Securing E-health Data using Ciphertext-Policy Attribute-Based Encryption with Dynamic User Revocation

Mohammed Ali Kamoona, Ahmad Mousa Al Tamimi

Abstract: E-health systems hold a massive amount of medical data that is stored and shared across healthcare service providers to deliver health facilities. However, security and privacy worries increase when sharing this data over distributed settings. As a result, Cryptography techniques have been considered to secure e-health data from unauthorized access. The Ciphertext Policy Attribute-Based Encryption (CP-ABE) is commonly utilized in such a setting, which provides role-based and fine-grained access control over encrypted data. The CP-ABE suffers from the problem of user revocation where the entire policy must be changed even when only one user is revoked or removed from the policy. In this paper, we proposed a CP-ABE based access control model to support user revocation efficiently. Specifically, the proposed model associates a unique identifier to each user. This identifier is added to the policy attributes and removed dynamically when the user is added/revoked. A tree structure (PolicyPathTree) is designed specifically for our model. It can facilitate fast access to policy's attributes during the verification process; The model is analyzed using Information Theory Tools. Results show that our model outperforms other notable work in terms of computational overheads.,

Keywords: Access Control, e-Health Security, Attribute-Based Encryption, User Revocation, CP-ABE.

I. INTRODUCTION

The idea of the e-health systems is to transform the traditional healthcare systems to a reliable digital system that can provide fast storing and searching of data over the network. Here, the patient's records are re-structured into a well standard digital format to facilitate data exchange across several healthcare providers [1]. E-Health is supported by the widespread of "wireless body area networks (WBAN) and wireless sensor network (WSN)" where health data is collected and stored in a standard format known as electronic health records (EHR) [2]. Storing health data in EHR format permits more efficiency through retrieving, sharing, and exchanging (interoperability). Specifically, using EHR has a significant advantage as many e-health systems operate in an isolated mode [3]. However, in some cases, patients' data must be shared with other systems to get proper service [4].

Several standards have been recommended to achieve e-health security requirements, and one can consider in this regard the "Health Level Seven (HL7) and Fast Healthcare Interoperability Resources (FHIR)" [5]. Even though these standards provide the interoperability feature; however, they do not offer advanced security, as a simple access control mechanism is presented [6]. As a result, Cryptography was considered to enhance the security of e-health data [7]. Here, the data is encrypted to unreadable form before outsourcing it. Then it is converted it back to its original shape upon received on the other side. It makes sure that data is available only to its rightful owners alongside the authorized users of the system [8]. There are three types of encryption techniques that are commonly used: Symmetric, Asymmetric, and attribute-based encryption (ABE). Although the symmetric an asymmetric provide sufficient security, both of them don't offer fine-grained access control efficiently when compared to ABE [9]. The cipher-text policy attribute-based encryption (CP-ABE) was considered as the best technique suited for the e-health setting [10].

Moreover, the CP-ABE provides a set of security features, besides the fine-grained access control, that are required for the e-Health systems. One can consider, in this regard, the role-based file decryption where the e-Health systems maintain many users with different roles and privileges [11]. The CP-ABE also reduces the number of keys where users with the same access privilege are sharing the same key. And can only decrypt the files when satisfying the set of the defined key's attributes [12]. Besides that, CP-ABE suffers from the user revocation problem, in which several users share the same attributes to access the same file. Thus, when a user is removed or revoked, all his attributes should also be revoked, to prevent the user from decrypting the shared data. However, this operation affects the (non-revoked) users who share same attributes of the revoked user. The user revocation causes two processes firstly; the same file must be re-encrypted using a newly generated key. Secondly, that key should get re-distributed and delivered to the non-revoked users; these two operations result in an overhead, thus effects system performance [13,14].

Manuscript published on 30 September 2019
* Correspondence Author

Mohammed Ali Kamoona*, Computer Science, Applied Science Private University, Jordan
Ahmad Mousa Al Tamimi, Information Technology Applied Science Private University, Jordan

A. Related Work

CP-ABE is “an access control scheme over encrypted data.” It depends on users' attributes to construct an access policy, where only authorized users determined by data owner can access the encrypted data. Both key and cipher-text are designated through group of expressive attributes. In which data owner can specify an access policy to the cipher-text embedded within itself. When there is a match between the user key and the policy-defined attributes inside the cipher-text, only then the user can decrypt that particular data [11]. That encryption techniques play a vital role when data are moved over the network as attackers cannot understand the transmitted data even if they have it [15,16]. In this regard, may encryption techniques have been utilized; however; the CP-ABE is considered one of the most suited method, specifically in the case of securing health data [17].

