An Access Control Model for Avoiding Outsourcing Risks

A.Meligy
Departement of Mathematics and Computer Science
Faculty of Science
Menofiya University, Shebin EL-Koom, Egypt.

H.DIAB
Department of Mathematics and Computer Science
Faculty of Science
Menofiya University, Shebin-Elkoom 32511, Egypt

M.Torky
Departement of Mathematics and Computer Science ,
Faculty of Science
Menofiya Univerity, Ashmoon,Egypt

ABSTRACT

Although the security Outsourcing companies can provide several security services to its customer organizations, but the customer organizations should avoid Outsourcing security risks which emerge from providing security services to the customers through open environments or malicious behaviors which the security providers at Outsourcing companies may carry out. For this problem we propose a new methodology represented in a security design model which combine Cryptography and Access Control techniques to prevent the external security providers of an Outsourcing company to access the sensitive data assets of the customer organizations. We could achieve the realism of this methodology through a proposed algorithm in MATLAP Language. Using our new access control model, the customer organizations can control and manage the external access rights of security providers of specific Outsourcing Company which the customer organization communicated with it

.General Terms

Security, Outsourcing, and Cryptography

Keywords

Outsourcing, Security Providers, Encryption /Decryption, access control, Two layer encryption model, access policy changes, AES algorithm, MATLAP Language.

1. INTRODUCTION

Several organizations depends on an external security outsourcing companies for providing security services and protecting their local data assets from malicious intruders across the Internet **[1]**. But these organizations should consider security risks of outsourcing companies especially its security providers who can carry out malicious access to the customer organizations, and subvert its sensitive data assets **[2].**In most Outsourcing transactions, the customer organizations transfer to a third party the total responsibility to handle security tasks or to manage security functions, this transition to security outsourcing companies will make valuable confidential information, trade secrets, personal contract information and sensitive financial or medical information available to the third party who can perform malicious behaviors against this sensitive assets **[3]**.

Therefore the customer organizations which communicated to security outsourcing company should have an access control technique enable them to prevent the malicious access of the external security providers and enable them to manage the access policy on their internal data assets. the traditional access control architectures assign a crucial role to the *reference monitor* **[4]**. for ensuring data confidentiality. The reference monitor is the system component responsible of the validation of access request in a closed domain systems such as in the Operating Systems. But, since the security outsourcing companies provide security services over open environments and open domains, so the reference monitor doesn't appropriate in Outsourcing approach. Hence, access control architecture can be based on the Cryptography as another security technique **[5]** and **[6].** Cryptography can be considered as a tool that transforms information in a way that its protection (i.e. confidentiality and integrity) depends only on the correct management of compact secret (i.e. the Key).Cryptography is typically used when information is transmitted on a channel, with the assumption that the channel lies outside of the trust boundary of the system. The aim of this paper is to introduce new access control as a security design model which depends on the cryptography as an efficient tool for enforcing the access policy. The customer organizations which communicated to external outsourcing companies can use this model to protect their data assets from malicious or unauthorized access of the external security provider **[6]**.

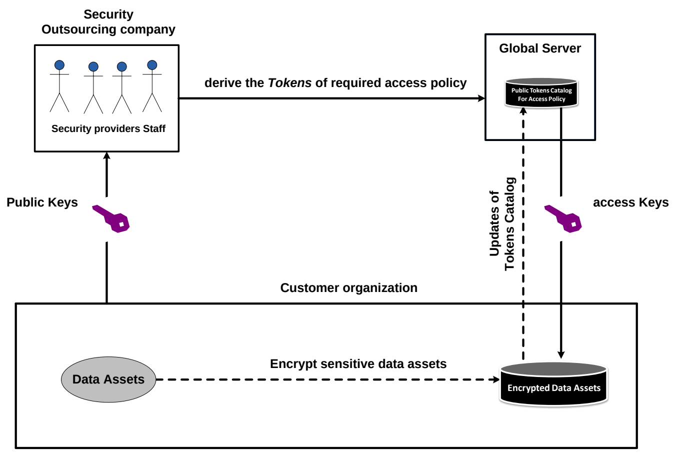
1.1 Related Work

The idea of combining access control and cryptography is in itself not new and has been suggested in different places. For instance, it was proposed in the context of: distributed file system **[7]**, where a classical client/server architecture is used; access control in a hierarchy **[8], [9]** where a centralized architecture is adopted; securing XML documents **[16],** where a data publishing" architecture is considered and where the data owner totally loses the control on her data. The combination of access control and cryptography in an outsourced architecture is introduced in **[10]** such that the authors introduces new model for enforcing authorization policies and supporting dynamic authorization and allowing policy changes on the outsourced resources which are available to several users across Internet such as YouTube, and Amazon Storage Services, etc. Previous work most directly related to ours is in the database outsourcing area**[11], [12].** In such a context, most proposals focused on query execution and propose to store additional indexing information together with the encrypted database. Such indexes can be used by the DBMS to select the data to be returned in reply to a query **[13], [14], [15].** Adding a traditional authorization level to the outsourcing scenario requires that when a client poses a query, both the query and its result has to be filtered by the data owner, who is in charge of enforcing the access control policy. A first attempt to solve this issue has been presented in

[16], where different keys are used for encrypting different data.

