INTEGRITY AUDITING BASED ON THE KEYWORD WITH SENSITIVE INFORMATION PRIVACY FOR ENCRYPTED CLOUD DATA

Guided by:

Dr. S.Milton Ganesh, M.E., Ph.D.,
Assistant Professor & HoD, of IT.

Presented by:

1. Dhilip Kumar D (422419205008)
2. Dhina P (422419205009)
3. Yuvaraj S (422419205043)

LITERATURE SURVEY

S.No.	Title & Author	Journal & Year	Problem	Solution	Parameters Measured	Advantages	Disadvantages
1	**Title :** Checking Only When It Is Necessary: Enabling Integrity Auditing Based on the Keyword With Sensitive Information Privacy for Encrypted Cloud Data **Authors :** Xiang Gao , Jia Yu ,Yan Chang, Huaqun Wang , and Jianxi Fan	IEEE transactions on dependable and secure computing, vol. 19, no. 6, Nov/Dec 2022	To check the integrity auditing of an encrypted file in a cloud.	RAL is a label that authenticates file relationships with queried keywords and generates auditing proof while preserving file identity privacy.	Reduce overall time complexity,computation time required for auditing.	Give security analysis that satisfies correctness, auditing soundness and sensitive information privacy.	It can only check the file that contain secure index keywords.
2	**Title :** Identity-Based Cloud Storage Auditing for Data Sharing With Sensitive Information **Authors :** Yang Yang, Yanjiao Chen, Fei Chen, Jing Chen	IEEE Internet of Things Journal (Volume: 9, Issue: 13, 01 July 2022)	The sensitive information should not be exposed to others when the cloud file is shared.	The user hide the data blocks corresponding to original file and the signatures, and then sends them to a receiver.	The proposed scheme is evaluated for its efficiency in terms of computational overhead, communication overhead, and storage overhead.	This method not only realizes the remote data integrity auditing, but also supports the data sharing on the condition that sensitive information is protected in cloud storage.	The data owner can decrypt this file. However, it will make the whole shared file unable to be used by others.
3	**Title :** Data Integrity Auditing without Private Key Storage for Secure Cloud Storage **Authors :** Wenting Shen; Jing Qin, Jia Yu, Rong Hao, Jiankun Hu, Jixin Ma	IEEE Transactions on Cloud Computing (Volume: 9, Issue: 4, 01 Oct.-Dec. 2021)	The user has to keep hardware token to store his private key and memorize a password to activate this private key. If it lost the current data integrity auditing would be unable to work.	We use biometric data (e.g. iris scan, fingerprint) as private key to avoid using the hardware token.	The result of integrity auditing is more accurate.	Adds reasonable overhead to realize data integrity auditing without private key storage compared with the existing schemes.	The user might need to remember multiple passwords for different secure applications

