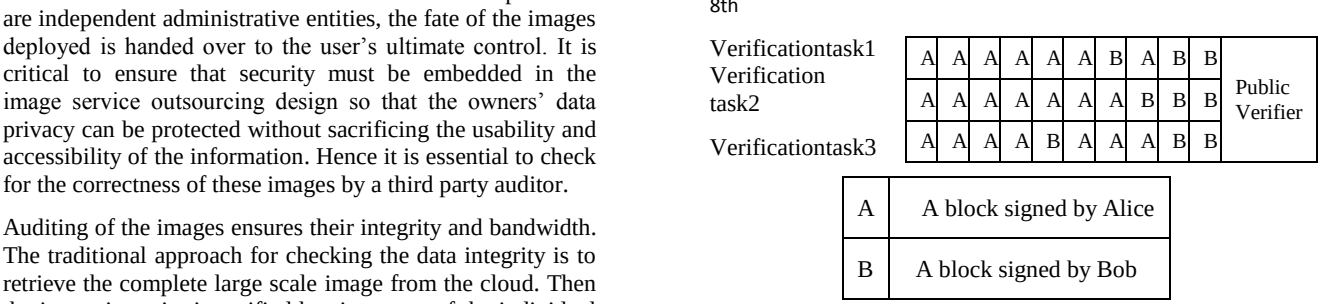

Fig. 1: Public Verifier checking the integrity of files shared by Alice and Bob

In this paper, to solve the issues of leakage of identity privacy of high definition images during public auditing, the system proposes Owner-Identity Confidentiality for Large Scale Image Public Verification. Here ring signatures are utilized to formulate homomorphic authenticators so that a public auditor is able to verify the integrity of shared large scale images without fetching the complete data. The identity of the signer on each block is kept private from the public auditor. The uploaded block stored in the cloud is compressed over the encrypted image. When the third party auditor is required to check the correctness of the blocks, the signature is verified. Once any user requests an image from the cloud the data is decompressed and downloaded via the BCIF (Bitmap Compressed Image File) framework.

2. PROBLEM STATEMENT

2.1 System Model

The system model consists of a user, administrator and a third party verifier. The administrator is the original user who creates shared data and uploads it. In the process, images are compressed automatically using BCIF framework in the cloud which is shared by the group users. These users have the access to modify the data. These members forming a group each have a private signature in the cloud. A third party auditor provides auditing services to check the integrity of shared data stored in the cloud.

The verifier sends an auditing request to the cloud and it receives a proof of the data availability from the cloud. Then the verifier checks the correctness of the auditing proof. The check is completed and when a group user requires a download of the image it is decompressed and downloaded.

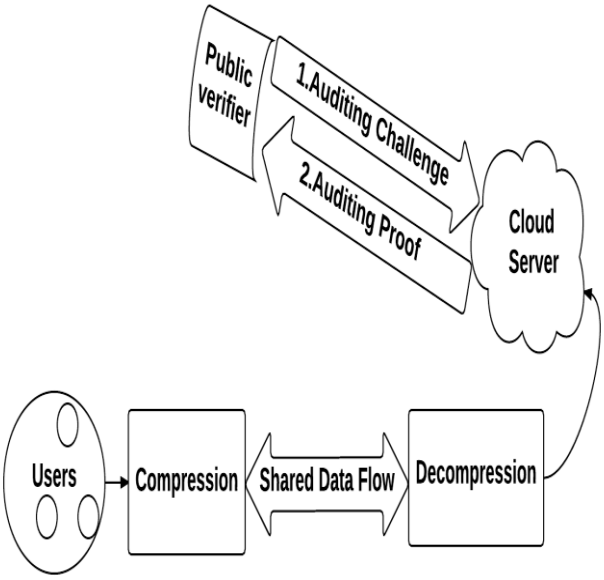

Fig.2: System Model inclusive of group users, administrator and public verifier

2.2 Threat Model

Integrity Threats: An adversary may try to corrupt the integrity of the shared data. The service provider may corrupt the data unintentionally (hardware or software failures).

Privacy Threats: The private identity of the signer may be revealed to the public verifier on any block by analyzing the signature of a particular user.

Storage Overhead: The storage and bandwidth overhead for high definition images is highly possible.

2.3 Design Goals

Verifiability: Auditing without retrieving the entire data from the cloud.

Integrity: The correct verification of the shared data.

Authenticity: Only a group user can generate ring signatures.

Confidentiality: The signer's identity is not revealed to the public verifier.

