Asian Journal of Computer Science and Technology
ISSN: 2249-0701 Vol.8 No.S3, 2019, pp. 117-121
© The Research Publication, www.trp.org.in

# Providing a Secure Cloud Storage by Using Attribute Based Temporary Key Word Search Scheme

**Likhita Meka**[1] **and Srivyshnavi Pagadala**[2]
[1]PG Student, [2]Senior Assistant Professor
[1&2]Department of Computer Science and Engineering, School of Engineering and Technology,
Sri Padmavati Mahila Visva Vidyalayam, Andhra Pradesh, India
E-Mail: likhimeka@gmail.com, vyshu.sri@gmail.com

*Abstract* - **The cloud providers are not fully trusted in the accept of temporary keyword search on confidential data. Hence this is the main focus of this research, it is necessary to outsource data in the encrypted format. In the attribute-based keyword search scheme the authorized users generate some tokens which were in encrypted format and send them to cloud for the search operation. These tokens can be used to extract all the cipher texts which are generated at any time and contain the search token which were generated by authorized users. Since this may lead to some information leakage, a new cryptographic primitive is introduced which is more secure to propose a scheme in which the search tokens can only extract the cipher texts generated in a specified time interval and that cryptographic primitive is called key-policy attribute-based temporary keyword search (KPABTKS) which provide this property. To evaluate the security, we have to prove that the proposed scheme achieves the keyword secrecy property and is secure against selectively chosen keyword attack (SCKA) both in the random oracle model and Decisional Bilinear Diffie-Hellman (DBDH) assumption.And at last the research will show the complexity of the encryption algorithm is linear with respect to the number of the involved attributes.**

*Keywords:* **Secure Cloud Storage, Key Policy, Security Analysis, Token Gen**

## I. INTRODUCTION

Today, cloud computing plays a crucial role in ourdaily life, as a result of it provides economical, reliable resources for knowledge storage and process activities at a reallylow price. However, the direct access of the cloud to the sensitiveinformation of its users threatens their privacy. A trivial answerto address this downside is encrypting knowledge before outsourcing itto the cloud. However, looking out on the encrypted knowledge is extremelydifficult.Public key cryptography with keyword search (PEKS)[12]may be a cryptographic primitive that was primarily introduced by Boneh *et al.*,[12]to facilitate looking out on the encrypted knowledge. In PEKS,each knowledge owner who is aware of the general public key of the supposed knowledgeuser generates a searchable ciphertext by means that of his/her publickey, and outsources it to the cloud.The notion of attribute-based keyword search(ABKS) [9]to permit a data owner to manage the accessof information users for looking on his/her outsourced encrypted data. They used attribute-based cryptography (ABE) [5] to construct a searchable scientific discipline primitive within the multi-sender/multireceiver model.

*A. Our Contribution:* Thescientificcontribution of the paper issummarizedasfollows:

1. We have at end encrypt introduce the novel notion of KPABTKS, and propose a concrete construction for this new crypto logical primitive which might be applied within the cloud storage services. The projected concrete theme is meant primarily based on linear pairing. Within the projected KP-ABTKS.
2. We have at end encryption formally outline two security definitions for KPABTKS [2] within the common place model. One in every of them defines its security against by selection chosen keyword attack (KPABTKSSCKA) and therefore the different one defines the keyword secrecy of KP-ABTKS.

## II. PRELIMINARIES

*A. Decisional Additive Diffie-Hellman (DBDH)Assumption:* The tuple, (e, P, aP, bP, cP, e(P, P)abc, e(P, P)z, in which a, b, c, z ∈R Zq square measure chosen uniformly every which way.Then, the Decisional additive Diffie-Hellman (DBDH) assumption implies that the success chance of D to tell apart between e(P, P) abc and e(P, P)z is a negligible perform of thesecurity parameter, λ.AdvDBDHD (λ) =|Pr[D(e, P, aP, bP, cP, e(P, P)abc : a, b, c ∈R Zq) = 1]− Pr[D(e, P, aP, bP, cP, e(P, P)z: a, b, c, z ∈R Zq) = 1]|≤ negl(λ) (1)

*B. Changed Decisional Diffie-Hellman Assumption:* The subsequent distributions square measure given to the PPT, D:The tuple, (e, P, aP, bP, cP, abcP, zP), in which a, b, c, z ∈R Zq square measure chosen uniformly every which way. The changed Decisional Diffie-Hellman (MDDH) assumption implies that the success chance of D to part between abcP and zP may be a negligible perform of the security parameter, λ.