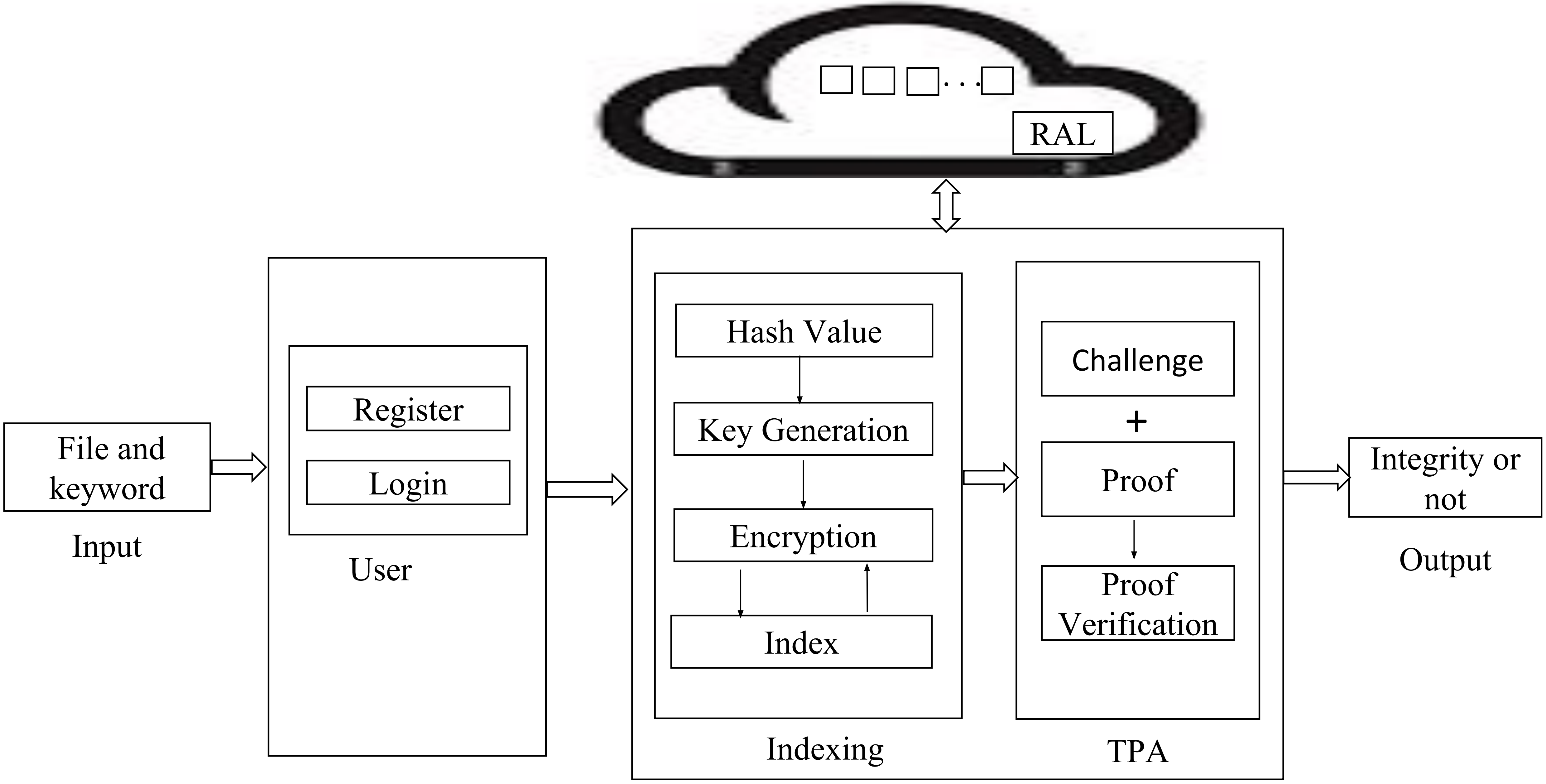
PROBLEM STATEMENT

Who does the problem affect?	Cloud users, Cloud service providers, Cloud auditors.
What are the boundaries of the problem?	Data sensitivity, Cloud storage, Auditing, Information privacy.
What is the issue?	Integrity of cloud data, Privacy of sensitive information, unauthorized access
When does the issue occur?	The issue may occur during the auditing of cloud data for integrity, as auditors may require access to sensitive information.
Where does the issue occur?	The issue may occur when the data is initially uploaded to the cloud.
Why is it important that we fix the problem?	Protecting the privacy of sensitive information is essential for maintaining data security, compliance, trust, and cost savings.
What solution to solve this issue?	A combination of encryption, authentication, access controls, monitoring, and audits can help to ensure the integrity of cloud data.
What methodology used to solve the issue?	Encryption, Keyword-based auditing, Relation Authentication Label, Auditing Proof Generation.