Increased Storage: The compression of large data reduces the storage capacity required.

2.4 Approaches

Every user has a private key. When a user leaves a group, this private key is removed and a fresh group of keys are generated to each of the existing users. This introduces a huge overhead on the users and in the cloud and hence is not recommendable. In the proposed solution the private keys can still be used.

Another approach is that a trusted proxy is introduced into the cloud and a new signature generated by the proxy is sent to the auditor. Thus the data is verified without any identity leakage. Yet, there is a limitation due to the probability of proxy failure. Not all group users would prefer to trust the proxy though. This can be overcome by using group signatures.

It is recommended that by using direct anonymous attestation mechanism a trusted computing group can preserve the privacy identity of the group users. A single signature from the group is created and that is introduced to the public auditor, thus sustaining the private identities.

3. RING SIGNATURE DESIGN

The secret key of one of the group members is used for calculating the digital signature that is verified by the auditor. But the signature that is used is undeterminable with a probability more than the reciprocal of the number of users i.e., 1/n, where n is the number of users in the group. Ring signatures are used to disguise the identity of the signer so as to preserve it from public auditors. Traditional ring signature do not support block verification and hence cannot be used for public auditing. Block verification does not require the complete data to be downloaded to check integrity. This method can be implemented by homomorphic authentication.

The homomorphic authentication is done by using three algorithms: KeyGen, RingSign and RingVerify.

KeyGen generates the public key and secret key for each user.

RingSign admits the group user to generate a signature on a block. It also allows the other group users who use block identifiers to generate their private keys. These block identifiers are strings that can distinguish one block from another. Lastly it creates the public key for all the group members.

RingVerify is used to check if a block is signed by the group member. This is required by the public verifier.

3.1 Homomorphic Authenticators Property

A verifier is correctly able to check the integrity of a block of image, if its block identifier id along with the ring signature is given (b, id). This is accomplished by the use of bilinear maps for multiplicative cyclic groups.

An adversary cannot forge a ring signature with homomorphic authenticators as long as the co diffie helman assumption holds. If an adversary knows the n user public key and has access to the hash values and ring signing values, he or she targets to output a valid ring signature over a block identifier (b, id) to breach through identity. An algorithm overcomes this hash query issued by the adversary by a probability of ½ randomly picking r from a set of prime numbers Z_p from a cyclic group. The algorithm returns $(g_1^{ab})^r$ to the adversary if the probability is 0, else returns $(g_2^a)^r$ for probability 1. Since r is from Z_p, $(g_1^{ab})^r$and $(g_2^{a})^r$are both from a cyclic group G and thus both the results are identical which implies that the adversary cannot distinguish the result of the hash query.

4. BCIF FRAMEWORK

Quality loss is never compromised in the encoding of images through the BCIF image compression algorithm. The design of this algorithm allows practical usage and it works fast. BCIF, a Huffman based framework is an open source lossless image compression extended from the PCIF algorithm. Images can be compressed and decompressed several times without any loss in quality unlike the JPEG format. The process of decompression is done in a very short span of time post the encoding of images since it is specifically designed for a quick decompression phase.

BCIF algorithm has been compared with significant lossless compression algorithms, PNG, JPEG200, JPEG-LS, Jasper and BMF. No other open source or closed source image compressors reach the BCIF compression ratio with the quick decompression. Various images tested benchmark the BCIF algorithm outperforming the other formats based on the compression ratios of JPEG and decompression speed of PNG, especially in high quality images.

The BCIF standard generally handles true colour images in the BMP format. The input files to be compressed must be of 24 bits per pixel hence 8 bits per colour i.e., Red, Green and Blue. Also it should not contain any alpha channel. An alpha channel specifies how one pixel's colours should be merged with another pixel when overlaid one of top of another. When the image file is compressed to BCIF file the data representing the image is alone stored, whereas the other auxiliary information from the original file is not saved. For example, some file formats contain a transparency channel or metadata about time and place like where the picture was taken, model of the used camera, etc.

4.1 Issues over Black and White Images

Since black and white images are described with 8 bit per pixel bitmaps, the BCIF program is actually cannot read them. The solution is that images should be converted to true colour with any image manipulation program. The drawback is that when the image is decompressed, the resulting BMP file will still be in true colour, resulting to be three times bigger than the original BMP.

