International Journal of Future Generation Communication and Network
Vol. 14, No. 1, (2021), pp. 578-591

Deduplication Of Data In Cloud With Enhanced Security

[1]mrs.M.Mythili, [2]mr.S.Simonthomas

[1]Assistant Professor, Department of Information Technology, Kongunadu College of Engineering and Technology, Trichy, Tamilnadu, India.
[2]Assistant Professor, Department of Information Technology, Kongunadu College of Engineering and Technology, Trichy, Tamilnadu, India.

Abstract

Data deduplication is the most important mechanism for data compression for removing identical copies of recurrent data, and it has been broadly used in cloud environment to reliquary the amount of storage warehouse and saves the bandwidth of data. To preserve the confidentiality of the sensitive and secure data as uphold de duplication, this convergent encryption algorithm has been proposed to convert cipher text before it outsourcing. To conserve the information security, this concept makes to appear the difficulty of authorized data de duplication. Dissension from conventional de duplication systems, the different access specifies to users are also considered in duplicate checking for the metadata. This concept also presents various new de duplication techniques for supporting the authorized duplicate investigate in hybrid cloud computing architecture. The security analysis helps to safeguard the proposed model of security protection. As a evidence of this concept, can develop a prototype for proposed authorized duplicate investigate mechanism and conduct of testing using this proposed prototype techniques. To present that the proposed method of authorized duplicate investigate scheme cause the minimal overhead reconciled to ordinary operations. To ensure data confidentiality this data is stored in an cipher technique using Advanced Encryption Standard (AES) algorithm.

Keywords: Data deduplication, Advanced Encryption Standard (AES), Warehouse, Metadata, Cloud Computing

1 Introduction

Cloud environment provides seemingly unlimited virtual resources to the users and providing services across the internet, while hiding the platform and implementation detail to the user. Nowadays a cloud service provider wide offers both high storage capacity and massively parallel computing of resources very low costs. As cloud computing becomes regular, an increasing the amount of data is keep on storing in the cloud and sharing by users with specified access privileges, which define the access rights of the stored data. One critical challenge of cloud storage services is the management of the ever-increasing volume of data. To improve data management scalable in the cloud computing, de duplication has been a well-known method and has attracted more attention. Data deduplication is an important data compression method for removing duplicate entry of repeating data entry in cloud storage. This mechanism is used to increase the storage capacity and also be applied to the transfers of data through network and to minimize the number of bytes

ISSN: 2233-7857IJFGNC

that must be sent. Instead of having multiple number of data copies with the similar content, deduplication discards the redundant data by keeping only one physical data and referring other unnecessary data to that copy. Deduplication take place at either the file level or the block level. For file level deduplication, it removes duplicate data in the same file. Deduplication in the block level, which removes duplicate blocks of data that occurs in non-identical files. Although data deduplication brings a lot of features like security and privacy concerns arise as user sensitive data are susceptible to both insider and outsider attacks. Traditional encryption, is to providing data confidentiality, incompatible with data deduplication.

Specifically, the encryption technique requires different users to encrypt their data with their private keys. So, the identical data copies of different users will cause to different cipher texts, making deduplication impossible. The Convergent encryption was proposed to enforce the data confidentiality while making deduplication process.

It encrypts/decrypts the data and copy with a secret key, which is received by computing the cryptographic hash key value of the content the the data copy. After data encryption and key generation, the users keep the keys and then sends the cipher text to the cloud environment. Since the encryption methods is to decisive and derived from the data content, and identical data copies can generate the same convergent key with the same cipher text. To determine unauthorized access, secure protection proof of owner's protocol is also needing for providing the user really owns the same file when a duplicate is found. After this proof, the subsequent users with unique file contents will give access to a pointer from the cloud server without the need of uploading the same content file. A user can access the encrypted file with the pointer from the cloud server, which can be decrypted by the coherent data owners with their convergent keys.

Thus, the convergent encryption technique allows the cloud to carryout deduplication on cipher texts and the proof of ownership is to protect the unauthorized access of user file. However, the previous deduplication systems will not carryout for differential authorization duplicate investigation, which is an important to many applications. In such a way an authorized deduplication system, each user is issued a set of privileges during system initialization. Every file stored to the cloud is also restricted by a set of privileges and describe which kind of users is allowed to do the duplicate check and accessing the files. Before submitting the file duplicate file check request for some file, the user needs to take this file and own the privileges as input access. The user is able to find a duplicate for this file if and only if there is a copy of this file and a matched privilege that are stored in cloud.