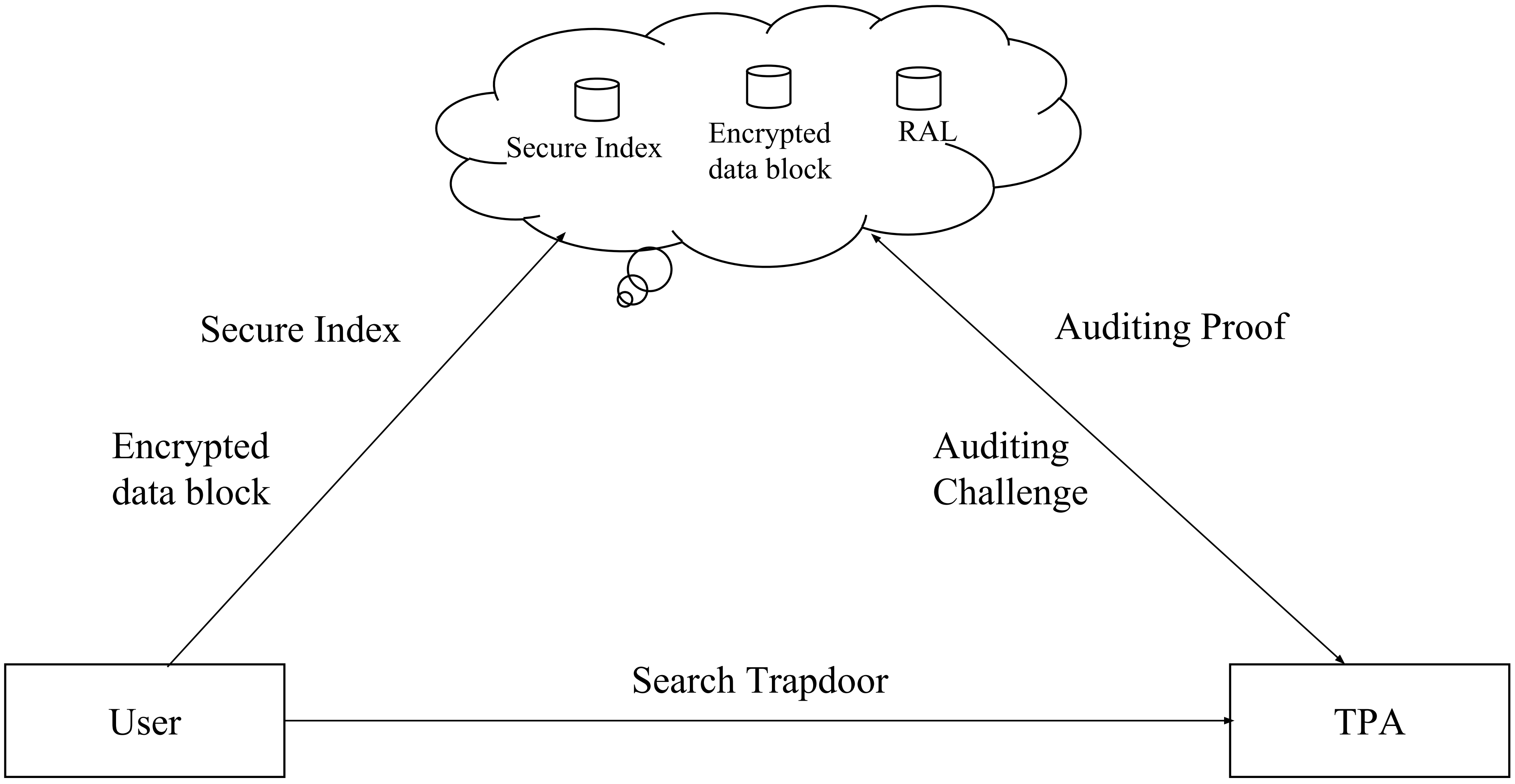
PROPOSED SOLUTION

Parameter	Description
Problem statement	To ensure the integrity of encrypted cloud data
Idea/Solution description	Solution is to check Integrity based on the queried keyword using RAL(Relational Authentication Label).
Novelty/Uniqueness	The uniqueness is keyword-based auditing approach.
Customer satisfaction	Ensures the privacy and security of sensitive information. This can increase customer satisfaction and trust.
Business model (Revenue model)	The company could charge a fee for each auditing request from the organizations.
Scalability of the solution	It should be able to handle large datasets and a high volume of auditing requests without a significant increase in computational overhead.

SYSTEM ARCHITECTURE



SYSTEM MODEL



MODULE SPLITS-UP

REGISTRAION

Input: Files and keyword

Output: Encrypted file block and keyword set

Algorithm:

1. The user register and creates account.
2. Then login to the account.
3. Generate the secret key x and public key y.
4. The file is splits into s blocks and then the blocks are encrypted by symmetric encryption.
5. Build the keyword set w_k from the files.

