

Achieving Confidentiality in Electronic Health Records using Cloud Systems

Robert French-Baidoo

Takoradi Technical University, Ghana
E-mail: robert.french-baidoo@tpoly.edu.gh

Dominic Asamoah and Stephen Opoku Oppong

Department of Computer Science, KNUST, Ghana; Faculty of Informatics, GTUC, Ghana
E-mail: dominic_asamoah@yahoo.co.uk; sopokuoppong@yahoo.com

Received: 26 August 2017; Accepted: 13 October 2017; Published: 08 January 2018

Abstract—Currently, existing methods for enforcing access to records in an Electronic Health Record system relies on a single Trusted Server which stores health records and mediates access. Such Trusted Servers employ either a Ciphertext-Policy Attribute-Based Encryption (CP-ABE) or Key-Policy Attribute-Based Encryption (KP-ABE) method for storing and controlling access. However, Trusted Server storage of health records is susceptible to single-point-of-threat attack and a successful attack invariably leads to compromising the integrity of records on the server. In this research work. This paper presents a methodology that defines and creates simple Access Structures and eliminates need for private keys during encryption and/or decryption of health records which is the Enhanced Ciphertext-Policy Attribute-Based Encryption (ECP-ABE). The ECP-ABE yields high cryptographic performance creates simple Access Structures, eliminates need for private keys and presents an implementation architecture that makes cloud-based EHR system secure and confidential. The ECP-ABE also performs cryptographic functions using less CPU time, minimal computer memory and produces high encryption and decryption throughput especially with increasing file sizes.

Index Terms—Trusted Authority, Advance Encryption System 256, Secure Hashing Algorithm 256, Virtual Electronic Medical Records, Public Key Infrastructure, Public Key Generator, Random Access Memory.

I. INTRODUCTION

The quest to store health records electronically so that management of such records can be done in a well-structured manner has received lots of attention lately. Cloud-based electronic health records system guarantees round-the-clock records availability regardless of your geographical location.

Electronic health management systems may be in the form of Electronic Health Records (EHR), where sensitive health records of patients are gathered, stored and managed by an enterprise such a hospital, or in the

form of Personal Health Records (PHR) where patients purchase the service of a third-party storage facility service provider so that the patient can create, update and generally manage their own health records independently. The third is a hybrid system where health care facilities and patients play the role of collecting health information of patients, updating and managing those records in tandem. Each party in the hybrid system has varying levels of rights and privileges accruing to him or her [1].

Unfortunately, cloud service providers who are trusted to secure stored records have not lived up to task. Twenty percent of health care providers have suffered security breaches between the years 2009 and 2013, and about Five Hundred health records have been breached through 804 attacks [2].

Security and data confidentiality remains the biggest threat to electronic health systems; attacks on cloud systems are relentless. Inappropriate user right definitions and permissions, ineffective access structure definition, and insecure implementation design are some of the problems bedeviling electronic health record systems.

II. RELATED WORKS

Related technologies are presented in this section.

A. Health Information Privacy

Health Insurance Portability and Accountability Act (HIPAA) 1996 addresses two major issues; Privacy Rule and Security Rule. The HIPAA document describes reasonable procedures to prevent exposure of protected health information [3]. The Act could not deal with privacy matters relating to health information. Researchers looked at cryptography to address health privacy and security gaps in the policy. Cryptography is used in insecure environment to store and scramble data so that only qualified persons can access, read and process data [4].

B. Patient-Centric Health Information System

Other researchers postulated the concept of a patient-centric system. The patient is the owner of the health

information, therefore maintains and manages a copy of his/her medical records [5]. The Indivo health information system is an example of a patient-centric system [6].

C. Smart Card

The use of electronic smart cards as a mechanism to guarantee security data, and to ensure privacy and confidentiality in EHRs have their own unique advantages. Features of smart cards such as; portability and mobility of the electronic smart cards are advantages that was harnessed to provide access security in an electronic patient health record system [7]. Smart cards among was used to store sensitive patient health data and to ensure data privacy and data security in an electronic health information system. However, due to frequent misplacing of smart cards and use of pin numbers to access health information on smart cards made in unreliable means.