2. THE GENERAL ARCHITECTURE OF THE PROPOSED ACCESS CONTROL MODEL

Our new model architecture depends on cryptography as an efficient tool can be used for access control of security outsourcer employees (i.e. security providers) on the sensitive assets within the customer organization. This architecture for the realization of cryptographic access protection are simply based on the use of one distinct key for the encryption of the sensitive assets within the customer organization which the customer want to grant to or prevent from security providers at Outsourcing company. The architecture should has a security protocol between the customer organization and the external security providers of the outsourcing company. the full realization of the cryptography-based access control architecture for secure data assets within the customer organization has to correctly manage different important requirements before it could be deployed in large scale applications with large security provider communities. First, different security providers may need to see or access different portion of data assets within the customer organization. To give security providers different access rights, the data asset owner (i.e. the customer organization) cannot then use a single key for encrypting the whole data assets; Instead, the data asset owner should be able to define access authorizations and to map them on the data through a *multi-key encryption* approach **[17]** and **[6]** Security providers use then multiple keys to extract from the query results the plaintext data they are entitled to see or access. Second, it is necessary to analyze the issue of realizing an efficient management of access policy updates, presenting a mechanism that should be able to offer a robust way between the customer (i.e. data asset owner) and security providers at the security outsourcing company. Fig. 1, illustrate the general architecture of the proposed access control model that consist of two parties: the customer organization which communicated to an external outsourcing company, and the security outsourcing company that provide security services to customer organization and may threaten the customer organization in the same time. Also, the model make up of two distinct repositories: one repository (repository at the bottom) to store the encrypted representations of the customer's data assets. the other repository is dedicated to the storage and management of the access policy expressed in a way that can be managed by *a global server* which connect the customer organization with the security outsourcing company, with adequate protections that guarantee that a malicious security provider or any external intruders cannot gain unauthorized access to data assets within the customer organization.


Security Outsourcing company
Security providers Staff
derive the *Tokens* of required access policy
Global Server
Public Tokens Catalog For Access Policy
Public Keys
Updates of Tokens Catalog
access Keys
Customer organization
Data Assets
Encrypt sensitive data assets
Encrypted Data Assets

.

Fig.1: The general architecture of the proposed access control model.

The Basic idea in our proposed access control model is:
each security provider in the security Outsourcing company which the customer organization connected to it, assigned a key by the data asset owner (i.e. the customer organization) and then a set of *tokens* is used to allow the derivation from one key to another key. *The token catalog* which resided in the global server is demonstrated to be usable only by set of security providers who already having access to the secret (i.e. only set of authorized security providers). As the Fig.1 shows, the customer (i.e. data assets owner) first, sends to each security providers at security outsourcing company a secret key, then the customer organization encrypt its required sensitive data assets and store the encrypted representation in an encrypted data repository, and send the corresponding representation of public tokens to the public token catalog that resides in the global server. hence, there is no external security provider can access or decrypt the encrypted data asset repository if and only if could compute the required *tokens* of specific *access key.*

3. THE PROPOSED ACCESS CONTROL MODEL- BASED ON TWO ENCRYPTION LAYERS.

To make the proposed Model generally applicable, we will follow a similar encryption methodology that explained in **[6]** which based on the key derivation process that introduced in **[18]** but with different assumptions. For example, suppose we have two sets ***SP*** and ***A* ,** such that ***SP*** is a set of security providers at Outsourcing company and **A** is a set of assets within the customer organization which the customer organization want to enforce the specific access policy on them. For instance ***SP***={ SP1, SP2,SP3……….SPn} and ***A***={ Asset a1, Asset a2, Asset a3,…………Asset am }.An *authorization policy* is a set of pairs of the form **(sp, a)** where **sp** ϵ **SP** and **a** ϵ **A** meaning that a security providers **sp** is authorized to access Asset **a.** An access policy can then be modeled via a traditional *access matrix X* [**6**]. The rows in access matrix represents the set of Assets within the customer organization (i.e. for each **a** ϵ **A)** and the columns represents the set of security providers at the outsourcing companies (i.e. for each **sp** ϵ **SP**).and ***X(sp, a)*** is set to '1' if specific security provider ***sp*** can access asset **a**, and set to '0' otherwise. also, each asset ***a*** within the customer organization should has an *access control list* **(** ***acl*(a))** which involves only the set of authorized security providers who can access asset ' ***a'***. Table 1 illustrates an example of access matrix with four**.** security providers (s*p1*, *sp2*, *sp3*,*sp4*) and six assets (*a1*, *a2*, *a3*,*a4*,*a5*, *a*6).

Table 1: An example of access matrix of four security providers and six data assets.

	a1	**a2**	**a3**	**a4**	**a5**	**a6**
Sp1	1	1	1	1	0	1
Sp2	0	0	0	0	1	1
Sp3	0	1	1	1	1	1
Sp4	0	0	0	0	1	1

The encryption methodology in our proposed access control model based on the *Key derivation method* which introduced in **[4].** This method exploits the definition and computation of *public tokens* that allow the derivation of keys from other keys. Let ***ki*** and ***kj*** be two keys for two security providers within outsourcing company, A token, denoted ***Ti,j*** , that allows the derivation of ***kj*** from ***ki*** is defined as:

$$T_{i,j} = K_j \oplus h(K_i, l_j)$$

Where ***lj*** is a publicly available label associated with ***kj*** and " $\oplus$ " is bit-a-bit xor operator, and '***h***' is a secure hash function(e.g. HMAC **[16]**). The key derivation process can be applied via a chain of tokens **T*i,l*,**..... ***Tn,j*** such that **T*c,d*** follows ***Ta,b*** in the chain if **d=a**. Graphically, a set of keys ***K*** and a set of tokens ***T*** can be represented through *a graph*, having a vertex ***vi*** associated with each key in the system, denoted ***vi.k***, and an edge **(*vi, vj*)** connecting ***vi*** and ***vj*** if token ***Ti,j*** belongs to ***T*** . Chains of tokens then correspond to *paths* in the graph. **[6].**

Our proposed model can be described through two encryption layers as shown in Fig.2:-

1-*local Encryption layer (LEL):*- in which the customer organization encrypt the sensitive data sets according to the local access policy (i.e. according to the customer's access policy on his data assets which he want to grant to or revoke from the hand of the external security providers).