INDEXING

- **Input:** Encrypted file block and keyword set

  Output: Secure index, Authenticator, Trapdoor

  Algorithm:

  1. Generate three secure Hash functions H1: {0,1}*→G_1, H_2 {0,1}*→G_1, H_3 {0,1}*→G_1
  2. Computes $\pi(w_k)$ as the address of each row in the secure index.
  3. Encrypts the index vector. $ev_\pi(w_k) = vw_k \oplus f\left(\pi(w_k)\right)$.
  4. Computes the RAL $\Omega_{\pi(w_k)=}\left\{\Omega_{w_{k,1}}, \Omega_{w_{k,2}}, \dots, \Omega_{w_{k,s}}\right\}$.
  5. Computes the Secure index $I = \left\{\pi(w_k), ev_{\pi(w_k)}, \Omega_{\pi(w_k)}\right\} k = 1,2, \dots., m$.
  6. Computes the authenticator $\sigma_{ij} = \left[H_1(ID_i||j) . u^{eij}\right]^x$.
  7. Computes the search trapdoor as $T_{w'} = \left\{\pi(w'), f\left(\pi(w')\right)\right\}$.

CLOUD

Input: Challenge, Secure index, Encrypted data blocks and Authenticators

Output: Proof

Algorithm:

1. It stores Auditing Challenge, Secure index I, encrypted data blocks and Authenticators in cloud.

$$Chal = \left\{T_{w'}, \{j, v_j\}_{j\in Q}\right\} \quad T_{w'=\{\pi(w'), f(\pi(w'))\}} \quad v_{wk} = ev_{\pi(w_k)} \oplus f\left(\pi(w_k)\right)$$

2. Using RAL, The cloud generate Auditing Proof and it sends to TPA.

$$T = \prod_{i\in S_{w_K}} \prod_{i\in Q} \sigma_{ij}^{vj} . \prod_{j\in Q} \Omega_{w_k,} j^{vj}, \mu = \Sigma_{i\in S_{w_k}} \Sigma_{j\epsilon Q} C_{ij} . v_{j.}$$

TPA(Third Party Auditor)

Input: Trapdoor and proof

Output: Challenge and proof verification

Algorithm:

1. It takes Trapdoor as input.
2. And it generate the Auditing Challenge then TPA sends challenge to Cloud.

$$Chal = \left\{T_{w''}, \{j, v_j\}_{j \in Q}\right\}.$$

3. TPA receives the Auditing proof from the cloud.
4. TPA verifies proof using Auditing Challenge sent by TPA and Proof generated by the cloud.

$$e(T, g) \stackrel{?}{=} e\left(\left(\prod_{j \in Q}\left(H_3(j).\ H_2(\pi(w')||j)\right)^{v_j}\right).u^{\mu}, y\right)$$

if Challenge == Proof

The file is intact

else

The file is corrupted.

