AN ADVANCED METHODOLOGY IN KEYWORD SEARCH ON ENCRYPTED DATA IN CLOUD ENVIRONMENT

Ms. Vajarala dhanalakshmi [1], Mrs. A. Chaithanya Sravanthi [2*]

1 Final Year MCA Student, QIS College of Engineering and Technology, Ongole

*2*Assistant Professor, MCA Dept., QIS College of Engineering and Technology, Ongole*

Abstract: *Accessible encryption enables a cloud server to conduct keyword search over encoded information for the benefit of the information clients without learning the hidden plaintexts. Nonetheless, most existing accessible encryption conspires just help single or conjunctive keyword search, while a couple of different plans that can perform expressive catchphrase look are computationally wasteful since they are worked from bilinear pairings over the composite-order groups. In this paper, we propose an expressive open key accessible encryption plot in the prime-request gatherings, which permits keyword look strategies (i.e., predicates, access structures) to be communicated in conjunctive, disjunctive or any monotonic Boolean equations and accomplishes critical execution improvement over existing plans. We formally characterize its security, and demonstrate that it is specifically secure in the standard model. Additionally, we execute the proposed plan utilizing a fast prototyping instrument called Charm, and lead a few examinations to assess it execution. The outcomes exhibit that our plan is substantially more proficient than the ones worked over the composite-order groups.*

Keywords: Cloud Computing, Keyword Search, Composite-order group.

I. INTRODUCTION

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a model for enabling ubiquishared 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. Driven by the abundant benefits brought by the cloud computing such as cost saving, quick deployment, flexible resource configuration, etc., more and more enterprises and individual users are taking into account migrating their private data and native applications to the cloud server. A matter of public concern is how to guarantee the security of data that is outsourced to a remote cloud server and breaks away from the direct control of data owners. Encryption on private data before outsourcing is an effective measure to protect data confidentiality.

However, encrypted data make effective data retrieval a very challenging task. To address the challenge (i.e., search on encrypted data), Song et al. first introduced the concept of searchable encryption [1][2][3] and proposed a practical technique that allows users to search over encrypted data through encrypted query keywords in. Later, many searchable encryption schemes were proposed based on symmetric key and public-key setting to strengthen security and improve query efficiency. Recently, with the growing popularity of cloud computing, how to securely and efficiently search over encrypted cloud data becomes a research focus. Some approaches have been proposed based on traditional searchable encryption schemes in [4], which aim to protect data security and query privacies with better query efficient for cloud computing. However, all of these schemes are based on an ideal assumption that the cloud server is an ”honest-but-curious” entity and keeps robust [5], and secure software/hardware environments.

As a result, correct and complete query results always are unexceptionally returned from the cloud server when a query ends every time. However, in practical applications [6], the cloud server may return erroneous or incomplete query results once he behaves dishonestly for illegal profits such as saving computation and communication cost or due to possible software/hardware failure of the server. Therefore, the above fact usually motivates [7], data users to verify the correctness and completeness of query results. Some researchers proposed to integrate the query results verification mechanisms to their secure search schemes, (e.g., embedding verification

information into the specified secure indexes or query results). Upon receiving query results, data users use specified verification information to verify their correctness and completeness.

There are two limitations in these schemes:

1) These verification mechanisms provide a coarse grained verification, i.e., if the query result set contains all qualified and correct data files, then these schemes reply yes, otherwise reply no. Thus, if the verification algorithm outputs no, a data user has to abort the decryption for all query results despite only one query result is incorrect.

2) These verification mechanisms are generally tightly coupled to corresponding secure query constructions and have not universality. In a search process, for a returned query results set that contains multiple encrypted data files, a data user may wish to verify the correctness of each encrypted data file (thus, he can remove incorrect results and retain the correct ones as the ultima query results) or wants to check how many or which qualified data files [22], are not returned on earth if the cloud server intentionally omits some query results. This information can be regarded as a hard evidence to punish the cloud server. This is challenging to achieve the fine-grained verifications since the query and verification are enforced in the encrypted environment. In, we proposed a secure and fine-grained query results verification scheme [23], by constructing the verification object for encrypted outsourced data files. When a query ends, the query results set along with the corresponding verification object are returned together, by which the query user can accurately verify:

1) The correctness of each encrypted data file in the results set;

2) How many qualified data files are not returned and

3) Which qualified data files are not returned? Furthermore, our proposed verification scheme is lightweight and loose-coupling to concrete secure query schemes and can be very easily equipped into any secure query scheme for cloud computing.

