International Journal of Technology (2015) 3: 327-335
ISSN 2086-9614

DATA SECURITY MODEL EMPLOYING HYPERELLIPTIC CURVE CRYPTOGRAPHY (HECC) AND SECURE HASH ALGORITHM-3 (SHA-3) IN CLOUD COMPUTING

Devi Thiyagarajan[1*], Ganesan R.[1]

[1] *School of Computing Science and Engineering, VIT University, Chennai Campus Vandalur - Kelambakkam Road, Chennai - 600127, India*

(Received: May 2015 / Revised: June 2015 / Accepted: July 2015)

ABSTRACT

Data owners use the huge space offered by 'Cloud' Computing for storage of data and also for carrying out computations. To eliminate the burden of storing file locally, cloud stores them on remote servers using virtualization concepts. Therein arises one of the major issues in the field of cloud computing: security. Data owners lack in having direct control over files stored in the cloud and consequently, the problem of data security arises. An efficient scheme to provide data security, while storing data in the cloud has been proposed which makes use of Hyperelliptic curve cryptography (HECC) for encryption and decryption and Secure Hash Algorithm-3 (SHA-3) for data integrity verification. Implementation results clearly illustrate that HECC remains as a good alternative asymmetric key technique rather than ECC and RSA when securing documents in cloud.

Keywords: Cloud; Curve; Cryptography; Encryption; Hyperelliptic; Integrity; Security

1. INTRODUCTION

Cloud computing offers resources as services to customers based on their demand. A 'cloud' environment is similar to a datacenter with hardware and software resources used over Internet to satisfy users' needs. The National Institute of Standards and Technology (NIST) defines that, "cloud computing enables ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Essential characteristics of the cloud which distinguishes it from traditional hosting are on-demand self-service, broadband network access, resource pooling, rapid elasticity and measured service. Three service delivery models in cloud computing are Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS). In SaaS, applications are hosted by cloud service providers and are offered to customers over the Internet. PaaS provides the platform and environment for developers to build applications and services over Internet. So, users create software applications with the tools provided by the service provider. In IaaS, consumers can provision processing, storage, network and other computing resources wherein the consumer can deploy and run arbitrary software, including operating systems and applications. The deployment models in cloud computing are public, private, hybrid and community clouds.

* Corresponding author's email: devi.janu@gmail.com, Tel. +98-41-560895, Fax. +91-44-39931555
Permalink/DOI: http://dx.doi.org/10.14716/ijtech.v6i3.1117

Another form of distributed computing is grid computing that includes co-ordination and sharing computing, application, data and storage or network resources across dynamic and geographically dispersed organization (Syed & Amid, 2012; Berman et al., 2003). Some of the significant grid features are geographical distribution, heterogeneity, multiple administrations and so on. Still grid computing differs from cloud computing in many aspects as shown in Table 1.

Table 1 Comparison of Grid and Cloud

Parameters	Grid Computing	Cloud Computing
Goal	Collaborative sharing of resources	Usage of Services
Abstraction level	Low	High
Accessibility of Portal	Via DNS system	Using IP only
Operating system	Any standard OS	Hypervisor supports running of multiple operating systems
User management	Decentralized	Centralized
Interoperability	Open grid forum standards	Web services (SOAP and REST)

Cloud stores outsource client data in untrusted remote servers. So, the client does not have any physical control on data stored. Security is one of the foremost issues faced in the adoption of cloud computing (Xiao & Xiao, 2013; Mather et al., 2009) by users as shown in Figure 1. Security is a combination of confidentiality, prevention of the unauthorized disclosure of information, integrity, and the prevention of the unauthorized amendment or deletion of information, and availability, including the prevention of unauthorized withholding of information (Aviˇzienis et al., 2004). Various surveys conducted also portray the same issues (Subashini & Kavitha, 2011; Modi et al., 2013; Gonzalez et al., 2012). In particular, the IT industry faces data security problems (Sun et al., 2014), as data is being scattered across different machines and storage devices like servers and mobile devices. Before adoption of cloud computing, security risks have to be assessed (Latif et al., 2014).