Adv MDDHD (λ) =|Pr[D(e, P, aP, bP, cP, abcP : a, b, c ∈R Zq) = 1]− Pr[D(e, P, aP, bP, cP, zP : a, b, c, z ∈R Zq) = 1]| ≤ negl(λ)(2)

*C. Access Management Policy: Access Tree:* Each tree contains some leaves and every leaf is related to associate in nursing attribute.If n is that the root or inner node with out-degree of numn, theneach of its branches square measure labelled from the proper to the left as1, 2 . . . , numn. Let kn, one ≤ kn ≤ numn, denote the brinkvalue associated to the inner node n, wherever kn = one represents the “OR” gate and kn = numn represents the “AND” gate. Trn(Atts) are often computed through one in all thefollowing procedures:

a. For every leaf node n: If att(n) ∈Atts, set Trn(Atts) = 1;otherwise, set Trn (Atts) = zero.
b. For every inner node n, with kids n1, n2, . . . , nnumn: If there exists a set I ⊆ specified |I| ≥ knand∀j∈ I, Trnj(Atts) = 1, then set Trn(Atts) = 1;otherwise, set Trn(Atts) = zero.

1. *Sharing a Secret through the Access Tree:* The tendency to use the algorithm n ∈lvs (Tr) ← Share (Tr,s) for allocating thesecret share of every attribute that is conferred within the accesstree Tr for AN whimsical secret worth s. In this rule, for every node n, the polynomial qn with degree kn − one is generated through the subsequent steps:• If the node, n, be the basis of the access tree Tr, then setqn(0) = s, and choose kn−1 coefficients for the polynomialqn uniformly willy-nilly.• If the node, n, is AN inner node, set qn(0) =qprnt(n)(lbl(n)), and choose kn − one coefficients for polynomial qn uniformly willy-nilly.

a. If the node, n, may be a leaf of the access tree, Tr, Then kn = one and qn(0) = qprnt(n)(lbl(n)). At the top of this rule, every leaf node n of the access tree Tris related to a price qn(0) because the secret share ofs.

## III. KEY-POLICY ATTRIBUTE-BASED TEMPORARYKEYWORD SEARCH (KP-ABTKS)

This theme consists of 4 entities together with data owner, datauser, cloud server and trustworthy Third Party (TTP)[2] that square measuredescribed as follows

1. *Data Owner:* Is AN entity UN agency encrypts its documents beneath AN discretional access management policy and outsources themto the cloud. He/She consider the time of encrypting ingenerating the ciphertexts.

2. *Data User:* Is AN entity UN agency is searching for documentswhich contains AN supposed keyword, and square measure encrypted in a determined amount. The amount is every which wayselected by the info user.

3. *Cloud Server (CS):* Is AN entity with powerful computation and storage resources. Atomic number 55 stores huge quantityof encrypted information, and receives the search tokens to appearfor the desired documents on behalf [4] of the info user. The cloud finds the relevant documents, and sends them backto the info user.