II RELATED WORK

a)Commonsense Techniques for Searches on Encrypted Data

It is alluring to store information on information stockpiling servers, for example, mail servers and document servers in encoded structure to lessen security and protection dangers. Be that as it may, this normally infers one need to forfeit usefulness for security. For instance, if a customer wishes to recover just archives containing certain words, it was not recently realized how to let the information stockpiling server play out the inquiry and answer the question, without loss of information secrecy. We depict our cryptographic [19], plans for the issue of looking on encoded information and give verifications of security to the subsequent crypto frameworks. Our procedures have various urgent points of interest. They are provably secure: they give provable mystery to encryption, as in the untrusted server can't get the hang of anything about the plaintext when just given the ciphertext; they give inquiry [20], detachment to looks, implying that the untrusted server can't pick up much else about the plaintext than the query item; they give controlled seeking, so that the untrusted server can't scan for a subjective word without the client's approval; they additionally bolster concealed inquiries, so the client may approach the untrusted server to scan for a mystery word without uncovering the word to the server. The calculations exhibited [21], are basic, quick (for an archive of length n, the encryption and inquiry calculations just need O(n) stream figure and square figure tasks), and present no space and correspondence overhead, and subsequently are down to earth to utilize today.

b) Programming Protection and Simulation on Oblivious Rams

Programming assurance is a standout amongst the most vital issues concerning PC practice. There exist numerous heuristics and specially appointed techniques for security, however the issue overall has not gotten the hypothetical treatment it merits. In this paper, we give hypothetical treatment of programming insurance. We decrease the issue of programming security to the issue of productive reproduction on unmindful RAM.A machine is negligent if the grouping in which it gets to memory areas is identical for any two contributions with a similar [24], running time. For instance, a negligent Turing Machine is one for which the development of the heads on the tapes is indistinguishable for every calculation. (In this way, the development is autonomous of the real information.) What is the log jam in the running time of a machine, on the off chance that it is required to be unaware? In 1979, Pippenger and Fischer indicated how a two-tape unaware Turing Machine can reenact, on-line, a one-tape Turing Machine, with a logarithmic stoppage in the running time. We demonstrate a similar to result for the arbitrary access machine (RAM) model of calculation. Specifically, we tell the best way to complete an on-line reenactment of a discretionary RAM by a probabilistic negligent RAM with a polylogaithmic stoppage in the running time. Then again, we demonstrate that a logarithmic stoppage is a lower bound.

c) Open Key Encryption with Keyword Search

We think about the issue of looking on information that is encoded utilizing an open key framework. Consider client Bob who sends email to client Alice encoded under Alice's [16], open key. An email door needs to test whether the email contains the watchword "dire" with the goal that it could course the email likewise. Alice, then again does not wish to enable the door to decode every one of her messages. We characterize and develop a system that empowers Alice to give

a key to the entryway that empowers the door to test whether "earnest" is a catchphrase in the email without picking up whatever else about the email. We allude to this system as Public Key Encryption [9][10], with watchword Search. As another precedent, consider a mail server that stores different messages openly encoded for Alice by others. Utilizing our component Alice can send the mail server a key that will empower the server to distinguish all messages containing some particular watchword, yet get the hang of nothing else. We characterize the idea of open key encryption with catchphrase hunt and give a few developments.

III. EXISTING SYSTEM

In a private-key SE setting, a user uploads its private data to a remote database and keeps the data private from the remote database administrator. Private-key SE allows the user to retrieve all the records containing a particular keyword from the remote database. However, as the name suggests, private-key SE solutions only apply to scenarios where data owners and data users totally trusted each other.

Private Information Retrieval. With respect to public database such as stock quotes, where the user is unaware of it and wishes to search for some data-item without revealing to the database administrator which item it is, private information retrieval (PIR) protocols were introduced, which allow a user to retrieve data from a public database with far smaller communication then just downloading the entire database. Nevertheless, in our context, the database is not publicly available, the data is not public, so the PIR solutions cannot be applied.

Disadvantages:

1. Private-key SE solutions only apply to scenarios where data owners and data users totally trusted each other.

2. Nevertheless, in our context, the database is not publicly available, the data is not public, so the PIR solutions cannot be applied.