To save cost and capable management, the data will be transferred to the storage server provider (SCSP) in the public cloud with specific privileges and the deduplication technique will be applied to keep only one copy of the same file contents. Because of privacy consideration, some files will be encrypted and allowed the duplicate check by employees with some specific discounts to realize the access control. Traditional de duplication methods are based on collection of encryption standards, although providing confidentiality to some extent; don't support the duplicate check with differential privileges. In alternate ways, no differential privileges have been considered in the deduplication based on convergent encryption mechanism. It appears to be considered as if we want to recognize both deduplication and differential authorization duplicate check at the same time.

SCOPE: The main goal is to access the de duplication and distributed storage of the data and meta data across multiple storage servers. Data deduplication mechanism is widely employed to backup data and reduce storage and network overhead by identifying and removing and redundancy data.

ISSN: 2233-7857IJFGNC

2. PROPOSED SYSTEM

Data de duplication is a specialized data compression technique for eliminating duplicate copies of repeating data in storage. The technique is used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. Instead of keeping multiple data copies with the same content, de duplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy.

Deduplication can take place at either the file level or the block level. For file level deduplication, it eliminates duplicate copies of the same file. To enhance our system in security. Specifically, we present an advanced scheme to support stronger security by encrypting the file. In this way, the users without corresponding privileges cannot perform the duplicate check. Furthermore, such unauthorized users cannot decrypt the cipher text even collude with the S-CSP.

Security analysis demonstrates that our system is secure in terms of the definitions specified in the proposed security model. AES algorithm is used for file-based encryption & decryption process.192-bit encryption, and is used in most modern encryption algorithms, protocols and technologies including AES.

Efficient storage allocation: Deduplication only writes unique data to disk, making it possible to greatly reduce the amount of capacity required for storage and allocate more space for backups.

Cost savings: Better storage allocation allows organizations to get much better mileage out of their storage devices. This can result in a significant cost savings.

Faster Access: By eliminating redundant data, it minimizes Access the data from the cloud.

3. SYSTEM ARCHITECTURE

INTRODUCTION

Design is a multi- step that focuses on data structure software architecture, procedural details, algorithm etc… and interface between modules. The design process also translates the requirements into presentation of software that can be accessed for quality before coding begins. Computer software design change continuously as new methods; better analysis and border understanding evolved. Software design is at relatively early stage in its revolution.

Therefore, software design methodology lacks the depth, flexibility and quantitative nature that are normally associated with more classical engineering disciplines. However, techniques for software designs do exit, criteria for design qualities are available and design notation can be applied.

ARCHITECTURE DIAGRAM:

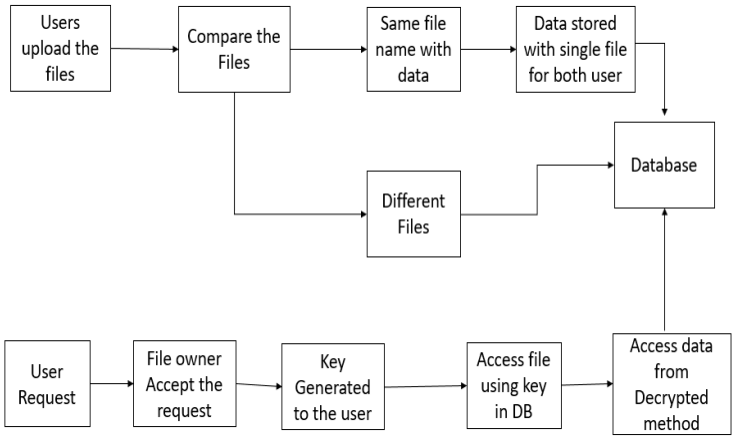


Figure 1: System Architecture

DESIGN STRUCTURE:

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

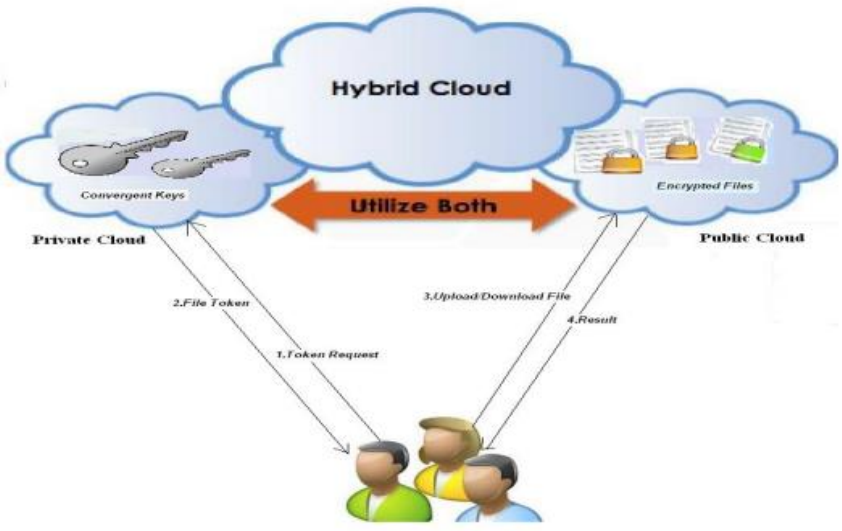


Figure 2: System Design

The corporate and private users outsource their data to cloud storage providers, recent data breach incidents make end to end encryption an increasingly prominent requirement. Unfortunately, semantically secure encryption schemes render various cost-effective storage optimization techniques, such as data deduplication, ineffective. To present a novel idea that differentiates data according to their popularity. Based on this idea, we design an encryption scheme that guarantees semantic security for unpopular data and provides weaker security and better storage and bandwidth benefits for popular data. This way, data deduplication can be effective for popular data, whilst semantically secure encryption protects unpopular content.

4. SYSTEM SPECIFICATION

MODULE DESCRIPTION

1. User Module
2. Server start up and Upload file
3. Secure DE duplicate System
4. Download file

1. USER MODULE

In this module, Users are having authentication and security to access the detail which is presented. If User Want to upload File he must verify his self using SMTP .Before accessing or searching the details user should have the account in that otherwise they should register first.

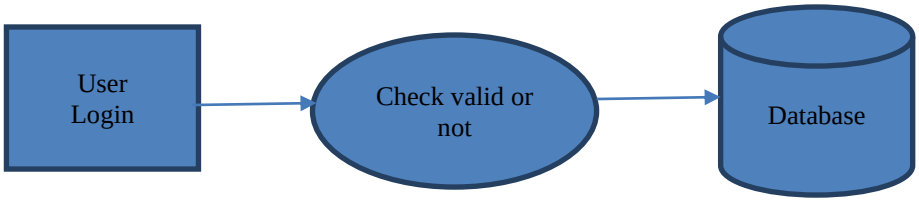


Figure 3: User Module

2. SERVER START UP AND UPLOAD FILE

The user can start up the server after cloud environment is opened. Then the user can upload the file to the cloud.

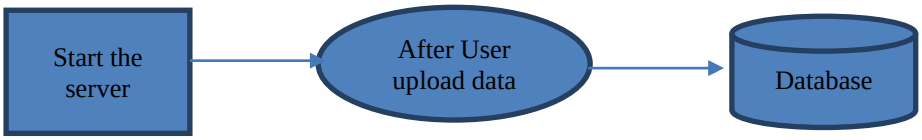


Figure 4: Server Start-up

3. SECURE DE DUPLICATION SYSTEM

Secure hashing algorithm And Minimum Hashing Algorithms has been used to solve the duplication functions. To support authorized access a secret key KP will be bounded with a privilege p to generate a file Token. De duplication exploits identical content, while encryption attempts to make all content appear random; the same content encrypted with two different keys results in very different cipher text. Thus, combining the space efficiency of de duplication with the secrecy aspects of encryption is problematic.

ISSN: 2233-7857IJFGNC

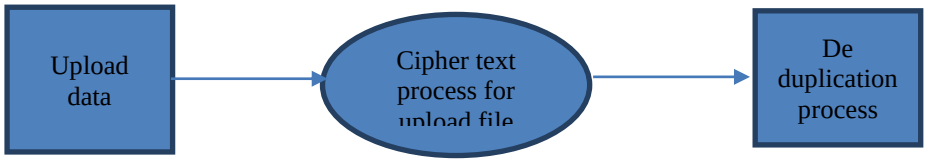


Figure 5: Secure Deduplication

4. DOWNLOAD FILE

After the cloud storage, the user can download the file based on key or token. Once the key request was received, the sender can send the key or he can decline it. With this key and request id which was generated at the time of sending key request the receiver can decrypt the message.