4.2 Implementation

In BCIF algorithm, the Java implementation allows embedding these images in websites. Thus, lossless highly compressed images can be embedded in a website with no requirement of client side compatibility, other than Java installation. When the page containing the BCIF image is loaded, it will be decompressed on the client side by the Java applet and be shown on the webpage. The first image is constructed and the Java is loaded. Then the browser takes up some time to load the succeeding images for which the visualization is quite fast.

In BCIF algorithm, the first stage involving filtering and second stage involving colour filtering are applied similarly but their efficiency has been improved. The filtered determination phase complexity is higher but there is an improvement of the compression ratios which does not alter the decompression speed. The compression procedure is the main difference between the proposed algorithm and the previously used PCIF standard, which used to decompose the image in bit planes to compress them independently. By compressing the prediction errors without this decomposition a faster and a more effective compression is implemented even if it does not allow a simple parallelization as in the PCIF algorithm.

Table2. Comparison of Images of Different format

Image Format	**Size Of Images**						
	Image 1	**Image 2**	**Image 3**	**Image4**	**Image 5**	**Image 6**	**Image 7**
Uncompressed	11714190	48529974	24400566	196662	24187518	23786550	82608294
Bcif	2570674	10521270	3881256	58647	7725664	7217574	12829120
Bmf	2620296	10671592	4252220	58356	7337040	6842660	13831068
Jasper	3412906	12143802	4866658	59647	7955315	7190730	15118994

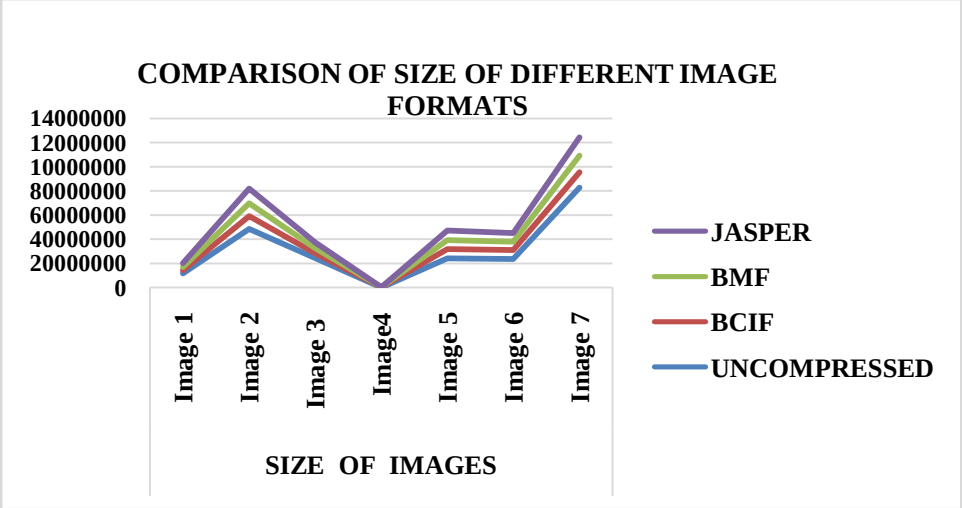

Fig.3. Comparison of Size of Different Image Formats

Table3. Comparison of Images against Compression Time							
Image format	**Compression time**						
	Image 1	**Image 2**	**Image 3**	**Image 4**	**Image 5**	**Image 6**	**Image 7**
Bcif	7.843	34.531	17.296	0.234	18.156	17.328	57.406
Bmf	3.203	19.812	5.625	0.156	6.703	6.843	27.593
Jasper	6.343	22.750	9.546	0.312	12.156	11.125	33.781