D. Attribute-Based Encryptions

- *Fuzzy Identity-Based Encryption*

Identity – Based Encryption (IBE) access control policy called Fuzzy Identity Based Encryption was developed to address privacy and security problems in health information systems [7]. In Identity-Based Encryption system, a sender can send encrypted message to an identity (receiver) without knowing the public key certificate of the receiver. With the IBE system identities are treated as strings. In Fuzzy IBE systems, identities are treated as a set of attributes; “a user with a secret key identity ω is able to decrypt a ciphertext encrypted with the public key ω' if and only if ω and ω' are within a certain distance of each other judged by some metric. Fuzzy IBE gave rise to numerous systems springing up to adopt biometric identities as a means of data encryption primarily because of the error tolerance capability that Fuzzy-IBE presented. One of the advantages in using the Fuzzy Identity – Based Encryption is that, the burden of making sure that third party servers (e.g. cloud servers) are trusted to perform authentication checks before delivering a document is alleviated. The downside of this system it that, even though it's a good cryptosystem that thrives very well in biometric applications, Fuzzy-IBE limits expressiveness and therefore limits the range of applications it can be put to.

- *Key-Policy Attribute-Based Encryption*

Key-Policy Attribute - Based Encryption (KP-ABE) is used in cryptosystem where an identity can encrypt and annotate each ciphertext with a set of descriptive attributes and each private key is associated to an access structure which determines which portions of ciphertext the private key can decrypt [9]. KP-ABE produced a standard model with generically large parameters and key sizes which made it quite impractical for reasonable expressive policies.

- *Ciphertext-Policy Attribute-Based Encryption*

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) introduced a scenario where an entity (patient) in a distributed system may wish to store his/her sensitive health records in a secure Web-Based Patient Health Record (web-PHR) and later may want to share same sensitive data with other users [10]. CP-ABE categorizes users into two mutually exclusive distinct domains i.e.

- Professional Domain (PD) which is a group of healthcare providers e.g. doctors and nurses and
- Social Domain (SD) which is a group comprising of patient's family, friends or fellow patients.

CP-ABE provides a solution for data security against network sniffers. The CP-ABE works with a third-party service managing web PHR which stores ciphered text.

E. Role Based Access Control (RBAC)

In RBAC, Entity Identity Assertion describes situation where a doctor wanting to add new medical records of a patient must authenticate himself/herself with the server through a username and a password [11]. Again, RBAC and Attribute-Based Access Control (ABAC) mechanisms can be employed to strictly tag specific roles and responsibilities to specific users (e.g. Doctor, Nurse, Teller, Manager) to enforce access control. In RBAC, users are assigned role – specific privileges, so that once a user is assigned a specific role, the user cannot perform anymore functionality other than what has been assigned the user [12]. Comparatively, ABAC is a better choice of access control mechanism over RBAC since ABAC can be seen as more applicable in the context of electronic health systems because it offers a more flexible policy description yet rigid document-portion-specific access controls [13]. Rules for ABAC are done using eXtensible Access Control Markup Language (XACML) which is easy to configure as a natural language. Again, ABAC prevents certain types of attacks like brute force and library attacks [14].

F. Limitations of the Reviewed Frameworks

Among some of the limitations discovered during the review were, cost of security implementation, especially with reference to Payment Card Industry (PCI) Data Security Standard and Payment Application Data Security Standard (PCI DSS) of the version 3.0, implementation cost of security for cloud is expensive regarding expert or personnel, security systems and modules [11]. Again, Security sub components interdependency is one of the key issues. The components of the frameworks are large and complicated, and therefore lead to misinterpretation of prescriptions [12] [13]. Researchers are of the view that, objective Specification and Misinterpretation adds up to the numerous setbacks, because frameworks contain long list of compliances and specifications that can lead to misinterpretations [13].

G. Conceptual Framework

The research work postulated a new attribute-based

encryption mechanism which is more secure and required minimal system resources for cryptographic functions. The new used Ciphertext – Policy Attribute – Based Encryption (CP-ABE) as the threshold encryption mechanism [15]. CP-ABE Encryption scheme uses expressible attributes to authenticate users and private keys to encrypt encrypt and/or decrypt Access Structures. It is useful because it is fault tolerant and prevents collusion attack.