4. *Trustworthy Third Party (TTP):* could be a totally trustworthy entity UN agencyreceives every user’s access tree, and generates their secretkeys like his/her attributes set conferred inhis/her access tree. Then, the TTP sends back the users’credentials through a secure and echt channel.

a. *Formal Definition of KP-ABTKS:* The projected KP-ABTKS theme consists of 5 algorithms [2]**,** Setup, KeyGen, Enc, TokenGen, and Search. These algorithms are described as follows

1. *(msk, pp) ← Setup (1λ):*This formula is travel by the TTP.It takes the safety parameter λ as input and generates themaster secret key msk and also the public parameter pp.
2. *sk ← KeyGen (msk, Tr):* This formula generates a secret key sk for the user with the access tree, Tr.
3. *cph ← Enc(ω, ti, Atts, pp):* This formula generates asearchable cipher text associated with the keyword ω and time of encrypting ti in step with associate attribute set, Atts which is determined by the info owner.
4. *st ← TokenGen(sk, ω, [ts, te]):* the info user runs thisalgorithm to get the search token st for looking outthe cipher texts that are encrypted within the interval[ts, te], and contain the keyword ω, in step with its secretkey sk.
5. *{0,1} := Search(cph,st):* for every hold on ciphertext cphand the received search token st that is related tospecific keyword ω and attribute set Atts, this formula returns one ifallofthe subsequent to conditions are met simultaneously:◦ Tr(Atts) = one,◦ cph∗ ← Enc(ω∗, ti, Atts)◦ st∗ ← TokenGen(sk, ω∗, [ts, te])◦ ti∈ [ts, te]Otherwise, it returns zero.

## IV. THE PROPOSED CONCRETE CONSTRUCTION OFKP-ABTKS

The detail of the development is conferred as follows

1. *(msk, pp) ← Setup (1λ):* This is often a randomized rule thatis travel by the TTP to come up with the master secret key and therefore the publicparameters. Supported the protection parameter λ, this rule selectsa additive map e: G1 ×G1 → G2, wherever G1 and G2 area unitcyclic teams of order λ-bit letter of the alphabet q. Let H1: ∗ →G1 and H2: ∗ → Zq be 2 cryptographically unidirectional hashfunctions. [6]Then, it sets the general public parameter and the master secret key as follows:

   pp:= (H1, H2, e, P, sP, srP, G1, G2)

   msk:= (s, sr) (3)

2. *skj ← KeyGen (msk, Trj ):* This rule runs Share(Trj , srs−1) as a package to allot the keyshare qn(0) to every leaf node n ∈ lvs(Trj ) with relevancy the access tree Trj [4]. For this aim, the TTP initial selects a random value t˜j∈R Zq, and computes associate = qn(0)P + t˜jH1(att(n))and Bn = t˜jsP for every leaf n ∈ lvs(Trj ). Then, the key keyskj is ready as follows:

skj := Trj , n ∈lvs(Trj ) (4)

3. *cph ← Enc(ω, ti, Atts, pp):*The information owner runs this ruleon the keyword ω, the time instance of encrypting ti, This randomized rule selects 2 random values r1, r2∈RZq,and encrypts the keyword ω in line withthe following steps:

W0 = r1r2sP
W0 = r1srP
W00 = r1H2(ω)sP + r1r2P
Wˆ = H2(ti)
∀attj∈Atts :
Wj = r1r2H1 (attj)
cph := (Atts, W0, W0, W00, W , ˆ attj∈Atts (5)

4. *st ← TokenGen(skj , ω, Tenc = [ts, te], pp):*An information user with the access tree Trj and therefore the secret key skj runs this randomized algorithm to come up with a research token for the keyword ω. For this aim, he/she selects z0 ∈R Zp, computes A0n = z0An and B0n = z0Bn for each leaf node n ∈ lvs(Trj ), and at last generates the searchtokens as follows:

l = te − tsSt(x)
= H2(ω) +lY−1j=0(x − H2(ts + j))
= (H2(ω) + a01) + a2x + · · · + alxl−1
= a1 + a2x + · · · + alxl−1
st1=nst1,j : st1,j = z0aj sP, ∀j ∈ I= o
st2 = z0srPst =: (st1,st2,Trj , n ∈ lvs(Trj )) (6)

5. *{0, 1}: = Search(st, cph):* This formula selects the most importantsubset S of the attribute set Atts satisfying the access tree Trj. IfS is empty, this formula returns 0; otherwise, acts as follows:

∀attj∈ S: En = e(A0n, W0)/e(B0n, Wj )= e(P, P)z0r1r2sqn(0)
It thought to be mentioned that we've got att (n) = attj, for n ∈lvs(Trj ).
Eroot:= Combine(Trj ,att(n) ∈ S)
= e(P, P)z0r1r2sqroot(0)
= e(P,P)z0r1r2ss−1sr
= e(P, P)z0r1r2sr(7)
Then, the cloud computes st∗as follows.
st∗ =Xlj=1Wˆ j−1st1,j(8)
Finally, this formula returns one if e(W0,st∗).Eroot =e(st2, W00), and 0, otherwise.

