International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Cryptoshare: Cryptosystem for Scalable Data Sharing in Cloud Storage

Ghilby Varghese Jaison

M.Tech Scholar, KMP College of Engineering & Technology, Cherukunnam, Perumbavoor, India

Abstract: ***Conceptual Data offering is a vital usefulness in distributed storage. In this paper, we demonstrate to safely, effectively, and adaptably offer information with others in distributed storage. We portray new open key cryptosystems that deliver consistent size ciphertexts such that proficient assignment of decoding rights for any set of ciphertexts are conceivable. The curiosity is that one can total any set of mystery keys and make them as reduced as a solitary key, yet incorporating the force of every last one of keys being accumulated. As it were, the mystery key holder can discharge a consistent size total key for adaptable decisions of ciphertext set in distributed storage, yet the other encoded records outside the set stay private. This conservative total key can be helpfully sent to others or be put away in a keen card with extremely restricted secure stockpiling. We give formal security investigation of our plans in the standard model. We too portray other application of our plans. Specifically, our plans give the first open key patient-controlled encryption for adaptable pecking order, which was yet to be known.***

Keywords: Cryptoshare, Scalable data sharing,cloud,data sharing,encryption,Scalable encryption

1. Introduction

Distributed storage is picking up ubiquity as of late. In big business settings, we see the ascent sought after for information outsourcing, which helps in the vital administration of corporate information. It is likewise utilized as a center innovation behind numerous online administrations for individual applications. These days, it is not difficult to seek free records for email, photograph collection, record imparting and/or remote access, with capacity measure more than 25 GB (or a couple of dollars for more than 1 TB). Together with the current remote innovation, clients can get to pretty much the greater part of their records and messages by a cell telephone in any corner of the world.

Considering information protection, a customary approach to guarantee it is to depend on the server to uphold the right to gain entrance control after verification (e.g., [1]), which implies any startling benefit heightening will uncover all information. In an imparted tenure distributed computing environment, things get to be surprisingly more terrible. Information from distinctive customers can be facilitated on particular virtual machines (VMs) yet live on a solitary physical machine. Information in a target VM could be stolen by instantiating an alternate VM coresident with the focus on one [2]. As to of documents, there are a progression of cryptographic plans which go the extent that permitting an outsider evaluator to check the accessibility of documents for the benefit of the information holder without releasing anything about the information [3], or without trading off the information holders secrecy [4]. Similarly, cloud clients presumably won't hold the solid conviction that the cloud server is making a decent showing regarding classifiedness. A cryptographic arrangement, for instance, [5], with demonstrated security depended on number-theoretic presumptions is more alluring, at whatever point the client is not superbly content with believing the security of the VM or the genuineness of the specialized staff. These clients are spurred to encrypy their information with their own keys before transferring them to the server.

Information offering is an imperative usefulness in cloud capacity. Case in point, bloggers can let their companions see a subset of their private pictures; an undertaking may concede her workers access to a segment of delicate information. The testing issue is the means by which to viably impart encoded information. Obviously clients can download the encoded information from the capacity, decode them, then send them to others for imparting, yet it loses the estimation of distributed storage. Clients ought to have the capacity to delegate the right to gain entrance privileges of the offering information to others so they can get to these information from the server specifically. On the other hand, discovering an effective and secure approach to offer halfway information in distributed storage is not inconsequential. Underneath we will take Dropbox1 as an illustration for representation.

Accept that Alice puts all her private photographs on Dropbox, and she would like to open her photographs to everybody. Because of different information spillage plausibility Alice can't feel calmed by simply depending on the security insurance instruments gave by Dropbox, so she encodes all the photographs utilizing her own particular keys before transferring. One day, Alice's companion, Bob, requests that her impart the photographs assumed control all these years which Bob showed up in. Alice can then utilize the offer capacity of Dropbox, however the issue now is the way to delegate the decoding rights for these photographs to Bob. A conceivable alternative Alice can pick is to safely send Bob thesecret keys required. Effortlessly, there are a pair of excessive methods to be with her beneath the conventional encryption paradigm:

Alice encrypts many records that has a individual encryption critical and gives Joe the particular equivalent secret critical directly.

