Enhanced Convergent Encryption Key Generation for Secured Data Deduplication in Cloud Storage

G. Uma¹, Dr. L. Jayasimman²
¹ Research Scholar, ²Assistant Professor
Bishop Heber College, Tiruchirappalli, Tamil Nadu, India.

Abstract. Data deduplication is broad for maintaining a sole copy of user’s data in cloud storage servers. Data Deduplication is a process of verify that whether the data are already stored in the cloud storage or not and maintain a single copy of data in the cloud. Deduplication is possible in cloud only with the use of convergent encryption (CE) technique. This technique is used to generate convergent encryption key from the user data which is to be uploaded to the cloud and the user’s data are encrypted with this convergent encryption key. This CE is exposed to different cryptography attacks. To maintain secure deduplication with convergent encryption in cloud, this paper proposes enhanced convergent encryption key generation technique. The paper introduces scheme with Key Generation and Management as a Service (KGMaaS) mechanism for generating and maintaining key for convergent encryption and deduplication process. Generation of convergent encryption key takes place in user’s side. In this process, a key is used for generating the convergent encryption key unlike existing method. Key used to generate convergent encryption key is received from KGMaaS. KGMaaS is assumed to be a trusted party for key generation and sharing. The proposed technique is analyst for security vulnerability, and it is ensure the security and efficient deduplication by procedure used in the overall data deduplication process.

Keywords: - Cloud Storage; Data Deduplication; Convergent Encryption; Key Generation;

1. Introduction
Cloud computing is place of huge amount computing energy ready to be available to users based on their necessity. The place of cloud is referred by the name of cloud data centre where high power computing servers and computers running for servicing consumers [1]. The physical computing resources in the cloud data centre are provisioned to user in virtualized manner [2]. The main objective of using cloud is to store data in the cloud storage [3]. Data storage and maintenance are crucial task in coming years. Because, based on the report of IDC, in recent year generation of data is increased and in the year of 2020 we are ready to store 40 trillion GB of data [4]. Obviously, cloud is the only option to maintain and store this huge amount data in future years. To support the cloud, and to maintain data efficiently, avoid storage of duplicate copies of data in the cloud using a technique called data deduplication [5].

Data deduplication eliminates the storage of redundant copy data in the cloud storage. It allows cloud to store unique copy of data in the cloud storage. Due the advancement of cloud, users are ready move their data to cloud, but the question is about security of data stored in the cloud. Data security is most ideal challenges in cloud to be addressed [6]. There are number research solutions for security of data in cloud [7][8][9], which are all addressed to use cryptography techniques. In cryptography technique, data are encrypted using an encryption algorithm and a secret key and stored it in the cloud. Data security should also be taken care by cloud users, thereby they are encrypted the data before it is uploaded to the cloud [10]. Encryption is the process in which each user uses a secret key to encrypt the data, for different secret key different cipher text is produced for same data copy and it makes data
deduplication impossible. So, the traditional encryption generates duplicates copies of data storage in the cloud storage. To avoid this problem and for efficient deduplication in cloud storage, convergent encryption [11] is used. Convergent encryption generates hash key from the user’s data, this key is considered as encryption key, which is used to encrypt the data with encryption algorithm. Convergent encryption techniques are very vulnerable to brute-force and dictionary attack[12]. Because the generation of convergent encryption key, also known as hash key, is not using any secret key to generate convergent encryption key. Hence, this key is easily hacked by the hackers. In some cases, a key is used for generating convergent encryption key. In this case, user should generate and maintain the key with them; if this key gets lost, then, the data is viable to hack. At the same time, increased numbers of data will also increase numbers of key to be stored in the user’s side. Users have more work burden to maintain the keys and metadata details [13].

To address the problems stated above, this paper proposes a convergent encryption key generation algorithm, which uses a key. This key is generated from KGMaaS, which is introduced in this paper as key generation and management service. In this approach, user no needs to maintain or generate key with them. Key is provided as a service from KGMaaS. The advantages KGMaaS it also provide keys to other user with same data copy by authenticating the users.

2. Related Work
Deduplication is an energetic research topic in cloud computing area of research. This section discusses some of the research works carried out in data deduplication with convergent encryption. Jin Jin Li et al. [13] proposed two different approaches for convergent encryption key management, baseline approach and Dekey. The baseline approach has two critical deployment issues. First method is not efficient due to the generation of number of keys. Next it is unreliable, because it requires each user to dedicately protect his/her own masterkey. The data is lost, when master key is lost. The proposed another method called Dekey, it uses RSSS (Ramp Secret Sharing Scheme). This is used to distribute the encryption key to different key server. Instead of sharing the key author may use a key generation system in cloud as a service to enable the key to authorized users.

