ISSN 2277-3061
Volume 15 Number 8
International Journal of Computers & Technology

Role-Attribute-Based Encryption (RABE) Access Control for Healthcare Cloud Systems

Monica Suleiman[1], Paolina Centonze[2]
1 Research Scholar, Department of Computer Science,
Iona College 715 North Avenue, New Rochelle NY 10801
msuleiman1@gaels.iona.edu
2 Professor, Department of Computer Science,
Iona College 715 North Avenue, New Rochelle NY 10801
pcentonze@iona.edu

ABSTRACT

In the medical industry, it is critical to ensure the confidentiality of patients' personal health records when storing and managing them. Before cloud computing surfaced, heath providers used local servers and hard drives to store their records and data. As cloud computing has been becoming more prominent many healthcare providers are using the cloud to store and manage their sensitive data. This journal compares and investigates two different access control models, in particular Role-Based Access Control and Attribute-Based Access Control, to validate the confidentiality of data when storing and managing personal health records on cloud services. The comparative analysis of these access control models is done to identify possible inefficiency and privacy restrictions in these two access control based models. In addition, in this journal we propose a new access control model, which we refer to as Role-Attribute-Based-Encryption Access Control (RABE), by combining some of the best aspects of both RBAC and ABAC in order to improve data privacy on cloud systems used in healthcare.

KEYWORDS:

Cloud Computing, Personal Health Records, Confidentiality, Role-Based Access Control (RBAC), Role-Based Encryption, Attribute-Based Access Control (ABAC), Attribute-Based Encryption, Role-Attribute-Based Encryption (RABE).

ACADEMEIC DISCIPLINE AND SUB-DISCIPLINES

Computer Science; Cloud Computing, Cryptgraphy.

SUBJECT CLASSIFICATION

Access Control.

TYPE (METHOD/APPROACH)

Theoretical Analysis.

1. INTRODUCTION

When managing and storing personal health records, it is very important to ensure the patients that their records are secure and confidential. In today's world, it is becoming very popular to use cloud computing services to store and manage data. In fact, 83% of healthcare providers are using the cloud to manage and store their medical data, and 9% intend to start to use the cloud in the future. [11] As a great percent of the healthcare industry is using cloud computing, the question that arises is how secure and confidential cloud computing actually is when storing and managing personal health. In the past year the number of health records breached increased drastically, which raises the question of the privacy of data being stored. Being that these records are so sensitive and personal, it is crucial to improve the privacy of data, such as personal health records, when being stored on these cloud computing services. In this journal, we will improve the confidentiality of of personal health records stored on cloud services by investigating and comparing the use of Role-Based Access Control and Attribute Based-Access Control. We will be proposing a new encryption scheme that combines the optimal concepts of Role-Based and Attribute-Based access control that validates the confidentiality of healthcare records being stored on cloud services.

2. BACKGROUND DEFINITIONS

2.1 Personal Health Record

A personal health record is a record of medical data or information pertaining to a particular individual that is managed and maintained on a system, in this case a cloud system. This system is a place where the individual can store and manage their health data wherever and whenever they wish as long as there is a connection to the internet present.

2.2 Cloud Computing

Cloud computing is the practice of storing, managing, and processing data on a network of remote servers hosted on the internet, instead of locally on servers or hard drives. Cloud computing has unlimited storage, capacity and scalability, as well as back up and recovery systems. It allows you to access your data anywhere with an internet connection. However, security and privacy is a big concern when managing confidential data. [8]

2.3 Data Confidentiality/Data Privacy

To ensure confidentiality and private, a system must be secure to protect sensitive data from being exposed to an unauthorized user. This is extremely important when data is stored on the cloud because the data owner is not aware of where their data is actually being stored and of who is able to access their personal data. This leaves the concerns of how confidential their data actually is. [8]

