Al-Rafidain Journal of Computer Sciences and Mathematics (RJCM)

www.csmj.mosuljournals.com

Multilevel Database Security for Android Using Fast Encryption Methods

Najla Badie AI Dabagh[1],* Mahmood S. Mahmood[2]

College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq [1] , College of Science, University of Mosul, Mosul, Iraq[2]

**Corresponding author. Email: najlabadie@uomosul.edu.iq[1]*

Article information

Article history:
Received : 12/12/2021
Accepted : 29/12/2021
Available online :

Abstract

Multilevel Security (MLS) is one of the ways that protects the stored information in the computer and mobile devices. It classifies users and information into levels of security; thus, the user can access information within its level or less.

A smartphone is used in managing some of businesses, controlling the home and car devices within the smart city environment by using a set of data stored in the database. The database is used by more than one authorized user some of this data is confidential and important that requires protection from un authorized users.

In this research a proposed system to implement the MLS principle within three levels of security is presented. The first level gives the user its own security level. The second level transfers users through the system parts according to their security level (system administrator or regular user). The third level allows users to manipulate the stored encrypted data in SQLite database by using a simple and quick cryptographic algorithm.

The proposed system is implemented in the smart mobile devices which are supported by the Android operating system. The experimental result showed that the proposed system has the ability to protect the data in the database and prevents users to view the data at upper levels. Also, the inability of users to change the security level of data that prevents the leak of data from the upper security levels to the lower level. Moreover, the proposed system works quickly and needs a little storage space.

Keywords:
Multilevel Security, Multilevel Database Security, Database Security, Information Security SQLite database.

Correspondence:
Author : Najla Badie AI Dabagh
Email: najlabadie@uomosul.edu.iq

1. INTRODUCTION

Many organizations such as (institutions and companies) need to protect confidential information which is transmitted through networks or stored in a database because these organizations may be exposed to a financial, commercial and scientific loss or leakage of information. The loss rate varies based on the level of information or data that have been exposed to theft or intrusion. Thus, these organizations must protect their information which is stored in the databases. In fact, there are three classical methods to protect information in the databases [1]: physical protection, protection of the operating system and DBMS (Database Management System), These methods are not enough to protect information in the database for many reasons [2]:

1. When the user has read permission only to access the data, he will access all data in the database.
2. When the user has the permission to make a backup of the database, an intruder may access the data by getting a copy of the backup file thus, the system will lose the reliability.

Many researchers used multiple classical methods to protect data, such as cryptography, data hiding and using passwords. These methods affect the efficiency of the system and require maintenance of the keys of encryption and passwords. Also, these methods waste time estimated by the complexity of encryption and decryption operations. These problems could be solved by using Multi level Security (MLS). MLS regulates the users in security levels and each level has own level of security to handle the shared data among users.

Today, most users deal with SQLite databases, through using smart mobile devices, this database may include confidential data, especially when using smartphone within smart cities environment. Smart cities include a set of data that is used to control the smart devices. This data must be stored in the databases and provided with protection. Due to the limited speed and small memory of the smart mobile devices, we need to build a simple and fast systems to protect data in database to enable these systems run on smart mobile devices efficiently.

Ramzi and Natalie [3] introduced a new definition of a relational database model based on data confidential in the rows and temporary data which called Temporal Multilevel Secure Database (TMSDB). TMSDB integrates the characteristics of the temporal database model and the database security levels model.

Abdulameer [4] introduced multilevel authentication method, which is considered necessary in sensitive system that contains a combination of security levels and data confidentiality. The proposed method divides the system into a set of security levels and checks the level of a user at each level to achieve reliability. Most levels include sub-security levels and define the security levels and data quality of each level by the Identity Manager (ID) that is responsible of user transferring between security levels in the system.

Shmueli, Vaisenberg, Elovici and Glezer [5] described the main challenges in data encryption, key management, encryption overhead, and review related academic work on alternative encryption configuration pertaining to encryption locus; indexing encrypted data; and key management. They concluded their work with a benchmark using the following design criteria: encryption configuration, encryption granularity and key storage.

Tzong-An and Hong-Ju [6] introduced a new mechanism of MLS based on Schema Level Classification. The security level in this mechanism depends on the tables and features in the databases that reduce the rules of inference and prevents the user from viewing the entire database.