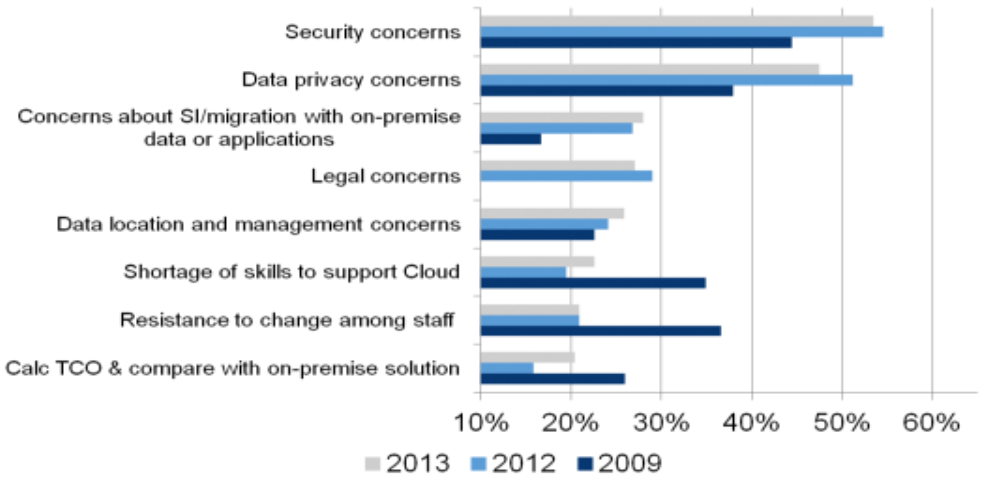

Figure 1 IDC Survey 2013

The paper proposes a new approach to secure data throughout its lifecycle in the cloud. Client data can be secured when stored in the cloud with the help of cryptographic techniques. Encryption is done on the client side to prevent certain problems, such as data loss and theft of keys by employing Hyperelliptic Curve Cryptography (HECC), (Koblitz, 1989). There is a need for verifying the integrity of stored data, as it is stored in remote servers, which may delete or modify those contents. Third-party auditing is not trustworthy at all times. Renting of auditing services can be prevented by employing the Secure Hash Algorithm 3 (SHA-3) (Fakhri et al., 2013).

The contributions of the paper are as follows:

1. Implementation of document encryption is done on the data owner side by using Hyperelliptic Curve Cryptography (HECC) using Spring Tool Suite and Java (jdk 8).
2. Data users authenticated by the data owner can use cloud services with the help of an Amazon server.
3. Data integrity is verified by users after the retrieval of desired documents by employing the Secure Hash Algorithm 3 (SHA-3).
4. The security analysis of the proposed model indicates that it can resist the attacks from users and hackers also.

The paper organization is as follows: Section 2 deals with the related work. Section 3 deals with security model in the cloud and Section 4 presents the implementation results with discussion. Finally, Section 5 concludes the paper.

2. RELATED WORK

Data integrity verification is focused on the Proof of Retrievability (POR) (Juels et al., 2007) along with spot checking and error-correcting codes. The Random Linear Function with a homomorphic authenticator (Shacham & Waters, 2008) is built which produces low overhead, while providing communication attributes. When the POR model is employed in distributed systems, pre-processing actions are taken by the users before they send data to the cloud. But, the model pays more attention to static data. To verify the accuracy of data stored in the cloud, a new proposed model (Ateniese et al., 2007) utilizes a homomorphic distributed verification scheme with pseudorandom data. This model does not guarantee data security.

A new model was proposed that maintains data integrity and also uses cryptographic techniques (Bowers et al., 2008) for data security. Search over encrypted data is done by asymmetric and symmetric searchable encryption, but certain complexities are faced at times. Various pitfalls which arise during the usage of cryptographic techniques in the cloud is deliberated (Wang et al., 2009) and a new algorithm named Order Preserving Symmetric Encryption (OSPE) was employed for a ranked keyword search (Kamara & Lauter, 2010). The model didn't focus much on security attacks and data integrity verification. CloudProof (Popa et al., 2010) is yet another model which makes use of a scalable framework to detect and prove the malfunctioning of the server.

A hybrid model with various techniques such as data classification, index creation and SSL encryption along with digital signature is employed to provide security of client data (Sood, 2012). Certain pitfalls are still there in the model. Another data security model proposed (Wang et al., 2010) deals mainly with small and medium enterprises and consists of three stages namely encryption/decryption, retrieval of ciphertext and verification of data integrity. Data encryption is carried out with fully a homomorphic symmetric encryption algorithm.