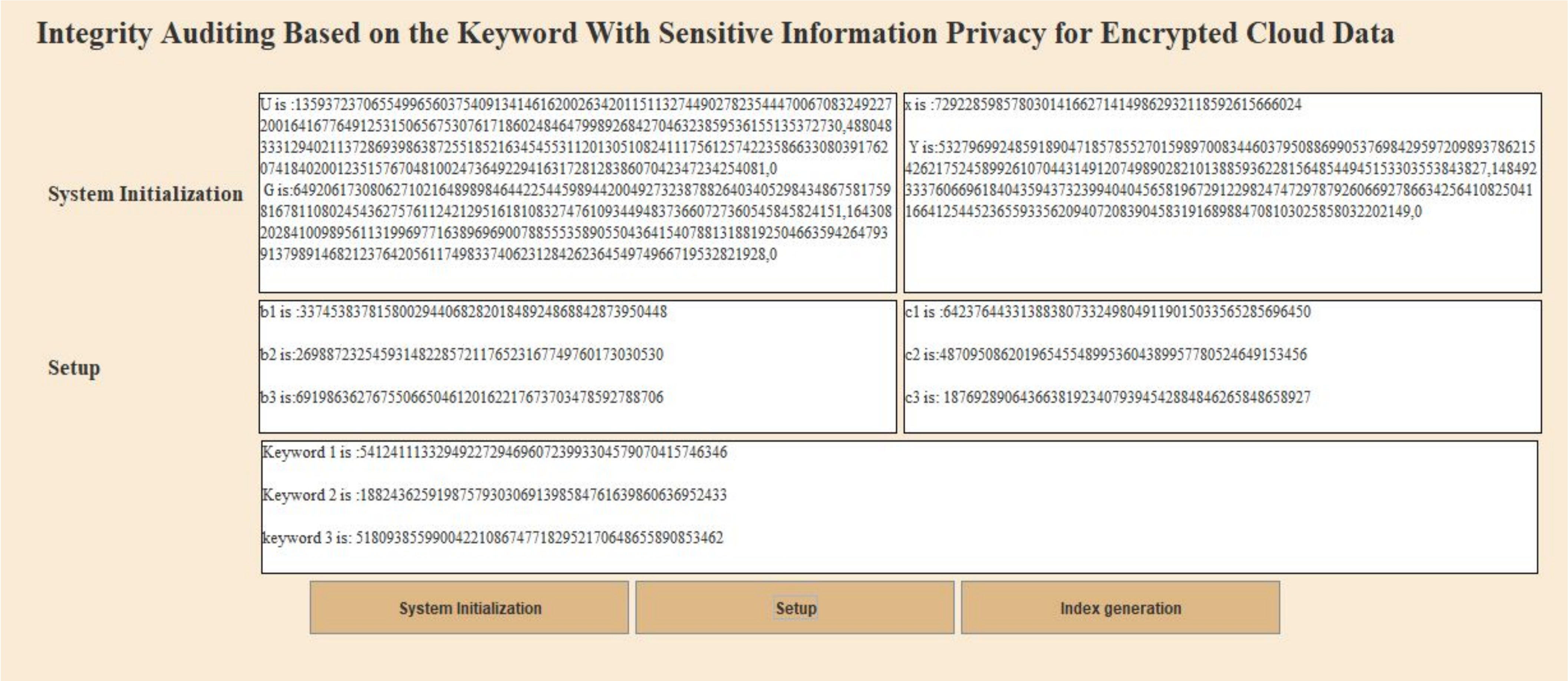
CODE LINK:

https://github.com/Dhina8801/Integrity--Auditing.git

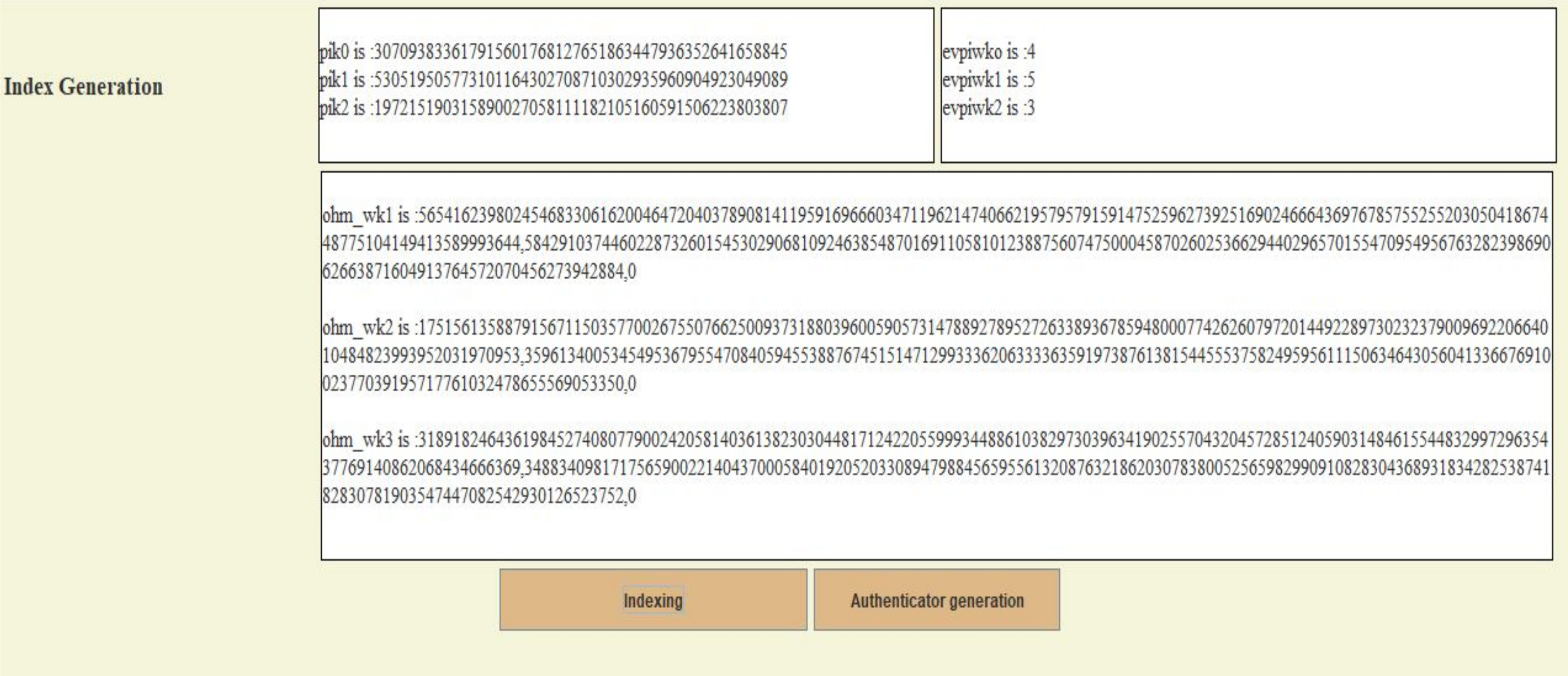
REGISTER


Register
Name:
ucet
Email:
ucet@gmail.com
Password:
Confirm Password:
Register
Clear
back

SYSTEM INITIALIZATION



INDEX GENERATION OUTPUT



AUTHENTICATOR AND TRAPDOOR OUTPUT

Authenticator

sigma_file1_block1 is :3011139061153026875403478770077408266267842849542051051304124263704340552049313352111456265839894046252211987814993026941444112626781809113283016724940084,2662274945582620951738710096289731000640974197449217866330455708883495405006461247661981967268827809868419911323170860419205917084746322887716248036973452,0

sigma_file1_block2 is :589040784921649017746607479461667232051589695133144373202195867851431442391030846026443936189087703405057508912877581082490153555641904906423002524886046,2094045951689692234641719426629945072874550805813591737309331300559194149753940631383247781868789522695758301464080658675202369670499421437548217576745141,0

sigma_file1_block3 is :4969521037402841377432265589742556596734641937589605250736415137274274785408989718107323169000458409288066931215913083189144456133897514698982004668688569,5382896945143842281598397991920931368628047312056363368999590138835074528299411375323849906941561501889156108041792918054902218607841638343975248805102729,0