Kaur and Bhardwaj [7] proposed technique to improve the security in a cloud computing environment that increases the flexibility of security levels using encryption algorithms which are RSA, Random Number Generator and DES.

BabuRaj and Babu [8] introduced a Schema that works to manipulate the database through the use of user levels and use of the master key to protect private key and private information. One of the disadvantages in master key scheme is that the authorized authorities cannot access database even with court search warrant. To overcome this disadvantage, key splitting method is introduced here. This scheme provides privileged access for designated authorities. Also, revocation list is maintained in the database to avoid unnecessary access when the user is revoked.

Yanjun and Chin-Chen [2] provided a schema for encrypting the database by encrypting the rows based on the GART algorithm. After analysis, the presented schema proved to be more efficient than the one provided by Lin et al [9] with the same level of confidentiality.

In this paper, fast and simple cryptography algorithms are suggested. These algorithms do not contain any explicit keys neither for encryption nor decryption process for both text and numbers. To investigate the performance of the proposed algorithms, the algorithms were applied to the stored data in SQLite databases that was loaded in a smart mobile based on multilevel database security principles. The rest of this paper includes: section 2 shows the Multilevel Security(MLS) that includes (definition of MLS, main goals of MLS, a comparison between traditional encryption methods with MLS). The proposed method has been offered in Section 3. Section 4 represents the conclusion.

2. Multilevel Security (MLS)

MLS was developed by US Military in 1970 [10], that is considered one of the computer applications that protects data in Operating Systems, Networks and Databases by classifying data and users to different security levels. Many organizations utilize MLS security during its operation, such that SELinux[11], Oracle Label Security (OLS)[12], MLChat[13] and cloud security[14]. Actually, there are four levels of security: Top Secret (TS), Secret (S), Confidential (C), Unclassified (U). The user must be appointed to a propitiate level of security by the system administrator before processing and sharing the data.

MLS was firstly used in military systems and later in reliable operating systems and databases, as well as in applications that operate on the network [15].

MLS has two main goals 1) preventing unauthorized users from accessing data with high security levels 2) preventing users from changing security level of data [16].

In implementation of MLS, traditional cryptographic methods have been used to protect important data, especially when the data is stored in the databases and shared by more than one user. Compared to conventional storage methods, all users can access data, non-repetition data as the number of users, also, provide data integrity and control access.

Most encryption methods use keys in encryption and decryption processes. These methods need to manage keys and maintain their confidentiality and complexity. So, some researchers use another encryption method to protect these keys [8]. The keys may need to be stored based on their size. Also, the algorithm should be used to distribute the keys safely such as the RSA algorithm. However, these methods are used to protect data in most systems, immediately. Using these methods in Android operating system, which runs on most mobile devices that have low-speed processor and a small memory, will be difficult when comes to processing large amounts of data and encryption every time, especially when dealing with database. Therefore, these systems need uncomplicated encryption methods that do not have keys. As a result, the management and distribution of keys will not be used.

3. Proposed System

In this research, new encryption methods are proposed, the main advantages of the proposed methods that they are easy and quick to implement. It contains an implicit key (extracted from plain text) so it is classified as substitution encryption algorithms because they replace the plain text characters by another's to produce the cipher text.

3.1. Encryption methods

The encryption process starts by converting the plain text to a set of characters. Then, encrypted each character by finding the encryption alphabetical from shifting the original alphabets base on the sequence of the character in the plain text. Later, taking the corresponding encryption character to the plain character. Encryption process applied according to the Eq. (1). Fig. 1 shows the flowchart of the encryption process.

$$C[I]= ((P[I] + (I \bmod 26)) \bmod 26 \qquad \dots (1)$$

Where C [] is an array holds Cipher text, and P [] is an array holds Plain text and I is character index in plain text or cipher text. Fig. 2 shown an example for the text encryption.