Alice encrypts records along with distinct keys along with communicates Joe the particular equivalent secret keys. Certainly, the very first technique is limited since many unchosen files can be furthermore released to be able to Joe. To the minute technique, there are practical issues about proficiency. How many these kinds of keys is as many as the quantity of the particular propagated images, say, a lot of. Transporting these types of secret keys inherently needs a secure channel, along with holding these types of keys requires instead high-priced secure hard drive. The price along with complexities required commonly boost along with the quantity of the particular decryption keys being propagated. To put it briefly, it is rather major along with high-priced to achieve that.

Encryption keys additionally accompany two flavors—symmetric key or hilter kilter (open) key. Utilizing symmetric encryption, when Alice needs the information to be begun from an outsider, she needs to give the encryptor her mystery key; clearly, this is not generally alluring. By complexity, the encryption key and unscrambling key are diverse in public key encryption. The utilization of open key encryption gives more adaptability for our applications. Case in point, in big business settings, each representative can transfer encrypyd information on the distributed storage server without the learning of the organization's expert mystery key.

Hence, the best answer for the above issue is that Alice encrypys records with unique open keys, however just sends Bob a solitary (steady size) unscrambling key. Since the unscrambling key ought to be sent by means of a secure channel and kept mystery, little key size is constantly alluring. Case in point, we can't expect substantial stockpiling for decoding keys in the asset requirement gadgets like shrewd telephones, brilliant cards, or remote sensor hubs. Particularly, these mystery keys are typically put away in the carefully designed memory, which is moderately extravagant. The present research endeavors basically center on minimizing the correspondence prerequisites, (for example, data transfer capacity, rounds of correspondence) like total mark [6]. Then again, very little has been carried out about the key itself (see Section 3 for more subtle elements).

Accordingly, the best answer for the above issue is that Alice encrypys documents with different open keys, yet just sends Bob a solitary (consistent size) decoding key. Since the unscrambling key ought to be sent through a protected channel and kept mystery, little key size is constantly attractive. Case in point, we can't expect extensive capacity for decoding keys in the asset stipulation gadgets like advanced mobile phones, keen cards, or remote sensor hubs. Particularly, these mystery keys are typically put away in the carefully designed memory, which is generally lavish. The present exploration endeavors for the most part spotlight on minimizing the correspondence prerequisites like total mark [6]. Nonetheless, very little has been carried out about the key itself

1.1 Our Contributions

In present day cryptography, a crucial issue we regularly study is about leveraging the mystery of a little bit of learning into the capacity to perform cryptographic capacities (e.g., encryption, validation) various times. In this paper, we ponder how to make an unscrambling key more compelling as in it permits unscrambling of numerous ciphertexts, without expanding its size. Particularly, our issue explanation is "To outline a proficient open key encryption plan which helps adaptable designation as in any subset of the ciphertexts (created by the encryption plan) is decryptable by a consistent size unscrambling key (created by the manager of the expert mystery key)."

We take care of this issue by presenting an exceptional sort of open key encryption which we call key-total cryptosystem (KTC). In KTC, clients encrypy a message not just under an open key, additionally under an identifier of ciphertext called class. That implies the ciphertexts are further classified into distinctive classes. The key manager holds an expert mystery called expert mystery key, which can be used to concentrate mystery keys for distinctive classes. More critically, the removed key have can be a total key which is as minimal as a mystery key for a solitary class, however totals the force of numerous such keys, i.e., the unscrambling power for any subset of ciphertext classes.

With our answer, Alice can essentially send Bob a solitary total key by means of a protected email. Bounce can download the encoded photographs from Alice's Dropbox space and afterward utilize this total key to decode these encrypyd photographs.
The sizes of ciphertext, open key, expert mystery key, what's more total enter in our KTC plans are all of consistent size. People in general framework parameter has size direct in the number of ciphertext classes, yet just a little piece of it is required each one time and it can be brought on interest from expansive (yet nonconfidential) distributed storage.

Past results may accomplish a comparable property emphasizing a steady size unscrambling key, however the classes need to fit in with some predefined various leveled relationship. Our work is adaptable as in this demand is disposed of, that is, no exceptional connection is needed between the classes. The subtle element and other related works can be found in Section 3.
We propose a few solid KTC plans with diverse security levels and expansions in this paper. All developments can be demonstrated secure in the standard model. To the best of our insight, our conglomeration mechanism2 in KTC has not been examined.