Taek-Young et al. [12] proposed an approach with access privilege to generate the key for convergent encryption. User without proper privileges will not able to generate a convergent key and thus can’t access the shared data. This approach got key from key server using RSA blind signature based oblivious PRF protocol defined by DupLESS scheme [14]. This scheme has applied the privileged information into the process in order to allow only the authorized user to access to data. The model contains entities like users, cloud storage providers and an authorization server (or a key server). AS is used to generate and manage the private key, and also computes a convergent encryption key for specific file based on the privileges of a user. Authors failed to describe how user with same data copy can get the convergent key.

Ashish Agarwala et al. [15] proposed a secure data deduplication protocol, DICE (Dual Integrity Convergent Encryption), in which it is focused on i) to prevent the duplicate-faking and erasure attacks, and ii) to provide integrity check at both client and server ends. Generated tag is uploaded; the server performs the integrity check of the received tag. The integrity check is performed when the client downloads the tag to retrieve the ciphertext. As a result, reduce the bandwidth consumption by sending only the tag instead of the long ciphertext while at the same time achieving deduplication. Key generation and management process is not clearly described in this paper.

More numbers researches are proposed different approach for deduplication with convergent encryption, but authors are concentrated on procedure of deduplication process. No researchers were tried to propose new convergent encryption key generation algorithm. Existing work describes the different ways of deduplication, and use existing algorithms for convergent key generation and encryption of user’s data. As said above, convergent key generation in existing approaches were not used any key; this is very vulnerable to brute force attack or confirmation of plaintext attack. To avoid the challenges in convergent key generation and deduplication, this paper proposes an enhanced convergent encryption key generation algorithm.

3. Problem definition
Data deduplication is only the right direction to maintain a unique copy of data in the cloud. The main task of deduplication is convergent encryption, but it is vulnerable to brute force attack and dictionary
attack [12]. In convergent encryption, there is a possibility to poison attack that includes duplicatefakingattack and erasure attack, where the malicious adversary replaces the original file with the corrupted one. As a result, honest users lose their files and are subjected to download the faked ones [15]. Another problem against convergent encryption is that confirmation of a file attack in which an attacker can effectively confirm whether the encrypted file same as original encrypted file [16][17].

Deduplication is not possible when the traditional cryptography techniques are used. But, Convergent encryption is more suitable because it derives a key from the user data and encrypts the user data with the convergent key [18]. Convergent key is a hash value and procedure of traditional convergent encryption technique not using any key to generate the convergent key. Generally, this type of algorithms, key plays a vital role to protect the data derived from the algorithm. But, without a key, it is very easy to find the way or procedure used to derive the output of the algorithm. In some cases, Convergent key is derived by using a hash algorithm with key. This time the key is generated and maintain by client itself, this should give more work burden to the users. To protect the convergent encryption from the attacks and maintain the keys used for convergent encryption, this paper proposes an enhanced convergent encryption key generation algorithm and also introduces Key Generation and Management service which provided as a service to the users.

4. Objective of the paper
The paper addresses issue in cloud storage for maintaining the data. The optimum aim of our research work is to develop framework for secured data deduplication in cloud environment.

The objective of this paper is to propose a convergent encryption to support secured data deduplication in cloud. Here, the paper also introduced an independent trusted service provider for key generation and maintenance.

5. Methodology
The user’s data should be stored in the cloud in a secured manner. Deduplication supports to avoid redundant copies of same data stored in the cloud. This paper proposes a scheme with introduction of KGMaaS. KGMaaS is an independent trusted key generation and management service provider. It helps to get the key for convergent encryption key and maintain a metadata about users and key generation.

5.1. Proposed Scheme’s System Model
Figure 1 shows abstract sample cloud system model, it consists of User(U), KGMaaS and Cloud Storage (CS), where how the deduplication is happened with convergent encryption is explained in the following procedures.