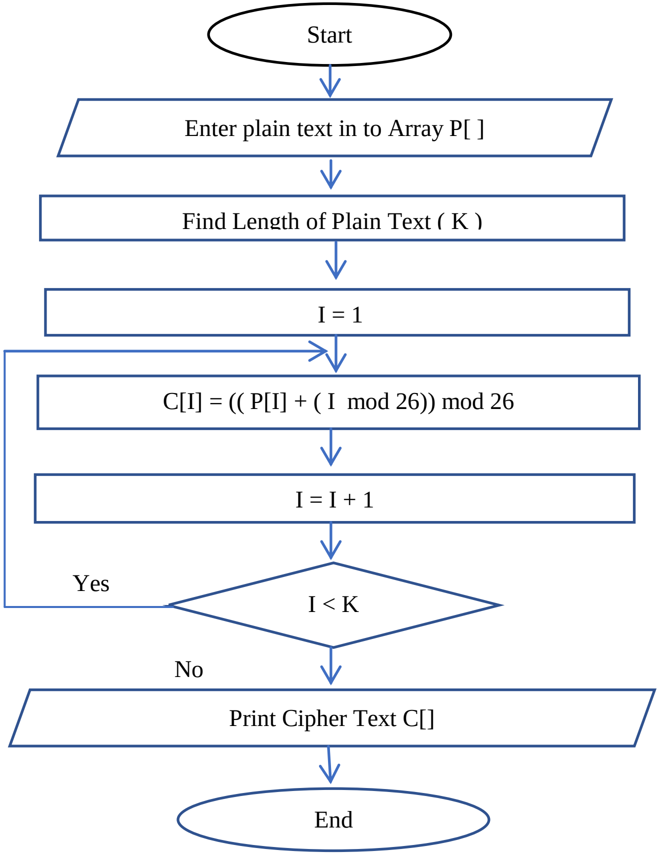

Start
Enter plain text in to Array P[ ]
Find Length of Plain Text (K)
I = 1
C[I] = ((P[I] + (I mod 26)) mod 26
I = I + 1
Yes
I < K
No
Print Cipher Text C[]
End

Fig. 1. Flow Chart of the encryption process

Plain Text = banana
1 2 3 4 5 6
b a n a n a

c[0] = ASC(b) + (0 mod 26)= b
c[1] = ASC(a) + (1 mod 26)= b
c[2] = ASC(n) + (2 mod 26)= p
c[3] = ASC(a) + (3 mod 26)= d
c[4] = ASC(n) + (4 mod 26) = r
c[5] = ASC(a) + (5 mod 26)= f
Cipher Text = bbpdrf

Fig. 2. Example of text encryption

3.2. Decryption methods

The decryption process starts in an opposite way to the encryption process by taking the cipher text then converting it to a set of characters. After that, decrypt the character by finding the decryption alphabetical from shifting the original alphabets base on the sequence of the character in the cipher text. The last step is taking the corresponding plain character to the cipher character. Decryption process is applied according to the Eq. (2). Fig. 3 shows the flowchart of decryption process.

$$P[I] = ((C[I] - (I \bmod 26)) \bmod 26 \qquad \dots (2)$$

Where C [] is an array holds Cipher text, and P [] is an array holds Plain text and I is character index in plain text or cipher text. Figure 4 shows an example for the text decryption.

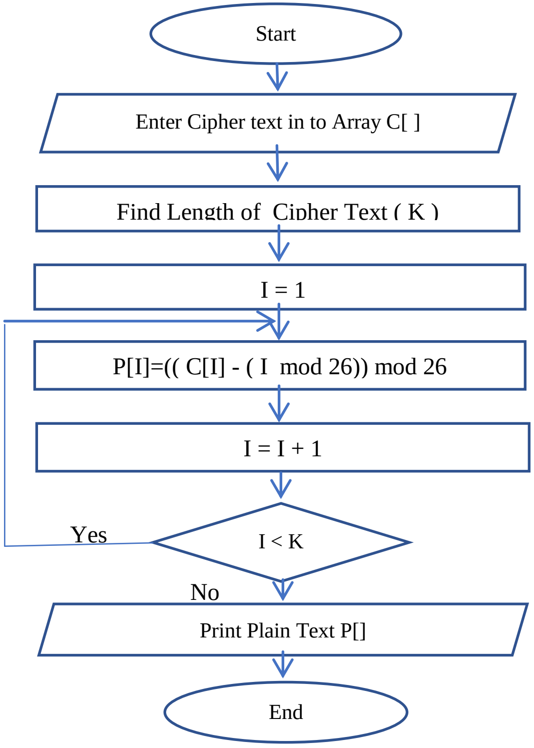

Start
Enter Cipher text in to Array C[ ]
Find Length of Cipher Text (K)
I = 1
P[I]=((C[I] - (I mod 26)) mod 26
I = I + 1
I < K
Yes
No
Print Plain Text P[]
End