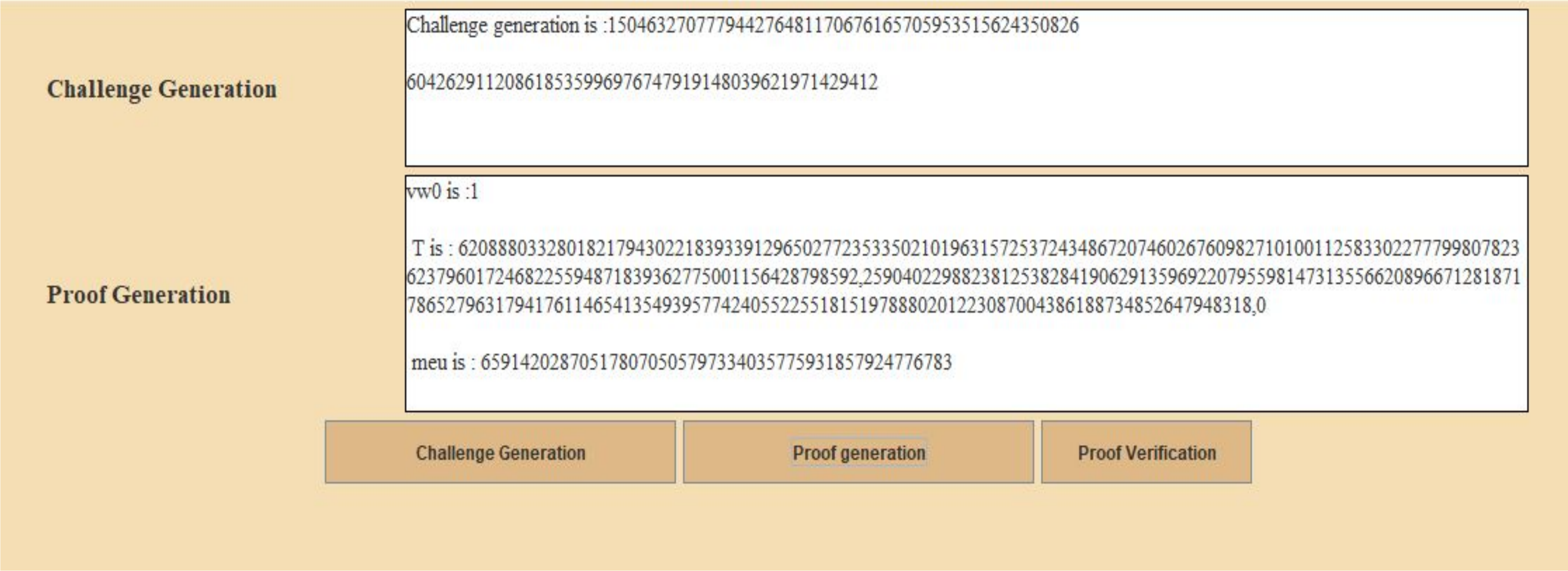
Trapdoor

Trapdoors are :
wpik0 is :307093833617915601768127651863447936352641658845

fpik0 is :55775095427622207636301035504698428519545230434

Authenticator | Trapdoor | Challenge generation

CHALLENGE GENERATION AND PROOF GENERATION OUTPUT



PROOF VERIFICATION OUTPUT

Proof Verification

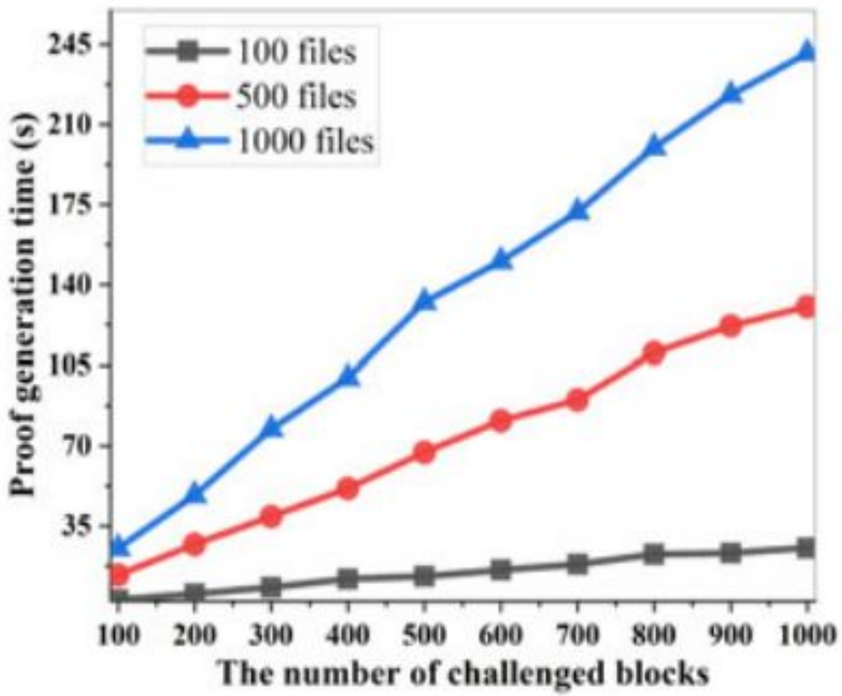
LHS is :{x=1158345588646789296702401035736027869730811201015807567570124
29507988314778931712162966135092375332719241693578794661318903306093957
677128486089107561078,y=79412477520504842064178972301776725105200635035
47791632454178240269445899097156057181413499355693360556517153593000832
655405598447148311526936275107746495}