Despite the features that CP-ABE able to provide, CP-ABE suffers from the problem of user revocation. Where revoking a user in CP-ABE means the access policy for all remaining users should be changed even when only one user revoked. User revocation is considered an open problem [18]. As a solution, the authors of [19] proposed a method based on linear secret sharing. Here, each user in the system is given an identifier (id), then a binary tree is used for searching. The user revocation process is done here by revoking the users according to their id's via a key-update for the non-revoked users.

authors of [20] proposed another work on revocation in CP-ABE. The method depends on using a non-monotonic access structure. User revocation process is made by revoking user’s attributes from the cipher-text itself, which makes revoked user's attribute removed from the list of users who can access and decrypt that file. Additionally, a method called the rekeying mechanism was proposed by authors of [21]. This method achieves revocation via a key update, in which if a user (i.e., Nurse) is revoked from the system, the key updated for other users' while keeping the revoked user key without an update. The revocation process is done by giving each user a secret key and an id. The revocation process is carried out by the system administrator based on the ids and through the modifying of users' keys except for the revoked user.

Moreover, another solution was proposed in [22]. This method is named “batch-based CP-ABE.” The main idea consists of having a batch-based model that splits time into several intervals called time slots. Any changes in policy can take place only between consecutive time slots. The process of revocation is done by making the attribute authority sends some of the key’s parts during each time slot to refresh it. While leaving the revoked user without this key update, therefore, remove his privileges from the system.

Likewise, a study has been proposed in [23]. The idea consists of two steps. The first step is concerned with data publication via splitting encrypted data into slices and selecting a random slice to be dynamic data while keeping the rest of the data as static data. The dynamic data is re-encrypted using AES while the static data is retained without further encryption. The second step involves the revocation process. It is done by changing the key for the dynamic data and re-encrypt it using the new key disinclining the revoked user.

Similarly, the author of [24] proposed a model named “traceable CP-ABE with attribute-level user revocation.” This method was done by “linear secret sharing schemes (LSSS)” to provide access structure while using ABE for fine-grained access control. This work enables tracing defectors where the authority is capable of sending the defector id. When a user is revoked, his attributes are revoked as well, a change in key and ciphertext has occurred within the user's group.

Besides that, the author of [25] proposed a model where a user is revoked using a proxy for re-encryption. The revocation process is done through two methods: the first method is the attribute revocation, in which a user is revoked by updating the attributes for the rest of users associated in the policy. The second method is withdrawing a user and his associated accessors by reversing all the attributes of that accessor. Also, in [26], a revocable CP-ABE was proposed called “Time-Based Proxy Re-Encryption” for User Revocation. This method utilized a “proxy re-encryption technique” by having the proxy re-encrypt data at the proxy server upon user revocation. Where each user will have time-based access and can only access in a particular time accosted with his attributes, this is done by giving users a User-Attribute-Secret-Keys (UAKs) that determine the access time for the user according to his attributes. Thus, revoking users is possible by removing the UAK and re-encrypt data via a proxy server.

Finally, the authors of [14] proposed a CP-ABE user revocation with key-escrow resistance. This system uses the user's identity to revoke users, so users are revoked via their specified identity without removing the user or his attributes from the system. It results in revocation free from re-encryption and key regeneration, also by making a decryption key called personal secret for each user.

In this paper, we choose to enhance the CP-ABE in another way focusing our work upon user revocation problem, this done via our new model (dynamic). Here, a unique identifier UID is added to each user attribute to determine if the user is revoked or not. The UID is a distinct value that identifies each user uniquely. It is constructed using personal information (e.g., SSN) as a seed to a random number generator (RC4 algorithm) [27]. Together, policy attributes and UID form policy threshold that must be satisfied to decrypt the file. Thus, in our model, each user should have a valid UID and should fulfill the policy attributes. We utilized a specific data stretcher to support our model called PolicyPathTree. To enhance user validation process, finally, our model supports three operations: add new user, move a user to another relation, revoke user. It is resulting in providing better key management and more crucial straightforward revocation.

