# An Efficient Attribute-Based Hybrid Encryption for Multi-Authority Access Control System

[1] S.Srinath, [2] P.Kishore, [3]K.R.Sarumathi

[1, 2] UG Scholar, Dept. of CSE, Sri Krishna College of Technology, Coimbatore.

[3]Assistant Professor, Dept. of CSE, Sri Krishna College of Technology, Coimbatore.

***ABSTRACT***-Cloud multiplying , as an incipient multiplying paradigm, supports users to slightly store their data in a cloud, so as to enjoy amenities on-demand. With rapid progress of cloud multiplying , more and more enterprises will farm out their delicate data for sharing in a cloud. To keep the shared data trustworthy against un trusted cloud service providers (CSPs), a natural way is to store only the scrambled data in a cloud. Attribute-based encryption has been proposed to protect the cloud stowage. In ABE earlier works to decide single-point bottleneck problem, several authorities separately maintain disjoint attribute subsets and its refuge and enactment still not resolved. In this work extended inception Multi Authority Scheme with hybrid encryption with certify able entrustment scheme are used to express the resilient form of authority access control. Combined certifiable reckoning and encrypt-then-MAC mechanism with our TMACS hybrid encryption, could delegate the certifiable partial decryption paradigm to the cloud server. An competent method to share and protect the trustworthy information between users with limited power and data owners with vast amount of data in the cloud. Combining the traditional multi-authority scheme with TMACS, we also construct a hybrid scheme that is more suitable for the real state, in which attributes come from different authority-sets and several authorities in an authority-set jointly maintain a subset of the whole attribute set.

## I. INTRODUCTION

Cloud multiplying is the use of multiplying assets (hardware and software) that are provided as a service over a network (typically theInternet). The name comes from the use of a cloud-shaped symbol as annotion for the complex setup it contains in system diagrams. Cloud multiplying delegates remote amenities with a user's data, software and reckoning.

There are many types of public cloud multiplying

- Infrastructure as a service (IaaS)
- Platform as a service (PaaS)
- Software as a service (SaaS)
- Stowage as a service (STaaS)
- Security as a service (SECaaS)
- Data as a service (DaaS)
- Test environment as a service (TEaaS)
- Desktop as a service (DaaS)
- API as a service (APIaaS)

## Cloud Stowage:

**Cloud stowage** is a model of schmoozed online stowage where data is stored in virtualized pools of stowage which are generally hosted by third parties. Compering companies maneuver large data centers, and people who require their data to be compered buy or lease stowage capacity from them. The data center operators, in the background, virtualized the assets according to the necessities of the customer and expose them as stowage pools, which the customers can themselves use to store files or data objects. Substantially, the resource may duration across several servers.

## Cloud Privacy:

The cloud model has been censured by privacy advocates for the greater ease in which the companies compering the cloud amenities control, thus, can monitor at will, legitimately or illegitimately, the communication and data stored between the user and the host company. Illustrations such as the furtive NSA program, working with AT&T, and Verizon, which chronicled over 10 million phone calls between American citizens, causes ambiguity among privacy advocates, and the greater supremacies it gives to telecommunication companies to monitor user bustle. Using a cloud service provider (CSP) can obfuscate privacy of data because of the scope to which virtualization for cloud dispensation (virtual machines) and cloud stowage are used to contrivance cloud service. The point is that CSP operations, customer or tenant data may not endure on the same system, or in the same data center or even within the same provider's cloud. This can lead to legitimate concerns over prerogative. While there have been efforts (such as US-EU Safe Harbor) to "harmonise" the legitimate environment, providers such as Amazon still gratify to major markets (typically the United States and the European Union) by arraying local infrastructure and allowing customers to select "obtainability zones." Cloud multiplying pretenses privacy concerns because the service provider may access the data that is on the cloud at any point in time. They could unintentionally or intentionally alter or even obliteratestatistics. Postage and delivery amenities company, Pitney Bowes propelled Volly, a cloud-based, digital mailbox facility to clout its communication management assets. They also faced the technical defy of on condition that strong data retreat and privacy. However, they were able to address the same concern by smearing adapted, application-level security, comprising encryption.