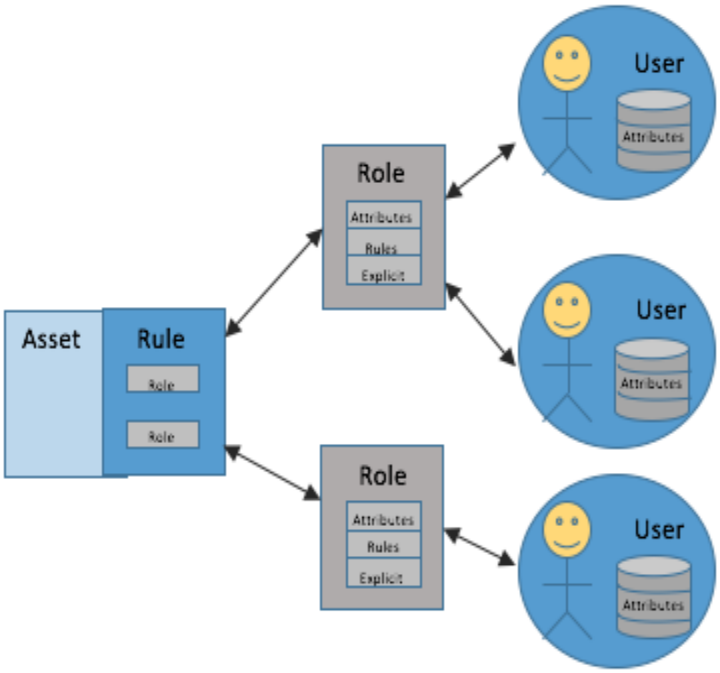
2.4 Access Control

Access control is the technique to ensure security in a system. To do this, a system either grants or revokes permissions or privileges to access some data resource. By doing this, it prevents unauthorized users to obtain access to sensitive data through authorization and authentication. Within the cloud, access control is needed to help keep data confidential and secure.

2.5 Role-Based Access Control

Role-Based Access Control(RBAC) is a model of access control in which privileges are granted to the appropriate roles in a system. Users are then assigned to the role that fits their responsibilities, making the management system of the permissions much easier to distribute. As shown in Figure 1, there is a user who wants to gain access to a certain asset. There are certain rules, or permissions, that controls who can access that data asset. These permissions are granted to roles. The roles are mapped to users in one of three ways. A role can be linked to a user based on attributes, rules, or it can be explicit to that user. Once It is mapped which particular role, the role gets mapped to the user and the user's attributes. By using roles, the relationship between users and access permissions is much simpler to manage. [3]

Figure N.1: Relationship between users, roles and permissions in Role-Based Access Control



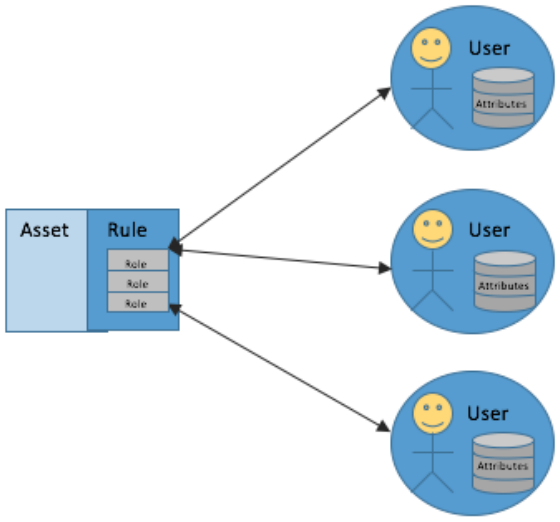
2.6 Role-Based Encryption

In Role-Based Encryption, the owners of the data can encrypt their data corresponding to the existing role-based policies for access control. This means that the data can only be accessed and decrypted by users who were granted the proper permissions or privileges according to their assigned role. The data owner can also revoke privileges, which takes away the access to the encrypted data, without affecting the other users of the same role in the system. [3,4]

2.7 Attribute-Based Access Control

Attribute-Based Access Control(ABAC) is a model of access control in which privileges are granted through a use of policies to users in a system. The policies are evaluated based on their on specific user attributes, such as subject attributes, resource attributes or environment and condition attributes. As show in in Figure 2, Attribute-based Access control removes the "man in the middle" and interacts the users directly with the assets. By doing this, the asset's rules can evaluate the user's attributes and determine the permissions granted. There is no need to add the role layer as the attributes are sufficient in determining who can access which asset.

Figure N.2: Relationship between users and attributes in Attribute-Based Access Control.



2.8 Attribute-Based Encryption

Attribute-Based Encryption is a relatively new way to enforce access control. In Attribute Based Encryption, a user can encrypt or decrypt data based on their attributes. In this one-to many public key based encryption process, private keys and cipher texts are dependent upon access policies or set of attributes. This means that when decrypting a cipher text, the user's private key and the cipher text must match according to the user's attributes, otherwise it will not decrypt the cipher text. [2]