2-*global Encryption layer (GEL):*- in which the global server which directly connected to the outsourcing company perform additional encryption layer over the encrypted data assets which already encrypted at local encryption layer (LEL) within the customer organization.

The rational in the two-layer encryption methodology in our proposed access control model as described in Fig 2 state that: each security provider at the outsourcing organization receive two keys, one to decrypt specific assets through Global Encryption Layer (GEL) and the other to decrypt this asset through Local Encryption Layer (LEL). In the encryption side the LEL encrypted first, but in the decryption side the GEL decrypted first. At each layer, each data asset is encrypted by using a single key. Also, sometime each security provider may receive only one key to access LEL and the key at GEL can be computed from the key at LEL layer by applying on it a hash function. As Fig. 2 illustrated, *in Local Encryption Layer (LEL)* the customer's data assets are encrypted according the local access policy which the customer organization specify. In LEL we have two kinds of keys: *access keys* are used to encrypt the customer's data assets which the customer want to protect against security risks of outsourcers, and *derivation keys* are used to provide the derivation capability via tokens. Each derivation key ***k*** is always associated with an access key ***k*a** that is obtained through a secure hash function from ***k*:**

$$K_a = h(K)$$

The rationale for this evolution is to distinguish the two roles associated with keys: enabling key derivation via the corresponding *tokens,* and enabling data asset access. It is then up to the customer organization to define required access policy and to generate the corresponding encryption keys and tokens in such a way that each security provider can derive the *right set of decryption keys*, meaning that each security provider can decrypt all and only the data assets for which he/she has the Authorization. The resulting is set of keys and tokens called LEL *encryption policy*.

Graphically, Local Encryption Layer (LEL) can represented as a graph to explain the key derivation method as described in Fig.3. where now there is a vertex ***L*** for each pair of keys ***(k, ka)*** and an edge **(L*i, Lj*)** connects ***Li*** and ***Lj*** if there is a *token* in the public token catalog allowing the derivation of either ***Lj .k*** or ***Lj*** **.*k*a** from ***Li.k.*** as we see in Fig. 3 which illustrates and graphically represent Local Encryption Layer (LEL) according to the access policy that defined in Table 1, Fig. 3(a) shows the graph representing the key derivation relationships, Fig. 3(b) shows the keys communicated to each security provider at outsourcing organization and Fig. 3(c) shows the keys used for encrypting the data assets of the customer organization. It is easy to see that, for example, since security provider **SP1** knows key ***L*1.*k*** and is able to derive tokens of keys ***L*5.*k*** and ***L*7.*k*** from the *public token catalogue* that in turn allow the computation of access keys ***L*1.*k*a, *L*5.*k*a**, and ***L*7.*k*a** (such that, Ka=h(k)),so, ***SP1***is allowed to access the set of data assets ***a*1,*a*2,*a*3,*a*4,*a*6**, which are exactly the data assets that security provider ***SP1*** is authorized to access in the customer organization.

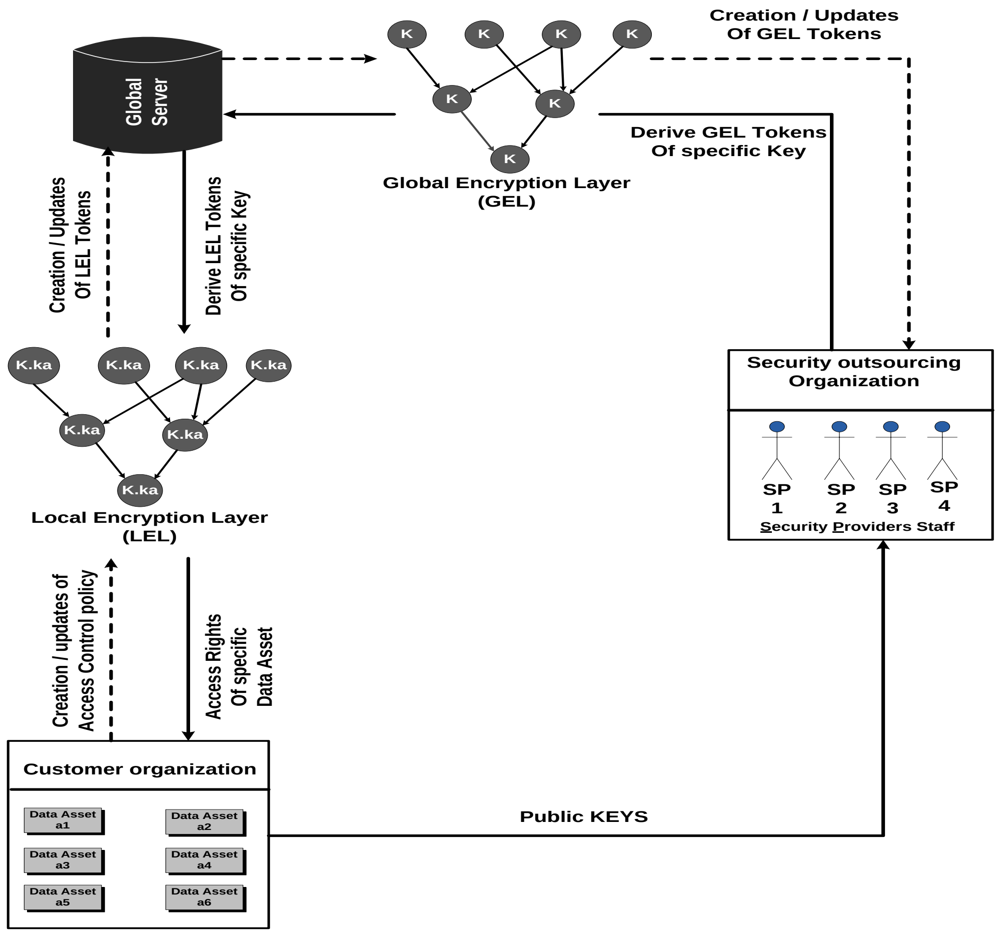
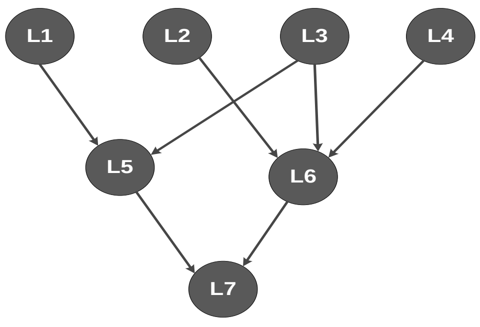