**Cloud multiplying techniques** are used to stake as sets. It conferrals the application software and databases to the centralized large data centers .Internet-based online amenities do afford huge amounts of stowage space and customizable multiplying assets, this multiplying platform shift, however, is jettisoning the accountability of local machines for data maintenance at the same time. Users are at the clemency of their cloud service providers (CSP) for the obtain ability and veracity of their data. Cloud stowage supports users to slightly store their data and enjoy the on-demand high eminence cloud applications without the encumbrance of local hardware and software management. Though the settlements are clear, such a service is also abandoning users' physical tenure of their farm out data, which inexorably poses new security menaces toward the exactitude of the data in cloud.

Several trends are opening up the era of Cloud Multiplying , which is an Internet-based progress and use of computer technology. The ever cheaper and more powerful workstations, together with the "software as a service" (SaaS) multiplying architecture, are transforming data centres into pools of multiplying service on a huge scale .In the interim, the increasing network bandwidth and steadfast yet pliable network connections make it even doable that clients can now subscribe high-quality amenities from data and software that dwell solely on remote data centers .Cloud multiplying , as an promising multiplying paradigm, supports users to remotely store their data into a cloud so as to enjoy scalable amenities on-demand. Principally for small and medium-sized enterprises with limited budgets, they can achieve cost savings and efficiency enhancements by using cloud-based

amenities to manage projects, to make collaborations, and the like. However, allowing cloud service providers (CSPs), which are not in the same trusted domains as enterprise users, to take care of trustworthy data, may raise potential security and privacy issues. To keep the sensitive user data trustworthy against untrusted CSPs, a natural way is to apply cryptographic approaches, by disclosing decryption keys only to authorized users. However, when enterprise users outsource trustworthy data for sharing on cloud servers, the assume encryption system should not only support fine-grained access control, but also provide high performance, full delegation, and scalability, so as to best serve the needs of accessing data anytime and anywhere, delegating within enterprises, and achieving a energetic set of users.

A large-scale enterprise with a high turnover rate, a scalable revocation scheme is a must. That is, the enterprise can revoke data access rights from users once they are no longer its employees. A user whose authorization is revoked will still retain the keys issued earlier, and thus can still decrypt data in the cloud. The traditional revocation scheme usually requires the AAs to sporadically re-encrypt data, and re-generate new secret keys to remaining authorized users. This approach will cause heavy workload on the AAs. A more scalable approach is to take lead of the abundant assets in a cloud by allowing the AAs to delegate the CSP to re-encrypt data and re-generate keys to users, under the environment that the CSP knows nothing about the data and keys.

**II.RELATED WORK**

***1. J.Bethencourt, a. Sahai, and b. Waters, "ciphertext-policy attribute-based encryption," proc. Ieeesymp. Security and privacy, 2007***

In this work, we provide the first construction of a ciphertext-policy attribute-based encryption (cp-abe) to address this problem, and give the first construction of such a scheme. In our system, a user's private key will be associated with an arbitrary number of attributes expressed as strings. On the other hand, when a party encrypts a message in our system, they specify an associated access structure over attributes. A user will only be able to decrypt a ciphertext if that user's attributes pass through the ciphertext's access structure. At a mathematical level, access structures in our system are described by a monotonic "access tree", where nodes of the access structure are composed of threshold gates and the leaves describe attributes. We note that and gates can be constructed as n-of-n threshold gates and or gates as 1-of-n threshold gates. Furthermore, we can handle more complex access controls such as numeric ranges by converting them to small access trees . At a high level, our work is similar to the recent work on key-policy attribute based encryption (kp-abe), however we require substantially new techniques. In key-policy attribute based encryption, ciphertexts are associated with sets of descriptive attributes, and users' keys are associated with policies (the reverse of our situation). We stress that in key-Policy abe, the encryptor exerts no control over who has access to the data she encrypts, except by her choice of descriptive attributes for the data. Rather, she must trust that the key-issuer issues the appropriate keys to grant or deny access to the appropriate users. In other words, the "intelligence" is assumed to be with the key issuer, and not the encryptor. In our setting, the encryptor must be able to intelligently decide who should or should not have access to the data

That she encrypts. As such, the techniques of do not apply to our setting, and we must develop new techniques.