Fig. 3. Flow Chart of the decryption process

Cipher Text = bbpdrf
1 2 3 4 5 6
b b p d r f

p [0] = ASC(b) - (0 mod 26) = b
p [1] = ASC(b) - (1 mod 26) = a
p [2] = ASC(p) + (2 mod 26) = n
p [3] = ASC(d) + (3 mod 26) = a
p [4] = ASC(r) + (4 mod 26) = n
p [5] = ASC(f) + (5 mod 26)= a
Plain Text = banana

Fig. 4. Example of text decryption

3.3. Encryption and Decryption Numbers methods

In addition, the same encryption algorithm was used to encrypt the numbers with simple changes, translate the number digits to char by adding (17) to number ASCII and encrypt the character. For example, to encrypt any number Eq. (3) is used.

$$C[I] = (((P[I]+17) + (I \bmod 26)) \bmod 26 \qquad \dots (3)$$

Figure 5 shows an example to encrypt a number.

Plain Number = 357621
C [0] = ASC (3) +17 + (0 mod 26) =D
C [1] = ASC (5) +17 + (1 mod 26) =G
C [2] = ASC (7) +17 + (2 mod 26) =J
C [3] = ASC (6) +17 + (3 mod 26) =J
C [4] = ASC (2) +17 + (4 mod 26) =G
C [5] = ASC (1) +17 + (5 mod 26) =G
Cipher Number = DGJJGG

Fig. 5. Example of a number encryption

And to decrypt any number Eq. (4) is used.

$$P[I]= (((C[I]-17) - (I \bmod 26)) \bmod 26 \qquad \dots (4)$$

Where C [] is an array holds Cipher number, and P [] is an array holds Plain number and I is a number index in plain number or cipher number.
Figure 6 shows an example of number decryption.

Cipher Number = DGJJGG
P [0] = ASC(D)-17 - (0 mod 26) =3
P [1] = ASC(G)-17 - (1 mod 26) =5
P [2] = ASC(J)-17 - (2 mod 26) =7
P [3] = ASC(J)-17 - (3 mod 26) =6
P [4] = ASC(G)-17 - (4 mod 26) =2
P [5] = ASC(G)-17 - (5 mod 26) =1
Plain Number = 357621

Fig. 6. Example of a number decryption

The proposed methods could be also deal with the real numbers.

3.4. Implementation Strategy of proposed System

The system is designed to control the access to the stored data in the SQLite database by using Multilevel security and this shown in Figure 7.

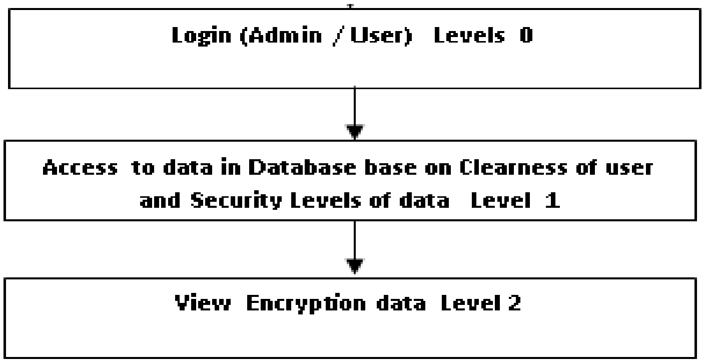

Login (Admin / User) Levels 0
Access to data in Database base on Clearness of user and Security Levels of data Level 1
View Encryption data Level 2

Fig. 7. Proposed System Security Levels

In order to deal with the encrypted stored data in the SQLite database by the user (system administrator / normal user), the user must pass through several levels of security, the first level represents login to the system, which includes inserting the user's name and password to verified user reliability. When the user login successfully, the system classifies the user either the system administrator (holds level 0) or the normal user (the level of 1 or 2). After that the user transfer to the next level of security. In the second level, the system administrator can manage the users, in addition to dealing with the system, while the normal user can be only able to access the stored data in the database that fall in its security level or the lower. Also, the user cannot delete or display the data that has higher security level than its security level. The last level includes displaying reports that include the query data from the database which appears in encrypted form to the user when the security level of data is higher than the security level of user, as outlined in Appendix A.