The first searchable encryption construction was presented whereby anyone with a public key can write to the data stored on the server, but only authorized users with private keys can search

(Boneh et al., 2004). Ranked keyword search in a secured manner utilizes keyword frequency to rank results instead of returning undifferentiated results (Wang et al., 2010, 2012). Data search service over encrypted content in the cloud made the work of data retrieval easy (Ning et al., 2014). Multi-keyword based ranked search over encrypted data improved the search experience with more search semantics.

3. DATA SECURITY MODEL IN THE CLOUD

The functioning of a security model is as follows (Figure 2):

1. The data owner encrypts the files before storing in the cloud using Hyperelliptic Curve Cryptography.
2. The hash value is calculated by the data owner using the SHA-3 algorithm and it is also sent along with the encrypted file for storage in the cloud.
3. Users register with the data owner and receive the valid credentials to retrieve files from the cloud and a corresponding decryption key to decrypt the files.
4. The data owner sends the details regarding the registered data users to the cloud for further verification during the cloud access.
5. Users request the cloud for downloading the particular file to be stored in the cloud with the credentials provided by the owner.
6. Upon receiving the file, the hash value is again calculated by the user to identify whether the file has been tampered or not while being stored in cloud.
7. If the file is tampered, then data owner will be notified by the user.

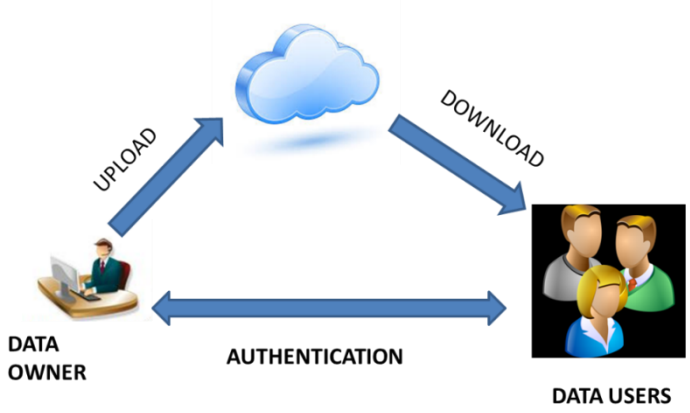

Figure 2 Data Security Model in the Cloud

The phases involved in the model are as follows:

1. *Key Generation*
   Client employs HECC algorithm for generation of public and private keys. The key set is defined as $\{p_k, d\}$ where p_k is public key and d is the private key.
2. *Encryption*
   The data owner encrypts the files before sending to the cloud with a public key p_k - -> E (F,pk) and also calculates the hash value of the file using SHA-3. The hash value is stored for future verification processes and encrypted data is uploaded to the cloud.
3. *Decryption*
   When the user is in need of a file, a download request is sent to the cloud and the retrieved content is decrypted with a decryption key. Upon retrieval, the hash value is again calculated. In comparison, file integrity can be verified. Since the file is stored in

the cloud, the hash value comparison helps to identify whether the file is intact while stored in the cloud.

4. RESULTS AND DISCUSSION

The framework for securing documents in the cloud with the use of HECC and SHA-3 was implemented in Java (jdk 8), Spring Tool Suite 3.6.1 is used as IDE with an Intel I5 processor. An Amazon server is mainly used for storage and testing purposes. HECC works much faster than the other encryption algorithms and on analyzing the results (Table 2); it is clear that HECC can be employed for encryption/decryption of files in cloud computing because of the hardness of discrete logarithm problems.

Table 2 Result Analysis of proposed framework

Public Key	3032301006072a8648ce3d020106052b81040006031e0004ba8186f1e48 3efeeabd1b884c7e9376209ffe57cf34094ec9c05e5c5
Private Key	302c020100301006072a8648ce3d020106052b8104000604153013020101040e47ede2ababe0c625d647e6e16b02
Encrypted Text	À÷~ÎŒ_ó5%¯%Z_ rî A[:Ñ´áçJÖéÝ, <»ßÜÖâw1Ô Áf,_wI_=
HECC Time Taken for encryption	562 msecs
HECC Time Taken for decryption	31 msecs
Remote SHA	13c5671b144c7a1cc11e5d0ce8765e6a0b667c5d5b50a87443d4c68d
Local SHA	13c5671b144c7a1cc11e5d0ce8765e6a0b667c5d5b50a87443d4c68d

From the above results it is clear that, HECC employed for data security in the cloud works efficiently. Moreover, SHA-3 helps in documentation for integrity verification, both locally and remotely. Table 3 shows the comparison of the model with other existing encryption methods and it is understood that the data security model employing HECC can be well utilized for document security when it comes to the cloud.