3. RELATED WORK

Many researchers have investigated the topics of role-based access control and attribute-based access control, but not much research ahs been done regarding healthcare. In 2015, Phyu Hnin et. al. [3] are one of the few who investigated fine-grained access control of healthcare data in the cloud by using Policy-Based and Attribute-Based access control on a XACML platform. Their system allowed staff as well patients to access medical records to add, delete, and modify them. The architecture proposed does need to be improved to adapt to a dynamic setting that is constantly changing by allowing dynamic access control policies, which they intend to do in future work. In 2011, Lan Zhou et. al. [1] did research on the storing data in the cloud securely by using Role-Based Access control. They proposed a Role-Based Encryption(RBE) scheme to secure data when storing in the cloud. In their scheme, the data owner stores encrypted data into the cloud and to access the data to specific user based on their role in the system, as they would have the appropriate permissions and corresponding private key to do so. [1] In 2007, John Benthencourt et. al. [2] proposed a similar system, but use a system with a more complex scheme. They used what they called to as Ciphertext-Policy Attribute-Based Encryption (CP-ABE), which uses ABAC to define a user's private keys based on their user attributes and establishes decryption policies over the attributes or establish access trees. In ABE, when data is encrypted, the data owner identifies an access tree that contains the appropriate attributes a user must possess to have access to the data and links it to the ciphertext, illustrating the concept of ciphertext-policies. The user's private key and attributes must match the ciphertext policy and access tree in order to decrypt and access data.[2] In 2006, Vipul Goyal et al, introduced another ABAC system they referred to as Key-Policy Attribute Based Encryption(KP-ABE) which differs from CP-ABE in that cipher texts associated with attributes and users secret keys are associated with policies. [4] Since ABAC has been introduced, the complex algorithms compete with the algorithms used in original RBAC, and provide a more intricate way to control private data. However, computationally, RBAC is more optimal.

In 2010, D. Richard Kuhn, et. al. proposed that it would indeed be beneficial to combine RBAC and ABAC because user provisioning be simplified by using a role structure and permissions would be distributed based on the intersect between the set of permissions assigned to a role and the permission needed to access a particular resource. These researchers determine that by establishing a role structure it can reduce the number of roles and rules used if only using one of the two access control methods. This proves that it is possible to combine the strengths of RBAC and ABAC to establish an approach that benefits access control in a system. It also proves that by combining the two, it simplifies managing users in a system while still distributing permissions based on user's attribuets in a system. However, these researchers did not assess how expensive it is to use one type of access control over the other. In addition, Kuhn et. al. did not evaluate the cryptographic perspective to improve the methods of access control in a system. [5]

Our Contribution. In this journal we propose a new Role-Attribute-Based Encryption Control (RABE) model of access control which combines the optimal characteristics of both RBAC and ABAC to improve data privacy in healthcare systems in cloud computing. In the RABE scheme, there is a role based structure that improves user provisioning while implementing an encryption scheme where the cipher texts are associated with policies and the users are associated with attributes. Through analyzing the effectiveness of Role-Based Encryption and Attribute-Based Encryption when storing

sensitive data on cloud services, we are able to make a new scheme that uses the best of both role-based and attribute-based access control techniques to improve the computation overhead for encrypting and decrypting and reduce the size of the cipher text while ensuring a secure method of access control is implemented that will keep personal health records secure on cloud based healthcare systems.

3.1 RBE

In research that has been done on the security of the cloud using role-based access control, Zhou *et al.* [1] proposed in 2011 a role based encryption scheme with five parties. One party was the set of data owners which are the ones who store their sensitive data on the cloud. The data owners can then share their data with a set of users U. These users are then assigned a role from a set of roles. The last two parties were the role manager and the group administrator. The group administrator is the one who has the ability and authority to generates the keys roles, and users in the system. The role manager distributes the roles to the appropriate users based on their qualifications. To enforce access control by using role based encryption, they defined the following algorithms.

Figure N.3 RBE Algorithms for Access Contro'l

- Setup(λ);
- CreateRole(mk, ID_R)
- CreateUser(mk, ID_u)
- AddUser(pk, pub_R, U, ID_u)
- Encrypt(pk, pub_R, M)
- Decrypt(pk, pub_R, dk, C)
- RevokeUser(pk, pub_R, U, ID_u)

Table N.1: Role-Based Access Control Algorithm Input Parameters

Inputs
λ = Security Parameter
mk = Master Key
ID_R = Role ID
ID_u = User ID
pk = Public Key
pub_R = Public Parameters of Role
U = User List
M = Message
C = Cipher Text

The group administrator uses the Setup algorithm to generate the master and public key. The group administrator also creates the roles and users by using CreateRole algorithm and the CreateUser algorithm, respectively. The role manager then distributes the roles to the users based on their qualifications by using the AddUser algorithm. The role manager also has the capability to revoke a user's role and privileges by using the RevokeUser algorithm. To encrypt sensitive data, the data owner can use the encryption algorithm to transform their data into cipher text that can be store securely on the cloud. For a user to decrypt the cipher text that they have access to in the cloud, a user can use the Decrypt algorithm. If that user is authorized and was given the proper decryption key to access that data and are assigned to a role that possesses the appropriate permissions, the user can view the data in plaintext. Otherwise, the user cannot access the data and receives an error. [1]