Fig 2: The proposed model as two encryption layer model



(A)

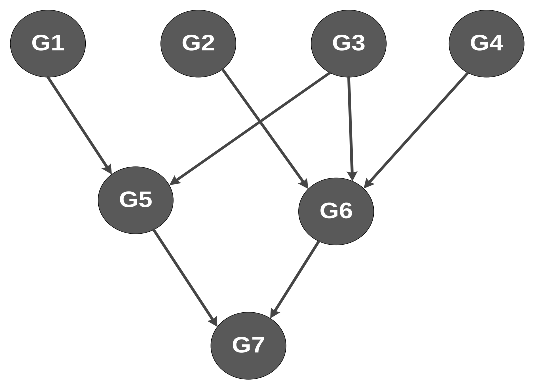
Security Providers SP	Personal Keys
Sp 1	L1.k
Sp 2	L2.k
Sp 3	L3.k
Sp 4	L4.k

(B)

Customer's assets	access Keys
Asset (a1)	L1.ka
Assets (a2,a3, a4)	L5.ka
Asset (a5)	L6.ka
Asset (a6)	L7.ka

(C)

Fig.3: LEL encryption access policy enforcing the authorization policy in Table 1.



(A)

Security Providers SP	Personal Keys
Sp 1	G1.k
Sp 2	G2.k
Sp 3	G3.k
Sp 4	G4.k

(B)

Customer's assets	access Keys
Asset (a1)	G1.ka
Assets (a2,a3, a4)	G5.ka
Asset (a5)	G6.ka
Asset (a6)	G7.ka

(C)

Fig. 4: GEL encryption access policy enforcing the authorization policy in Table 1.

Again, in *Global Encryption Layer (GEL)* which performed at the global server which the customer organization connected to the external security Outsourcing company through it, for each pair of keys ***(k, ka)*** in LEL a corresponding ***key*** is defined in GEL, for each *token* in LEL, a *corresponding token* is defined in GEL, The key derivation relationship is then represented through a graph that is *isomorphic* to the graph existing at the LEL level. Each security provider is assigned the unique key corresponding to his key at LEL. Each data assets is encrypted with the unique key corresponding to the unique access key at LEL layer. So the Global Encryption Layer (GEL) model exactly the LEL policy according to the original authorization policy, this strategy called FULL_GEL which state that each item in LEL level has an isomorphic item in GEL level. Fig. 4 illustrates and graphically represents Global Encryption Layer (GEL) according to the original authorization policy that defined in Table 1 and graphically modeled in LEL in Fig. 3.

3.1 Access Policy Updates Enforcement

For more control capabilities of customer organization against security risks of external security providers and outsourcing companies, we proposes new strategy called *Delta_GEL* which enable the customer organization to make changes and updates in access policy enforcement. The changes in access policy involve *the insertion/deletion* of specific security providers within authorization policy or *grant/revoke* specific data assets of customer organization. We focus on *grant/revoke* operations. **[6],**

1-Grant *Operation*:- is performed when the customer organization want to make specific data asset is *authorized* to specific security provider at the outsourcing company which the customer organization is communicated with it. *The rational* to perform ***grant operation*** judge that to make specific data asset ***a*** is authorized to specific security provider ***sp***, the status of this data asset ***a*** should be compared with the current configuration of access policy in Local Encryption layer (LEL) and Global Encryption Layer (BEL). So the element **X[sp,a]** in access matrix X is set to 1 (*i.e.* $acl(a) = acl(a) \cup \{sp\}$) and if security provider ***sp*** cannot derive the LEL key used for encrypting ***a,*** a LEL *token* from the LEL key associated with ***sp*** is added to the *access key* which used to encrypt the data asset ***a***. then the Global Encryption Layer (GEL) receives the new encryption method request about granting the data asset ***a*** for security provider ***sp*** to synchronize the GEL layer with the new access policy update about data asset ***a*** and security provider ***sp***. meaning that the graph corresponding to the key derivation relationship needs to be properly updated to reflect the new policy **[17].** To this aim, it may be necessary to create a new GEL key, along with the tokens for its derivation. Analogously, whenever a GEL key is not used for encrypting any data asset, it can be removed from the layer, along with the corresponding public tokens.