***2. A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Proc. 24th Annu. Int. Conf. Theory Appl. Cryptographic Techn., 2005, pp. 457–473.***

a new type of Identity-Based Encryption (IBE) scheme that we call Fuzzy Identity-Based Encryption. In Fuzzy IBE we view an identity as set of descriptive attributes. A Fuzzy IBE scheme allows for a private key for an identity, !, to decrypt a ciphertext encrypted with an identity, !0, if and only if the identities ! and !0 are close to each other as measured by the "set overlap" distance metric. A Fuzzy IBE scheme can be applied to enable encryption using biometric inputs as identities; the error-tolerance property of a Fuzzy IBE scheme is precisely what allows for the use of biometric identities, which inherently will have some noise each time they are sampled. Additionally, we show that Fuzzy-IBE can be used for a type of application that

we term "attribute-based encryption".In this paper we present two constructions of Fuzzy IBE schemes. Our constructions can be viewed as an Identity-Based Encryption of a message under several attributes that compose a (fuzzy) identity. Our IBE schemes are both error-tolerant and secure against collusion attacks. Additionally, our basic construction does not use random oracles. We prove the security of our schemes under the Selective-ID security model. Fuzzy Identity-Based Encryption and provide a construction for a Fuzzy Identity-Based Encryption scheme. Our construction uses groups for which an efficient bilinear map exists, but for which the Computational Diffie-Hellman problem is assumed to be hard. Our primary technique is that we construct a user's private key as a set of private key components, one for each attribute in the user's identity. We share use Shamir's method of secret sharing to distribute shares of a master secret in the exponents of the user's private key components. Shamir's secret sharing within the exponent gives our scheme the crucial property of being error-tolerant since only a subset of the private key components are needed to decrypt a message. Additionally, our scheme is resistant to collusion attacks. Different users have their private key components generated with different random polynomials. If several users collude they will be unable to combine their private key components in any useful way. In the first version of our scheme, the public key size grows linearly with the number of potential attributes in the universe. The public parameter growth is manageable for a biometric system where all the possible attributes are defined at the system creation time. However, this becomes a limitation in a more general system where we might like an attribute to be defined by an arbitrary string. To accommodate these more general necessities we additionally provide a Fuzzy-IBE system for large universes, where attributes are defined by arbitrary strings. We prove our scheme secure under an adapted version of the Selective-ID security model first proposed. Additionally, our construction does not use random oracles. We reduce the security of our scheme to an assumption that is similar to the Decisional Bilinear Diffie- Hellman assumption.

***3. B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in Proc. 14th Int. Conf. Practice Theory Public Key Cryptography, 2011, pp. 53–70.***

A new methodology for realizing Ciphertext-Policy Attribute Encryption (CP- ABE) under concrete and noninteractive cryptographic assumptions in the standard model. Our solutions allow any encryptor to specify access control in terms of any access formula over the attributes in the system. In our most efficient system, ciphertext size, encryption, and decryption time scales linearly with the complexity of the access formula. The only previous work to achieve these parameters was limited to a proof in the generic group model.We present three constructions within our framework. Our first system is proven selectively secure under a assumption that we call the decisional Parallel Bilinear Diffe-Hellman Exponent(PBDHE) assumption which can be viewed as a generalization of the BDHE assumption. Our next two constructions provide performance tradeoffs to achieve provable security respectively under the (weaker) decisional Bilinear-Diffe-Hellman Exponent and decisional Bilinear Diffe- Hellman assumptions. a new methodology for realizing Ciphertext-Policy ABE systems from a general set of access structures in the standard model under concrete and non-interactive assumptions. Both the ciphertext overhead and encryption time scale with O(n) where n is the size of the formula. In addition, decryption time scales with the number of nodes.Our first system allows an encryption algorithm to specify an access formula in terms of any access formula. In fact our techniques are slightly more general. We express access control by a Linear Secret Sharing Scheme (LSSS) matrix M over the attributes in the system. Previously used structures such as formulas (equivalently tree structures) can be expressed succinctly [6] in terms of a LSSS. We do not lose any efficiency by using the more general LSSS representation as opposed to the previously used tree access structure descriptions. Thus, we achieve the same performance and functionality as the Bethencourt, Sahai, and Waters construction, but under the standard model. In addition, we provide two other constructions that tradeo some performance parameters for provable security