B. Contribution

As mentioned before, the user's revocation in CP-ABE remains an open problem; thus, in this work, a dynamic user revocation model (Dynamic UID) Is proposed. The main contribution:

- Enhancing the standard CP-ABE by offering a model to optimize the process of user revocation. As a result, the model eliminates the need for file re-encryption and key distribution. It reduces the revocation impact over the system.

- Users are revoked \ removed without affecting any other users. It is done by assigning a unique identifier for each user who is considered as one of the policy attributes. So, when the user is revoked, his id is only removed from the policy.
- A data structure is specifically designed called PolicyPathTree (PPT) To help to implement our model. PPT is an access structure that can achieve fast user validation. Moreover, the PPT support three operations: add user, move user, and revoke user, from a specific policy with neglectable time. More details are given in section3.

The UID is generated from any distinct user's personal information (i.e., SSN) and conjunct with other attributes of the user's key. So any user who satisfies the cyphertext attributes and has a valid UID can get access and decrypt the encryption data, where UIDs is a list of legitimate users' identifiers. In the case of revocation, the user can be revoked by removing his UIDi from the valid list and keeping the rest of the users' key's attributes without any modification. Finally, our model is combined with Benthencourt scheme [13].

The remaining of this works is presented as follows: preliminaries used are presented in section 2. the proposed model (Dynamic UID) is explained in Section 3. Section 4 presents the system construction; analysis is offered in section 5. finally; the conclusion and future work are given in Section 6.

II. PRELIMINARIES

A background overview of some concepts related to the standard CP-ABE is presented in this section. We begin by (bilinear maps and bilinear pairings) that will be used in the CP-ABE encryption process, after that some notations are given in table1.

A. Bilinear map [13]:

Let, $G1$ and GT be two multiplicative cyclic groups of prime order p. Let g be a generator of $G1$ and e be a bilinear map, where $e: G1 \times G1 \rightarrow GT$. The bilinear map e then it has the propriety of Bilinearity for all $u, v \in G1$ and $a, b \in Z\,p,$ we have $e\,(ua, vb) = e\,(u, v)ab$, it is nondegenerate where $e(g.g) \neq 1.$ We note that $G1$ is a bilinear group if the group operation in $G1$ and the bilinear map $e: G1 \times G1 \rightarrow GT$ are both efficiently computable.

B. Access Tree [13]:

Let AT represents an access tree, where AT is one of many forms of access policies or access trees. Each access tree consists of several leaf nodes and non-leaf nodes. The nodes use a conjunction (AND) or disjunction (OR) gates between them. non-leaf nodes include a threshold gate and a threshold value. Assume having a non-leaf node x as one of the nodes in the access tree it consists of two values, the first is the number of children nx , and the second is threshold value $Kx. If\ 0 < Kx \leq Nx\ were\ Kx = 1$ then the threshold gate represents an OR gate, else when $Kx = Nx$ then the threshold gate represents an AND gate. Additionally, leaf nodes say x represents an attribute having the value $KX = 1$. Finally, parent nodes are denoted by the $parent\,(x)$ and child nodes are denoted by the $child\,(x)$ where the $child\ (x)$ represents an attribute of the access tree.

C. Satisfying the Access Tree [13]:

Let AT denotes an access tree having a root node r. Let $AT\ x$ be a subtree of the access tree AT having a root node x, thus $AT\ =\ AT\ r$. If the access tree Tx can be satisfied by a set of attributes says A , it can be formed as $AT\ x(A) = 1. AT\ x(A)$ is calculated reclusively using the following method: if x is a non-leaf node, then calculate $AT\ x'\,(A)$ for all children of x' of the node $x. AT\ x(A)\ = 1$ only if at least Kx return $1, AT\ x(A) = 1\ only\ if\ att\,(X)\ is\ in\ (A).$

Table1. Notations

NOTATION	DESCRIPTION
AT	Access tree
Zp	Result of the hash function
p	Prime order
$G1, GT$	multiplicative cyclic groups of prime order p
g	Generator of G1
$H1, H2, Hx$	Cryptographic hash function
U	A Set of all users in the system
UID	Unique user identity for each user
$UIDr$	Revoked user UID
α, β	Random numbers

III. THE DYNAMIC UID MODEL

A. Dynamic UID

Figure 1 illustrates our proposed model, which consist of four components: attribute authority (AA), server, file owner, users. Attribute Authority initiates the process (AA), where the public parameters (policy's attributes) and keys are generated and sent to the file owner and users, respectively. The file owner then encrypts the data file according to the received parameters and sends it to the server while the user uses the received key to decrypt the encrypted file. At the server-side, the Policy Path Tree (PPT) is maintained and updated periodically by the AA in case of adding or revoking a user.