## V. SECURITY ANALYSIS

To providesecurity of the KP-ABTKS theme against A, our system styleshould at the same time satisfy the subsequent needs.

*A. Selective Security against Chosen Keyword Attack:* This requirement implies that the person, A, [5] cannot inferany info regarding the keyword from its cipher textin the selective security model while not being given anymatching search trapdoor. This property is formalized via by selection chosen keyword attack game.

*B. Keyword Secrecy:* This security demand implies thatthe person, [3]A cannot verify the keyword from therelated cipher text and valid search tokens with a chance overarandomkeywordguess.

*1. Security Definitions*

*a. Security Against by Selection Chosen Keyword Attack:* The Selectively chosen keyword attack (SCKA) game is in betweenthe PPT person, A, and therefore the competitor C, and contains 5Steps: Setup, Phase 1, Challenge, part two and Guess.

*a.* *Setup:* The person, A, selects the challenge attributes set,Att∗, and sends it to the competitor, C.Then, C runs the setup algorithm, (msk, pp) ← Setup(1λ). It stores the master secretkey msk,and publishes the general public parameter pp.

*b.* *Phase 1:* The person, A, is allowed to access to thefollowing oracles for polynomially over and over. At first, thechallenger C selects associate degree empty keyword list, Lω.

i. *OKeyGen(Tri):* If Tri(Att∗ ) = 1, then this oracle halts toanswer; otherwise, the competitor Cruns the key generation algorithmic rule, ski ← KeyGen(msk, Tri), and returns the secret key, skito the person, A
*ii.* *OTokenGen(Tri, Tij , ωi, pp):*
*stij* ←TokenGen(ski, Tij , ωi, pp) If Tri(Att∗ ) = 1, then the challenger, C adds ωito the list, Lω, selects the at the startempty set, Sωi, and updates Sωi by adding Tij thereto, i.e., Sωi ← Sωi ∪ Tij.
iii. *Challenge:* The person, A, outputs the tuple (ω0, ω1, t∗ ) such that if ωb ∈ Lω then t∗ cannot belong to the set Sωb wherever b ∈. Then, the competitor, C selects the random bit, b ∈R, encrypts ωb by running the coding algorithmic rule, Cb ←Enc(ωb, t∗ , Att∗ , pp), and sends Cb to A.

*3. Phase 2:* The person, A continues to question the oracles OKeyGen and OTokenGen an equivalent as part one. The sole restrictionis that the tuples (Tr, T, ω0) and (Tr, T, ω1) don't seem to be allowed to be queried to the oracle OTokenGen if Tr(Att∗ ) = one and t∗ ∈ T.

*4. Guess:*The resister, A, guesses b0as the worth of b. It winsthe game if b = b'.The advantage of the resister, A, to win the sport is outlinedas follows:
Advkp−abtks−sckaABT KS,A (1λ)
=|Pr[AOKeyGen, OTokenGen (1λ, pp) = b0:b=b0] −1/2| (9)

*5. Keyword Secrecy:* The keyword secrecy game is command between the PPT resister, A, and therefore the competitor C, and containsfour steps: Setup, Query, Challenge and Guess.

*6. Setup:* During this a part of the sport, the competitor, C runsthe algorithmic rule (pp, msk) ← setup(1λ), and sends the general publicparameter pp to the resister, A.

*7. Query:* The resister, A, is allowed to access the subsequent oracles polynomially over and over. The resister, A, by selection [4]chooses its supposed keywords or access trees and receives thevalid search tokens and secret keys, severally.