Figure 6: Download File

ALGORITHMS

MINIMUM HASHING

```
S ← S0;                 // Initialize the state.
for k in 1, 2, ..., m do   // Scan the input data units:
S ← F(S, b[k]);         // Combine data unit k into the state
return G(S, n)          // Extract the hash value from the state.
```

Let h be a hash function that maps the members of A and B to distinct integers, and for any set S define hmin(S) to be the member x of S with the minimum value of h(x). Then hmin(A) = hmin(B) exactly when the minimum hash value of the union A ∪ B lies in the intersection A ∩ B. Therefore,

Pr[hmin(A) = hmin(B)] = J(A,B).

SECURE HASHING

Secure Hashing Algorithms, also known as SHA, are a family of cryptographic functions designed to keep data secured. It works by transforming the data using a hash function: an algorithm that consists of bitwise operations, modular additions, and compression functions. The hash function then produces a fixed size string that looks nothing like the original. These algorithms are designed to be one-way functions, meaning that once they're transformed into their respective hash values, it's virtually impossible to transform them back into the original data. A few algorithms of interest are SHA-1, SHA-2, and SHA-5, each of which was successively designed with increasingly stronger encryption in response to hacker attacks. SHA-0, for instance, is now obsolete due to the widely exposed vulnerabilities.

ISSN: 2233-7857IJFGNC

A common application of SHA is to encrypting passwords, as the server side only needs to keep track of specific user's hash value, rather than the actual password. This is helpful in case an attacker hacks the database, as they will only find the hashed functions and not the actual passwords, so if they were to input the hashed value as a password, the hash function will convert it into another string and subsequently deny access. Additionally, SHA exhibit the avalanche effect, where the modification of very few letters being encrypted cause a big change in output; or conversely, drastically different strings produce similar hash values. This effect causes hash values to not give any information regarding the input string, such as its original length. In addition, SHAs are also used to detect the tampering of data by attackers, where if a text file is slightly changed and barely noticeable, the modified file's hash value will be different than the original file's hash value, and the tampering will be rather noticeable.

Initialize hash value for this chunk:

Initialize hash value for this chunk:

```
a = h0
b = h1
c = h2
d = h3
e = h4
Main loop:
for i from 0 to 79
    if 0 ≤ i ≤ 19 then
        f = (b and c) or ((not b) and d)
        k = 0x5A827999
    else if 20 ≤ i ≤ 39
        f = b xor c xor d
        k = 0x6ED9EBA1
    else if 40 ≤ i ≤ 59
        f = (b and c) or (b and d) or (c and d)
        k = 0x8F1BBCDC
    else if 60 ≤ i ≤ 79
        f = b xor c xor d
        k = 0xCA62C1D6
    temp = (a leftrotate 5) + f + e + k + w[i]
    e = d
    d = c
    c = b leftrotate 30
    b = a
    a = temp
Add this chunk's hash to result so far:
h0 = h0 + a
h1 = h1 + b
h2 = h2 + c
h3 = h3 + d
h4 = h4 + e
```

ISSN: 2233-7857IJFGNC

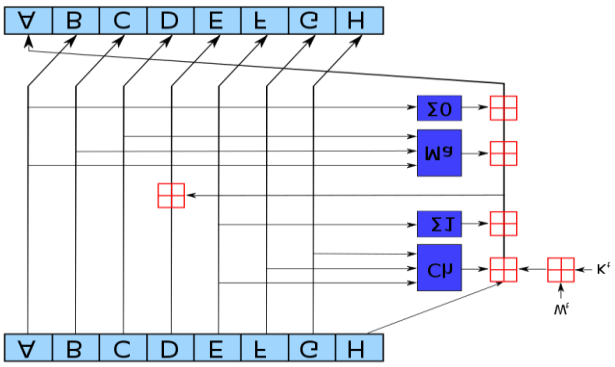


Figure 7: Secure Hashing

ADVANCED ENCRYPTION STANDARD

Rijndael is a family of block ciphers developed by Belgian cryptographers Vincent Rijmen and Joen Daemen. It was submitted as an entry to the National Institute of Standards and Technology's (NIST) competition to select an Advanced Encryption Standard (AES) to replace Data Encryption Standard (DES). In 2001, Rijndael won the competition and the 128, 192, and 256-bit versions of Rijndael were officially selected as the Advanced Encryption Standard.

The three variants of AES are based on different key sizes (128, 192, and 256 bits). In this article, we will focus on the 128-bit version of the AES key schedule, which provides sufficient background to understand the 192 and 256 bit variants as well. At the end, we'll include a note the other variants, and how they differ from the 128-bit version.