The postulated Enhanced Ciphertext – Policy Attribute – Based Encryption scheme (ECP-ABE) improves security mechanism by splitting access into two ways. Firstly, to access health records, a guest is required to authenticate with an Escrow Server which is a separate server. Secondly, a guest must express exactly the same attributes that was used to encrypt a record, once verified, depending on the Access Structure which is identifiable by a set of attributes, the Escrow Server seamlessly transfers the guest together with determined Access Structure to the Data Server to retrieve corresponding records. The Data Server is a separate server which stores health records of patients.

III. METHODOLOGY

This paper postulated a new encryption mechanism to enhance security and improve confidentiality in cloud based EHR. Derived from Ciphertext – Policy Attribute – Based Encryption (CP-ABE), the Enhanced Ciphertext – Policy Attribute – Based Encryption (ECP-ABE) harnessed the efficiencies of CP-ABE and removed the complexities of CP-ABE. The end result was an improved cryptographic function that was more secure which could be used in cloud-based EHR systems.

Enhanced Ciphertext – Policy Attribute – Based Encryption (ECP-ABE) was formulated using Ciphertext – Policy Attribute – Based Encryption (CP-ABE) as a threshold mechanism. ECP-ABE eliminates the weaknesses of CP-ABE and modifies how encryption using expressible attributes could be done in a rather simple manner yet enhancing security of encrypted files and improving confidence in implementation

The construction of ECP- ABE summarily is as follows;

1. ECP-ABE uses attributes to describe a user's credentials.

2. Described attributes are used to encrypt and decrypt ciphertexts.
3. User creates Access Structures specifying who can decrypt in during encryption.
4. User secures Access Structures by encrypting them to with a set of expressed attributes.
5. No private key(s) is needed to further encrypt defined Access Structures.

The study adopted an experimental research approach to simulate an EHR system which was constructed based on the postulated ECP-ABE. A proof of concept was employed for purposes of simulation and approval of postulated encryption mechanism. Fig. 1 describes the Performance indicators of adopted encryption mechanism.

A. Construction of CP-ABE

In CP-ABE, a user is required to express attributes to be used to encrypt a file, after which private keys are used to further secure defined access structures created within a whole file. Private keys are necessary for the elimination and prevention of collusion attacks because, it is expected that each private key is different from the other. Therefore, users cannot combine to view unauthorized data pages.

B. Construction of ECP-ABE

ECP-ABE creates access structures (policies) out of a full document, each access structure is created as a result of policy definitions. Each policy definition creates a separate file along with its expressible attributes which are used for encryption and/or decryption. There is no need for setting private keys on respective policies. ECP-ABE eliminates the need to create private keys for policies to reduce requirement for more CPU time and other resources such as memory space to perform cryptographic functions. Because ECP-ABE requires minimal CPU time, it makes it efficient and yields high cryptographic throughputs. Collusion attack is also eliminated in ECP-ABE through the separation of distinct access structures which are created for every policy defined. Each policy exists mutually exclusive of the next policy and different multiple sets of expressible attributes are used to encrypt different policies, this makes it impossible for guests to collude to decrypt policies.

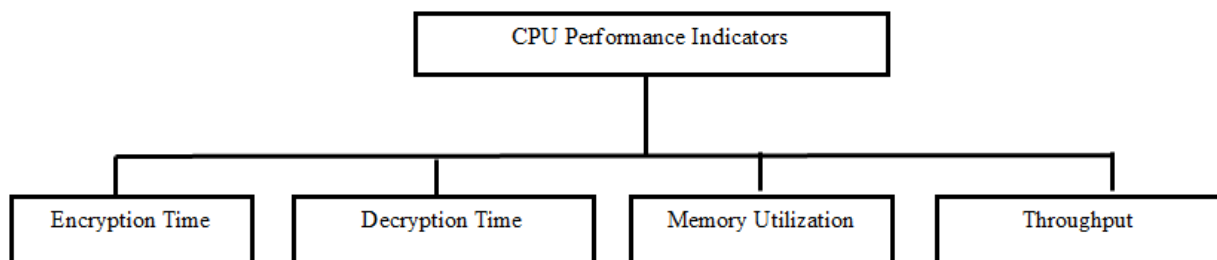


Fig.1. CPU Performance Indicators Columns

IV. RESULTS AND DISCUSSION

Results from the simulation were observed and recorded for analysis. The analysis of the study was based on descriptive analysis.