*8. Challenge:* Chooses a challenge attributes set Att∗ such that Tri(Att∗) = zero for all Tri belongsto LTr, and sends it to the competitor C. Then, the competitorC every which way selects a challenge keyword, ω∗, from the messagespace, M, a interval T∗ = [ts, te], and therefore the time instance ofencrypting, t∗∈R T∗. It additionally every which way selects the access tree, Tr∗, specified Tr∗(Att∗) = 1. Then, it runs the coding algorithmic rule, cph∗ ← Enc (ω∗, t∗, Att∗, pp) and therefore the token generation algorithm st∗ ← TokenGen(sk∗, ω∗) specified sk∗is associated to the access tree Tr∗. Finally, the competitor C sends the tuple, (cph∗,st∗), to the resister A.

*9. Guess:* The competitor, C, computes cph0 ←Enc(ω0, t∗ , Att∗ , pp), and runs the search algorithmic rule,
b:=Search(st∗ , cph0). It wins the sport if b = one. The advantageof A to win this game is outlined as follows:
Advkp−abtks−ksgKP−ABT KS,A(1λ)=
Pr[AOKeyGen,OTokenGen (pp, 1λ) = w0:cph0 ← Enc(ω0, t∗ , Att∗ , pp), one := Search(st∗ ,cph0)](12)

*10. Definition 2:* A KP-ABTKS theme provides the keywordsecrecy property, if the advantage of the PPT individual, A, towin the keyword secrecy game is at the most a negligible perform,negl(λ) wherever λ is that the security parameter:
Advkp−abtks−ksgKP−ABT KS,A(1λ) ≤ negl(λ) (13)

*B. Security Proof*

*1.Theorem 1:*The projectedKP-ABTKS theme is by selection secure against chosen keyword attack within the random oracle model.

*2. Proof:* To prove this theorem, suppose that our theme isn't secure against SCKA, thus there exists a PPT individual sort of a who wins the SCKA game with a non-negligible advantage, i.e.,Advkp−abtks−sckaKP−ABT KS,A(1λ) = (λ), wherever (λ) could be a non-negligiblefunction. Since this contradicts with the MDDH assumption. The distinguisher, D is given aMDDH instance, (G1, P, r1P, r2P, r3P, Q), where P, Q ∈R G1and r1, r2, r3 ∈R Zq, and acts as follows to simulate the SCKA game for the individual, A.

*3. Setup:* The distinguisher, D, selects s, sr ∈R Zquniformly every which way, and computes s−1. Ten, it sets msk:=(s, sr) because the master secret key. It conjointly selects a additive map, e: G1 × G1 → G2, computes sP, srP, and sets the pp:=(H1, H2, e, P, sP, srP, G1, G2)

*4. Phase 1:* The distinguisher D selects the keyword list, Lw, which is at first empty, and answers A's queries by simulating OKeyGenand OTokenGen as follows.

a. *OKeyGen(Tri):* The KeyGen algorithmic program first runs Share(Tr, srs−1) to cipher the quota of every leaf noden∈lvs(Tr), i.e., qn(0). Then, once choosing t˜ ∈R Zq, it computes associate = qn(0)P + t˜OH1(att(n)) and Bn = tsP ˜for all leaves in Tri. The ensuing secret key areski:= (Tri, n ∈lvs(Tri)). This oracle haltsto answer if Tri(Att∗) = 1.
b. *OTokenGen (Tri, Tij, ωi):* The distinguisher, D, first runs OKeyGen(Tri) to urge the key key,ski:= (Tri, n ∈ lvs(Tri)). Then, itgenerates the search token, stij, by choosing the exponent, z0 ∈R Zq, and computing A0in = z0Ain andB0in = z0Bin. After that, it computes:

lij = teij–tsij
Stij (x) = H2(ωi) + Yj∈ Tij(x − H2 (tij))
= (H2(ωi) + a0i,1) + ai,2x + · · · + ai,lij xlij−1
= ai,1 + ai,2x + · · · + ai,lij xlij−1
st1,i =nst1,ij:st1,ij = z0ai,j sP,
∀ j ∈ I = ost2,
i = z0srPstij =: (st1,i, st2,i,Tri, n ∈ lvs(Tri))(14)