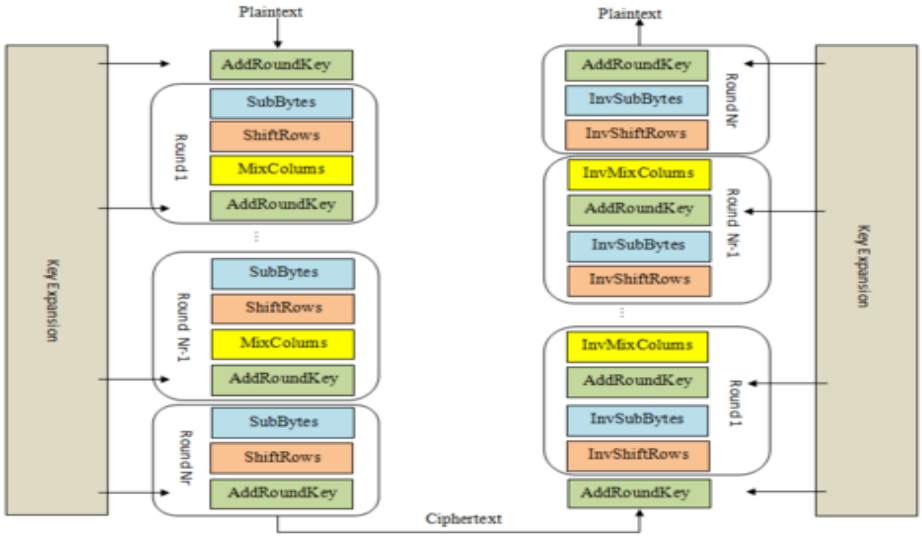


Figure 8: Overall structure of encryption and decryption in Advanced Encryption Standard

ENCRYPTION WITH AES

The encryption phase of AES can be broken into three phases: the initial round, the main rounds, and the final round. All of the phases use the same sub-operations in different combinations as follows:

- Initial Round
- AddRoundKey

- Main Rounds
- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey
- Final Round
- SubBytes
- ShiftRows
- AddRoundKey

DECRYPTION WITH AES

To decrypt an AES-encrypted cipher text, it is necessary to undo each stage of the encryption operation in the reverse order in which they were applied. The three stage of decryption are as follows:

- Inverse Final Round
- AddRoundKey
- ShiftRows
- SubBytes
- Inverse Main Round
- AddRoundKey
- MixColumns
- ShiftRows
- SubBytes
- Inverse Initial Round
- AddRoundKey

ADDROUNDKEY

The AddRoundKey operation is the only phase of AES encryption that directly operates on the AES round key. In this operation, the input to the round is exclusive-ored with the round key.

SUBBYTES

The SubBytes phase of AES involves splitting the input into bytes and passing each through a Substitution Box or S-Box. Unlike DES, AES uses the same S-Box for all bytes. The AES S-Box implements inverse multiplication in Galois Field 28.

To read this Table, the byte input is broken into two 4-bit halves. The first half determines the row and the second half determines the column. For example, the S-Box transformation of 35 or 0x23 can be found in the cell at the intersection of the row labeled 20 and the column labeled 03. Therefore decimal 35 become 0x26 or decimal 38. The AES S-Box is shown in the Table below;

ISSN: 2233-7857IJFGNC

	0	1	2	3	4	5	6	7	8	9	0a	0b	0c	0d	0e	0f
0	63	7c	77	7b	f2	6b	6f	c5	30	1	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	4	c7	23	c3	18	96	5	9a	7	12	80	e2	eb	27	b2	75
40	9	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	0	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	2	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	6	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	8
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	3	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure 9: SubBytes

SHIFTROWS

In the ShiftRows phase of AES, each row of the 128-bit internal state of the cipher is shifted. The rows in this stage refer to the standard representation of the internal state in AES, which is a 4x4 matrix where each cell contains a byte. Bytes of the internal state are placed in the matrix across rows from left to right and down columns.