IV. PROPOSED SYSTEM

We propose an public key based expressive SE plot in prime-request groups, which is particularly reasonable for catchphrase look over encoded information in situations of numerous information proprietors and different information clients, for example, the cloud-based medicinal services data framework that has re-appropriated PHRs[8], [11], [12], [13], [14], [15], [22], [23] from different human services suppliers.

Our expressive SE [17], conspire comprises of a trusted trapdoor age focus which distributes an open framework parameter and keeps an ace key covertly, a cloud server which stores and hunts encoded information for information clients, numerous information proprietors who transfer scrambled information to the cloud, and different information clients who might want to recover encoded information containing certain catchphrases. To redistribute an encoded record to the cloud, an information proprietor annexes the scrambled archive with watchwords encoded under the open parameter and transfers the joined encoded report and encoded catchphrases to the cloud. To recover all the scrambled records containing watchwords fulfilling a specific access structure (i.e., predicate or strategy) such as("Illness = Diabetes" AND ("Age = 30" OR "Weight = 150-200")), an information client initially gets a trapdoor related with the entrance structure from the trapdoor age focus and after that sends the trapdoor to the cloud server. The last will lead the hunt and return the relating scrambled records to the information client.

The design goals of our proposed scheme are

a)Expressiveness The proposed scheme should support keyword access structures expressed in any Boolean formula with AND and OR gates.

b) ***Efficiency*** The proposed scheme should be adequately efficient in terms of computation, communication and storage for practical applications.

c) Keyword privacy First, a ciphertext without its corresponding trapdoors should not disclose any information about the keyword values it contains to the cloud server and outsiders. Second, a trapdoor should not leak information on keyword values to any outside attackers without the private key of the designated cloud server. We capture this notion of security for the SE scheme [20][21], in terms of semantic security to ensure that encrypted data does not reveal any information about the keyword values, which we call "selective indistinguishability against chosen keyword-set attack [22], (selective IND-CKA security)".

d) Provable security The security of the proposed scheme should be formally proved under the standard model rather than the informal analysis.

Advantages:

1. We define a security model for expressive SE, which takes into account all adversarial capabilities of the standard SE security notion.

2. Using a randomness splitting technique, our scheme achieves security against offline keyword dictionary guessing attacks to the ciphertexts.

V ARCHITECTURE & SYSTEM COMPONENTS

Below architecture diagram represents mainly flow of request from the users to database through servers. In this scenario overall system is designed in three tiers separately using three layers called presentation layer, business layer,

data link layer. This paper was developed using 3-tier architecture.

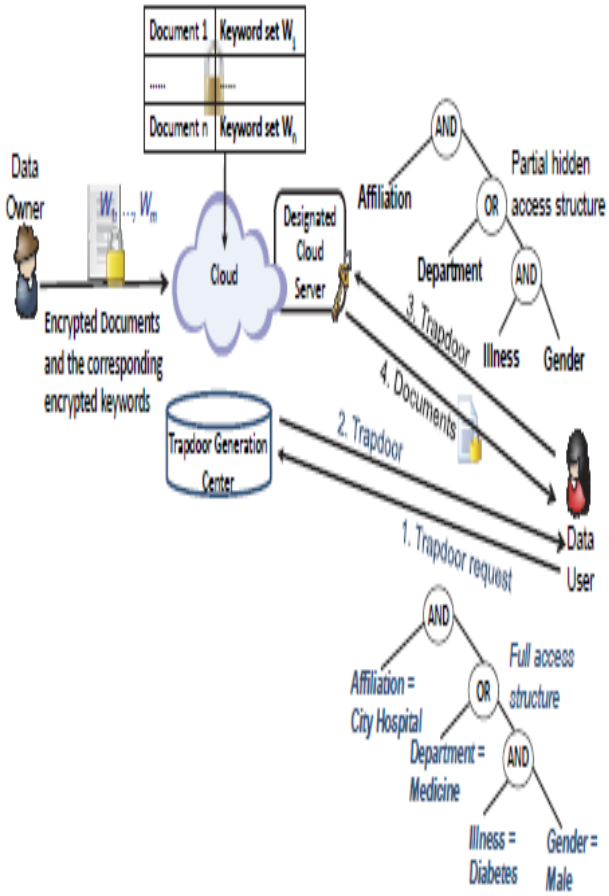


Fig: Architecture of keyword search System

The Major components of the architecture are given by

1. Data Owner:

Data owners who outsource encrypted data to a public cloud. Data owner consists of two parts: the encrypted document generated using an encryption scheme and the encrypted keywords generated

2. Data User:

Data users who are privileged to search and access encrypted data.