1. Initially, users want upload a data (UD) to the cloud, as per the new procedure, they must generate token (TKN) from the user’s data using TKN = Tkn_Gen(UD) primitive function.
2. The token TKN is forwarded to the KGMaaS through a secure channel to get a key for generating convergent encryption key (CEK).
3. KGMaaS verify the metadata whether TKN already exist in the database or not, if it is already exist in the database, then, KGMaaS forward the corresponding key previously generated for same TKN to the user. If the TKN does not exist in the metadata then, KGMaaS generates a key (GCk) for TKN send to the user. KGMaaS maintain metadata for each user TKN and its corresponding key details.
4. KGMaaS forward the corresponding key GCK to the user for generating convergent encryption key CEK.
5. Users generate the convergent encryption key using the proposed algorithm CEK = CEkey_Gen(GCK) with the key GCK received from the KGMaaS.
6. After the key generation, UD is encrypted with the convergent key using cryptography encryption technique E_D = CEEnc_Alg(U_D, CEK). After encryption, a Tag (TG) is generated from the encrypted data (E_D) using TG = Tag_Gen(E_D) for verifying the duplication of data.
7. Users forward the TG to cloud to check whether the data is already stored in the cloud or not.
8. Cloud CS verify the $T_G$ with $T_G'$ which one derived from the cloud, if $T_G==T_G'$ then data is already in the cloud storage. Now a link is generated for this user for the same data. If $T_G!=T_G'$ then cloud system asked user to forward the data $E_D$.

9. Users forward the $E_D$ to CS, now deduplication is verified and only one copy of data is stored in the CS.

Figure 1. Abstract Methodological diagram of Cloud Environment for Deduplication

Figure 1 represents the work flow procedure of the proposed scheme and the paper also proposed an enhanced convergent encryption key generation algorithm, which is described in the next section.

6. Proposed Enhanced Convergent Encryption Key Generation (ECEKGA)

Convergent Encryption is process of generating key (hash value) from the user’s original data and encrypts the data with the generated key. This process will produce the same encrypted data which will help to identify the duplicate data. This paper proposed algorithm for generating the convergent key by digest the user’s data. The convergent encryption key generation algorithm is represented as $CE_k = CEKeyGen(GC_K)$. $CE_k$ is convergent encryption key. $CEKeyGen()$ primitive function for generating convergent encryption and $GC_K$ is a key which is generated from KGMaaS for the token $T_K$ sent by the users. User gets a key $GC_K$ from the KGMaaS and using the key and with the proposed convergent encryption key generation algorithm; he/she can generate a $CE_k$. The proposed convergent encryption key generation is digested algorithm, which reads the user’s data and follows the unique steps in the algorithm to generate the $CE_k$. The following pseudo represents the proposed convergent encryption key generation algorithm.

6.1. Pseudo code ECEKGA

Sub $CEKeyGen(GC_K)$
1. $U_D \leftarrow$ user’s data
2. $N \leftarrow$ sizeof($U_D$)
3. $U_D[N] \leftarrow$ array($U_D$)
4. for $i \leftarrow 1$ to $N$
   $A_{SCU_D[i]} \leftarrow$ ascii($U_D[i]$)
   next $i$
5. $j \leftarrow 1$
6. $k \leftarrow 1$
7. for $i \leftarrow 1$ to $N$
   if ($i \% 2 == 0$) then
      $E_{BLOCK}[k] \leftarrow A_{SCU_D[i]}
      k \leftarrow k+1$
   else
      $O_{BLOCK}[j] \leftarrow A_{SCU_D[i]}
      j \leftarrow j+1$
   end if
next $i$
8. Mid$\leftarrow$N/2
9. for $i \leftarrow 1$ to Mid
   $S_{BLock}[i] \leftarrow E_{BLock}[j]+O_{BLock}[j]$
   next i
10. for $i \leftarrow 1$ to Mid
    BinUp$\leftarrow$append(binary($S_{BLock}[i]$))
    next i
11. BinN$\leftarrow$sizeof(BinUp)
12. Blck$\leftarrow$BinN/256
13. m$\leftarrow$1
14. Bin$\leftarrow$block of 256 0’s
15. for $i \leftarrow 1$ to Blck
    Binbk[i]$\leftarrow$split(BinUp, m, m+255)
    m$\leftarrow$m+256
    Bin$\leftarrow$binary_addition(Bin,Binbk[i])
    next i
16. BinUp$\leftarrow$Bin $\oplus$ GCK
17. Blck$\leftarrow$sizeof(BinUp)/8
18. m$\leftarrow$1
19. for $i \leftarrow 1$ to Blck
    DecUp[i]$\leftarrow$decimal(split(BinUp, m, m+7))
    AscUp[i]$\leftarrow$ascii(DecUp[i])
    AscBuff$\leftarrow$append (AscUp[i])
    m$\leftarrow$m+8
    next i
20. CEK$\leftarrow$AscBuff
21. End sub