Table 3 Comparison with other encryption methods

Encryption Technique	Authentication	Data Protection	Smaller key size
Homomorphic encryption (Tao and Xinjun, 2013)	Yes	Partial	No
Incremental encryption (Zhao et al., 2010)	Yes	Partial	No
Broadcast encryption (Liu et al., 2011)	Yes	Complete	No
SSL encryption (Sood, 2012)	Yes	Partial	No
Proposed Framework	Yes	Complete	Yes

Even if the timestamp increases, HECC encryption/decryption time (ms) decreases as shown in the Figure 3, whereas ECC encryption/decryption time (ms) increases. And key generation time in milliseconds also differs based on timestamp. The *x*-axis represents a timestamp interval and the *y*-axis represents overall time taken for performing encryption and decryption of data in the cloud. From Figure 3, it is clear that, HECC incurs reduced time for both encryption and decryption of documents in cloud computing.

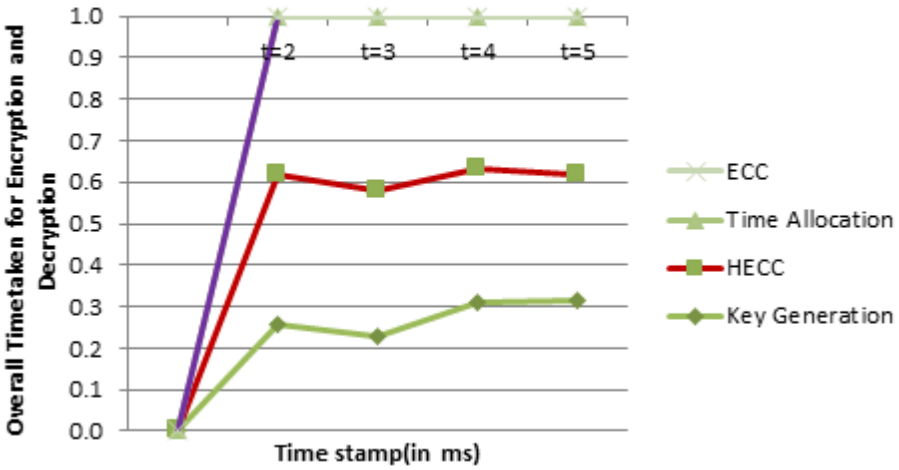

Figure 3 Time Evaluation in HECC and ECC

Performance analysis of HECC and ECC is depicted in the graph (Figure 4). As key size decreases, time taken for encryption/decryption also decreases. The ECC with 160-bit key size takes more time for key generation and encryption/decryption, whereas HECC with 80-bit key size takes less time.

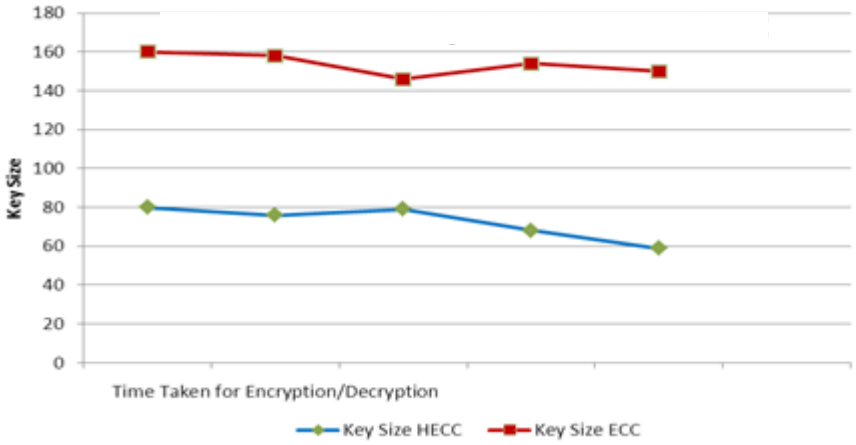

Figure 4 Performance Analysis of HECC and ECC

Figure 4 represents the performance analysis of both ECC and HECC. The *x*-axis represents the size of the key and the *y*-axis represents time taken for HECC and ECC encryption of data in the cloud. It is clear that HECC with a minimal key size offers stronger security.