A. Simulation Results for Encryption Times

Table 1. Encryption Times for CP-ABE, KP-ABE and ECP-ABE Mechanisms

Text Size (bytes)	CP-ABE (ns)	KP-ABE (ns)	ECP-ABE (ns)
8	0.65	0.65	0.61
10	0.75	0.78	0.74
13	0.87	0.88	0.80
16	1.03	1.15	0.85
20	1.32	1.51	1.24
30	2.11	2.54	2.05
45	3.42	4.01	2.95
50	3.44	4.13	2.98

Table 1 represents the encryption times for CP-ABE, KP-ABE and ECP-ABE during the simulation. From the table, it could be seen that as the size of text kept increasing, CP-ABE and KP-ABE used more time to perform encryption. However, the table further reveals that ECP-ABE used relatively fairly less amount of time to encrypt the same size of text that CP-ABE and KP-ABE encrypted. This is because, it is faster to perform encryption on small blocks of text organized as access structures than to encrypt large blocks of text since more CPU time is required to perform ciphering of large blocks of text. CP-ABE just as KP-ABE, does not define access structures on small modularized files sizes, rather, they define access structures on an entire document. Processing large documents need more CPU resources and more computer memory to be encrypted. The data in Table 1 is also represented in Fig 2

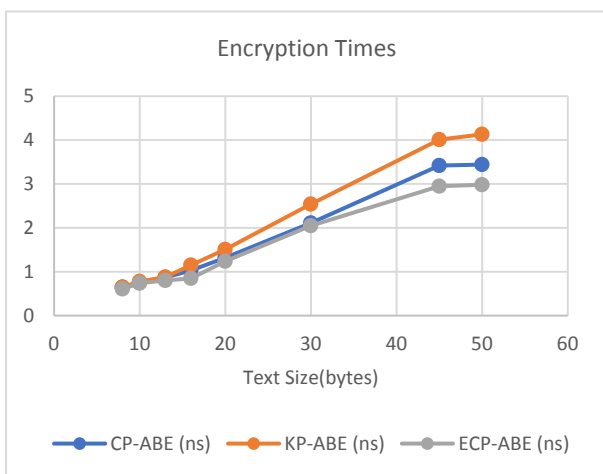


Fig.2. Encryption Times for CP-ABE, KP-ABE and ECP-ABE Mechanisms

B. Simulation Results for Decryption Times

Table 2. Decryption Times Based On CP-ABE, KP-ABE and ECP-ABE Mechanisms

Text Size (bytes)	CP-ABE (ns)	KP-ABE (ns)	ECP-ABE (ns)
8	0.23	0.22	0.23
10	0.61	0.63	0.43
13	0.73	0.75	0.54
16	0.89	0.91	0.65
20	1.19	1.17	0.91
30	1.49	1.56	1.25
45	2.23	2.33	1.84
50	2.31	2.48	1.98

Table 2 describes the decryption times for CP-ABE, KP-ABE and ECP-ABE during the simulation and is also shown in Fig 3. KP-ABE required more CPU time and resources to perform decryption as the size of text increases. This was because, KP-ABE required private keys to be set on specific access structures. CP-ABE used fairly the same amount of time to perform decryption compared with KP-ABE. Because CP-ABE organizes access structure within large blocks of texts as is the case of KP-ABE, CP-ABE required fairly similar CPU resources to perform decryption. Comparing ECP-ABE to CP-ABE and KP-ABE revealed that, ECP-ABE used less CPU resources to perform decryption of text. As the size of text increased, ECP-ABE used less time to decrypt texts. The reasons are that, because ECP-ABE eliminates the need to use private keys to be set on specific access structures and also creates manageable blocks of text (access structures) into small individual files, decrypting these small file sizes needed less CPU time and resources.

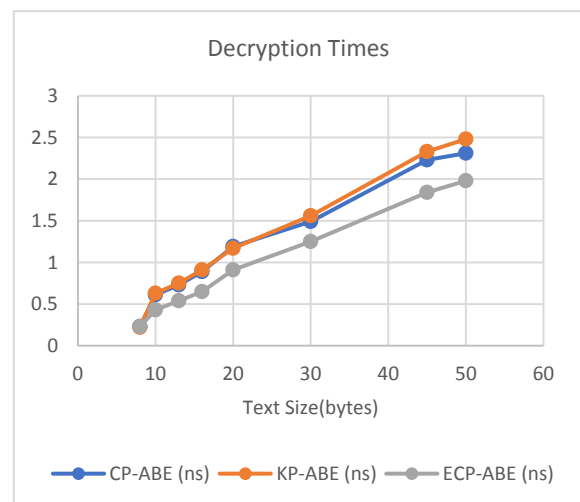


Fig.3. Decryption Times for CP-ABE, KP-ABE and ECP-ABE Mechanisms

C. Simulation for Memory Utilization (Encryption)

Table 3 and Fig 4 shows that both CP-ABE and KP-ABE needed large memory space to perform encryption. ECP-ABE used small amount of memory spaces to