The pseudo generate the convergent encryption key, user can use this key to encrypt the user’s data using encryption algorithm. This proposed algorithm is digest algorithm, it digest the user’s data input and generate the key based content of the data. For generating the CEK proposed algorithm uses a key and it is received from KGMaaS. Users no need to keep the key with them. They can securely the delete this key. Because KGMaaS takes care for maintaining the key and it ready to forward the key to users when they submit and proof their own ship of data. Generated CEK also stored in KGMaaS associated with GC_K. To retrieve these keys from KGMaaS, users are verified with their data ownership. Users with same data copy can generate the same TK from the data. By submitting the TK, they could prove their data copy authentication to KGMaaS. If the TK is already available in the KGMaaS, then it forwards the the corresponding GC_K and DE_K to user. If it is not exist in the KGMaaS, then, it generates a GC_K for TK and sends GC_K only to the user. If the TK does not exist in the KGMaaS, then, it means that, for a new data copy user is requested for a key, this time KGMaaS, only forwards the GC_K. If TK does exist in the KGMaaS then, it means that, CE_K is already generated for data copy and KGMaaS forwards GC_K and CE_K to the users.

7. Simulation Details
Simulation environment is discussed in this section; proposed work in this paper is one of the works in our research work. For our overall research work, we setup the simulation environment. Windows server 2008 micro instance with 30GB storage and 1GB RAM is rented from Microsoft Azure Cloud infrastructure which is the cloud server for storage. Proposed work for convergent key generation is developed using ASP.Net in Visual Studio 2012 and it is hosted in the platform provided by Microsoft Azure. There is main cloud service which contains all the process of the research work, where this key generation is also included with in. KGMaaS is developed as a separate web service which is connected to main cloud service. In the main cloud service, there are provisions overall research work, which consists of generation of Token TK from the user’s data, forward requisition for uploading data,
received key GC_K from KGMaaS, generation of convergent key CE_K, encrypt the data ED using CE_K, 
generation Tag TG, verify tag data for deduplication, uploading the data to cloud server, verify the user 
for proof of owner ship and for user data is download from the server. All these process are coded in 
the main cloud service which deployed in the Azure Platform as a Service from Microsoft. Figure 1 
shows the diagrammatical representation of the simulation environment.

![Diagram](https://example.com/diagram.png)

Figure 2. Simulation Environment of our overall Research Work

This paper concentrates only on portion of KGMaaS and convergent encryption key generation. This 
same simulation setup can also be created in Amazon and Google Cloud Platform.

8. Analysis Proposed Work

Proposed work is analyst security of convergent encryption and efficient deduplication. As it is said 
early, convergent encryption is vulnerable to different attacks, like brute force and dictionary or poison 
attack. The proposed key generation generates the key from user’s data by digest the input user’s with 
proposed procedure of ECEKGA. ECEKGA uses a key GC_K for generating CE_K. The GC_K generated 
from KGMaaS by requesting through TKN. By submitting the data to cloud application for upload, 
process is start to generate the TKN. Original data owner only can generate the TKN, and forward it the 
KGMaaS and Generated GC_K for TKN and it is securely forwarded to the user. User generates the CE_K 
using proposed algorithm with GC_K. Where, without knowing the GC_K no one can generate the CE_K. 
Hence, it avoids attacks like brute force, dictionary and poison attack by duplicate faking and erasure 
attack is impossible. Flow of process is start from the user’s data UD \(\rightarrow\) TKN \(\rightarrow\) GC_K \(\rightarrow\) CE_K \(\rightarrow\) ED \(\rightarrow\) TG. 
TG is finally forwarded to cloud server for deduplication checking, if it is duplicated TG, then, a new 
link is created to the existing copy of data in server. If TG is not exist in the server users can upload the 
ED to the cloud server. To retrieve the data, users are authenticated where they want proof their owner 
ship of data. Hence, the deduplicated data are protected from unauthorized access. By the procedures 
for proposed scheme ensure the security of convergent key and deduplicated data.