When a user requests a file from the server, the server is first verifying the user against the PPT. If the user is authenticated, the server returns the encrypted data to the user, which can be decrypted using his key. Otherwise, if the verification process failed the user request will be denied preventing him from accessing the file.

Retrieval Number: C6309098319/2019©BEIESP
DOI:10.35940/ijrte.C6309.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

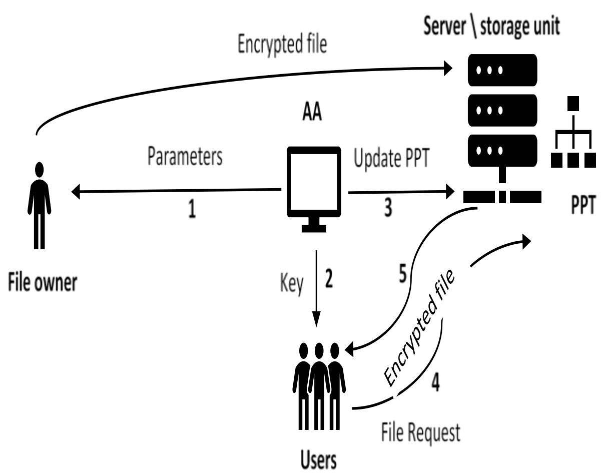


Fig. 1.Dynamic UID Model

B. Dynamic UID Security Model

In our model, the user revocation process is done by removing the user's unique identity from the PPT; for example, let $U = \{UID1, UID2 \ldots UIDN\}$ be the set of all UID in the system, let $Pleaf$ be the leaf node of PPT where $Pleaf$ is a subset of U, and let Ai be the set of attributes of the access policy, $A\{a1, a2, \ldots ai\}$ when a user r is revoked, his $UIDr$ is removed from the PPT as we show in figure2.

Assumption: a revoked user cannot access file using $UIDr$

Proof: when a revoked user with $UIDr$ and attribute set Ai want to obtain a data file the model verifies the user against the PPT using Ai. Because the verification process failed at the leaf node $UIDr \notin Pleaf$ that is,

$PPT = \perp, UID = \perp, UIDr \notin Ai$ and the user can't compute the decryption key as $(CT, SK, X) = \perp$.

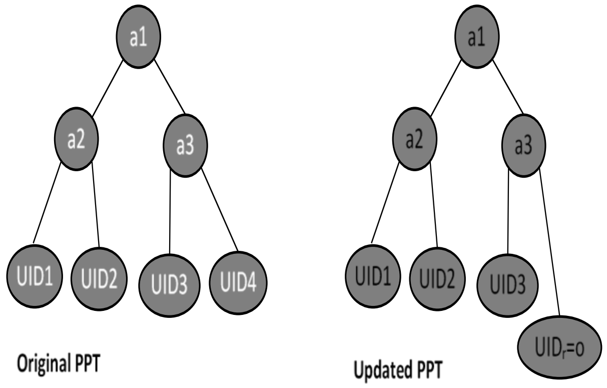


Fig. 2. Policy Path Tree

IV. SYSTEM CONSTRUCTION

A. Setup:

In this step, the public key, as well as the master key, are generated by the AA,

phase1: AA chooses a bilinear group $G1$ and GT of prime order P where its generator is g and a map $\hat{e}\,(G1 \times G1) = GT$. Then AA chooses two random exponents $\alpha, \beta \in ZP$. And purplish the public key as follows:

$$pk = (G1, GT, g, \hat{e}, h = g\beta, \hat{e}\,(g, g)\,\alpha, H1, H2, Hx)$$

note: $(H1, H2, Hx \; are \; hash \; functions)$

while the master key is generated as follows:

$MK = (\beta, g\alpha)$.

Phase2: the second phase is the construction of the PolicyPathTree (PPT) according to the access policy and users UID. The tree is used in the verification process by finding a path from the root attribute to the leaf if such a path exists the PPT returns a valid UID along with the encrypted file.