perform the same cryptographic function using the same size of text. Comparatively CP-ABE and KP-ABE used more memory space because, both mechanisms deal with large documents (texts), and both set different access structures (policies) on different portions of an entire documents. The whole document is therefore kept in memory during this process thus occupying more memory space. Unlike CP-ABE and KP-ABE, ECP-ABE allowed users to define separate individual small access structures from health records. Only the necessary columns (data pages) needed to create access structure specifications were used. Due to this reason, file sizes in ECP-ABE were tiny and therefore occupied less amount of space in the computer’s memory.

Table 3. Memory Utilization for CP-ABE, KP-ABE and ECP-ABE during Encryption

Text Size (bytes)	CP-ABE (KB)	KP-ABE (KB)	ECP-ABE (KB)
8	4866	4727	4270
10	4872	4734	4278
13	4885	4743	4284
16	4896	4756	4293
20	4918	4768	4311
30	4937	4787	4331
45	4962	4812	4356
50	4975	4825	4364

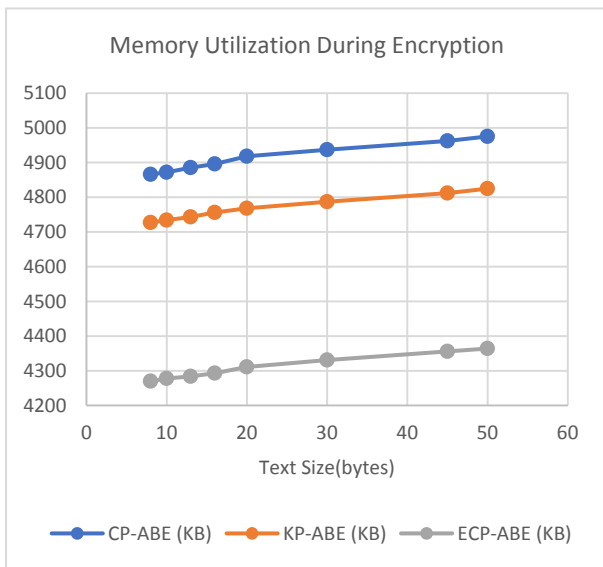


Fig.4. Memory Utilization for CP-ABE, KP-ABE and ECP-ABE during Encryption

D. Simulation for Memory Utilization (Decryption)

From Table 4, it was realised that decrypting blocks of files used small memory spaces during the decryption simulation when ECP-ABE was used. ECP-ABE by its design required small access structures to be carved out of an entire file, ECP-ABE therefore required minimal CPU time to perform decryption.

Table 4. Memory Utilization for CP-ABE, KP-ABE and ECP-ABE during Decryption

Text Size (bytes)	CP-ABE (KB)	KP-ABE (KB)	ECP-ABE (KB)
8	5023	5069	4092
10	5059	5086	5025
13	5084	5097	5048
16	6011	6021	5065
20	6042	6061	5088
30	6069	6097	6021
45	7015	7028	6074
50	7031	7040	6092

CP-ABE and KP-ABE on the other hand, works quite the opposite. Both mechanisms use large files, and also require private keys to create access structures(s) which is included in the same document. CP-ABE and KP-ABE therefore took much time to decrypt the files and the private keys when the simulator was used. A graphical representation is given in Fig 5