under the respective weaker assumptions of decisional-Bilinear Diffe-Hellman Exponent (d-BDHE) and decisional-Bilinear Diffe-Hellman assumptions. In summarize the comparisons between our schemes and the GJPS and BSW CP-ABE systems in terms of ciphertext and key sizes and encryption and decryption times. Taken all together our first scheme realizes the same efficiency parameters as the BSW encryption scheme, but under a concrete security assumption. At the same time, our d-BDH construction is proved under the same assumption as the GJPS system and achieves significantly better performance. In our systems, the ciphertext distributes shares of a secret encryption exponent s across different attributes according to the access control LSSS matrix M.A user's private key is associated with a set S of attributes and he will be able to decrypta ciphertext his attributes \satisfy" the access matrix associated with the ciphertext. As inprevious ABE systems, the primary challenge is to prevent users from realizing collusion attacks.Our main tool to prevent this is to randomize each key with an freshly chosen exponent t. During decryption, each share will be multiplied by a factor t in the exponent. Intuitively, this factor should \bind" the components of one user's key together so that they cannot be combined with another user's key components. During decryption, the di_erent shares (in the exponent) that the algorithm combines are multiplied by a factor of t. Ultimately, these randomized shares are only useful to that one particular key

***4.V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. 13th ACM Conf. Comput. Commun. Security, 2006, pp. 89–98.***

Sensitive data is shared and stored by third-party sites on the Internet, there will be a need to encrypt data stored at these sites. One drawback of encrypting data, is that it can be selectively shared only at a coarse-grained level (i.e., giving another party your private key). We develop a new cryptosystem for fine-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE). In our cryptosystem, ciphertexts are labeled with sets of attributes and private keys are associated with access structures that control which ciphertexts a user is able to decrypt. We demonstrate the applicability of our construction to sharing of audit-log information and broadcast encryption. Our construction supports delegation of private keys which subsumes Hierarchical Identity-Based Encryption (HIBE). a much richer type of attribute-based encryption cryptosystem and demonstrate its applications. In our system each ciphertext is labeled by the encryptor with a set of descriptive attributes. Each private key is associated with an access structure that specifies which type of ciphertexts the key can decrypt. We call such a scheme a Key-Policy Attribute-Based Encryption (KP-ABE), since the access structure is specified in the private key, while the ciphertexts are simply labeled with a set of descriptive attributes.We note that this setting is reminiscent of secret sharing schemes. Using known techniques one can build a secret-sharing scheme that specifies that a set of parties must cooperate in order to reconstruct a secret. For example, one can specify a tree access structure where the interior nodes consist of AND and OR gates and the leaves consist of different parties. Any set of parties that satisfy the tree can reconstruct the secret. In our construction each user's key is associated with a tree-access structure where the leaves are associated with attributes.2 A user is able to decrypt a ciphertext if the attributes associated with a ciphertext satisfy the key's access structure. The primary difference between our setting and secret-sharing schemes is that while secret-sharing schemes allow for cooperation between different parties, in our setting, this is expressly forbidden. For instance, if Alice has the key associated with the access structure \X AND Y", and Bob has the key associated with the access structure \Y AND Z", we would not want them to be able to decrypt a ciphertext whose only attribute is Y by colluding. To do this, we adapt and generalize the techniques introduced by to deal with more complex settings. Will show that this cryptosystem gives us a powerful tool for encryption with fine-grained access control for applications such as sharing audit log information. In addition, we provide a delegation mechanism for our construction. Roughly, this allows any user that has a key for access structure X to derive a key for access structure Y, if and only if Y is more restrictive than