In the Role Based Encryption Scheme, the researchers accomplished a number of important aspects of security that should be noted for ensuring access control. One benefit from using this encryption is that there is a hierarchy of roles that are defined and a role can have successor roles. This means when confidential data is encrypted to a particular role, only the users who belong to that role and its predecessors are able to decrypt it. An example of this is if the patient wants to share her records stored on the cloud with her doctor, but not her Nurse, role based access control helps determine who is allowed access. The patient encrypts his personal health records to the role of the Doctor, but not to the Nurse. The doctor can use his private key along with the public parameters of the doctor role to access his personal health records from the cloud, however, the Nurse cannot because her public parameters from those of the doctor. If a successor role to the Doctor and Nurse role called Employee is created and the data owner encrypts their personal health record in the cloud to the Employee role, then the Doctor and Nurse can use their own private keys and have access to that role being that they are its predecessors. Another beneficial element of security that role based access control enforces is that even after the data owner encrypts his data, a user can be added to a role and be able to access the data that was already previously encrypted. When a user is revoked from the role, the user will be stripped of all of the privileges previously granted to access encrypted data for that role. This is also beneficial because the data owner does not need to encrypt their data again. Also, this means that the role manager does not have to change the private key because they are only created when a user is created. This shows that role-based access control is a less expensive way

3.2 CP-ABE

Another group of researchers Benthencourt et al. [2] studied attribute-based access control and proposed a ciphertext policy attribute based encryption scheme to address the issue of securely storing sensitive data. In this system a user will have a private key that is affiliated with a number of attributes. When the data owner encrypts sensitive data, they specify an access tree that contains attributes and link it to the cipher text. If a user wants to decrypt it, a user must have a set of attributes and the corresponding private key that matches the access tree and

attributes pertaining to that data. To enforce access control using attribute based encryption, they defined the following algorithms.

Figure N.4 ABE Algorithms for Access Control

- Setup(λ);
- Encrypt(pk, M, A)
- KeyGen(mk, S)
- Decrypt(pk, C, sk)
    - DecryptNode(C, sk, x)

Table N.2: Attribute-Based Access Control Algorithm Input Parameters

Inputs
λ = Security Parameter
mk = Master Key
pk = Public Key
M = Message
A = Universe of Attributes
S = Set of Attributes that Describes Key
C = Cipher Text
sk = Private Key
x = Node from access tree

In this scheme, the algorithm for Setup generates the public and master keys. By using the set of descriptive attributes for the key and the maser key, secret keys for users are generated accordingly using KeyGen. To encrypt sensitive data, the data owner can create a cipher text according to the access tree and its attributes. Access trees exist in this scheme to determine who is able to access an encrypted message according to their attributes and secret keys. To decrypt the cipher text, the attributes of the user must go through the access tree and only if it reaches the leaves of the tree and possesses each necessary attribute and the corresponding private key does the decryption take place. During the decryption process a recursive algorithm is used to decrypt data called DecryptNode which takes in the node from the access tree as x if it is a leaf node in the tree and determines if the user is able to decrypt the data if they have the satisfying set of attributes. [2]

In the Attribute-Based Encryption Scheme presented by Benthencourt et. al. [2], the aspect of revoking privileges from a user differs from the Role-Based scheme. When revoking a user in the role based encryption scheme, the scheme supports dynamic user revocation. However, in attribute based encryption scheme all users are effected. This is because in attribute based encryption there may be a number of different users who are associated with the same access tree. In addition, every time a user is revoked from certain permissions, all other users need to update their private keys because the parameters of the system need to be redeveloped. However, in the role based scheme, the only thing needed to be updated if a user is revoked is the role's public parameters.