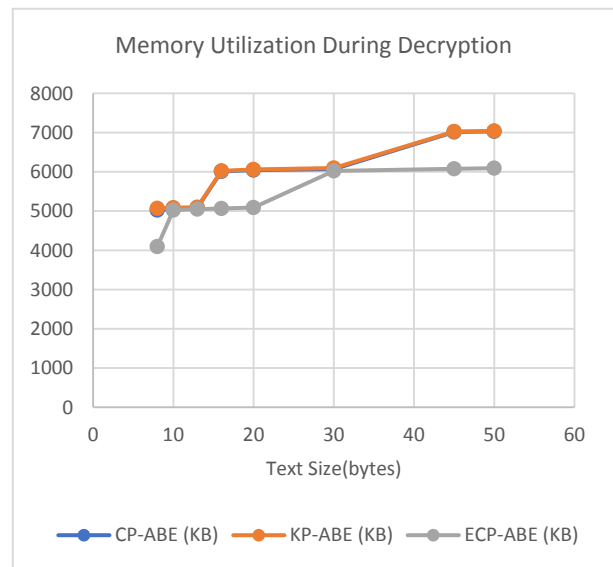


Fig.5. Memory Utilization for CP-ABE, KP-ABE and ECP-ABE during Decryption.

E. Simulation for Throughput (Encryption)

Table 5. Throughput for CP-ABE, KP-ABE and ECP-ABE during Encryption

Text Size (Bytes)	CP-ABE (KB/Sec)	KP-ABE (KB/Sec)	ECP-ABE (KB/Sec)
8	7727.69	7732.81	6602.25
10	6743.33	6482.05	6790.54
13	5843.68	5792.05	5938.82
16	5835.92	5235.65	5958.82
20	3062.12	4013.91	4103.22
30	2876.30	2400.39	2937.07
45	2015.17	1752.62	2038.26
50	2020.42	1704.60	2044.23

Table 5 and Fig 6 describes the throughput for CP-ABE, KP-ABE and ECP-ABE. The table shows that encryption throughput falls with respect to size of text. The table further reveals that, generally ECP-ABE recorded marginally high encryption throughput as the file size begun to increase, Reason is, ECP-ABE deals with small sized access structures at a time and because access structures are defined and stored separately, encrypting such file sizes yield high throughput. In the case of both CP-ABE and KP-ABE, files sizes are relatively larger when compared with file sizes of ECP-ABE. This makes encrypting files with CP-ABE and KP-ABE use more CPU time and resources hence yielding low throughput.

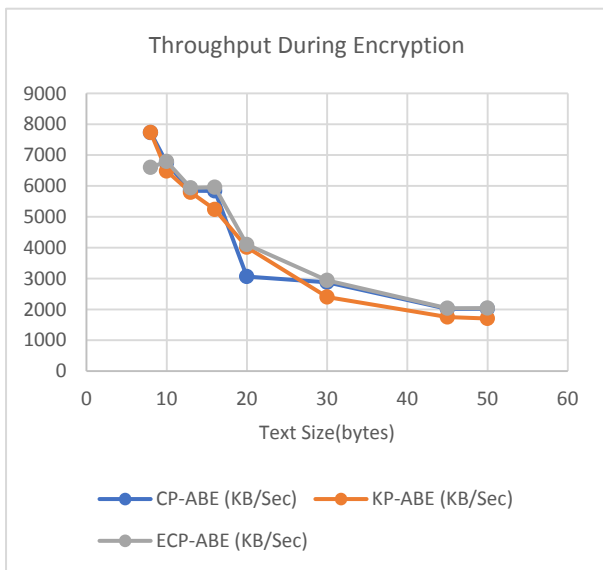


Fig.6. Throughput for CP-ABE, KP-ABE and ECP-ABE during Encryption.

F. Simulation for Throughput (Decryption)

Table 6. Throughput for CP-ABE, KP-ABE and ECP-ABE during Decryption.

Text Size (Bytes)	CP-ABE (KB/Sec)	KP-ABE (KB/Sec)	ECP-ABE (KB/Sec)
8	21156.52	21486.36	18565.22
10	15716.13	13525.71	15278.28
13	11360.47	10310.87	12600.00
16	9066.67	8343.86	9540.00
20	6557.33	6192.21	8133.96
30	5252.13	4835.35	5698.68
45	4470.27	4181.35	4188.46
50	4111.57	3799.21	3967.27

Table 6 and Fig 7 describes results of simulation for throughput of CP-ABE, KP-ABE and ECP-ABE during the decryption throughput simulation. The illustration revealed that, comparatively, ECP-ABE generally has fairly high decryption throughput over CP-ABE and KP-ABE when files sizes begin to increase. Minimal CPU time and resources are needed to perform decryption of

small file for processing. Because ECP-CP works on small files sizes, it has a better decryption time hence yields high throughput.