X. Somewhat surprisingly, we observe that our construction with the delegation property subsumes Hierarchical Identity-Based Encryption In SSS, one can specify a tree-access structure where the interior nodes consist of AND and OR gates and the leaves consist of different parties. Any set of parties that satisfy the tree can come together and reconstruct the secret. Therefore in SSS, collusion among different users (or parties) is not only allowed but required. In our construction each user's key is associated with a tree-access structure where the leaves are associated with attributes. A user is able to decrypt a ciphertext if the attributes associated with a ciphertext satisfy the key's access structure. In our scheme, contrary to SSS, users should be *unable* to collude in any meaningful way.

***5. R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in Proc. 14th ACM Conf. Comput. Commun. Security, 2014, pp. 195–203.***

An Attribute-Based Encryption (ABE) scheme that allows a user's private key to be expressed in terms of any access formula over attributes. Previous ABE schemes were limited to expressing only monotonic access structures. We provide a proof of security for our scheme based on the Decisional Bilinear Diffie-Hellman (BDH) assumption. Furthermore, the performance of our new scheme compares favorably with existing, less-expressive schemes In this work we present a new Attribute-Based Encryption scheme where private keys can represent any access formula over attributes, comprising non-monotone ones. In particular, our construction can handle any access structure that can be represented by a Boolean formula involving AND, OR, NOT, and threshold operations. As mentioned above, the main technical obstacle we overcome is finding a way to make use of secret sharing schemes to yield non-monotonic access structures. At a high level, the technical novelty in our work lies in finding a way to (implicitly) make a share "available" to the decryptor only if a given attribute is not present among the attributes of the ciphertext. To accomplish this we adapt an idea from the broadcast revocation scheme of Naor and Pinkas to our setting of Attribute-Based Encryption based on bilinear groups. Every negative attribute node in a key is tied to a degree d polynomial (in the exponent) that was created by the authority at setup (where d is the maximum number of attributes used to describe a ciphertext). To access the secret share corresponding to this node, the decryptor will need to make use of at least d+1 different points from the polynomial in order to perform an interpolation, where we map attributes to distinct points on the polynomial. The decryption algorithm will be able to gather d different points of the polynomial from the attributes of the ciphertext. To get the remaining point, the decryptor must examine the one point that corresponds to the negative attribute in this particular node of the access formula.

If this attribute is distinct from all the attributes in the ciphertext — that is, if the attribute is not present — then the decryptor will have d+1 points of the polynomial and be able to decrypt; otherwise, if the key's attribute appears in the ciphertext, then the decryption algorithm will have only points (one particular point will have been given twice) and the decryption algorithm willnot be able to interpolate the polynomial and thereby access the secret share corresponding to the node. In designing our construction several challenges arise from adapting these negation techniques while preserving the collusion resistance features that are necessary for Attribute-Based Encryption systems.

## III.AN EFFICIENT ATTRIBUTE-BASED HYBRID ENCRYPTION FOR MULTI-AUTHORITY ACCESS CONTROL SYSTEM:

Secure access control problem has become a critical taxing issue in public cloud stowage, in which traditional security technologies cannot be directly applied

The problem of single-point bottleneck on both security and concert .Decisional Bilinear Diffie-Hellman Exponent Assumption.It is a discrete logarithm problem to calculate the master key.To make ABE satisfy the state of affairs where attributes come from several authorities has been proposed as an open problem.Multi-authority access

control schemes, the whole attribute set is divided into several disjoint subsets and maintained by several establishment, but each attribute subset is still preserved by only one authority, which makes the problem of single-point bottleneck on both security and recital.