2. Key-Total Encryption

We first give the structure and definition for key total encryption. At that point we portray how to utilize KTC in a situation of its application in distributed storage.

2.1 Framework

A key-total encryption plan comprises of five polynomial-time calculations as follow:

The information holder builds people in general framework parameter through Setup and creates an open/expert secret key pair through KeyGen. Messages can be encrypyted by means of Encrypt by any individual who additionally chooses what ciphertext class is related with the plaintext message to be encrypted. The information holder can utilize the expert mystery to create an total unscrambling key for a set of ciphertext classes through Remove. The created keys can be gone to delegatees safely using an in system mailing service .Finally, any client with a total key can decrypt any ciphertext given that the ciphertext's class is contained in the total key by means of Decrypt.

- Setup(1^{λ}; n): executed by the information manager to setup an record on an untrusted server. On info a security level parameter 1^{λ} and the quantity of ciphertext classes n (i.e., class file ought to be a number limited by 1 and n), it yields general society framework parameter param, which is discarded from the info of alternate calculations for quickness.
- .KeyGen: executed by the information manager to haphazardly produce an open/expert mystery key pair (pk; msk).
- Encrypt(pk; i;m): executed by any individual who needs to encode information. On data an open key pk, a record i meaning the ciphertext class, and a message m, it yields a ciphertext C.
- Extract(msk; S): executed by the information manager for designating the decoding force for a certain set of ciphertext classes to a delegatee. On info the expert mystery key msk and a set S of files relating to distinctive classes, it yields the total key for set S meant by KS.
- Decrypt(KS; S; i; C): executed by a delegatee who gotten a total key KS produced by Extract. On info KS, the set S, a list i indicating the ciphertext class the ciphertext C fits in with, and C, it yields the unencrypyd result m.

2.2 Sharing Encrypted Data

A standard application of KTC is information imparting. The key conglomeration property is particularly valuable when we anticipate the assignment to be proficient and adaptable. The plans empower a substance supplier to impart her information in a classified what's more particular path, with a settled and little ciphertext development, by dispersing to each one approved client a solitary what's more little total key.

Here, we depict the fundamental thought of information offering in cloud capacity utilizing KTC, . Assume Alice needs to impart her information m1;m2; . . .;m on the server. She to begin with performs Setup(1^{λ}; n) to get param and execute KeyGen to get people in general/expert mystery key pair (pk; msk). The framework parameter param and open key pk can be made open and expert mystery key msk ought to be kept mystery by Alice. Anybody (counting Alice herself) can then encrypy every mi by Ci ¼ Encrypt(pk; i;mi). The encrypyted information are transferred to the server.

With param and pk, individuals who collaborate with Alice can upgrade Alice's information on the server. When Alice is eager to impart a set S of her information with a companion Bob, she can figure the total key KS for Bob by performing Extract(msk; S). Since KS is simply a steady size key, it is simple to be sent to Bob by means of a safe email. In the wake of acquiring the total key, Bob can download the information he is approved to get to. That is Bob downloads Ci (and some required values in param) from the server. With the total key KS, Bob can unencrypy every Ci by Decrypt(KS; S; i; Ci) for every i E s.

3. Related Work

This area we contrast our fundamental KTC plan and other conceivable arrangements on imparting in secure distributed storage.

3.1 Cryptographic Keys For A Predefined Hierarchy

We begin by examining the most significant study in the writing of cryptography/security. Cryptographic key task plans (e.g., [11], [12], [13], [14]) plan to minimize the cost in putting away and overseeing mystery keys for general cryptographic utilization. Using a tree structure, a key for a given extension can be utilized to determine the keys of its relative hubs (yet not the other route round). Just allowing the guardian key verifiably allows all the keys of its relative hubs. Sandhu [15] proposed a technique to produce a tree chain of importance of symmetric-keys by utilizing rehashed assessments of pseudorandom capacity/blockcipher on a settled mystery. The idea can be summed up from a tree to a diagram. More progressed cryptographic key task plans help access strategy that can be demonstrated by a non-cyclic diagram or a cyclic chart [16], [17], [7]. The majority of these plans produce keys for symmetric-key cryptosystems, even despite the fact that the key determinations may require secluded number-crunching as utilized as a part of open key cryptosystems, which are by and large more costly than "symmetric-key operations, for example, pseudorandom capacity.