4. Conclusions

The proposed system provided high performance in multilevel database security with the following properties: Firstly, prevent users from switching between security levels. Secondly, prevent users from transferring data from one level to another, such as sending data from the upper level to the lower level and vice versa. Thirdly, Protect the data in the database from the access by unauthorized users even if they have a copy of the backup of database because of the ease and quick use of the proposed new encryption algorithms, in addition, it has an implicit key which make it a lightweight method, not need large storage space, robust and unbreakable by the cryptanalyst.

References

[1] Guo, C.; and Chang, C.C. An authenticated group key distribution protocol based on the generalized Chinese remainder theorem, international journal of communication system, 27(1), 126-134, 2014.

[2] Yanjun, L.; and Chin-Chen, C. A Database Encryption Scheme Based on the Generalized Aryabhata Remainder Theorem. Journal of Information Hiding and Multimedia Signal Processing, 5(4), 603-613, 2014.

[3] Ramzi, A.H. and Natalie, B. Towards a Temporal Multilevel Secure Database (TMSDB). Journal of computer Science, 2(1), 19-28, 2006.

[4] Abdulameer, K.H. Enhanced Authentication Mechanism Using Multilevel Security Model. International Arab Journal of e-Technology, 1(5), 49-57, 2009.

[5] Shmueli, E., Vaisenberg, R., Elovici, Y., and Glezer, C. Database Encryption – An Overview of Contemporary Challenges and Design Considerations. ACM SIGMOD Record, 38(3), 29-34, 2009.

[6] Tzong-An, S., and Hong-Ju, L. A Schema Classification Scheme for Multilevel Databases. Computing Sciences and Software Engineering. Springer, Dordrecht, DOI 10.1007/978-90-481-9112-3_72, 427-431, 2010.

[7] Kaur, A.; and Bhardwaj, M. hybrid encryption for cloud database security. International Journal of Engineering science & advanced technology, 2(3), 737 – 741, 2012.

[8] BabuRaj, S.; and Babu, P. Zero private information leak using multi-level security and privileged access for designated authorities on demand. (IJCSIT) International Journal of Computer Science and Information Technologies, 5 (4) ,4970-4974, 2014.

[9] Lin, C.H.; Chang, C.C.; and Lee, R.C.T. A record-oriented cryptosystem for database sharing. The Computer Journal, 35 (6), 658-660, 1992.

[10] Bell, D. and LaPadula, L. Secure computer systems: Unified exposition and multics interpretation. MITRE technical report, MITRE Corporation, Bedford Massachusetts, 2997: ref A023 588, 1976.

[11] Petersen, Richard. Fedora 14 Administration and Security. Surfing Turtle Press. p. 298. ISBN 9781936280223. Retrieved 2012-09-13. The SELinux reference policy [...] Multi-level security (MLS) adds a more refined security access method. MLS adds a security level value to resources. 2011.

[12] https://www.oracle.com/database/technologies/security/label-security.html.

[13]http://www.sse.gr/NATO/EreunaKaiTexnologiaNATO/36.Coalition_C4ISR_architectures_and_information_exchange_capabilities/RTO-MP-IST-042/MP-IST-042-12.pdf.

[14] Tallapally, Sampath Kumar, and B. Manjula. "Competent multi-level encryption methods for implementing cloud security." IOP Conference Series: Materials Science and Engineering. Vol. 981. No. 2. IOP Publishing, 2020.

[15] Ramachandran, R.; Pearce, D.J.; and Welch, I. AspectJ for Multilevel Security, ACP4IS, 20(6),13-17, 2006.

[16] George, M. Multilevel Security. SHARE Washington DC, Session 1736. RACF Development, 2003.

[17] Chinetha, K.; Daphney, J.; and Shalini, A. An Evolution of Android Operating System and Its Version. (IJEAS) International Journal of Engineering and Applied Sciences,2(2), 30-33, 2015.