A data user issues a trapdoor request by sending a keyword access structure to the trapdoor generation center.

After obtaining a trapdoor, the data user sends the trapdoor and the corresponding partial hidden access structure (i.e., the access structure without keyword values) to the designated cloud server.

3. Tapdoor Generation Center:

Trusted trapdoor generation center who publishes the system parameter and holds a master private key and is responsible for trapdoor generation for the system.

Trapdoor generation centre which generates and returns a trapdoor corresponding to the access structure to data user.

Trapdoor generation center has a separate authentication mechanism to verify each data user and then issue them the corresponding trapdoors.

4. Cloud Server:

Designated cloud server who executes the keyword search operations for data users. To enable the cloud server to search over ciphertexts, the data owners append every encrypted document with encrypted keywords. The latter performs the testing operations between each ciphertext and the trapdoor using its private key, and forwards the matching cipher texts to the data user.

The three-tier software architecture (a three layer architecture) emerged in the 1990s to overcome the limitations of the two-tier architecture. The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users (as compared to only 100 users with the two tier architecture) by providing functions such as queuing, application execution, and database staging.

The three tier architecture is used when an effective distributed client/server design is needed that provides (when compared to the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user. These characteristics have made three layer architectures a popular choice for Internet applications and net-centric information systems.

Unbounded keyword search

In "small universe" KP-ABE constructions [18], the size of the keyword space were polynomially bounded in the security parameter and the keywords were fixed at the setup phase. Moreover, the sizes of the public parameters grow linearly with the number of keywords [8], [14], [15]. On the contrary, in "large universe" constructions, the size of the keyword space can be exponentially large, so it is much more desirable in the real-world applications. Our construction of the expressive SE scheme inherits the advantages of the Rouselakis-Waters scheme [18]. Thus, it is straightforward to see that in our SE scheme, the size of the public parameter is immutable [23],with the number of keywords, and the number

of the keywords allowed for the system is unlimited and can be freely set.

Extensions

Our expressive SE system can be extended in several ways. Expressive searchable encryption for the range search. Range search is an important requirement for searchable encryption in many applications. By defining keywords in a hierarchical manner as shown in [24], we can directly expand our SE system to support a class of simple range search [24]. Take a keyword name "Age" with keyword values from 0 to 100 as an example. The path of the leaf node "11-20" is ("0-100", "0-30", "11-20"), and "0-30", "0-10" are simple ranges from level-2 and level-3, respectively. Anonymous KP-ABE. Our SE system is built by anonym zing the Rouselakis-Waters KP-ABE scheme [18]. Therefore, our scheme can be easily extended to obtain an unbounded and anonymous KP-ABE scheme in the prime-order group without random oracles, in which an adversary, given a ciphertext, cannot learn any information about the associated attribute set.Anonymous hierarchical identity-based encryption (HIBE). The Rouselakis-Waters KP-ABE scheme in [18] can be converted to an HIBE scheme using nonrepeating identities, "AND" policies and delegation capabilities [19]. Since our SE scheme can be used to construct an anonymous KP-ABE scheme, it can be further converted to an anonymous HIBE scheme using the same method as in [19].

Comparison of Expressive Keyword Search Schemes

Comparisons of expressive keyword search schemes

	Keyword Privacy	Expressiveness	Bilinear Group	Security	Unbounded keywords
BDOP04 [7]	keyword guessing attacks on trapdoors	AND	prime	full random oracle	yes
KSW13 [16]	keyword guessing attacks on trapdoors	AND, OR	composite	full standard model	no
LZDLC13 [8]	keyword guessing attacks on trapdoors	AND, OR	composite	full standard model	no
LHZF14 [14]	no keyword guessing attacks on trapdoors	AND, OR, NOT	composite	full standard model	no
Our scheme	keyword guessing attacks on trapdoors by designated server only	AND, OR	prime	selective standard model	yes

Table 1: Comparison of expressive keyword search schemes

Comparison:

Let |pars|, |msk |, |CT|, $|T_{M;}|$, |M| be the sizes of the public parameter, the master private key, the ciphertext, the trapdoor and the access structure, respectively. Let k be the length of the vector corresponding to the ciphertext in [16], l be the number of keywords in an access structure, n be the maximum number of keywords allowed for the system, and m be the size of a keyword set ascribed to a ciphertext. Denote E as an exponentiation operation, P as a pairing operation, $_1$ as the number of elements in $I_{Mp} = \{I_1, \ldots, Ix_1\}$, X_2 as $|I_1| + \ldots +$ $|Ix_1|$, and X_3 as the number of primed keywords [14] in a search predicate.