*5. Challenge:* If att∗ j ∈ Att∗ was queried before, D retrieves αj from OH1and computes Wj = αjQ; otherwise, D selects the random exponent, αj ∈ R Zq, computesWj = αjQ, and adds αj to the table of OH1. Then, Dsets W' = sQ, W0 = sr(r1P), W00 = H2(ωb)s(r1P) + letter and Wˆ =H2(t∗ ). Therefore, the ensuing cipher text is going to beCb: = (Att∗ , W', W', W'', W, ˆ att∗ j ∈ Att∗ ). Then, Dreturns Cb to A. Note that if letter = r1r2r3P, then Cb could be a valid cipher text by considering r'1 = r1 and r'2 = r2r3.

*6. Phase 2:* The soul A continues to question identical asPhase 1. We tend to prompt that the sole restriction for A is that she cannot question (Tr, T, ω0) and (Tr, T, ω1) to OTokenGen.

*7. Guess:* The soul A outputs b0as a guess for the worthof b. Then, the distinguisher D checks whether or not b = b0 or not. Ifb = b0, it will notice that letter = r1r2r3P with a non-negligibleprobability; otherwise, letter could be a random component in G1.

AdvMDDHD (λ) =|Pr[D(P, r1P, r2P, r3P, r1r2r3 P: r1, r2, r3 ∈ R Zq) = 1]
− Pr[D (P, r1P, r2P, r3P, Q: r1, r2, r3 ∈ R Zq, letter ∈ R G1) = 1]|(15)
As letter is willy-nilly chosen from G1, then we've
Pr[D (P, r1P, r2P, r3P, Q: r1, r2, r3 ∈ R Zq, letter ∈ R G1) = 1] = 1 /2.
Also, we have:
Pr[D(P, r1P, r2P, r3P, r1r2r3P: r1, r2, r3 ∈ R Zq) = 1]=
|Pr[D(P, r1P, r2P, r3P, r1r2r3P) = 1|A wins] Pr[A wins]+
Pr[D(P, r1P, r2P, r3P, r1r2r3P) = 1|A wins] Pr[A win]=
1.(λ) + 12(1 − (λ)) = (λ)2+12(16)
Therefore,
AdvMDDHD (λ) = (λ)2+1/2−1/2=(λ)2(17)

## VI. PERFORMANCE EVALUATION

The KP-ABTKS theme consists of five algorithms: Setup, KeyGen, Enc, TokenGen and Search. Since the

Setupalgorithmic rule is run offline, we tend to exclude its process cost in analyzing the performance of our theme. The KeyGen algorithmic rule, |S| hash functions and 3|S| modular exponentiations in G1 are run. The Enc algorithmic rule needs to execute (4 + N) modular exponentiations in G1 and (N + 2) hash functions. In the TokenGen[6] algorithmic rule, (2|S| + l + 1) modular exponentiations in G1 and l hash functions are computed. Finally, the Searchalgorithm is executed by running 2(N + 1) pairings and l exponentiations.

TABLE I THE PERFORMANCE EVALUATION OF PROPOSED KP-ABTKS SCHEME

| Algorithm | Computational Cost | Output Length |
|---|---|---|
| KeyGen | 3\|s\|exp+\|s\|H | 2\|s\|log2\|G1\| |
| Enc | (4+N)exp+(N+2)H | (N+4)log2(G1) |
| TokenGen | 2\|s\|+l+1)exp(l+1)H | (2\|s\|+l+1)log2\|G1\| |
| Search | (2N+1)pair+iexp | - |