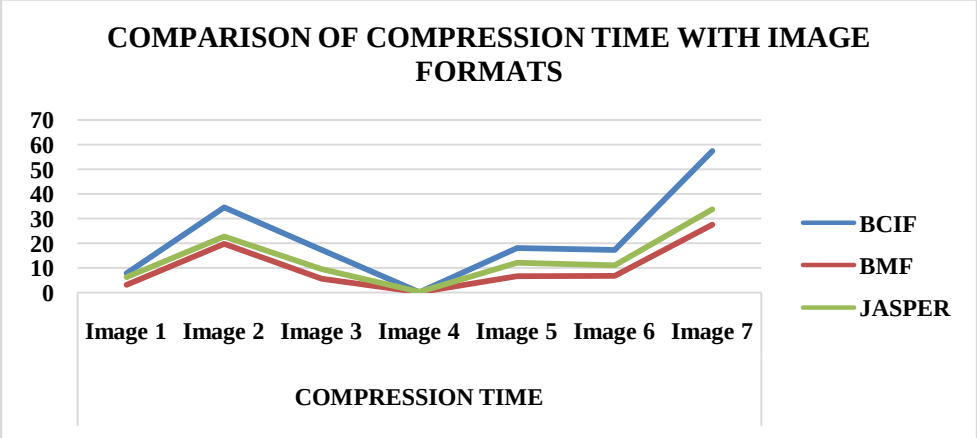

Fig.4. Comparison of Compression Time with Image Formats

Table4. Comparison of Images against Decompression Time							
Image format	**Decompression time**						
	Image 1	**Image 2**	**Image 3**	**Image 4**	**Image 5**	**Image 6**	**Image 7**
Bcif	1.640	7.796	3.718	0.046	3.906	3.765	12.765
Bmf	1.343	6.953	3.109	0.031	3.625	3.671	9.984
Jasper	5.312	17.843	8.062	0.265	10.796	9.718	28.906

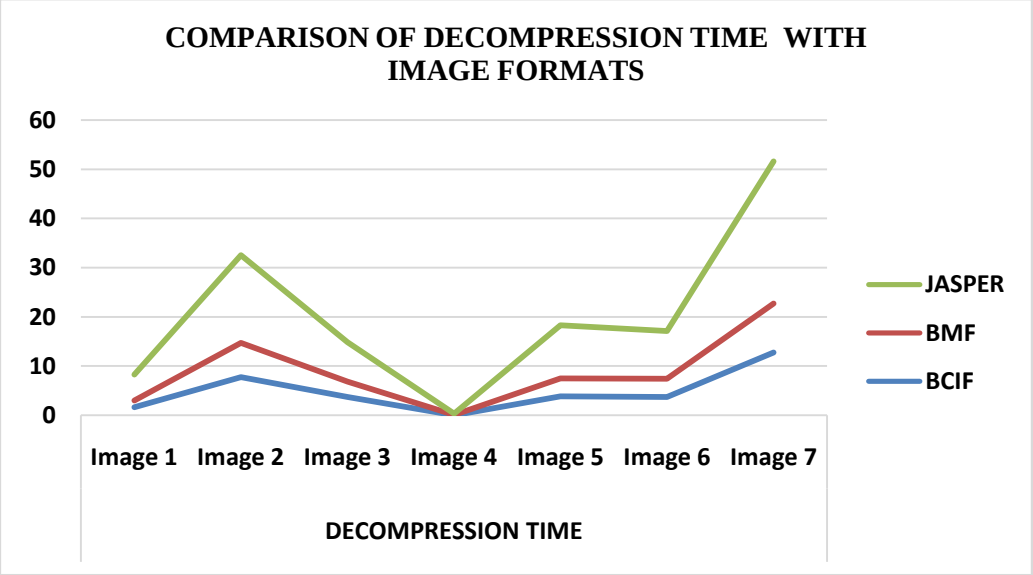

Fig.5. Comparison of Compression Time with Image Formats

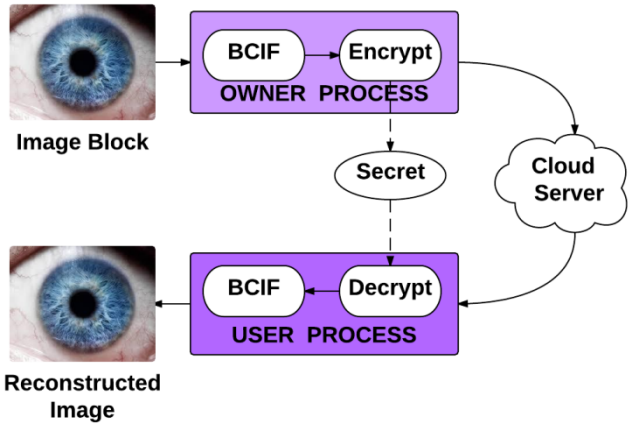

Fig 6: Image blocks compressed and uploaded fetching decompressed reconstructed image