Table 2: Comparison of Storage and communication overhead

	Public parameter |pars|	Master private key |msk|	Trap-door $\downarrow_{M;}$	Cipher-text |CT|
KSW13 [16]	2k +3	2k +4	2k + 1 + |M|	2k + 1
LZDLC13 [8]	n + 5	n + 4	2l + |M|	m + 2
LHZF14 [14]	n + 4	n + 2	3l + |M|	m + 2
Our Scheme	9	5	6l + |M|	5m + 2

We compare our searchable encryption system with the other three known expressive SE schemes [8], [14], [16] in Table 2 which are all constructed over composite order groups. From Table 2, it is not difficult to see that our construction is the only one that supports unbounded number of keywords in the expressive keyword search systems. Note that our scheme is measured in terms of number of elements in prime order groups while the other three schemes are measured in terms of number of elements in composite order groups. According to the analysis in [24], in terms of the pairing-friendly elliptic curves, prime order groups have a clear advantage in the parameter sizes over Composite order groups.

VI. RESULT

We implement our scheme in Charm [39]8, which is a framework developed to facilitate rapid prototyping of cryptographic schemes and protocols. Based on the Python programming language, Charm enables one to implement a cryptographic scheme with very few lines of code, significantly reducing development time. Meanwhile, computationally intensive mathematical operations are implemented with native modules, so the overhead due to Python in Charm is less than 1%. Since all Charm routines are designed under the asymmetric groups, our construction is transformed to the asymmetric setting before the implementation. That is, three groups G, ^G and G1 are used and the pairing ^e is a function from G × ^G to G1. Notice that it has been stated in [18] that the assumptions and the security proofs can be converted to the asymmetric setting in a generic way.

We use Charm of version charm-0.43 and Python 3.4 in our implementation. Along with charm-0.43, we install the latest PBC library for underlying cryptographic operations. Our experiments run on an all-in-one desktop computer with

Intel Core i7-4785T CPU (4 core 2.20GHz) and 8GB RAM running 64-bit Ubuntu 15.10.

ECs(time in ms)	Exp. G	Exp. $\hat{G}$	Exp. G_1	Pairing
SS512	0.194	0.194	0.027	0.881
MNT159	0.068	0.584	0.160	3.148
MNT201	0.101	0.762	0.207	4.194
MNT224	0.131	0.968	0.252	5.169

Fig. 2. Computational costs for the group operations and pairings over different elliptic curves on a desktop with 2.2GHz 4 core CPU.

Fig. 3 shows the computational overhead for generating trapdoors containing 2 keywords to 10 keywords, from which we can see that the computation time for the trapdoor generation is almost linear to the number of keywords associated with the access structure in the trapdoor. The MNT curves with higher security levels have longer computation time, so MNT224 has higher computation cost among all curves. The computation time of SS512 is close to that of MNT224 due to its higher exponentiation cost over G. The computation time of generating a trapdoor with 10 keywords is only 0.22s for MNT224, which is quite modest for a powerful trapdoor generation centre.

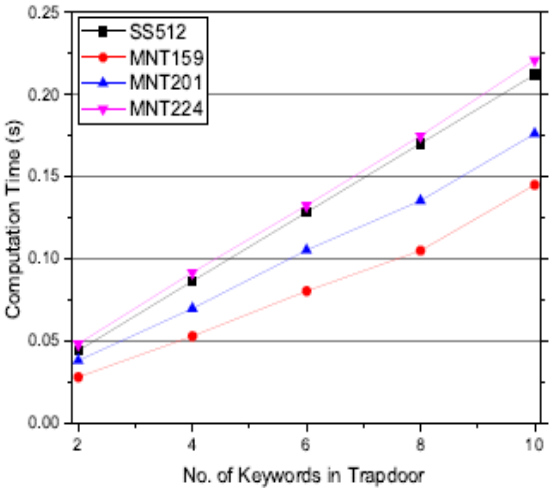
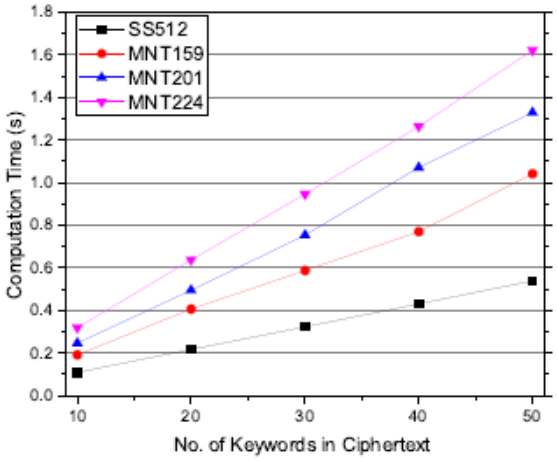