## IV.IMPLEMENTATION AND RESULTS

### A.Developing a cloud environment

Initially the basic network model for the cloud data stowage is developed in this module. Four different network entities can be identified as follows: Client(Data Owner): an entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations; Cloud Stowage Server (CSS): an entity, which is managed by Cloud Service Provider (CSP), has significant stowage space and computation resource to maintain the clients' data; Certificate Authority: an entity, which has proficiency and capabilities that clients do not have, is trusted to assess and expose risk of cloud stowage facilities on behalf of the clients upon request. In the cloud paradigm, by putting the large data files on the remote servers, the clients can be relieved of the burden of stowage and computation; Public User: The one who access the cloud data which is the private data of cloud data owners. The public data is stored in the cloud by data owners for business purposes it can be accessed by any user for their needs.

### B.Proposes TMACS

Threshold secret sharing, based on redundant several AAs, then propose a threshold multi-authority CPABE and the relevant access control scheme TMACS in public cloud stowage TMACS, several establishment jointly manage the whole attribute set but no one has full control of any specific attribute. In TMACS, a global certificate authority is responsible for the construction of the system, which avoids the extra overhead caused by AAs' conciliation of system parameters. CA is also responsible for the muster of users, which avoids AAs synchronized maintaining a list of users. However, CA is not involved in AAs' master key sharing and users' secret key generation, which avoids CA becoming the security vulnerability and performance bottleneck. In TMACS, AAs must first register to CA to gain the corresponding identity and credential (aid, aid.cert). Then AAs will be involved in the construction of the system, supporting CA to finish the establishment of system parameters. CA accepts users' registration and issues the certificate (uid, uid.cert) to each legitimate user. With the certificate, the user can contract with any t AAs one by one to gain his/her secret key. Owners who want share their data in the cloud can gain the public key from CA. Then the owner can encrypt his/her data under predefined access policy and upload the cipher text (CT) to the cloud server. User can freely download the cipher texts that he/she is interested in from the cloud server. However, he/she can't decrypt the cipher text unless his/her attributes satisfy the access procedure unseen inside the cipher texts. to guarantee the flexibility of the system in users' secret key generation is another exigent issue. In traditional (t; n) threshold secret sharing, the secret can be reconstructed unless there are at least t participants cooperating with each other. This means that, if just simply introducing traditional (t; n) threshold secret sharing into our multi-authority CP-ABE design, the user should contact with t AAs during the secret key generation for each time, and the chosen t AAs also have to contact with each other to unconditionally reconstruct the master key. This will bring too much communication overhead, which is not stretchy for system performing. To reduce the minor communication overhead, in TMACS, rather than the master key, the entire secret key is reconstructed by collecting t secret key shares generated by AAs. Furthermore, the recreated process can be done by the user rather than the specific t AAs. By this means, the user can contact with the t AAs one by one, which is suit for real application scenarios, augments the flexibility of the system, avoids the extra communication overhead and synchronization issues among AAs.

**C.Data Access Control Scheme**

System Initialization is divided into three sub-processes: CASetup1, AASetup, and CASetup2. The operation of CASetup1 is mainly responsible for establishment of system parameters and accepting listing of users and AAs. AAs cooperate with each other to share the master key in AASetup, while the analogous public key is generated by CA in CASetup2. The operation of CASetup1 is run by CA. First, CA chooses two multiplicative cyclic groups G and GT with the same prime order p, then defines a binary map e : G x G -> GT on G. CA chooses a haphazard a € Zp as the master key, and then calculates the germane public key part $g^a$. Here, the parameter g is a generator of G. CA engenders a pair of keys ($sk_{CA}$ ,$vk_{CA}$) to sign and verify, in which, $vk_{CA}$ is publicly known by each entity in the system.The operation of AASetup is run by each one of all n AAs. These n AAs cooperate with each other to call (t; n) verge secret sharing. After finishing the operation of AASetup, each AA (AAi, i = 1, 2 . . . n) gains a pair of keys (ski, pki). Here, the public key share pki can be shared with any other entities, comprising CA. The operation of CASetup2 is run by CA. To calculate the global public key, CA randomly chooses t out of n AAs' public key shares, denoted as pki, i = 1,2 . . . t.