3.3 Efficiency

The researchers Kuhn et. al. [5] by using ABAC it may require 2n access rules for n attributes, meaning if there are 10 attributes, it may require up to 1024 access rules. Meanwhile, in RBAC it could require 2n roles for each group of attributes when trying to apply the access control, meaning that if there are 10 attributes, in the worst case scenario require up which is 1024 roles. In general, this means that in ABAC it is easy to set up access rules, but it is difficult to modify user's access permissions with ease. However, in RBAC, it is the opposite where user provisioning alleviates the issue of modifying access permissions without affecting other users in the same system. These researchers identify that there are many different approaches to control the mapping between roles and attributes, while still effectively fulfilling user provisioning in a system. By combining the system and making a role structure that is based on static attributes, such as job, or worksite, a system with 4 static attributes and 6 dynamic attributes, such as time of day, it benefits the efficiency of the system in which it results in 24 or 16 roles, instead of having 210 or 1024 roles. It also benefits the efficiency of the system in which it results in 26 or 64 access rules instead of 210 or 1024 access rules, making the system easier to manage. This proves that it is beneficial to combine RBAC and ABAC in a system to better the access control in a system [5].

The efficiency of the scheme presented by Zhou et. al. has very clear advantages over the scheme proposed by Benthencourt et. al. Unlike attributed based encryption, in role based encryption the size of the cipher text and the decryption key is not reliant on the number of roles and users. The cipher text size in role based encryption is O(1), meaning it will always take the same amount of time to decrypt a cipher text [1]. Attribute based encryption is inefficient being that the size of the cipher text is O(n) and is directly proportional to the number of users in the system. This would be a disadvantage in a healthcare system as there is a large number of users in the system, which would make the size of the cipher text large as well [2]. Another advantage of using role based encryption for access control, the encryption complexity is O(1), meaning regardless of the input it will always take the same amount of time to encrypt data[1]. Meanwhile in CP-ABE it is O(n), which grows in direct proportion and linearly to the size of the data [2]. One similarity between the two encryption schemes, as well as in Goyal et. al.'s Key-Policy attribute based encryption scheme, is that the complexity of decryption are O(n), however role based only needs one round of computation while in attribute based the number of computation depends on the number of nodes in the access tree. To decrease the time, Goyal et. al. suggests to determine which nodes are and are not satisfied in the access tree of the cipher text somehow before performing any encryption or decryption [4]. This is an important quality to have in a healthcare system because it means

that the person who is trying to gain access will be able to perform cryptographic operations on personal health records quicker to be able to evaluate a situation faster.

4. RABE METHODOLOGY

After researching and evaluating the two techniques for access control mentioned in the section above, we propose it is more valuable to combine the best characteristics of the two previously described schemes to make managing personal health data on the cloud more secure and confidential. In RBAC, the user does not need to interfere with the distribution of privileges and roles because this is done by the administrator. In ABAC the data owner is the one who has access to the data. In ABAC, data is encrypted based on a set of attributes and if the user who is looking to gain access, possesses those attributes, they will be able to decrypt and access data.

We refer to the new proposed scheme as Role-Attribute-Based Encryption (RABE) for access control. The new access control RABE scheme that we are proposing in this journal is the hierarchy system used in role-based access control combined with the attribute-based encryption used in attribute-based access control. A scenario sample of the new proposed scheme may be, for example when a user, such as a doctor, wants to access a patient's personal health records in the cloud, they log onto the role based health care system. The system gets a set of users who are authorized to access the personal health record based on their role and the attributes they possess and check if it satisfies the access policy tree for that record. If it satisfies the access policy tree, the doctor would be able to decrypt the data from the cloud using their private key and access the data.

To enforce the proposed Role-Attribute-Based-Encryption for Access Control System, the following algorithms would need to be used to securely store and manage personal health records in the cloud:

Figure N.5 RABE Algorithms for Access Control

- Setup(λ);
- CreateRole(mk, ID_R)
- CreateUser(mk, ID_u)
- AddUser(pk, pub_R, U, ID_u)
- Encrypt(pk, M, A)
- Decrypt(pk, C, sk)
- RevokeUser(pk, pub_R, U, ID_u)

Table N.3: Role-Attribute-Based-Encryption for Access Control Algorithm Input

Inputs
λ = Security Parameter
mk = Master Key
ID_R = Role ID
ID_u = User ID
pk = Public Key
pub_R = Public Parameters of Role
U = User List
M = Message
A = Universe of Attributes
S = Set of Attributes that Describes Key
C = Cipher Text
sk = Private Key

In the proposed scheme, group administrator would set up the master and public key, as well as create the roles and users based on the corresponding algorithms. The role manager would add the users to the roles which they qualify for based on the attributes that are associated with that particular role. The user is given a private key f from the role manager through a secure channel based on the set of attributes associated with that user. After the user receives their private key, the user is accountability to store and keep it safe from a malicious user. If it gets into the wrong hands and patient's personal health records become disclosed to a malicious user, the user would be accountable for those actions. In the event this occurs, the user should inform the role manger as soon as possible to temporarily remove the user from that role to protect the security of the personal health records in the system. The role manager would need to then re-add the user and assign a new private key. When a personal health record is encrypted to the cloud, it is assigned an access tree. The access tree consists of the attributes that users who are authorized access are associated with. This ensures that the right roles and users will only be able to access the data. To decrypt the data, the user needs use their private key to see if they are able to gain access. The user will only be granted access if the attributes associated with their private key fulfills the attributes in that record's access tree.