2-*Revoke Operation*:- is performed when the customer organization want to make specific data asset ***a*** *unauthorized* to an external specific security provider. *The rational* to perform ***revoke operation*** judge that to make specific data asset ***a*** became unauthorized to specific security provider ***sp***, the element **X[sp,a]** in access matrix **X** is set to 0 (*i.e.* $acl(a) = acl(a) - \{sp\}$) and the new encryption policy in LEL is sent to GEL to make *the data asset* ***a is*** accessible only to the new set of security providers that are authorized to access ***a.***

3.2 Grant / Revoke Operations Example

As an example, consider the initial configuration of both Local Encryption Layer (LEL) and Global Encryption Layer (GEL) encryption policies represented in fig. 3 and fig. 4 respectively. Fig. 5 , Fig. 6, Fig. 7, and Fig. 8 illustrate the evolution of both LEL and GEL encryption policies to accommodate a series of *grant* and *revoke* operations. Note that, in the graphical representation, *dashed edges* represent *tokens* leading to access keys.

- Example 1: *Revoke (a1, sp1):* in this example the customer organization want to remove the security provider ***sp1***from access control list of data asset ***a1*** (i.e. *acl*(*a*1) = *acl*(*a1*) - *{sp1}*). So, the *acl*(*a*1) becomes empty because it was involving only ***sp1*** as an authorized security provider to this data asset ***a1***. So the new encryption policy judge that a data asset ***a1*** has to be over encrypted with a key that no user can compute or derive through the graph representation in LEL and GEL. Consequently a new vertices **LT** and **GT** is created and its key is used to encrypt data asset ***a1,*** as described in Fig. 5.

- Example 2: *Grant (a4, sp4):* this in example the customer organization want to make the security provider ***sp4*** is authorized to access ***a4****(i.e. acl(a4)=acl(a4) U {sp4}),* but key ***L*5.*k*a** used to encrypt *data asset* ***a4*** cannot be derived from ***L*4.*k*** which assigned to security provider ***sp4.*** The customer organization therefore should modifies the encryption policy in Local Encryption Policy (LEL) by adding a new Vertex **L8** to the LEL along with two token a t ***T5,8*** and ***T4,8*** in LEL graph. Also , in Global Encryption layer (GEL) ***, G*5.*k*** used to over-encrypt *data asset* ***a4*** cannot be derived from ***G*4.*k*** assigned to security provider ***sp4***, therefore, a new vertex **G8** is added to the GEL along with two tokens, **T5,8** and ***T4,8***. Hence, the data asset ***a4*** is then over-encrypted by using ***G*8.*k*,** which is accessible to security providers ***sp1, sp3*** and ***sp4*** as described in Fig.6.

- Example 3: *Revoke (a6, sp1):* in this example the customer organization want to remove the security provider ***sp1***from access control list of data asset ***a6*** (i.e. *acl*(*a*6) = *acl*(*a*6) - *{sp1}*). This change means ***a*6** needs to be over-encrypted with a key accessible only to security providers ***sp2, sp3*** *and* ***sp4.*** Since ***L6.k*a** and ***G6.ka*** can be accessed by these three security providers only, therefore, data asset ***a6*** is over-encrypted by using ***L6.ka*** and ***G6.ka.*** After ***a*6** over-encryption, no resource is over- encrypted using ***L7.ka*** and ***G7.ka.*** Consequently, ***L7*** and **G7** is removed from the Local Encryption Layer (LEL) and Global Encryption Layer (GEL) respectively from the graph, as explained in Fig.7.

- Example 4: *Grant (a3, sp4):* in this example the customer organization want to make the security provider ***sp4*** is authorized to access ***a3****(i.e. acl(a3)=acl(a3) U{sp4})*. In LEL, key ***L*5.*k*a** used to encrypt ***a*3** can't be derived from ***b*4.*k*** assigned to ***sp4***, consequently, vertex **L8** in LEL which associated with ***L8.Ka*** should encrypt also ***a3*** which became accessible by ***sp1,sp3***, and ***sp4***, hence, ***L5.Ka*** will encrypt only ***a2***. On the other hand, in GEL level, key ***G*5.*ka*** used to over encrypt ***a3*** cannot be derived from ***G*4.*k***, assigned to ***sp4***. Data asset ***a3*** needs to be over-encrypted with a key derivable from ***sp1, sp3***, and ***sp4***. Since ***G8.ka*** is accessible by exactly these security providers, ***a3*** is over-encrypted by using ***G8.Ka,*** as described in Fig. 8.

4. ENCRYPTION / DECRYPTION CASE STUDY ON THE PROPOSED ACCESS CONTROL MODEL IN MATLAP LANGUAGE.

To achieve the realism of our proposed encryption methodology that represented in our proposed two layer encryption model, we produced an algorithm in MATLAP which forced by the Advanced Encryption Standard (AES) algorithm **[19]**. AES is a symmetric key block cipher which published by a national institute of standard and technology (NIST) in December 2001. AES is non feistel cipher that encrypt and decrypt data block of 128 bits through four functions, at the encryption site the four functions are :

(1) *Sub Byte function:* which used to substitute each byte as two hexadecimal digits at the encryption site

(2*) Shift Row function*: which used to permute and shift bytes in each row.

(3) *Mix Column function*: it operates at column level and transforms each column of specific state to a new column,

(4) *Add Round key function*. Adds round key words with each state column matrix , the operation in this function is matrix addition.