We take the tree structure as a case. Alice can first order the ciphertext classes as per their subjects.Every hub in the tree speaks to a mystery key, while the leaf hubs speaks to the keys for individual ciphertext classes. Filled circles speak to the keys for the classes to be appointed and circles dodged by dabbed lines speak to the keys to be allowed. Note that each key of the nonleaf hub can infer the keys of its relative hubs.

If Alice needs to impart all the documents in the "individual" class, she just needs to allow the key for the hub "individual," which consequently gives the delegatee the keys of all the relative hubs ("photograph," "music"). This is the perfect case, where most classes to be imparted have a place with the same limb and in this manner a guardian key of them is sufficient. Nonetheless, it is still troublesome for general cases.

If Alice imparts her demo music at work ("work"! "easygoing"! "demo" and "work"! "classified" ! "demo") with a partner who likewise has the rights to see some of her individual information, what she can do is to give more keys, which prompts an increment in the aggregate key size. One

can see that this methodology is not adaptable when the arrangements are more intricate and she needs to impart diverse sets of records to diverse individuals. For this delegatee in our case, the number of conceded mystery keys turns into the same as the number of classes.

By and large, various leveled methodologies can tackle the issue part of the way if one means to impart all documents under a certain limb in the pecking order. By and large, the quantity of keys increments with the quantity of limbs. It is unrealistic to concocted an order that can spare the quantity of aggregate keys to be allowed for all people (which can get to a distinctive set of leaf-hubs) at the same time.

3.2 Compact Key In Symmetric-Key Encryption

Roused by the same issue of supporting adaptable pecking order in decoding force assignment (yet in symmetric-key setting), Benaloh et al. [8] displayed an encryption plan which is initially proposed for briefly transmitting huge number of keys in telecast situation [18]. The development is basic and we quickly survey its key inference transform here for a cement portrayal of what are the alluring properties we need to attain to. The deduction of the key for a set of classes (which is a subset of all conceivable ciphertext classes) is as per the following: A composite modulus N = p * q is picked where p and q are two vast arbitrary primes. An expert mystery key Y is picked at irregular from Zn. Each one class is connected with an unique prime e_i. All these prime numbers can be placed in general society framework parameter.

3.3 Compact Key In Identity-Based Encryption (Ibe)

IBE (e.g., [20], [21], [22]) is a kind of open key encryption in which people in general key of a client can be set as an identitystring of the client (e.g., an email address). There is a trusted gathering called private key generator in IBE which holds an expert mystery key and issues a mystery key to every client concerning the client personality. The encryptor can take people in general parameter and a client character to encode a message. The beneficiary can unencrypy this ciphertext by his mystery key.

Guo et al. [23], [9] attempted to assemble IBE with key accumulation. One of their plans [23] expect irregular prophets however an alternate [9] does not. In their plans, key total is obliged as in all keys to be amassed must originate from diverse "personality divisions." While there are an exponential number of characters and hence mystery keys, just a polynomial number of them can be accumulated. In particular, their key-collection [23], [9] has a go at to the detriment of Oðnþ sizes for both ciphertexts and people in general parameter, where n is the quantity of mystery keys which can be accumulated into a steady size one. This incredibly expands the expenses of putting away and transmitting ciphertexts, which is unreasonable by and large, for example, imparted distributed storage. As we specified, our plans characteristic steady ciphertext size, and their security holds in the standard model.

In fluffy IBE [21], one single minimized mystery key can decode ciphertexts encrypyd under numerous characters which are close in a certain metric space, yet not for a discretionary set of personalities and, thusly, it doesn't match with our concept of key total.

4. Basic Construction

Key total encryption plans comprise of five polynomial time calculations and they are portrayed as five modules:

- Setup Phase
- KeyGen Phase
- Encrypting Phase
- Extract Phase
- Decrypting Phase

1 Setup (1^{λ},n)
It is executed by the information holder to setup a the record on the untrusted server for imparting of information. On info a security level parameter 1^{λ} and the quantity of ciphertext classes n (i.e., class record ought to be a number limited by 1 and n), it yields general society framework parameter param, which is overlooked from the data of alternate calculations for quickness.