B. Key Generation:

In this step, the decryption keys are generated for the users by the AA, the key generation algorithm takes a set of attributes Ai that is assigned to each user, and it outputs a key according to that set. The algorithm selects random numbers $\{r, rj \in ZP\}\, \forall j \in Ai$ the

SK is computed as follows:

$$SK = (D = g(\alpha + r)/\beta, Dj = gr.H(j)rj, D'j = grj)\,\forall j \in Ai$$

C. Encrypt:

In this step, the file owner encrypts the message M according to an access policy tree AT using the public parameters PK. Then the algorithm chooses a polynomial qx for each node x in the access tree AT. Then set the degree dx of the polynomial qx as follows:

$dx = Kx - 1$ where Kx represents the threshold value.

Then the polynomials are chosen in a top-down manner, starting from the root node R.The algorithm starts with the root node R and chooses a random number $s \in ZP$ and sets $qx(0) = qparent(x)(index(x))$ for each node x and chooses dx other points until qx is wholly defined.

Let, Y represents the set of leaf nodes in AT. The ciphertext is computed as the following:

$$CT = (AT, C' = M\hat{e}(g, g)\alpha s, C = hs, Cy = gqy(0), C'y = H.(att\,(y))qy(0)).$$

After this process is completed, the CT is sent to the storage server by the user.

D. User Revocation:

In this step, a user is revoked from the system by updating the PPT at the server-side.

That is $UIDr$ is updated to new value $UIDr'$ so the verification process returns $\perp$ for that user.

Let, pleaf represents the leaf node of the PPT where $pleaf = (UID1, UID2, \ldots, UIDi)$. When a user is revoked from the relationship, the $PPT \; leaf$ is updated to be

$p'leaf = (UID'1, UID'2, \dots, UID'I)$ thus, when the revoked user with $UIDr$ (where $UIDr$ denotes revoked UID) want to access the data file the PPT will return $UIDr = \perp$. And by having the UID as one of the leaf nodes in the tree-threshold of the decryption key, this will results:

$i \notin Ai$ where Ai is the set of attributes in the access policy and $i = UID$.

Then, we can define the decrypting node $(CT.SK.X) = \perp$. And then the revoked user cannot calculate the secret key.

E. Decrypt:

In this step, send an access request to the storage server to gain data, the user first gets verified against the PolicyPathTree using his key, PPT will return $UID = true$ if the user UID is verified and then the user will gain access to the encrypted data file. Otherwise, the request will be canceled, and the PPT will return $UID = \perp$. After receiving the ciphertext CT, the decryption process occurs as follows:

First, define Decryptode (CT, SK, x) recursively that takes as input $CT = (AT, C', C, \forall y \in Y: CY, CY'), a$ secret key SK associated with a set Ai of attributes, and node x from AT there are two cases:

First, if $i \in s$ and $node\ x$ is a leaf node then: $i = att(x), i \in Ai$ then

$$Decryptnode\ (CT, SK, x) = \frac{e(Di, Cx)}{e(Di', Cx')}$$

$$= \frac{e\left(gr.H(i)ri, h\,qx(0)\right)}{e\left(gri.H(i)\ qx(0)\right)}$$

$$= e(g, g)rqx(0).$$

If $i \notin s\ (CT, SK, X) = \perp$.

Second, if x is a non-leaf node, the algorithm will run Decryptnode (CT, SK, X) recursively as follows:

For all nodes z that are children of x, it calls Decryptnode (CT, SK, z) and stores the output as Fz. Let $, Aix$ be an arbitrary kx-sized set of child nodes z such that $Fz \neq \perp$. If no such set exists, then the node was not satisfied, and the function returns ⊥.

Else Fx is computed as follows:

$$F_x = \prod_{z \in S_x} F_z^{\Delta_{i,S'_x}(0)}, \quad \text{where } \begin{array}{c} i = \text{index}(z) \\ S'_x = \{\text{index}(z) : z \in S_x\} \end{array}$$

$$= \prod_{z \in S_x} \left(e(g,g)^{r \cdot q_z(0)}\right)^{\Delta_{i,S'_x}(0)}$$

$$= \prod_{z \in S_x} \left(e(g,g)^{r \cdot q_{\text{parent}(z)}(\text{index}(z))}\right)^{\Delta_{i,S'_x}(0)} \quad \text{(by construction)}$$

$$= \prod_{z \in S_x} e(g,g)^{r \cdot q_x(i) \cdot \Delta_{i,S'_x}(0)}$$

$$= e(g,g)^{r \cdot q_x(0)} \qquad \text{(using polynomial interpolation)}$$

And return the result.