The produced algorithm design can be described in Fig. 9 that show the flow chart of Encryption side and Fig.10 that show the flow chart of Decryption side of our proposed two layer encryption model:

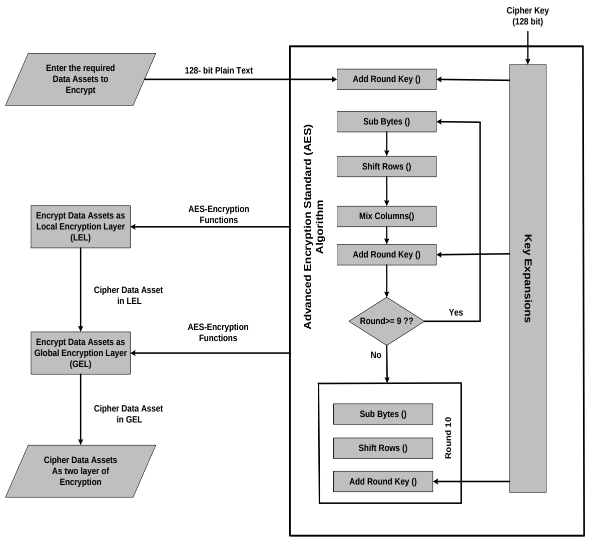
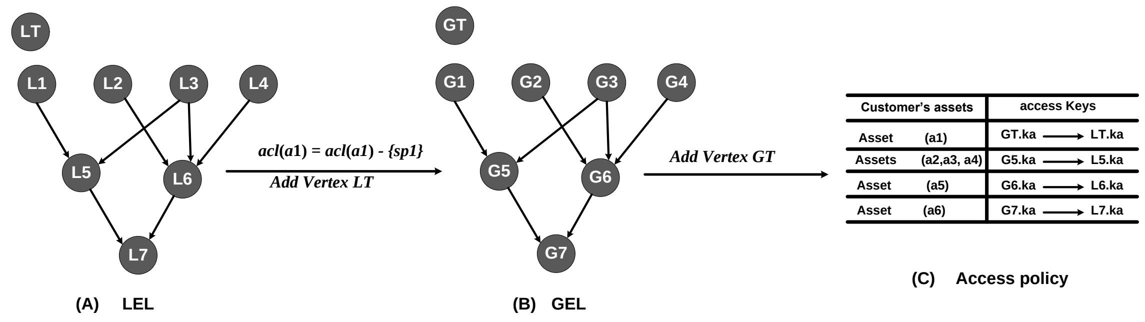


Fig.9: Flow Chart of the Encryption side of the proposed algorithm.

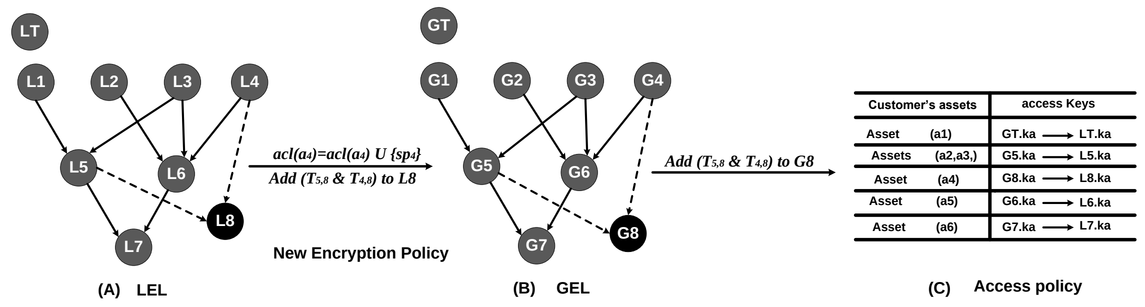
Example 1:



Customer's assets	access Keys
Asset (a1)	GT.ka $\longrightarrow$ LT.ka
Assets (a2,a3, a4)	G5.ka $\longrightarrow$ L5.ka
Asset (a5)	G6.ka $\longrightarrow$ L6.ka
Asset (a6)	G7.ka $\longrightarrow$ L7.ka

Fig. 5: Revoke (a1, sp1).

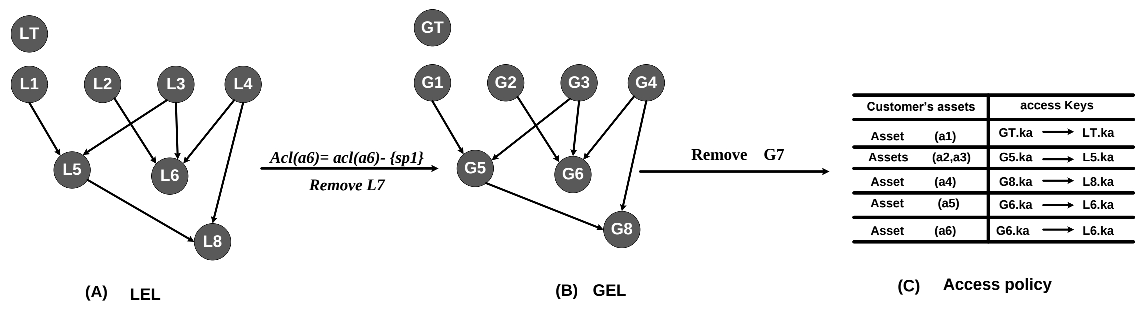
Example 2:



Customer's assets	access Keys
Asset (a1)	GT.ka $\longrightarrow$ LT.ka
Assets (a2,a3,)	G5.ka $\longrightarrow$ L5.ka
Asset (a4)	G8.ka $\longrightarrow$ L8.ka
Asset (a5)	G6.ka $\longrightarrow$ L6.ka
Asset (a6)	G7.ka $\longrightarrow$ L7.ka

Fig. 6: Grant (a4,sp4).