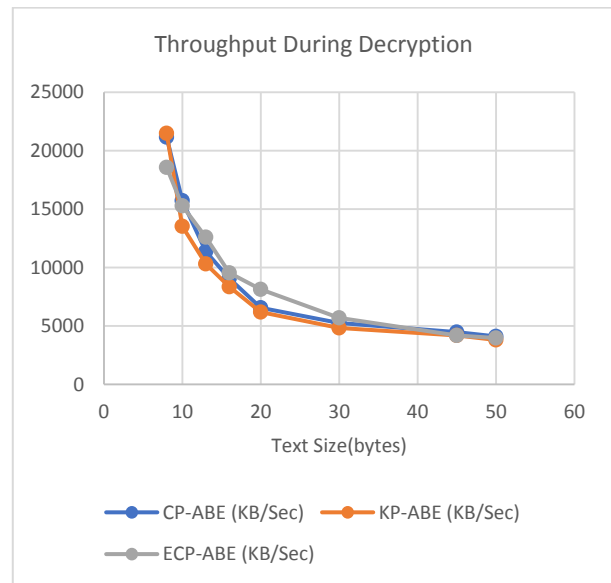


Fig.7. Throughput for CP-ABE, KP-ABE and ECP-ABE during Decryption.

G. Discussion

In course of the research, it was found out that;

CP-ABE and KP-ABE functions almost similar to each other. Both encryption mechanisms need more CPU time to encrypt and to decrypt texts. This is because both technologies create large record sizes. Records were embedded with Access Structures; a record could define as many Access Structures. This makes CP-ABE and KP-ABE create larger file sizes hence use more time for encryption and/or decryption. Also, both mechanisms require more memory space during either encryption or decryption, this was yet due to the large records that were created under CP-ABE and KP-ABE regimes.

ECP-ABE mechanism proves to have high performance capability. Because ECP-ABE defines small manageable units of Access Structures, it requires less system resources. From the experiments conducted, ECP-ABE required less CPU time to perform encryption and/or decryption. It needed relatively minimal memory space and yielded better throughput with larger text sizes.

Prevention of collusion attack is one advantage with CP-ABE was achieved also with ECP-ABE. CP-ABE uses oracles of randomization to generate distinct private keys which are used to further encrypt Access Structures in a file (health record). Therefore, it is impossible to combine attributes of multiple guests to retrieve unauthorized Access Structures. During the research work, it came out that ECP-ABE mechanism could also prevent collusion attack by separating each defined Access Structure from the next. Users therefore, could not combine attributes to retrieve information which they are not permitted to.

Throughout the research, it became apparent that users

are averred towards use of EHR systems because of security reasons. Existing cloud-based EHR systems are susceptible to malicious attacks, storing sensitive information such as health records in the cloud meant that those records are open to everyone. Human involvement in storing and managing health records further poses security and confidentiality challenges; humans could be corrupted to divulge sensitive information. Current CP-ABE and KP-ABE are designed to be implemented on a single server. Both patient health data and patient personal information are stored on the same server, a successful single-point-of-attack could compromise user's health records. This design therefore presents security threats thus does not inspire confidence to using cloud-based EHR systems.

ECP-ABE implemented a different approach; a domain called escrow server is used to store and manage access structures and access permissions. The data server was concerned with storing only health records. Implementing this architecture in the software revealed that:

1. The escrow server which is separate from the data server, stored user attributes in a ciphertext so that those stored attributes made no meaning when hacked. Successfully attacking this server only revealed patient's personal information without corresponding patient's health records.
2. Data server domain stored only ciphered health information about users who were identifiable through the escrow sever domain. The health records stored on this server were secure when attacked, an attacker could only see ciphered health records without corresponding patient's personal information.

Separating attributes away from health records proves ECP-ABE to be more secure and confidential than CP-ABE and KP-ABE.

V. CONCLUSION

ECP-ABE encryption mechanism could be adopted in the construction of cloud-based EHR systems to leverage the high-performance standard that the mechanism presents. Using ECP-ABE, an EHR system needs less CPU time, less computer memory and yield high throughput during encryption and/or decryption operations to process increasing user requests.

The ECP-ABE mechanism creates small units of Access Structures and eliminates need for private keys to encrypt and/or decrypt Access Structures. This greatly improve performance of ECP-ABE yet does not compromise security; collusion attacks are prevented by the separation of Access Structures.