**D.Operation of Encryption and Decryption**

The operation of Encryption is implemented by a specific data owner independently. To recover the system's performance, the owner first chooses a random number k €Zp as the symmetric key and encrypts the plaintext message M using k with the symmetric encryption algorithm, such as AES. The encrypted data can be denoted as Ek(M), then the owner encrypts the symmetric key k using CP-ABE under an access policy defined by himself/herself. The owner first identifys an easy expressed monotone boolean formula. By following the method defined in [26], he/she can turn it to a LSSS access structure, which can be denoted as (M, r).Mis a l _ k matrix, where l is the scale of a specific attribute set and k is variable that is depend on the monotone boolean formula definition and the LSSS turning method. The function r maps each row of M to a specific attribute, marked as P(i)(€ {Att1; Att2; . . . ; $Att_U$}). A haphazard secret parameter s is chosen to encrypt the symmetric key k. To hide the parameter s, a random vector v = (s, y2, y3. . . . ; yk) € $Z^n_p$ is selected, where y2, y3; . . . , yk are randomly chosen and used to share the perameter s.

The Secret Key Generation operation is run by one user and any t out of n AAs. Less than t AAs, user's secret key cannot be generated. In this operation, there is no interaction between any two of t AAs, so the user can select t AAs according to his/her own preference, and then separately contact with each of these t AAs to get the secret key share. After getting t secret key shares separately from t AAs, the user can generate his/her secret key. To gain the secret key share from AAi, the user uidj first sends his/her signed request comprising his/her identity and his/her certificate to AAi. After receiving the request, AAi verifies uidj's certificate by using CA's public verification key vkCA, then authenticates the user by verifying his/ her signature over the request. If the user is an illegitimate one, the operation aborts. Otherwise, AAi assigns an attribute set S to the user according to the role he/she plays in the domain1 and engenders the secret key share for him/her.

The Decryption operation is run by each user. The user can freely query and download any encrypted data that he/she is interested in from the cloud server. However, he/she can't decrypted the data unless his/her attribute set satisfies the right to use structure hidden inside the cipher text.

## V.CONCLUSION:

In this work extended threshold Multi Authority method with hybrid encryption with certifiable delegation scheme are used to express the burly form of authority access control. Combined certifiable computation and encrypt-then-MAC mechanism with our TMACS hybrid encryption, could delegate the certifiable partial decryption paradigm to the cloud server. TMACS, in public cloud

stowage, in which all AAs jointly manage the whole attribute set and share the master key a. Taking advantage of (t, n) threshold secret sharing, by interacting with any TAAS, a legitimate user can spawn his/her secret key. Thus, TMACS avoids any one AA being a single-point bottle neck on both security and routine.

## *REFERENCES:*


1. J. Bethencourt, a. Sahai, and b. Waters, "ciphertext-policy attribute-based encryption," proc. Ieeesymp. Security and privacy, 2007
2. A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Proc. 24th Annu. Int. Conf. Theory Appl. Cryptographic Techn., 2005, pp. 457–473.
3. B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in Proc. 14th Int. Conf. Practice Theory Public Key Cryptography, 2011, pp. 53–70.
4. V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. 13th ACM Conf. Comput. Commun. Security, 2006, pp. 89–98.
5. R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in Proc. 14th ACM Conf. Comput. Commun. Security, 2014, pp. 195–203.
6. M. Chase, "Multi-authority attribute based encryption," in Proc. 4th Theory Cryptography Conf., 2007, pp. 515–534.
7. A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in Proc. 30th Annu. Int. Conf. Theory Appl. Cryptographic Techn., 2011, pp. 568–588.
8. H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multiauthority attribute based encryption without a central authority," Inf. Sci., vol. 180, no. 13, pp. 2618–2632, 2010.
9. T. Pedersen, "A threshold cryptosystem without a trusted party," in Proc. 10th Annu. Int. Conf. Theory Appl. Cryptographic Techn., 1991, pp. 522–526.
10. A. Shamir, "How to share a secret," Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979.