9. Conclusion

Data duplication with convergent encryption support the cloud to eliminates the duplicates copies of 
user’s data. This paper proposed an enhanced convergent encryption key generation algorithm, this 
algorithm use a key with the user’s data to generate the convergent key unlike existing approach. In 
the proposed scheme, using KGMaaS and convergent encryption, should protect the key and data from 
attack describe in the previous sections. The proposed work is simulated using the Azure cloud 
platform. All the works are coded and developed as cloud service using ASP.Net Application. The proposed scheme efficiently manages the key and reduce work burden to the users from generating 
and maintaining the key. ECEKGA is effectively executed for generating the key CE_K by using GC_K. By the analysis the proposed scheme, security will be ensured for convergent key and deduplicated 
data stored in the cloud storage. Further, this work will implement in the cloud environment and also 
will compare with existing approaches.

10. References

[1] Arockiam, L, Monikandan, S, & Parthasarathy G 2011 Cloud computing: a survey Int. J. Internet 
Comput 1 pp 26-33.
[2] Uma G, Jayasimman L 2018 Survey on Data Deduplication Techniques used for Efficient Management of Cloud Storage International Journal of Scientific Research in Computer Science Applications and Management Studies 72(2018) pp 163-170

[3] Mather T, Kumaraswamy S and Shahed L 2009 Cloud security and privacy Chapter 4 O’Reilly Media Inc, pp 61-71.

[4] David Reinsel, John Gantz and John Rydning 2017 Data Age 2025: The Evolution of Data to Life-Critical Don’t Focus on Big Data; Focus on the Data That’s Big, https://www.seagate.com/www-content/our-story/trends/files/Seagate-WP-DataAge2025-March-2017.pdf pp 1-25.

[5] Mark W. Storer, Kevin Greenan, Darrell D. E. Long, Ethan L. Miller 2008 Secure Data Deduplication, ACM proceeding of Storage pp1-10.

[6] Arockiam L, Monikandan S 2013 Data security and privacy in cloud storage using hybrid symmetric encryption algorithm International Journal of Advanced Research in Computer and Communication Engineering 28 pp 3064-3070.

[7] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou 2014 A Hybrid Cloud Approach for Secure Authorized Deduplication, IEEE Transactions on Parallel and Distributed Systems pp 1-12.

[8] Monikandan S, Arockiam L 2015 Confidentiality technique to enhance security of data in public cloud storage using data obfuscation Indian Journal of Science and Technology 824 pp 1-10.

[9] Megala D, S. Pothumani, C. Anuradha 2017 Secure and Efficient De-Duplication System in Cloud Computing, International Journal of Pure and Applied Mathematics, 11620 pp 531-535.

[10] Arockiam L, Monikandan S 2014 Efficient cloud storage confidentiality to ensure data security. IEEE International conference prod. in Computer Communication and Informatics pp 1-5.

[11] J.R. Douceur, A. Adya, W.J. Bolosky, D. Simon, and M. Theimer 2002 Reclaiming Space from Duplicate Files in a Serverless Distributed File System in Proc. ICDCS pp 617-624.

[12] Taek-Young Youn, Ku-Young Chang, Kyung Hyune Rhee, and Sang Uk Shin 2016 Authorized convergent encryption for client-side deduplication IT CoNvergencePRACTice (INPRA) 2 pp 9-17.

[13] Jin Li, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick P.C. Lee, and Wenjing Lou 2014 Secure Deduplication with Efficient and Reliable Convergent Key Management IEEE Transactions on Parallel and Distributed Systems 256 pp 1615-1625.

[14] Mihir Bellare, Sriman Keelveedi, Thomas Ristenpart 2013 DupLESS: Server-Aided Encryption for Deduplicated Storage, Proc. of the 22th USENIX Security Symposium (SEC’13) pp 179–194.

[15] Ashish Agarwala, Priyanka Singh and Pradeep K. Atrey 2017 DICE: A Dual Integrity Convergent Encryption Protocol for Client Side Secure Data Deduplication, IEEE International Conference on Systems, Man, and Cybernetics (SMC) pp 2176-2181.

[16] Zooko Wilcox-O’Hearn, 2008 Drew Perttula and Attacks on Convergent Encryption, 2018 https://tahoe-lafs.org/hacktahoeafs/drew_perttula.html, accessed on September 2018.

[17] Convergent encryption, 2018 https://en.wikipedia.org/wiki/Convergent_encryption, accessed on April 2018.

[18] M. Bellare, S. Keelveedi, and T. Ristenpart, 2013 Message-Locked Encryption and Secure Deduplication in Proc. IACR Cryptology 631 pp 296-312.