Now the decryption algorithm starts by calling the function on the root node R of the tree AT, if the tree is satisfied by s we set

$A = Decryptnode\ (CT, SK, r) = e(g.g)rqR(0) = e(g.g)rs.$

The algorithm now decrypts by computing $C'/(e(C.D)/A) = \tilde{}\ C'/(e(hs.g(\alpha + r)/\beta)/e(g.g)rs) = M.$

V. Analysis

In In this section, we are going to analyze our system as well as compare it to one of the existed CP-ABE methods. To do that we have noticed that the closest scheme to our scheme is the Nazatul scheme. The Nazatul scheme is adopted from the same base system we adapted, which is the Bethencourt scheme [13]. Both Nazatul and our system are based on using user IDs. However, Nazatul model embeds the user's IDs within the cipher-text itself whereas, our proposed solution differs from where the UID represents one of the policy attributes.

Moreover, the paring time required for the encryption, decryption, and user revocation operation alongside the size of both ciphertext and key are illustrated using group elements adopted from information theory tools [27]. To simplify the terms, the following notations are used: $TmulG1\ and\ TmulGT$ denote required computational time for one exponentiation/ scalar multiplication paring operation. Tp signifies the time for one paring operation. $|Z * q|$ indicates the size of $Z * q. |G1|\ and\ |GT|$ represents the size of $G1, and\ GT$ elements respectively. $CT, NC, and\ nu$ represent ciphertext, the number of attributes within CT, and the number of user's associated attributes, respectively. In comparison, it can be noted that the proposed model has less encryption, decryption, user revocation cost. Also, it has a smaller key and cipher-text size, as we demonstrate in table 2.

Table 2: Comparison of The Proposed Scheme to Nazatul Scheme

Scheme	**Nazatul scheme [14]**	**Proposed model**
Encryption cost	***(2nc +1) TmulG1 +2 TmulGT***	***(nc +1) TmulG1 + TmulGT***
Decryption cost	***2nu Tp + TmulG1***	***nu Tp + TmulG1***
User revocation cost	***TmulG1***	***nu (PPT update)***
Cipher-text size	***(2nc + 1) |G1|+|GT|+tc |Z*q|***	***(nc + 1) |G1|+|GT|***
Key size	***(2nu+1) |G1|***	***(nu+1) |G1|***

Further, the proposed scheme based on an access-control over encrypted data via user's validation. The only overhead is the validation process against the PPT, which require a linear time over user's attributes (nu) that equals to 0.092 milliseconds which is a neglectable time. Compared to Nazatul model, the proposed scheme outperforms it by illuminating the required server re-encryption, user revocation polynomial, or users' secret keys which are considered an overhead operation.

Furthermore, to ground our conceptual work, we have implemented the proposed model and Nazatul model. Time is measured in a millisecond, and results are illustrated in figure 3. comparison of computational cost. It was measured using a java development kit (NetBeans IDE 8.2). Using a laptop was running on Microsoft Windows 10-64bit platform with Intel Core i7-6700 HQ CPU with 2.6 GHz microprocessor and 16 GB RAM.