Cloud-based EHR system could adopt the implementation architecture that ECP-ABE presents. The split design separates health records away from user's personal login information on two different servers so that, attacking any of the storage facilities will not lead to

compromising security of health records.

ECP-ABE overall helps to improve data privacy and boost confidentiality in cloud-based EHR systems.

REFERENCES

- [1] Jafari, M., Safavi-Naini, R. & Sheppard, N. P., A Rights Management Approach to Protection of Privacy in a Cloud of Electronic Health Records. Chicago, Association for computing Machinery, pp. 23-30, 2011.
- [2] Brino, A., Cloud still sparks fear of breaches. [Online] Available at: <http://www.healthcareitnews.com/news/cloud-still-sparks-fear-breaches>, 2014.
- [3] Rodzinka, M.. United States Legislation and HIPAA. In: Cross-Enterprise Access Control: Security for Electronic Health Records: Technical, Practical and Legislation Impact. Rochester, New York: s.n., pp. 7-11, 2012.
- [4] Rouse, M. & Pawliw, B., cryptography. [Online] Available at: <http://searchsoftwarequality.techtarget.com/definition/cryptography>, 2014.
- [5] Szolovit, P. et al., Guardian Angel: Patient Centered Health Information Systems. Massachusetts Institute of Technology Laboratory for Computer Science, 1994.
- [6] Narayan, S., Gagné M. & Safavi-Naini, R., Privacy Preserving EHR System Using Attribute-based Infrastructure. Canada: University of Calgary, Alberta, Canada, 2010.
- [7] Chien-Ding, L., A Cryptographic Key Management Solution For HIPAA Privacy/Security Regulations, 1994.
- [8] Sahai, A. & Waters, B., Fuzzy Identity Based Encryption. pp. 469-472, 2005.
- [9] Goyal, V., Pandey, O., Sahai, A. & Waters, B., Attribute Based Encryption for Fine-Grained Access Control of Encryption Data. Virginia, 2006.
- [10] Ibraimi, L. et al., Ciphertext-Policy Attribute-Based Threshold Decryption with Flexible Delegation and Revocation of User Attributes (extended version), Enschede: UT Publications, 2009.
- [11] Scholl, M., Stine, K., Lin, K. & Steinberg, D., Draft Security Architecture Design Process For Health Information Exchanges. Gaithersburg, MD: National Institute of Standards and Technology, 2009.
- [12] Microsoft Corporation, A Brief Introduction to Role-Based Access Control – Part 1. [Online] Available at: <http://blogs.technet.com/b/nexthop/archive/2010/06/06/refrbac1.aspx>, 2010.
- [13] Li, M., Yu, S., Ren, K. & Lou, W. Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-owner Settings. Security and Privacy in Communication networks, Volume 50, 2010.
- [14] Janssen, C., Attribute-Based Access Control (ABAC). [Online] Available at: <http://www.techopedia.com/definition/29706/attribute-based-access-control-abac>, 2015.
- [15] Bethencourt, J., Sahai, A. & Waters, B. Ciphertext-Policy Attribute-Based Encryption. Los Angeles: IEEE Computer Society, 2007.

Authors' Profiles



Robert French-Baidoo is a researcher at Takoradi Technical University in the Department of ICT. His main research interests include cloud computing and network security



Stephen Opoku Oppong received his Bsc degree in Actuarial Science from Kwame Nkrumah University of Science and Technology (KNUST), Ghana in 2012 and Masters of Philosophy (MPhil) degree in Information Technology also from KNUST in 2015. His research areas include Algorithms and Mathematical Computations.



Dominic Asamaoh received his BSc and MPhil Degree in Computer Science from Kwame Nkrumah University of Science and Technology (KNUST), Ghana. He is a Lecturer in the Department of Computer Science, KNUST. He has an extensive career of over 15 years teaching experience in Computer Science. Research areas include Image processing, Data Structures and Algorithms and Computer Systems Architecture.

How to cite this paper: Robert French-Baidoo, Dominic Asamaoh, Stephen Opoku Oppong, "Achieving Confidentiality in Electronic Health Records using Cloud Systems", International Journal of Computer Network and Information Security (IJCNIS), Vol.10, No.1, pp.18-25, 2018. DOI: 10.5815/ijcnis.2018.01.03