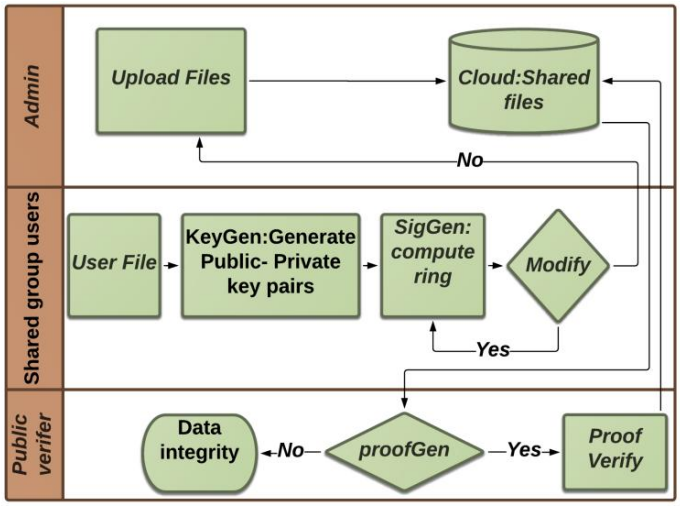

Fig 7: Working of Ring Signature

5. RING PUBLIC VERIFICATION

The construction of the ring signature includes five algorithms: KeyGen, SigGen, Modify, ProofGen and ProofVerify.

KeyGen: Algorithm generates the public and private key pairs.

SigGen: A user from the group can calculate the ring signatures on blocks using his or her private key and all the members' public keys. All of the users can insert, delete or update and also calculate the new ring signatureover a block.

ProofGen: Used by the public auditor and the cloud server. It generates the proof of possession of the shared file.

ProofVerify: Required for auditing the integrity of the shared data.

The data is compressed and encrypted before outsourcing in the cloud. An auditor can identify any corrupted block in shared data with a high probability by selecting a subset of a block. All ring signatures over the shared data need to be re-calculated with the secret key of the signer and all the public keys. During auditing, the probability to reveal the signer to the verifier is $1/d^b$, where d is the data and b is the selected blocks.

5.1 Reducing Storage

It will be very discouraging forusers, since cloud service providers, such as Amazon,will charge users based on the storage space they use.

To reduce the storage of ring signatures on shared dataand still allow the public verifier to audit shared data efficiently,an aggregated approach from [10] is exploited toexpand the size of each block in shared data into k bits.

To generate a ring signature on block m_j with homomorphic authenticators, a user aggregates block $m_j = (m_{j,1} \ldots, m_{j,k})$ as $\Pi^k_{l=1} h_l^{m,l}$ instead of computing g_1^min Equation (1), where$h_1, \ldots, h_k$ are random values of G_1. With the aggregation ofa block, the length of a ring signature is only d/k of thelength of a block.

Index	Block	V	R
1	m1	ℓ	r1
2	m2'	[3ℓ/2]	r2'

3	m2	2ℓ	r2
4	m3	3ℓ	r3
. .	. .	. .	. .
N+1	mn	nℓ	rn

6. FUTURE WORK

The further developments for the project to be focused on is to introduce a compressed sensing framework to improve upon the efficiency of the compression. Also to improve upon the requirement that the ring signature needs to be recalculated every time after a modification by one of the group user, thus reducing the overhead.

7. CONCLUSION

Thus the ring structure enables for preserving the privacy of image sharing over the cloud. The compression and decompression accomplished by the BCIF framework is done by encryption of the blocks and each of these is signed by the group user to whom the file belongs. The block is also signed by the public keys of the other group users. Then a random signature is uploaded into the cloud from where the public auditor challenges the integrity of the file. The file is fetched from the cloud as proof of existence. The integrity is checked by the verifier only by referencing the signature which cannot tell the author of the image. But it can give proof of existence and can be verified. Thus the integrity of the file is preserved. When a random user wants to download the file, it is auto decompressed and thus saves the overhead of space for the user.

8. REFERENCES

[1] Dr. Sandeep Sharma&Navdeep Kaur Khiva Pune University, India Secure Cloud Architecture for Preserving Privacy in CloudComputing using OTP/WTP Global Journals Inc. (USA) Volume 13 Issue 3 Version 1.0 Year 2013

[2] Sathiskumar R1, Dr.Jeberson Retnaraj2, Secure Privacy Preserving Public Auditing for Cloud storage ,IJIRSET Volume 3, Special Issue 1, January 2014

[3] M. Kavitha Margret, Secure Policy Based Data Sharing for Dynamic Groups in the Cloud,(IJARCET)Volume 2, Issue 6, June 2013

[4] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," Proc. IEEE Fifth Int'l Conf.Cloud Computing, pp. 295-302, 2012

[5] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the PublicCloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.

[6] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing,"Proc. IEEE INFOCOM, pp. 525-533, 2010.

[7] Stefano Brocchi and Elena Bracucci , Department of Systems and Computer Science university of Florence,Italy , bcif: another algorithim for lossless true colour image compression.