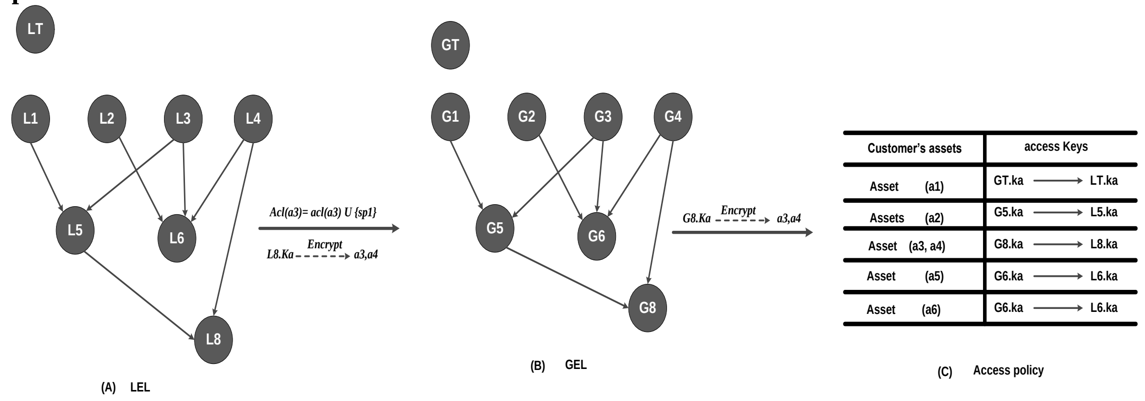
Example 3:



Customer's assets	access Keys
Asset (a1)	GT.ka $\longrightarrow$ LT.ka
Assets (a2,a3)	G5.ka $\longrightarrow$ L5.ka
Asset (a4)	G8.ka $\longrightarrow$ L8.ka
Asset (a5)	G6.ka $\longrightarrow$ L6.ka
Asset (a6)	G6.ka $\longrightarrow$ L6.ka

Fig. 7: Revoke (a6, sp1).

Example 4:



Customer's assets	access Keys
Asset (a1)	GT.ka $\longrightarrow$ LT.ka
Assets (a2)	G5.ka $\longrightarrow$ L5.ka
Asset (a3, a4)	G8.ka $\longrightarrow$ L8.ka
Asset (a5)	G6.ka $\longrightarrow$ L6.ka
Asset (a6)	G6.ka $\longrightarrow$ L6.ka

Fig. 8: Grant (a3,sp4)

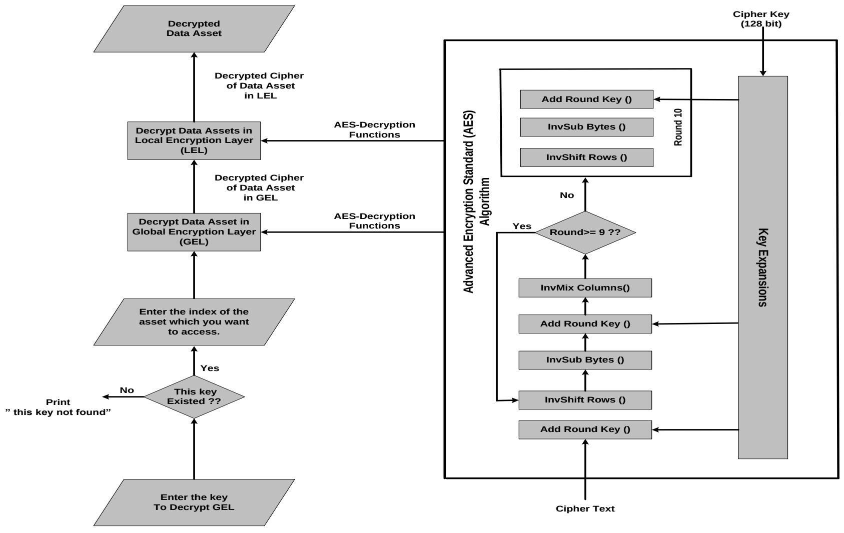

Decrypted Data Asset
Decrypted Cipher of Data Asset in LEL
Decrypt Data Assets in Local Encryption Layer (LEL)
AES-Decryption Functions
Decrypted Cipher of Data Asset in GEL
Decrypt Data Asset in Global Encryption Layer (GEL)
AES-Decryption Functions
Enter the index of the asset which you want to access.
Yes
No
Print " this key not found"
This key Existed ??
Enter the key To Decrypt GEL
Cipher Key (128 bit)
Advanced Encryption Standard (AES) Algorithm
Add Round Key ()
InvSub Bytes ()
InvShift Rows ()
Round 10
No
Yes
Round>= 9 ??
InvMix Columns()
Add Round Key ()
InvSub Bytes ()
InvShift Rows ()
Add Round Key ()
Key Expansions
Cipher Text

Fig.10: Flow Chart of the Decryption side of the proposed algorithm.

As we saw in Fig 10, at the decryption site there exist four functions to decrypt the ciphered data assets:

(1) *InvSubByte function*: used to reverse work of SubByte function.
(2) *InvShift Row function*: used to reverse work of Shift Row function.
(3) *Inv MixColumn function*: used to reverse work of Mixcolumn function.
(4) *Add Round Keyfunction*: used to adds round key words with each state column matrix.