2 KeyGen stage
This calculation is utilization for the era of open key. The information holder creates an open discharge key to encrypy the information over cloud. Information holder to arbitrarily produce an open/expert mystery key pair (pk , msk).

3 Encrypt (pk, i, m)
This calculation encodes the information gave by the information manager by utilizing the discharge key. This encoded information is then impart among the cloud. executed by any individual who needs to encode information. On info an open key pk, a record i indicating the ciphertext class, and a message m, it yields a ciphertext C.

4 Extract (msk,S)
It is executed by the information manager for assigning the decoding force for a certain set of ciphertext classes to a delegatee. On information the expert mystery key msk and a set S of files relating to distinctive classes, it yields the total key for set S indicated by KS. The total key is utilization for removing the specific square of the figures from the figure document. At the same time other encoded information stays secure.

5 Decrypt (Ks, S, i, C)
It is executed by a delegatee who got a total key KS created by Extract. On info Ks, the set S, a list i meaning the ciphertext class the ciphertext C has a place with, and C, it yields the decoded result m in the event that i ϵ S.The encoded information is then unencrypyd by utilizing the same discharge key which is utilization for encrypt.

5. Conclusion

Step by step instructions to ensure clients' information security is a focal inquiry of distributed storage. With more

scientific devices, cryptographic plans are getting more adaptable and regularly include different keys for a solitary application. In this paper, we consider instructions to "pack" mystery enters in broad daylight key cryptosystems which help appointment of mystery keys for diverse ciphertext classes in distributed storage. Regardless of which one among the force set of classes, the delegatee can simply get a total key of steady size. Our methodology is more adaptable than progressive key task which can just spare spaces if all key-holders impart a comparative set of benefits.

A constraint in our work is the predefined bound of the number of most extreme ciphertext classes. In distributed storage,the quantity of ciphertexts normally becomes quickly. So we need to save enough ciphertext classes for what's to come augmentation. Else, we have to extend general society key .Despite the fact that the parameter can be downloaded with ciphertexts, it would be better on the off chance that its size is autonomous of the greatest number of ciphertext classes applications and extensions. Authors are strongly encouraged not to call out multiple figures or tables in the conclusion—these should be referenced in the body of the paper.

References

[1] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, "SPICE – Simple Privacy-Preserving Identity-Management for Cloud Environment," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.
[2] L. Hardesty, Secure Computers Aren't so Secure. MIT press, http://www.physorg.com/news176107396.html, 2009.
[3] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.
[4] B. Wang, S.S.M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS), 2013.
[5] S.S.M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R.H. Deng, "Dynamic Secure Cloud Storage with Provenance," Cryptography and Security, pp. 442-464, Springer, 2012.
[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '03), pp. 416-432, 2003.
[7] M.J. Atallah, M. Blanton, N. Fazio, and K.B. Frikken, "Dynamic and Efficient Key Management for Access Hierarchies," ACM Trans. Information and System Security, vol. 12, no. 3, pp. 18:1-18:43, 2009.
[8] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," Proc. ACM Workshop Cloud Computing Security (CCSW '09), pp. 103-114, 2009.
[9] F. Guo, Y. Mu, Z. Chen, and L. Xu, "Multi-Identity Single-Key Decryption without Random Oracles," Proc. Information Security and Cryptology (Inscrypt '07), vol. 4990, pp. 384-398, 2007.
[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06), pp. 89-98, 2006.
[11] S.G. Akl and P.D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," ACM Trans. Computer Systems, vol. 1, no. 3, pp. 239-248, 1983.
[12] G.C. Chick and S.E. Tavares, "Flexible Access Control with Master Keys," Proc. Advances in Cryptology (CRYPTO '89), vol. 435, pp. 316-322, 1989.
[13] W.-G. Tzeng, "A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy," IEEE Trans. Knowledge and Data Eng., vol. 14, no. 1, pp. 182-188, Jan./Feb. 2002.
[14] G. Ateniese, A.D. Santis, A.L. Ferrara, and B. Masucci, "Provably- Secure Time-Bound Hierarchical Key Assignment Schemes," J. Cryptology, vol. 25, no. 2, pp. 243-270, 2012.
[15] R.S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," Information Processing Letters, vol. 27, no. 2, pp. 95-98, 1988.
[16] Y. Sun and K.J.R. Liu, "Scalable Hierarchical Access Control in Secure Group Communications," Proc. IEEE INFOCOM '04, 2004.
[17] Q. Zhang and Y. Wang, "A Centralized Key Management Scheme for Hierarchical Access Control," Proc. IEEE Global Telecomm. Conf. (GLOBECOM '04), pp. 2067-2071, 2004.
[18] J. Benaloh, "Key Compression and Its Application to Digital Fingerprinting," technical report, Microsoft Research, 2009.
[19] B. Alomair and R. Poovendran, "Information Theoretically Secure Encryption with Almost Free Authentication," J. Universal Computer Science, vol. 15, no. 15, pp. 2937-2956, 2009.
[20] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Advances in Cryptology (CRYPTO '01), vol. 2139, pp. 213-229, 2001.
[21] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '05), vol. 3494, pp. 457-473, 2005.
[22] S.S.M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, "Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions," Proc. ACM Conf. Computer and Comm. Security, pp. 152-161, 2010.
[23] F. Guo, Y. Mu, and Z. Chen, "Identity-Based Encryption: How to Decrypt Multiple Ciphertexts Using a Single Decryption Key," Proc. Pairing-Based Cryptography Conf. (Pairing '07), vol. 4575, pp. 392-406, 2007.
[24] M. Chase and S.S.M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," Proc. ACM Conf. Computer and Comm. Security, pp. 121-130. 2009,
[25] T. Okamoto and K. Takashima, "Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption," Proc. 10th Int'l