In the ShiftRows operation, each of these rows is shifted to the left by a set amount: their row number starting with zero. The top row is not shifted at all, the next row is shifted by one and so on. This is illustrated in the Figure below

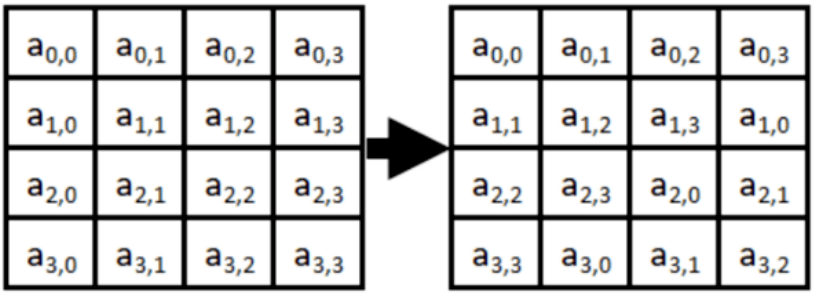


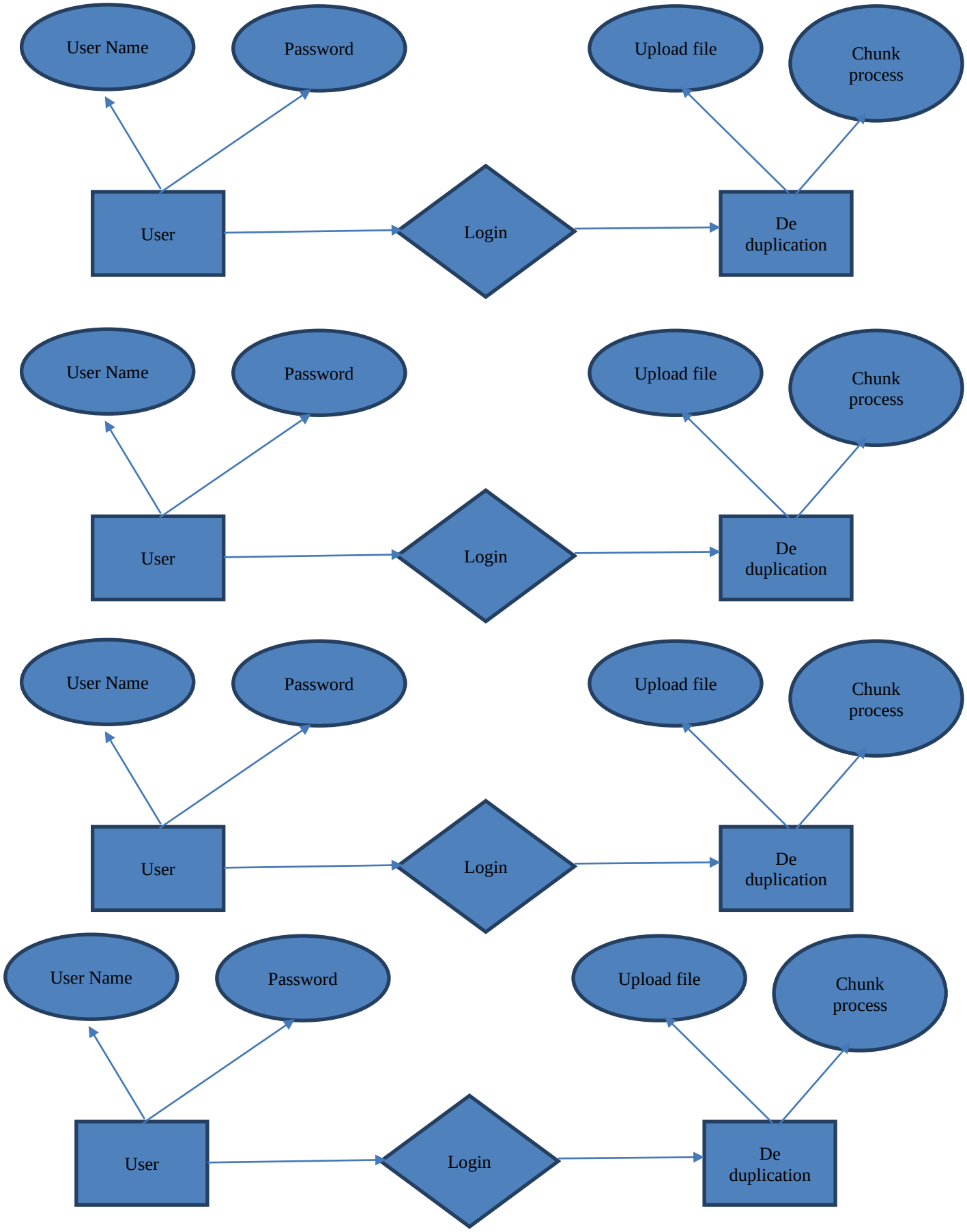
Figure 10: ShiftRows

In the Figure, the first number in each cell refers to the row number and the second refers to the column. The topmost row (row 0) does not shift at all, row 1 shifts left by one, and so on.