4.1. Analysis of the Model

The security model employing HECC has the following important features that make it distinct from other models. They are as follows:

1. *Encryption performed on client side:* Mostly servers are untrusted in cloud environments.

Hence, data is encrypted on the client side and uploaded in the cloud for removing the storage burden on the client side.

2. *Integrity verification of data:* Data integrity plays a major role on storage in the cloud. SHA-3 is employed for such a purpose. Local hash value and remote hash value is compared to identify tampering of data.
3. *Confidentiality:* Encrypted data is stored in the cloud which prevents the cloud service provider from knowing about the file contents. Also it is not easy to decipher the encrypted content during transmission.
4. *Attack prevention:* HECC is a proven algorithm as it utilizes a minimal key size, but at the same time offers more security than other cryptographic algorithms. Hence, most of the attacks are prevented by employing HECC.

Only storage space is offered by the cloud provider and they do not perform any operations. It is only the data owner who does encryption and decryption works, thereby reducing the storage requirements of the data owner.

The proposed framework is said to be secure if it satisfies certain properties.

Known-key attack: In this proposed framework, new public and private keys are generated for every new session. Since a discrete logarithm problem is intractable, the model is secure against known-key attacks.

Man-in-the-middle attack: On compromising a public key, values of random prime number k and the divisor D cannot be easily calculated as the security of HECC depends on a discrete logarithm calculation. As the file cannot be decrypted by an attacker, the model resists the 'man-in-the-middle' attack.

Replay attack: An unauthorized attacker sends duplicate data to the receiver frequently or on a delayed basis. A proposed framework protects against replay attacks as each time a new key is generated for encryption.

4.2. Computational Complexity

Usage of HECC for encrypting and decrypting documents by the data owner lowers the computational costs when compared to other data security models. The reason for such lower costs is a minimal key size and also because of the hardness of a discrete logarithmic problem, the algorithm remains strong under various attacks. The proposed framework has a low computational load as it uses 2 point multiplication, 1 encryption/decryption, 1 signature generation/verification and finally 2 SHA-3 operations. Lesser time is spent in many operations which clearly indicate that HECC is best suited for scalable and faster cloud environments.

5. CONCLUSION

HECC is best suited for secure communication in cloud environments as HEC operand size is only a fractional amount of EC operand size. Also a standard discrete logarithm based protocols like Diffie-Hellman and ElGamal can be planted to HEC. The framework proposed in the paper utilizes HECC for encryption/decryption and SHA-3 for data integrity verification. Efficiency of the framework is analyzed by a time comparison for encryption and decryption with that of ECC. HECC of 80-bit operand length provides the same security level with ECC of 160-bit, thereby making it more suitable for cloud environments. Thus, data owners and users can utilize such frameworks for securing their data in the cloud.

6. REFERENCES

Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., Song, D., 2007. Provable Data Possession at Untrusted Stores. *In*: *Proceedings of the 14th ACM Conference on Computer and Communications Security* (CCS '07), pp. 598–609, New York, USA

Aviˇzienis A, Laprie, J., Randell, B., Landwehr, C., 2004. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing*, Volume 1(1), pp. 11–33

Berman, F. Fox, G., A. J. G., 2003. *Hey, Grid Computing: Making the Global Infrastructure a Reality*, Volume 2, John Wiley and Sons

Boneh, D., Crescenzo, G.D., Ostrovsky, R., Persiano, G., 2004. Public Key Encryption with Keyword Search. *Proc. Int'l Conf. Theory and Applications of Cryptographic Techniques EUROCRYPT*

Bowers, K.D., Juels, A., Oprea, A., 2008. *HAIL: A High-availability and Integrity Layer for Cloud Storage*. Cryptology e-Print Archive, Report 2008/489

Fakhri, I.A-S., Alahmad, M.A., Munthir, K., 2013. Hash Function of Finalist SHA-3: Analysis Study. *International Journal of Advanced Computer Science and Information Technology* (IJACSIT), Volume 2(2), pp. 1–12