4.1 Algorithm Overview of RABE

The following section gives descriptive definitions of each algorithm in the new proposed Role-Attribute-Based Encryption model:

Setup(λ): This algorithm takes a security parameter λ and generates a master key mk and a public key pk by using three bilinear groups and mapping them to one another in a deterministic function e: G1 X G2 → GT which chooses two secret

values for the keys. To do this two generators are chosen g1,g2, in which g1 is an element in G1 and g2 is an element in G2. Once chosen the algorithm outputs a bilinear pair e:(g1,g2) if and only if it satisfies the following: 1. If x,y are elements of Zp* where Zp* is a multiplicative group that uses only integers from 1 to p-1 and p is prime, then e:(g1x, g2y) =e:(g1,g2)xy, proving its bilinearity. 2.If If e:(g1,g2) $\neq$ 1 unless g1 = 1 or g2 = 1, proving it is nondegenerate. 3. If e:(g1,g2) is computable in polynomial time. After the pair is output and it is indeed, bilinear, nondegenerate and computable, the master key is given to the group administrator and the public key is made public to all users.

CreateRole(mk, ID_R): This algorithm creates a role *R* by first generating an empty user list U and chooses a random secret value to send over a secure channel to the group administrator. The group administrator generates the private key sk and the role's public parameters pub_R. Note: The role *R* with ID_u has predecessors with identities $(ID_{R1}, ID_{R2}…ID_{Rn})$.

CreateUser(mk, ID_u, S): The group administrator uses this algorithm which takes in the set of user descriptive attributes *S* and creates a private key *sk* that is associated with that set of attributes and gives it to user with identity ID_u.

AddUser(pk, pub_R, U, ID_u): To add a user U_1 to a role *R*, the role manger first checks to see if U_1 exists in *U*, the set of users in *R*, and if U_1 is not in U the role manager adds the user with ID_{U1} with this algorithm to the user lists U and returns the role's public parameters pub_R.

- By creating users and roles and adding the user to a role, this maps the user to a role and then a role to a cipher text's access policies. This makes user provisioning simpler in that when a user needs certain permissions granted to them, the user is assigned to the role, which holds the appropriate privileges. The complexity of these operations of creating and adding a user to a role are linearly proportionate to the number of users who are authorized in the system.

Encrypt (pk, M, A): To encrypt a message *M* under access tree *T*, we first choose a polynomial p_x for each node *x* and set the degree d_x of the polynomial one less than the value of the threshold k_x of the node. Each inner node of the tree is a threshold gate and the leaves of the tree are associated with attributes. The algorithm starting at the root R chooses a random s and sets $p_R(0)=s$. It chooses all other points of the polynomial d_R and randomly defines them. For node *x*, it sets $p_x(0)= p_{parent(x)}(index(x))$, with *index(x*) being a function that returns number of nodes children, and chooses all other points d_x randomly. The cipher text *C* then is made by giving the access tree T.

- The encryption complexity it is O(n), which means that it the complexity increases in direct proportion and linearly to the size of the data being encrypted to a cipher text. However, since the size of the cipher text is reduced because it is not in direct proportion to the users in the system and is instead in proportion with access policies regarding to that data, the encryption complexity and the time it takes to encrypt data reduces as well.

Decrypt(pk, C, sk): To decrypt a cipher text *C* this algorithm takes in as parameters a public key *pk*, secret key *sk* of the user, and a cipher text *C* If the private key *sk* is associated with the set of attributes *S* that corresponds the access tree *T*, then the cipher text *C* will be decrypted.

- The decryption complexity is O(n), which grows in proportionally to the size of the cipher text, but because the size of the cipher text is reduced because it is not in direct proportion to the users in the system. Instead the size of the cipher text is proportional to policies in access tree corresponding to that cipher text, which is smaller than the number of users in the system, making the time that it takes to decrypt data is reduced as well.

RevokeUser(pk, pub_R, U, ID_u): This algorithm revokes a user U_1 from a role *R* To revoke a user U_1 to a role *R*, the role manger first checks to see if U_1 exists in the set of users in *R*, and if U_1 is in the set, then role manager removes the user with ID_{U1} with this algorithm from the user lists U. The role manager chooses a random secret value and sends it over a secure channel to the group administrator who updates the role's public parameters pub_R.