We could implement our proposed algorithm in MATLAP Language to test our proposed algorithm's behavior; the output of one implementation case is shown in Fig 11:

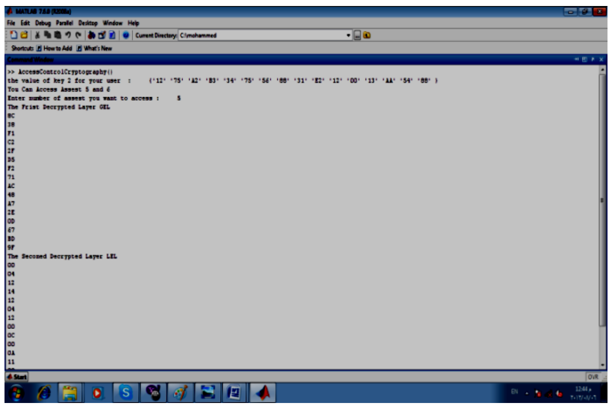

```
>> AccessControlCryptography()
the value of key 2 for your user  :     {'12' '75' 'A2' 'B3' '34' '75' '56' '88' '31' 'E2' '12' '00' '13' 'AA' '54' '88' }
You Can Access Assest 5 and 6
Enter number of assest you want to access :     5
The Frist Decrypted Layer GEL
8C
38
F1
C2
2F
D5
F2
71
AC
48
A7
2E
0D
67
BD
9F
The Seconed Decrypted Layer LEL
00
04
12
14
12
04
12
00
0C
00
0A
11
```

Fig 11: An application case of our proposed algorithm in MATLAP Language.

5. CONCLUSION

This paper introduced new access control system design model which combine cryptography and access control technique to enable the customer organizations that communicated with an external security outsourcing companies to manage and control in access policy of their internal data assets which want to protect against the malicious behaviors of the external security providers at Outsourcing Company. This paper also introduced an algorithm to implement the behavior of the proposed Access control model and to test its realism. The future work will involve more improvements in the proposed algorithm about data assets encryption and updating the algorithm to encrypt and decrypt the data assets which in the form images or sounds.

6. ACKNOWLEDGMENT

I would like to express my deep thanks to my colleague ***Marwa EL-Danaf*** who helped us to implement our proposed model using MATLAP language**.**

7. REFERENCES

[1] C.Warren, A, Computer security series: Outsourcing Information Security ARTECH HOUSE,INC (2004).

[2] Ian.Tho," Managing the Risks of IT Outsourcing", Computer Weeklly PROFFESSIONAL SERIES. 1 st, ed, Elsevier –Butterworth-Heinemann,UK (2003).

[3] (2009), The IT Law Group Web Site (Online) , Available, http://Itlawgroup.com / resources/ articles.

[4] J. Anderson. Computer security planning study.Technical Report 73-51, Air Force Electronic System Division, 1972.

[5] J. Saltzer and M. Schroeder. The protection of information in computer systems. *Communications of the ACM*, 17(7), July 1974.

[6] S. De CapitaniForestiJajodia, S. Foresti,and S. Jajodia," A Data Outsourcing Architecture Combining Cryptography and Access Control." Work shop, On computer security architecture. New York ,USA (2007).

[7] A. Harrington and C. Jensen. Cryptographic access control in a distributed ¯le system. In *Proc. of the 8th SACMAT*, Como, Italy, June 2003.

[8] S. Akl and P. Taylor. Cryptographic solution to a problem of access control in a hierarchy. *ACM TOCS*,1(3):239{248, August 1983..

[9] J. Crampton, K. Martin, and P. Wild. On key assignment for hierarchical access control. In Proc. of the 19th IEEE CSFW'06, Venice, Italy, July 2006.

[10] G. Miklau and D. Suciu. Controlling access to Published data using cryptography. In *Proc. of the 29th VLDB Conference*, Berlin, Germany, September 2003.

[11] H. HacigÄumÄus, B. Iyer, and S. Mehrotra. *Providing database as a service*. In *Proc. of 18th ICDE*, SanJose, CA, USA, February 2002.

[12] H. HacigÄumÄus, B. Iyer, S. Mehrotra, and C. Li. Executing SQL over encrypted data in the database-service-provider model. In *Proc. of the ACM SIGMOD 2002*, Madison, Wisconsin, USA, June 2002.

[13] R. Agrawal, J. Kierman, R. Srikant, and Y. Xu. Order preserving encryption for numeric data. In *Proc. of ACM SIGMOog2004*, Paris, France, June 2004..

[14] A. Ceselli, E. Damiani, S. De Capitani di Vimercati,,S. Jajodia, S. Paraboschi, and P. Samarati. *Modeling and assessing inference exposure in encrypted databases*. *ACM TISSec*, 8(1):119{152, February 2005.

[15] H. HacigÄumÄus, B. Iyer, and S. Mehrotra. Efficient execution of aggregation queries over encrypted relational databases. In *Proc. of the 9th International Conference on Database Systems for Advanced Applications*, Jeju Island, Korea, March 2004.

[16] E. Damiani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. *An* experimental evaluation of multi-key strategies fordata outsourcing. In Proc. of the 22nd IFIP TC-11 International Information Security Conference, South Africa, May 2007.

[17] E. Damiani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. *Key* management for multiuser encrypted databases. In Proc. of the International Workshop on Storage Security and Survivability, Fairfax, Virginia, USA,November 2005.

[18] M. Atallah, K. Frikken, and M. Blanton. Dynamic and efficient key management for access hierarchies. In *Proc. of the 12th ACM CCS05*, Alexandria, VA, USA,November 2005.

[19] William.S." Cryptography and Network Security Principales and Practices." 4 th ed, ,Prentic hall, USA, November,16, 2005.

.