TABLE II TIME EXECUTION OF THE PROPOSED KP-ABTKS SCHEME THE VALUE OF THE INTENDED TIME UNITS IS FIXED WITH L=10

| | N=\|s\| | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 10 | 20 | 30 | 40 | 50 |
| KeyGen(ms) | 0.3010 | 3.0100 | 6.0200 | 9.0300 | 12.0400 | 15.0500 |
| Enc(ms) | 0.5030 | 1.4120 | 2.4220 | 3.4320 | 4.4220 | 5.4520 |
| TokenGen(ms) | 1.5110 | 3.3110 | 5.3110 | 7.3110 | 9.3110 | 11.3110 |
| Search(ms) | 7 | 4.3 | 8.3 | 123 | 163 | 203 |

To simulate the important state of affairs as closely as attainable, we considered AN Intel 64-bit CoreTMi7-2670QM CPU at 2:20GHz with quad-core processor as a high-computational resource and computed the execution time of core operations on it using Multiprecision number and Rational Arithmetic Cryptographic Library (MIRACL).[7]Moreover, to receive the 80-bit security level, associate elliptic curve cryptosystem with 160-bit key length is required. Therefore, we have a tendency to set log2 |G1| = 160 bits, and log2 |G2| = 320 bits.

## VII.CONCLUSION

Securing cloud storage is a very important drawback in cloud computing. This researchaddressed the issue and introduced the notion of key-policy attribute-based temporary keyword search (KPABTKS). According to this notion, every data user will generate a search token that is valid just for a restricted time interval.[10]. We tend to plan the primary concrete construction for this new cryptographically primitive bilinear map. We tend to formally showed that our theme is provably secure within the random oracle model. The complexness of encryption algorithm of our proposal is linear with respect to the amount of the involved attributes.

## REFERENCES

[1] Yong Yu, Jianbing Ni, Haomiao Yang, Yi Mu, and Willy Susilo, “Security and Communication Networks”, *Security Comm. Networks*, No.7, pp. 466–472, DOI: 10.1002/sec.790, 2014.

[2] Mohammad Hassan Ameri, MahshidDelavar, JavadMohajeri, and Mahmoud Salmasizadeh, “A Key-Policy Attribute-Based Temporary Keyword Search Scheme for Secure Cloud Storage”, *DOI: 10.1109/TCC.2018.2825983, IEEE*, 12 April 2018.

[3] K. Liang and W. Susilo, “Searchable attribute-based mechanism with efficient data sharing for secure cloud storage,”*IEEE Transactions on Information Forensics and Security*, No. 9, pp. 1981–1992, 2015.

[4] J. Han, W. Susilo, Y. Mu, and J. Yan, “Attribute-based data transfer with filtering scheme in cloud computing,” *The Computer Journal*, Vol. 57, No. 4, pp. 579–591, 2014.

[5] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proceedings of the 13th ACM conference on Computer and communications security. Acm*, Vol. 89–98, 2006

[6] W. Sun, S. Yu, W. Lou, Y.T. Hou, and H. Li, “Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 27, No. 4, pp. 1187–1198, 2016.

[7] Shamus, “Multiprecision integer and rational arithmetic c/c++ library (miracl),”*MIRACL, 2014*. [Online] Available at: http://www.certivox.com/miracl/miracl-download/.

[8] Sneha R. Ghorpade, and S.N. Kini “Dual Server Hybrid Key Encryption with Multi Keyword Search”, *an ISO 3297: 2007 Certified Organization*, Vol. 5, No. 6, June 2017.

[9] H. Deng, Q. Wu, B. Qin, J. Domingo-Ferrer, L. Zhang, J. Liu, and W. Shi, “Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts,” *Information Sciences*, Vol. 275, pp. 370–384, 2014.

[10] Y. Yu, J. Ni, H. Yang, Y. Mu, and W. Susilo, “Efficient public key encryption with revocable keyword search,”*Security and Communication Networks*,Vol. 7, No. 2, pp. 466–472, 2014.

[11] E.J. Goh*et al*., “Secure indexes”, *IACR Cryptology e-Print Archive*, Vol. 216, 2003.

[12] S.T. Hsu, C.C. Yang, and M.S. Hwang, “A study of public key encryption with keyword search”, *IJ Network Security*, Vol. 15, No. 2, pp. 71–79, 2013.