- The mapping from user to role and role to attributes is also beneficial when revoking a user, as it does not affect the other users assigned in that role, it only removes the user from that role, and does not revoke the privileges from the role itself. It only removes the user from the role, which then removes the role's privileges from the individual user being removed. The complexity of revoking a user to a role are linearly proportionate to the number of users who are authorized in the system.

4.2 Benefits and Efficiency of RABE

By combining role-based access control and attribute-based access control, the proposed new scheme RABE seems to be beneficial in many ways.

- One way is that by using the role-based structural hierarchy, the successor and predecessor roles still apply.
- In addition, when adding a user to the RABE system, the user is able to access any data that was previously added to the system, without the data owner needing to re-encrypt their data. This would save the data owner time, making the proposed system less expensive.
- Also, when revoking a user, it uses the same concepts of role-based access control, and it does not affect any of the previous users who are associated with that role. Unlike in attribute based where all users who are associated with the same access tree are affected. This means that due to the change in the system, all users would need to update their private keys. in this new RABE system, since the users are not linked directly to the attributes themselves, when revoking a user, the other users are not effected. Only the role's public parameters are modified due to the revocation of a user.

- When checking to see who is able to access a personal health record, this scheme not only checks to see if the user who wants to gain access possesses the correct role that should be authorized access, but also checks if that particular role has the right attributes to satisfy the access tree and be granted access to view that record.
- By using this new proposed scheme, the size of the cipher text should be smaller than in CP-ABE, being that the it is no longer directly proportional to the number of users in the system. Instead, it should be directly proportional to the number of policies in the access tree.
- In this new proposed scheme RABE, the overhead time of computation should also decrease for encryption and decryption complexities when comparing to the results for CP-ABE, as a result of having smaller cipher texts. The complexity is still O(n), with n being the size of the cipher text. However, since the the size of the cipher text is reduced because it is proportional to the cipher text's access policies and not to the number of users in the system, the time that it takes to decrypt data is reduced as well. This means that the number of rounds of computation depends on the number of policies in the access tree of the cipher text, instead of the number of users in the role hierarchy.

These beneficial characteristics will not only better a healthcare system in user provisioning, but will make the new RABE scheme less expensive when performing encryption and decryption on personal health records than the previously proposed schemes. The rounds of computation will depend on the number of policies corresponding to that particular cipher text, rather than the number of users that pertain to the role in the system. Since the time it takes to decrypt a record is proportional to the size of the encrypted record, it will reduce the time it takes to decrypt a personal health record in the cloud. This is extremely beneficial in the field of health care because in some situations, a doctor's time is very valuable and some patient's may have life-threatening conditions that require quick decision making. Therefore, by reducing the size of the encrypted record, or cipher text, the time it will take to decrypt personal health records would be O(n), with n being the size of the encrypted record, therefore the RABE scheme will reduce the overhead computation time.

5. FUTURE WORK

In the future, testing the two encryption schemes separately to validate the access control models when dealing with personal health records would be significant to research in this field alone. In addition, in the future constructing the new RABE scheme is a must to improve data privacy on cloud systems used in healthcare. This would be an important contribution to be made in this field because personal health records are very valuable. Some even say that these records are more valuable than a social security number or other personal information. This is because from a personal health record you not only learn about health issues, but also about other personal details including the patient's social security number, address, medical history, family contact information, family history, etc. In addition to the construction, proving correctness is also a must to prove the satisfaction of the access tree when a user is requesting access to a patient's records. By constructing and proving the correctness of this scheme is less expensive based on overhead computations, it will be a great improvement as doctors would be able to decrypt a patient's personal health records faster so they will be able to evaluate a patient's health status and take the appropriate actions.

6. CONCLUSION

In this journal, we compared role based access control to attribute based access control in regards to validating that personal health records will be confidential in a healthcare cloud system. Two different encryption schemes were evaluated and compared in attempts to show the advantages that RBAC has over ABAC. The overhead computations of encryption and decryption complexities, as well as the size of the cipher text are significantly different in the two processes. After evaluating the two systems, we propose a combined role-attribute-based-encryption to enforce access control system to make a more secure system for managing and storing personal health records on the cloud. The system should reduce the size of overhead computations of encrypting and decrypting data, as well as reduce the size of the cipher text. In more recent times, it is less common to hear about role based access control being used in systems since attribute based access control came into existence. Overall, Attribute-Based Access Control is a more intricate way to control sensitive data, such as personal heath records on the cloud, because of their complex algorithms and encryption schemes. Role-based access control does seem to have advantages over attribute-based access control, like when dealing with large scale systems, however, attribute-based access control is more useful in securing sensitive data due to the use of access trees and it is more resistant to the malicious users. It is crucial to ensure the privacy of personal health records in cloud systems. Being that not much research has been done on this particular subject, it needs to be further addressed before everyone makes the move to the cloud to ensure that sensitive data stored on these cloud services are secure. By combing the two previously mentioned systems, the proposed RABE will improve data privacy on cloud systems used in healthcare.