Conf. Cryptology and Network Security (CANS '11), pp. 138-159, 2011.
[26] R. Canetti and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 185-194, 2007.
[27] C.-K. Chu and W.-G. Tzeng, "Identity-Based Proxy Re-encryption without Random Oracles," Proc. Information Security Conf. (ISC '07), vol. 4779, pp. 189-202, 2007.
[28] C.-K. Chu, J. Weng, S.S.M. Chow, J. Zhou, and R.H. Deng, "Conditional Proxy Broadcast Re-Encryption," Proc. 14th Australasian Conf. Information Security and Privacy (ACISP '09), vol. 5594, pp. 327-342, 2009.
[29] S.S.M. Chow, J. Weng, Y. Yang, and R.H. Deng, "Efficient Unidirectional Proxy Re-Encryption," Proc. Progress in Cryptology (AFRICACRYPT '10), vol. 6055, pp. 316-332, 2010.
[30] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Trans. Information and System Security, vol. 9, no. 1, pp. 1-30, 2006.
[31] D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," Proc. Advances in Cryptology Conf. (CRYPTO '05), vol. 3621, pp. 258-275, 2005.
[32] L.B. Oliveira, D. Aranha, E. Morais, F. Daguano, J. Lopez, and R. Dahab, "TinyTate: Computing the Tate Pairing in Resource- Constrained Sensor Nodes," Proc. IEEE Sixth Int'l Symp. Network Computing and Applications (NCA '07), pp. 318-323, 2007.
[33] D. Naor, M. Naor, and J. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," Proc. Advances in Cryptology Conf. (CRYPTO '01), pp. 41-62, 2001.
[34] T.H. Yuen, S.S.M. Chow, Y. Zhang, and S.M. Yiu, "Identity-Based Encryption Resilient to Continual Auxiliary Leakage," Proc. Advances in Cryptology Conf. (EUROCRYPT '12), vol. 7237, pp. 117-134, 2012.
[35] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical Identity Based Encryption with Constant Size Ciphertext," Proc. Advances in Cryptology Conf. (EUROCRYPT '05), vol. 3494, pp. 440-456, 2005.
[36] D. Boneh, R. Canetti, S. Halevi, and J. Katz, "Chosen-Ciphertext Security from Identity-Based Encryption," SIAM J. Computing, vol. 36, no. 5, pp. 1301-1328, 2007.

Author Profile

Ghilby Varghese Jaison received the B.Tech (CS) degree from Musaliar College of Engineering and Technology, Pathanamthitta, India in 2009-20013. In 2013-2015 pursuing M.Tech Computer Science and Engineering Specialization in Cyber Security. His research interests include Ethical hacking, Network security and GUI based software development.