MIXCOLUMNS

Like the ShiftRows phase of AES, the MixColumns phase provides diffusion by mixing the input around. Unlike ShiftRows, MixColumns performs operations splitting the matrix by columns instead of rows

ISSN: 2233-7857IJFGNC



ISSN: 2233-7857IJFGNC

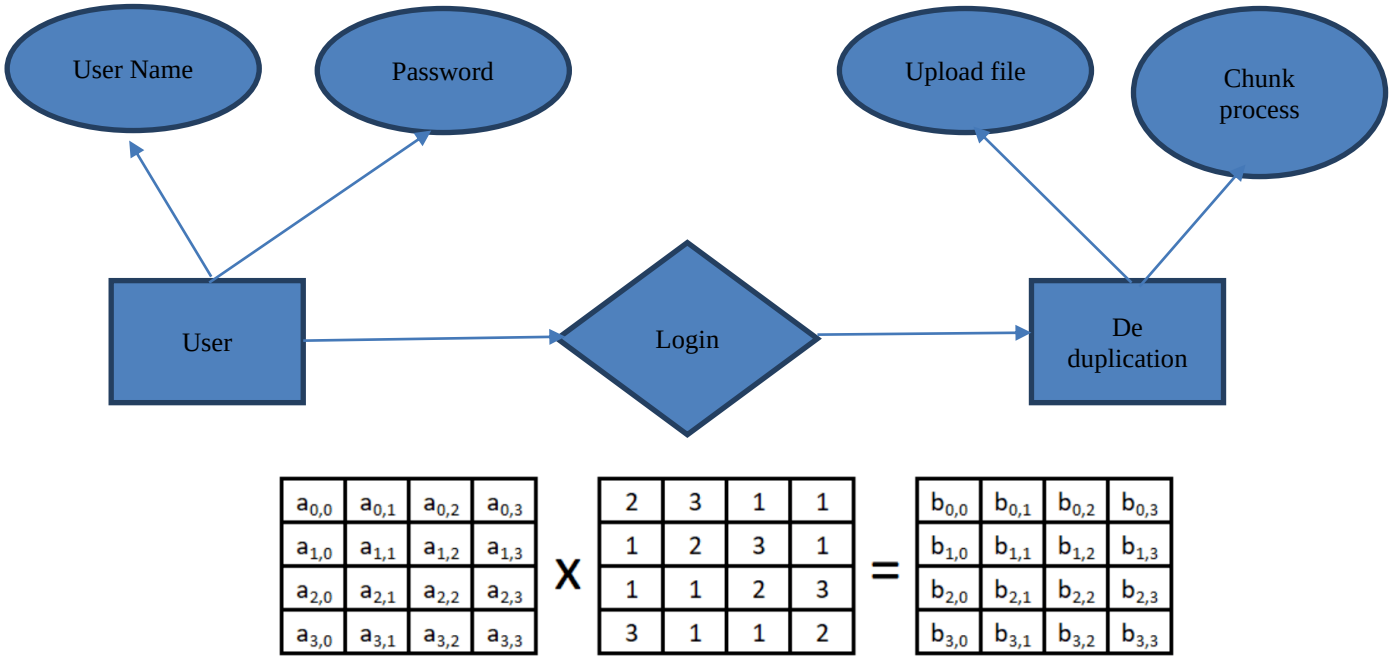


Figure 11: MixColumns

A visual representation of the MixColumns operation is shown above. Unlike standard matrix multiplication, MixColumns performs matrix multiplication as per Galois Field 28. Although we won't describe this step in detail, it is important to note that this multiplication has the property of operating independently over each of the columns of the initial matrix, i.e. the first column when multiplied by the matrix produces the first column of the resultant matrix.

Result:

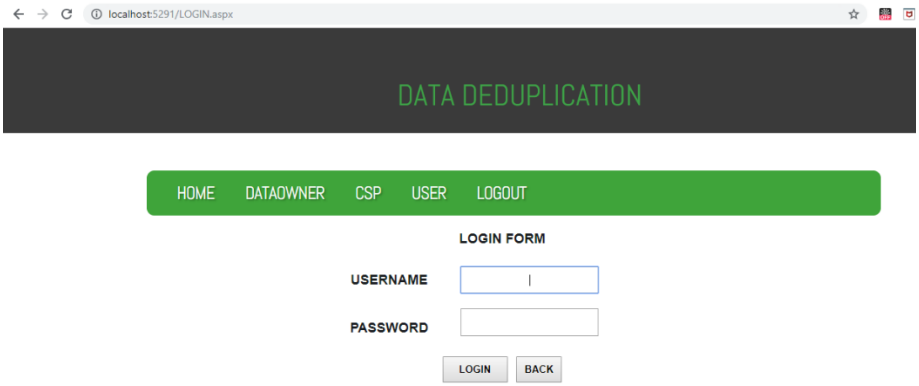


Figure 12: Login Form

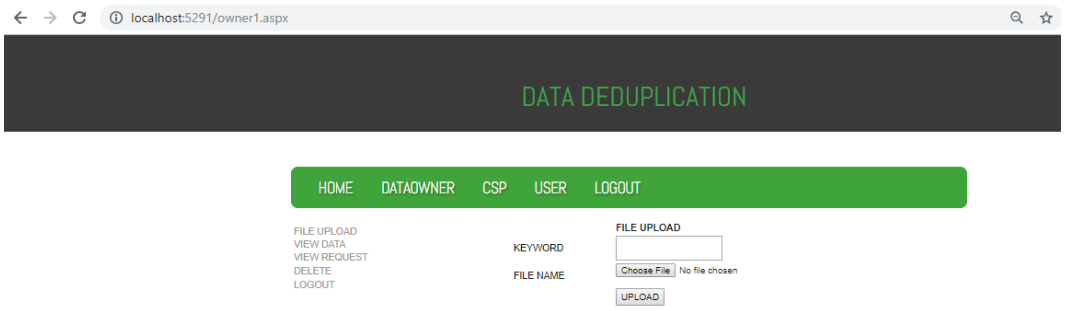


Figure 13: Upload File

CONCLUSION

In this paper, the notion of authorized data de duplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new de duplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct test bed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

REFERENCES

[1] Anderson P and L. Zhang. (2017) 'Fast and secure laptop backups with encrypted de-duplication.' In Proc. of USENIX LISA.

[2] Bellare M, Keelveedhi S, and Ristenpart T. Dupless: (2013) 'Serveraided encryption for deduplicated storage'. In USENIX Security Symposium.

[3] Bellare M, Keelveedhi S, and Ristenpart T.(2013) 'Message-locked encryption and secure deduplication.' In EUROCRYPT, pages 296–312.

[4] Bellare M, S. Keelveedhi, and T. Ristenpart. Dupless (2015) 'Serveraided encryption for deduplicated storage'. In USENIX Security Symposium.

[5] Bugiel S, Nurnberger S, Sadeghi A, and Schneider T. (2015) 'Twin clouds: An architecture for secure cloud computing.' In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.

[6] Bugiel S, S. Nurnberger, A. Sadeghi, and T. Schneider. (2017) 'Twin clouds: An architecture for secure cloud computing'. In Workshop on Cryptography and Security in Clouds (WCSC 2011)

[7] C. Ng and P. Lee. Revdedup: (2016) 'A reverse deduplication storage system optimized for reads to latest backups'. In Proc. of APSYS, Apr.

[8] Douceur J R, A. Adya, Bolosky W J, Simon D, and M. Theimer. (2017) 'Reclaiming space from duplicate files in a serverless distributed file system'. In ICDCS, pages 617– 624.

[9] Ferraiolo D and R. Kuhn (1992) 'Role-based access controls'. In 15th NIST-NCSC National Computer Security Conf.

[10] Halevi S, Harnik D, Pinkas B, and ShulmanPeleg (2011) ’A. Proofs of ownership in remote storage systems’. In Y. Chen, G. Danezis, and V. Shmatikov, editors, ACM Conference on Computer and Communications Security, pages 491–500.
[11] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou (2013). ‘Secure deduplication with efficient and reliable convergent key management. In IEEE Transactions on Parallel and Distributed Systems’.
[12] Namprempre C and G. Neven (2014) ‘Security proofs for identity-based identification and signature schemes’. J. Cryptology, 22(1):1–61.
[13] Pietro R D and A. Sorniotti (2012) ‘Boosting efficiency and security in proof of ownership for deduplication’. In H. Y. Youm and Y. Won, editors, ACM Symposium on Information, Computer and Communications Security, pages 81–82. ACM.
[14] Quinlan S and Dorward S. Venti (2011) ‘A secure cloud backup system with assured deletion and version control’. In 3rd International Workshop on Security in Cloud Computing.

ISSN: 2233-7857IJFGNC