18. Hipp, R.D.; Kennedy, D.; and Mistachkin, J. SQLite. Retrieved November 2nd, 2012, from www.sqlite.org, 2000.

Appendix A

Implementation and Figures of Proposed System

A.1. Introduction

Android Studio V.3 was used to implement the proposed system which works under Android Environment. Java programming language was used with some of the implicit libraries to associative Android Studio with SQLite.

A.2. Android and SQLite Database

Android is the most widely deployed mobile devices operating system in the world. It is used in the smart phones, tablets and other devices because it is easy to use and an open-source code for developers and nowadays it has 1.5 billion users [17].

Most running applications in the Android environment need to process data and information in database. Android uses a library that enables it to build SQLite databases that operate according to the SQL rules [18].

Many complex database applications developed in the Android environment such as shopping, warehouse managing, banking, business managing and others, which makes SQLite a useful tool for developers. SQLite is the smallest machine to manage databases and has the following characteristics [18]:

1- Serverless (the process that wants to access the database reads and writes directly from the database files on disk. There is no intermediary server process).

2- Self-Contained (it requires very little support from the operating system).

3- Zero-Configuration (no configuration is required).

4- Transactional (all changes and queries are Atomic, Consistent, Isolated and Durable (ACID)).

A.3. Implementation

Initially, two tables are created in the database. The first one included information about the users of the system (system administrator or normal user) as shown in Table 1, which contains the following fields (User ID, User Name, Password, Privilege). The second table includes employee information as shown in Table 2, which contains the following fields (Employ ID, Employ name, Salary, Address, Row privilege). We noticed that the fields (Salary, Address) contain encrypted data and the field (Row privilege) is not visible to users and take the same level of security to the user who inserted it.

Table 1. Information of System's users

ID	***NAME***	***PASSWORD***	***PRIVILEGE***
4	mahmood subhy	12345	0
8	zaid subhy	761421	1
9	yaser subhy	77621	2
10	yousif mahmood	86713	3

Table 2. Employee information

E-ID	***E-NAME***	***E_Salary***	***E_address***	***E_priv***
8	Salim	DIFIE	Isct1Ggnpmko	0
9	Aws	EBJFE	Eh{sxx2Ihq{y	0
10	Ban	GHJEL	Iofle2Jltqs	1
11	Maryam	CDLGG	Fscqgj3Wi{s~	1
12	Mayar	JIEEGIM	Tvtni~3Hvtk}m	2
13	Yousif	IEEELDNM	Gftpes 4JjIwu{	2

when the Multilevel Database System is executed, the first interface of the system appears as shown in Fig. A-1-a, which includes Login information to the system by using the username and password.