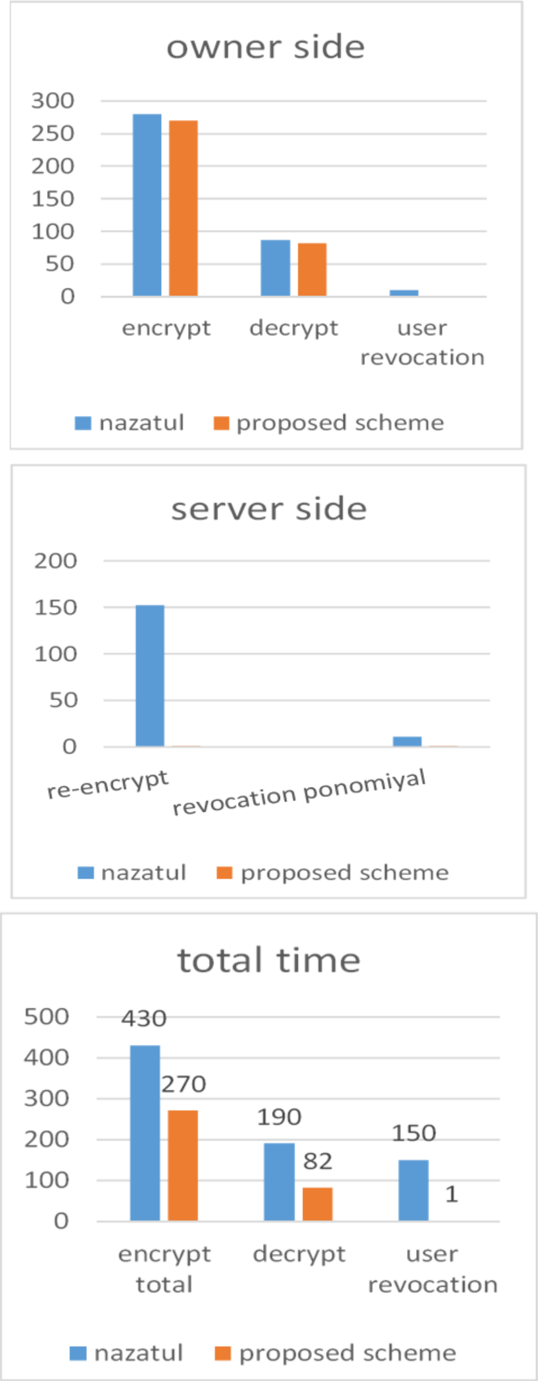


Fig. 3. Comparison of Computational Cost

VI. Conclusion and Future Work

In this paper, we proposed a CP-ABE based access control scheme for distributed e-health data. The main focus on offing an optimum solution for the problem of user revocation via applying a unique id-based CP-ABE model. It offers an effective user revocation process disinclining any complicated operations, such as key re-distribution and data re-encryption, with the least possible overhead. Besides, it supports three operations such as adding, removing, or changing user privilege back and forth effectively, even for fresh revoked or added users. Further, we have categorized users into multiple domains to offer more ease of role determination. And by utilizing a particular tree data structure is developed (PolicyPathTree) based on the policies attributes and used to provide a fast verification process over the security policies. Furthermore, the analysis of the proposed scheme showed that it has the least possible computational overhead when compared to other existing systems.

Acknowledgments

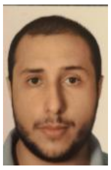
The authors are grateful to the Applied Science Private University, Amman-Jordan, for the full financial support granted to cover the publication fee of this research article.

REFERENCES

1. Rani, A. Antony Viswasa, and E. Baburaj. "An efficient secure authentication on cloud-based e-health care system in WBAN." Biomedical Research (2016).
2. Kamoona et al. "Importance of WBAN and Its Security: An Overview." International Journal of Advanced Research in Computer Science and Software Engineering 8(8) ISSN (E): 2277-128X, ISSN (P): 2277-6451, pp. 30-34 ,(August 2018).
3. Al Ameen, Moshaddique, Jingwei Liu, and Kyungsup Kwak. "Security and privacy issues in wireless sensor networks for healthcare applications." Journal of medical systems 36.1 (2012): 93-101.
4. Antony, Sajan. "SECURE SHARING OF PERSONAL HEALTH RECORD IN CLOUD COMPUTING." International Journal of Computer Application Issue 4, Volume 2 (March - April 2014).
5. Benson, Tim. Principles of health interoperability HL7 and SNOMED. London: Springer, 2010.
6. Franz, Barbara, Andreas Schuler, and Oliver Krauss. "Applying FHIR in an integrated health monitoring system." EJBI 11.2 (2015): 51-56.
7. Swathi, S. V., P. M. Lahari, and A. Thomas Bindu. "Encryption algorithms: a survey." International Journal of Advanced Research in Computer Science & Technology 4.2 (2016).
8. Altamimi, Ahmad Mousa. "Security and Privacy Issues In eHealthcare Systems: Towards Trusted Services." International Journal of Advanced Computer Science and Applications, Vol. 7, No. 9, 2016.
9. Yang, Yang. "Attribute-based data retrieval with semantic keyword search for e-health cloud." Journal of Cloud Computing 4.1 (2015): 10.
10. Qiao, Zhi, et al. "Survey of attribute-based encryption." 2014 15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD). IEEE, 2014.
11. Tian, Ye, et al. "An attribute-based encryption scheme with revocation for fine-grained access control in wireless body area networks." International Journal of Distributed Sensor Networks 10.11 (2014): 259798.
12. Yuan, Wei. "Dynamic Policy Update for Ciphertext-Policy Attribute-Based Encryption." IACR Cryptology ePrint Archive 2016 (2016): 457.
13. Bethencourt, John, Amit Sahai, and Brent Waters. "Ciphertext-policy attribute-based encryption." 2007 IEEE symposium on security and privacy (SP'07). IEEE, 2007.
14. Sultan, Nazatul Haque, Ferdous Ahmed Barbhuiya, and Nityananda Sarma. "A universal cloud user revocation scheme with key-escrow resistance for ciphertext-policy attribute-based access control." Proceedings of the 10th International Conference on Security of Information and Networks. ACM, 2017.