Gonzalez, N., Miers, C., Redigolo, F., Simplicio, M., Carvalho, T., Naslund, M., Pourzandi, M., 2012. A Quantitative Analysis of Current Security Concerns and Solutions for Cloud Computing. *Journal of Cloud Computing*, Volume 1(11), pp. 1–18

Juels, A., Burton, J., Kaliski, S., 2007. PORs: Proofs of Retrievability for Large Files. *Proceedings of CCS*, pp. 584–597

Kamara, S., Lauter, K., 2010. Cryptographic Cloud Storage, *Lecture Notes in Computer Science* 6054, pp. 136–49

Koblitz, N., 1989. Hyperelliptic Cryptosystems. *Journal of Cryptology*, Volume 1(3), 1989, pp. 129–150

Latif, R., Abbas, H., Assar, S., Ali, Q., 2014. *Cloud Computing Risk Assessment: A Systematic Literature Review*. in Future Information Technology, pp. 285–295, Springer, Berlin, Germany

Liu, Q., Tan, C.C., Wu, J., Wang, G., 2011. Reliable Re-encryption in Unreliable Clouds. *IEEE Global Telecommunications Conference*, pp: 1–5

Mather, T., Kumaraswamy, S., Latif, S., 2009. *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*. O'Reilly Media

Modi, C., Patel, D., Borisaniya, B., Patel, A., Muttukrishnan Rajarajan, J., 2013. A Survey on Security Issues and Solutions at Different Layers of Cloud Computing. *Journal of Supercomputer*, Volume 63, pp. 561–592

Ning, C., Wang, C., Li, M., Ren, K., Lou, W., 2014. Privacy-preserving Multi-keyword Ranked Search over Encrypted Cloud Data. *IEEE Transactions on Parallel and Distributed Systems*, Volume 25 (1), pp. 829-837

Popa, R.A., Lorch, J.R., Molnar, D., Wang, H.J., Zhuang, L., 2010. *Enabling Security in Cloud Storage SLAs with Cloudproof*. Technical Report, Microsoft Research

Shacham, H., Waters, B., 2008. Compact Proofs of Retrievability. *Proceedings of Asiacrypt 5350*, pp. 90–107

Sood, Sandeep, K., 2012. A Combined Approach to Ensure Data Security in Cloud Computing. *Journal of Network and Computer Applications*, Volume 35(6), pp. 1831–1838

Subashini, S., Kavitha, V., 2011. A Survey on Security Issues in Service Delivery Models of Cloud Computing. *Journal of Network and Computer Applications*, Volume 34(1), pp. 1–11

Sun, Z., Xiong, Zhu, 2014. Data Security and Privacy in Cloud Computing. *International Journal of Distributed Sensor Networks*, Volume 2014, pp. 1-9

Syed, Amid, 2012. Cloud Computing Vs. Grid Computing. *ARPN Journal of Systems and Software*, Volume 2(5), pp. 188-194

Tao, S., Xinjun, W., 2013. Research of Data Security Model in Cloud Computing Platform for SMEs. *International Journal of Security and its Applications*, Volume 7(6), pp. 97–108

Wang, C., Cao, N., Li, J., Ren, K., Lou, W., 2010. Secure Ranked Keyword Search over Encrypted Cloud Data. *Journal of the ACM*, Volume 43(3), pp. 431–73

Wang, C., Cao, N., Li, J., Ren, K., Lou, W., 2010. Secure Ranked Keyword Search over Encrypted Cloud Data. *Proc. IEEE 30th Int'l Conf. Distributed Computing Systems* (ICDCS)

Wang, C., Cao, N., Li, J., Ren, K., Lou, W., 2012. Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data. *IEEE Transactions of Parallel and Distributed Systems*, Volume 23(8), pp. 1467–1479

Wang, C., Wang, Q., Ren, K., Lou, W., 2009. Ensuring Data Storage Security in Cloud Computing. Quality of service, IWQoS. *IEEE 17th International Workshop*, pp. 1–9

Xiao, Z., Xiao, Y., 2013. Security and privacy in cloud computing. *IEEE Communications Surveys & Tutorials*, Volume 15(2), pp. 843–859

Zhao, G., Chunming, R., Jin, L., Feng, Z., Yong, T., 2010. Trusted Data Sharing over Untrusted Cloud Storage Providers. *Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science*, pp. 97–10