Fig. 4 demonstrates the computation time for the Encrypt algorithm over 10 keywords to 50 keywords. As expected in our analysis, it shows that the computation time is approximately linear to the number of keywords used to generate the ciphertext. The MNT curves with higher security the encryption cost of SS512 is much less than that of MNT curves. This is due to the fact that (4m+1) exponentiations are done in ^G for the total (7m + 2) exponentiations (see Table 3). To encrypt a document with 50 keywords using MNT224 curve, the computation time is about 1.6s, which is acceptable for most applications.



10 20 30 40 50

0

VII. CONCLUSION

In this paper, So as to enable a cloud server to seek on scrambled information without learning the basic plaintexts in the public key setting, Boneh proposed a cryptographic crude called open key encryption with watchword look (PEKS). From that point forward, considering diverse prerequisites in practice,e.g., correspondence overhead, seeking criteria and security upgrade, different sorts of accessible encryption frameworks have been advanced. Be that as it may, there exist just a couple of open key accessible encryption frameworks that help expressive watchword seek arrangements, and they are altogether worked from the wasteful composite-request bunches. In this paper, we concentrated on the plan and investigation of open key accessible encryption frameworks in the prime-request bunches that can be utilized to look through numerous watchwords in expressive seeking equations. In view of an expansive universe key-approach trait based encryption plot given in [18], we exhibited an expressive accessible encryption framework in the prime order assemble which underpins expressive access structures communicated in any monotonic Boolean equations. Likewise, we demonstrated its security in the standard model, and broke down its productivity utilizing PC reproductions.

VIII. REFERENCES

[1] O. Goldreich and R. Ostrovsky, “Software protection and simulationon oblivious rams,” J. ACM, vol. 43, no. 3, pp. 431–473, 1996.

[2] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques forsearches on encrypted data," in 2000 IEEE Symposium on Securityand Privacy, Berkeley, California, USA, May 14-17, 2000. IEEEComputer Society, 2000, pp. 44–55.

[3] E. Goh, "Secure indexes," IACR Cryptology ePrint Archive, vol.2003, p. 216, 2003.

[4] C. Cachin, S. Micali, and M. Stadler, "Computationally privateinformation retrieval with polylogarithmic communication," inSAdvances in Cryptology - EUROCRYPT '99, International Conferenceon the Theory and Application of Cryptographic Techniques, Prague,Czech Republic, May 2-6, 1999, Proceeding, ser. Lecture Notes inComputer Science, vol. 1592. Springer, 1999, pp. 402–414.

[5] G. D. Crescenzo, T. Malkin, and R. Ostrovsky, "Single database privateinformation retrieval implies oblivious transfer," in Advancesin Cryptology - EUROCRYPT 2000, International Conference on theTheory and Application of Cryptographic Techniques, Bruges, Belgium,May 14-18, 2000, Proceeding, ser. Lecture Notes in Computer Science,vol. 1807. Springer, 2000, pp. 122–138.

[6] W. Ogata and K. Kurosawa, "Oblivious keyword search," J. Complexity,vol. 20, no. 2-3, pp. 356–371, 2004.

[7] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Publickey encryption with keyword search," in Advances in Cryptology- EUROCRYPT 2004, International Conference on the Theory andApplications of Cryptographic Techniques, Interlaken, Switzerland, May2-6, 2004, Proceedings, ser. Lecture Notes in Computer Science, vol.3027. Springer, 2004, pp. 506–522.