ACKNOWLEDGMENTS

I extend my words of gratitude toward my advisor, Dr. Centonze for her commitment and support. I would also like to thank my family for their continuous support and motivation. In addition, I would like to thank all of the faculty as well as students of the Computer Science Department at Iona College for sharing their knowledge and experiecnes with me in support of my scholarly career.

REFERENCES

[1]] Zhou, L., Varadharajan, V., & Hitchens, M. (september 2011). Enforcing Role-Based Access Control for Secure Data Storage in the Cloud. The Computer Journal, 54(10).
[2] Bethencourt, J., Sahai, A., & Waters, B. (2007). Ciphertext-Policy Attribute-Based Encryption. 2007 IEEE Symposium on Security and Privacy (SP '07).
[3] Thike, P. H., & Oo, N. (2015). Ensuring Fine-Grained Authorized Access Control for Healthcare Applications on Cloud Provisioned Platform. Proceedings of 2015 International Conference on Future Computational Technologies (ICFCT'2015), 184-190.
[4] Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006). Attribute-based encryption for fine-grained access control of encrypted data. Proceedings of the 13th ACM Conference on Computer and Communications Security - CCS '06.
[5] Kuhn, D. Richard, Edward J. Coyne, and Timothy R. Weil. "Adding Attributes to Role-Based Access Control." IEEE Computer 43.6 (2010): 79-81.
[6] Shah, P. (2015). Data Security for Cloud Storage System Using Role Based Access Control. International Journal of Science and Research (IJSR), 4(Q), 305-307.
[7] Ganorkar, G., Deshmukh, A., Prof., & Tambhakhe, M., Prof. (2015). Implementation of Role Based Access Control on Encrypted Data in Hybrid Cloud. International Journal of Engineering Research and General Science, 3(4), 50-58Y.
[8] Green, M., Hohenberger, S., & Waters, B. (2011). Outsourcing the Decryption of ABE Ciphertexts. Proceedings of the 20th USENIX Conference on Security.
[9] Rao, R. V., & Selvamani, K. (2015). Data Security Challenges and Its Solutions in Cloud Computing. Procedia Computer Science, 48, 204-209.
[10] Shah, J. R., Murtaza, M. B., & Opera, E. (2014). Electronic Health Records: Challenges and Opportunities. International Information Management Association, 189-203.
[11] "2014 HIMSS Analytics Cloud Survey." HIMSS Analytics, June 2014. Web.

Authors' Biographies with Photo

Monica Suleiman received her Bachelors Degree of Science in Computer Science from Iona College in New Rocelle, NY in 2015. Her research is centered on healthcare systems and privacy in various computing systems, including cloud, web, and mobile applications. She previously presented her researach at the 2015 National Coference of Undergraduate Research(NCUR) with fellow colleagues from Iona College. She has also presented her work at numerious school-based events in both 2015 and 2016, such as Iona Scholar's Day(ISD), Iona's Science Symposium and the Iona College Computer Science Alumni Event. She will be receiving her Masters Degree of Science in Computer Science with a Concentration in Cybersecurity in May of 2016.

Paolina Centonze has been a professor in the Computer Science Department of Iona College since August 2011. Her areas of research include language-based security and mobile computing. At Iona College, she has been responsible for extending the Computer Science curricula into the field of Cyber Security.

Before joining Iona College, Dr. Centonze was a researcher at IBM's Thomas J. Watson Research Center in Yorktown Heights, N.Y. She has published extensively at numerous conferences worldwide, such as ISSTA, ECOOP, ACSAC, MDM, MOBILESoft, MobileDeLi. Dr. Centonze is also the author of two book chapters in the area of cloud and mobile security, which will appear in 2016 in books published by IGI Global and John Wiley & Sons. She is also the inventor of 10 patents issued by the United States Patent and Trademark Office.
Dr. Centonze received her Ph.D. in Mathematics and MS degree in Computer Science from New York University (NYU) Tandon School of Engineering in Brooklyn, N.Y., and her BS degree in Computer Science from St. John's University in Queens, N.Y.