15. Senthilkumar, S.. “SCALABLE AND PROTECTED SHARING RECORDS IN CLOUD COMPUTING USING ABE.” International Journal For Technological Research In Engineering Volume 1, Issue 6, February , (2014)
16. Hong, Hanshu, Di Chen, and Zhixin Sun. "A practical application of CP-ABE for mobile PHR system: a study on the user accountability." SpringerPlus 5.1 (2016): 1320.
17. Choure, Namdev, and Shrikant Dhamdhere. "SURVEY ON DIFFERENT TYPE OF ENCRYPTION ALGORITHM’S." Open Access International Journal of Science& Engineering (OAIJSE) ,Volume 3, Special Issue 1, March (2018).
18. Moffat, Steve, Mohammad Hammoudeh, and Robert Hegarty. "A Survey on Ciphertext-Policy Attribute-based Encryption (CP-ABE) Approaches to Data Security on Mobile Devices and its Application to IoT." Proceedings of the International Conference on Future Networks and Distributed Systems. ACM, 2017.
19. Xie, Xingxing, et al. "An Efficient Ciphertext-Policy Attribute-Based Access Control towards Revocation in Cloud Computing." J. UCS 19.16 (2013): 2349-2367.
20. Zhao, Yang, et al. "Attribute-based Encryption with Non-Monotonic Access Structures Supporting Fine-Grained Attribute Revocation in M-healthcare." IJ Network Security 19.6 (2017): 1044-1052.
21. Shynu, Padinjappurathu Gopalan, and Kumaresan John Singh. "An Enhanced ABE based Secure Access Control Scheme for E-health Clouds." February 3, 2017.
22. Touati, Lyes, and Yacine Challal. "Batch-based CP-ABE with attribute revocation mechanism for the Internet of Things." International Conference on Computing, Networking, and Communications (ICNC, 2015). 2015.
23. Cheng, Yong, et al. "Efficient revocation in ciphertext-policy attribute-based encryption based cryptographic cloud storage." Journal of Zhejiang University SCIENCE C 14.2 (2013): 85-97.
24. Wang, Shangping, Keke Guo, and Yaling Zhang. "Traceable ciphertext-policy attribute-based encryption scheme with attribute level user revocation for cloud storage." PloS one 13.9 (2018): e0203225.
25. Zheng, Hongying, et al. "Modified Ciphertext-Policy Attribute-Based Encryption Scheme with Efficient Revocation for PHR System." Mathematical Problems in Engineering 2017 (2017).
26. Prasad, VenkataVara, Lokeswari Y. Venkataramana and Pandiyan Muthuraj. "Time-Based Proxy Re-Encryption for User Revocation with Reduced UAKs in Cloud Storage." International Journal of Applied Engineering Research 13.8 (2018): 6138-6150.
27. SALHAB, OMAR, et al. "SURVEY PAPER: PSEUDO RANDOM NUMBER GENERATORS AND SECURITY TESTS." Journal of Theoretical & Applied Information Technology 96.7 (2018).
28. T. Cover and J. Thomas. Elements of Information Theory. John Wiley and Sons, Inc., 1991.

AUTHORS PROFILE



First Author Mohammed Ali Kamoona has been received his master’s degree in Computer Science from applied science privet university in Jordan His research interests are primarily in information security and data privacy, cryptography and steganography specifically in e-health related application to develop a standard security setting for such environment.

Second Author Dr. Ahmad AL Tamimi has been received his PhD degree in Computer Science from Concordia University - Canada in 2014.He has joined the Faculty of Information Technology as an assistant professor since July 2014. His research interests are primarily in preserving privacy for interactive and non-interactive database environments. Specifically, providing access and inference control mechanisms, and designing and enforcing security policies for OLAP cubes and relational databases. Currently, Dr. AL Tamimi is working on interoperability healthcare systems to develop a standard architecture supported by advanced security models.