[8] J. Lai, X. Zhou, R. H. Deng, Y. Li, and K. Chen, "Expressivesearch on encrypted data," in 8th ACM Symposium on Information,Computer and Communications Security, ASIA CCS '13, Hangzhou,China - May 08 - 10, 2013. ACM, 2013, pp. 243–252.

[9] P. Golle, J. Staddon, and B. R. Waters, "Secure conjunctive keywordsearch over encrypted data," in Applied Cryptography and Network Security, Second International Conference, ACNS 2004, YellowMountain, China, June 8-11, 2004, Proceedings, ser. Lecture Notes inComputer Science, vol. 3089. Springer, 2004, pp. 31–45.

[10] D. J. Park, K. Kim, and P. J. Lee, "Public key encryption with conjunctivefield keyword search," in Information Security Applications,5th International Workshop, WISA 2004, Jeju Island, Korea, August 23-25, 2004, Revised Selected Papers, ser. Lecture Notes in ComputerScience, vol. 3325. Springer, 2004, pp. 73–86.

[11] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctivekeyword search and its extension to a multi-user system," in Pairing-Based Cryptography - Pairing 2007, First International Conference,Tokyo, Japan, July 2-4, 2007, Proceedings, ser. Lecture Notesin Computer Science, vol. 4575. Springer, 2007, pp. 2–22.

[12] B. Zhang and F. Zhang, "An efficient public key encryption withconjunctive-subset keywords search," J. Network and Computer Applications, vol. 34, no. 1, pp. 262–267, 2011.

[13] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings, ser. Lecture Notes in Computer Science, vol. 4392. Springer, 2007, pp. 535–554.

[14] Z. Lv, C. Hong, M. Zhang, and D. Feng, "Expressive and secure searchable encryption in the public key setting," in Information Security - 17th International Conference, ISC 2014, Hong Kong, China, October 12-14, 2014. Proceedings, ser. Lecture Notes in Computer Science, vol. 8783. Springer, 2014, pp. 364–376.

[15] J. Shi, J. Lai, Y. Li, R. H. Deng, and J. Weng, "Authorized keyword search on encrypted data," in Computer Security - ESORICS 2014 - 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part I, ser. Lecture Notes in Computer Science, vol. 8712. Springer, 2014, pp. 419–435.

[16]J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," J. Cryptology, vol. 26, no. 2, pp. 191–224, 2013.

[17] D. M. Freeman, "Converting pairing-based cryptosystems from composite-order groups to prime-order groups," in Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings, ser. Lecture Notes in Computer Science, vol. 6110. Springer, 2010, pp. 44–61.

[18] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in

2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013. ACM, 2013, pp. 463–474.

[19] A. B. Lewko and B. Waters, "Unbounded HIBE and attributebased encryption," in Advances in Cryptology - EUROCRYPT 2011- 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings, ser. Lecture Notes in Computer Science, vol. 6632, 2011, pp. 547–567.

[20] X. Boyen and B. Waters, "Anonymous hierarchical identity-based encryption (without random oracles)," in Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings, ser. Lecture Notes in Computer Science, vol. 4117. Springer, 2006, pp. 290–307.

[21] J. Lai, R. H. Deng, and Y. Li, "Expressive CP-ABE with partially hidden access structures," in 7th ACM Symposium on Information, Compuer and Communications Security, ASIACCS '12, Seoul, Korea, May 2-4, 2012. ACM, 2012, pp. 18–19.

[22] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Improved searchable public key encryption with designated tester," in Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2009, Sydney, Australia, March 10-12, 2009. ACM, 2009, pp. 376–379.

[23] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings, ser. Lecture Notes in Computer Science, vol. 4622. Springer, 2007, pp. 535–552.

[24] C. Gu, Y. Zhu, and H. Pan, "Efficient public key encryption with keyword search schemes from pairings," in Information Security and Cryptology, Third SKLOIS Conference, Inscrypt 2007, Xining, China, August 31 - September 5, 2007, Revised Selected Papers, ser. Lecture Notes in Computer Science, vol. 4990. Springer, 2007, pp. 372–383.

Authors Profile

Ms. **Vajarala dhanalakshmi** pursuing MCA 3rd year in Qis College and Engineering and Technology in Department of Master of Computer Applications, Ongole.

Mrs. **A. Chaithanya Sravanthi** is currently working as an Assistant Professor in Department of Master of Computer Applications in QIS College of Engineering & Technology with the Qualification of MCA.