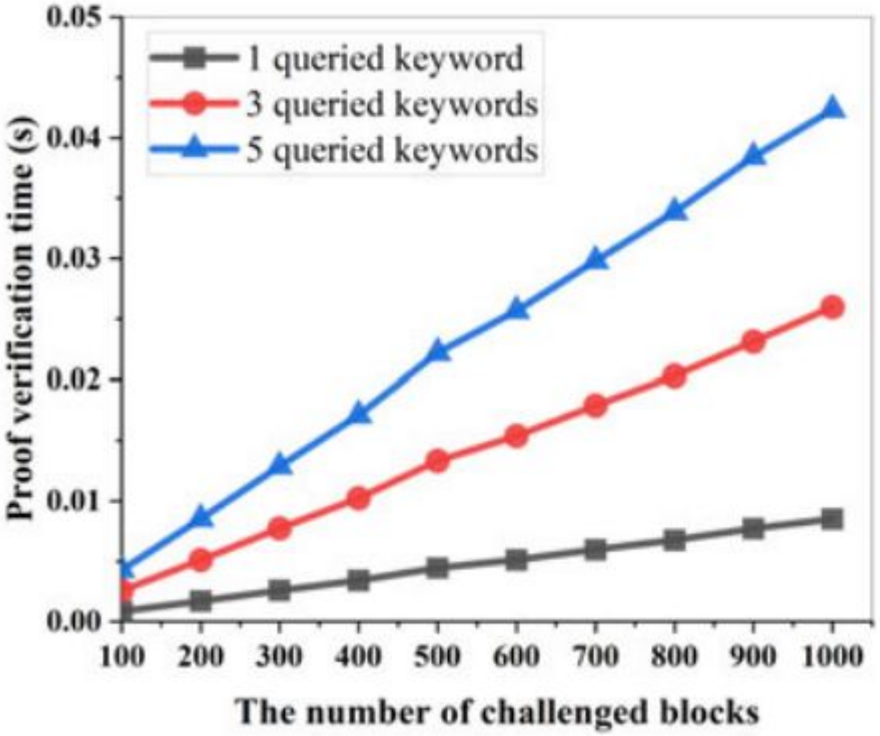
RHS is :{x=115834558864678929670240103573602786973081120101580756757012
42950798831477893171216296613509237533271924169357879466131890330609395
7677128486089107561078,y=7941247752050484206417897230177672510520063503
54779163245417824026944589909715605718141349935569336055651715359300083
2655405598447148311526936275107746495}

LHS

RHS

PERFORMANCE EVALUATION





CONCLUSION

Cloud data integrity auditing techniques are important for ensuring the security of outsourced data, but auditing all files in the cloud can be costly and inefficient. Third Party Auditor (TPA) to audit the integrity of all encrypted cloud files containing a specific keyword, without knowing which files contain the keyword or how many files contain it. The scheme uses a newly proposed Relation Authentication Label (RAL) to authenticate the relation that files contain the queried keyword and generate the auditing proof without exposing sensitive information.

FUTURE WORK

- Support multiple keyword queries or SQL-like queries without exposing sensitive information
- Investigate ways to reduce the computation and storage overhead of the RAL in the proposed scheme.
- Design new schemes that can protect forward and backward privacy in the context of data dynamics

REFERENCES

1. G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu, and R. Hao, ‘Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability’, J. Syst. Softw., vol. 113, pp. 130–139, 2016.
2. R. Bost, P.-A. Fouque, and D. Pointcheval, ‘Verifiable dynamic symmetric searchable encryption: Optimality and forward security’, IACR, Lyon, France, Rep. 2016/062, 2016.
3. R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, ‘Searchable symmetric encryption: Improved definitions and efficient constructions’, J. Comput. Secur., vol. 19, no. 5, pp. 895–934, 2011.
4. X. Ge, J. Yu, C. Hu, H. Zhang, and R. Hao, ‘Enabling efficient verifiable fuzzy keyword search over encrypted data in cloud computing’, IEEE Access, vol. 6, pp. 45725–45739, 2018.
5. X. Zhu, Q. Liu, and G. Wang, ‘A novel verifiable and dynamic fuzzy keyword search scheme over encrypted data in cloud computing’, in Proc. IEEE Trustcom/BigDataSE/ISPA, 2016, pp. 845–851.
6. Y. Yu, J. Ni, M. H. Au, Y. Mu, B. Wang, and H. Li, ‘Comments on a public auditing mechanism for shared cloud data service’, IEEE Trans. Serv. Comput., vol. 8, no. 6, pp. 998–999, Nov./Dec. 2015.

THANK YOU