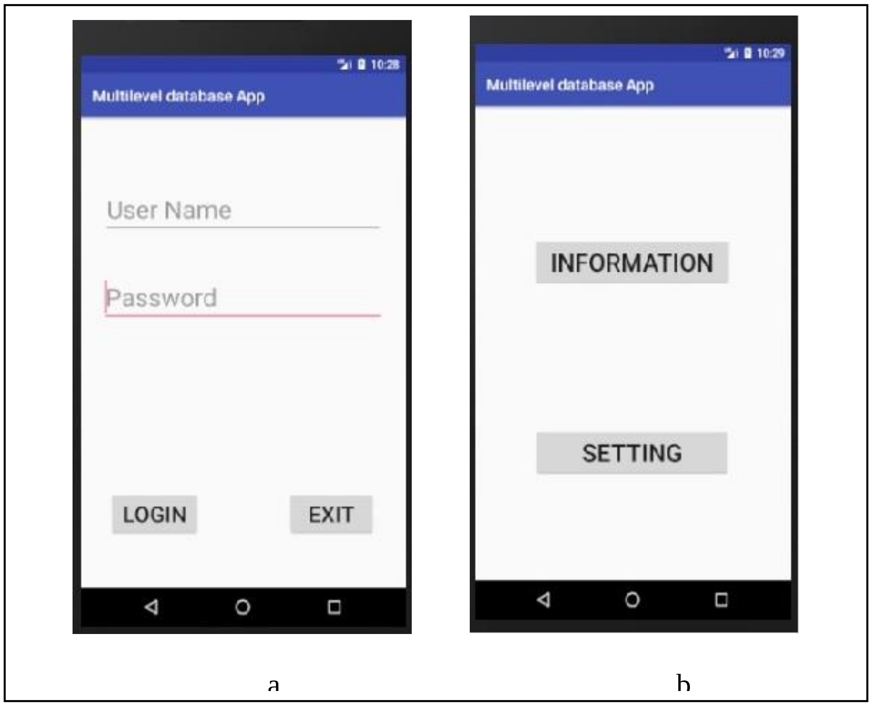
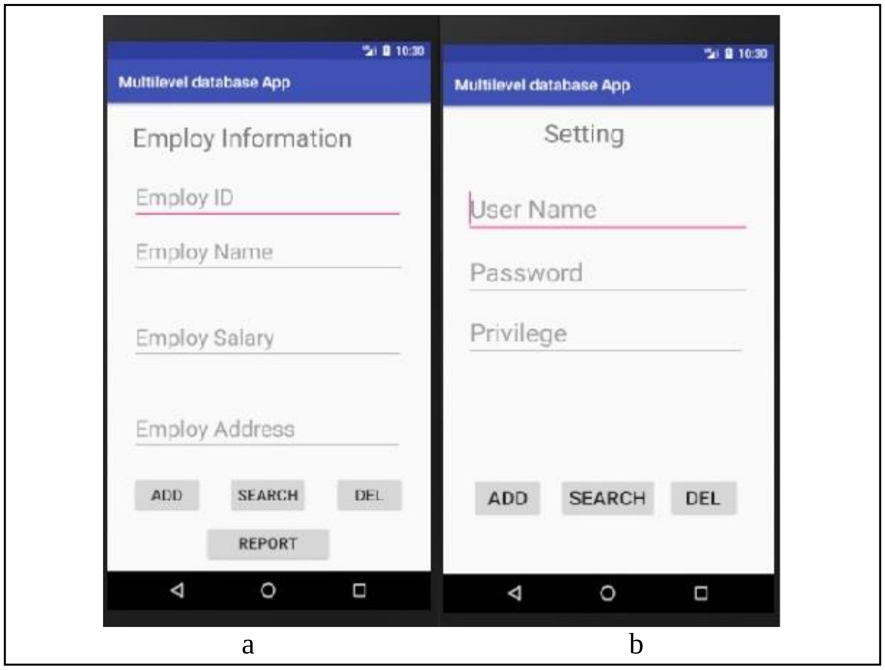


Fig. A-1. Login to Multilevel Database System

When the username and password are entered correctly, the second interface will appears as shown in Fig. A-1-b, which contains two buttons, the first button used to move the user to the Employees information interface as shown in Fig. A-2-a, while the second button used to move the user to the user information interface as shown in Fig. A-2-b. This interface enables the system administrator (Level 0) only, manage users accounts

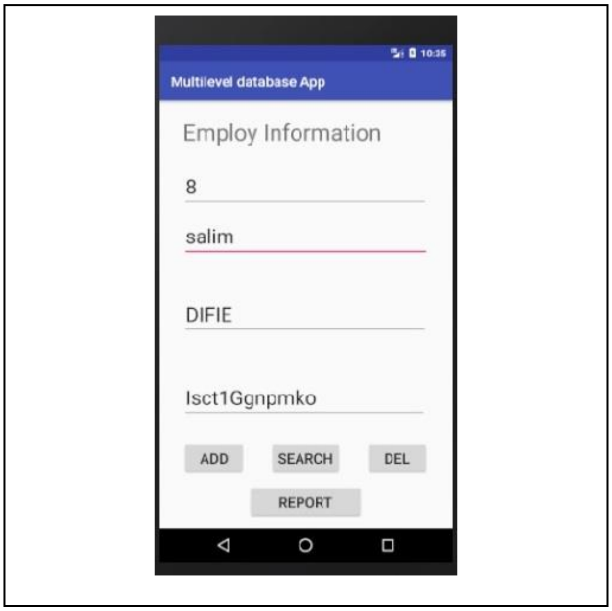


a

b

Fig. A-2. Employee Information and Setting Interfaces

When the user passes to the Employee Information Interface as shown in Fig. A-2-a, then the employee's name in the Employee Name field is inserted and press Search button. All employee information will appear and may appear in encrypted form such as fields (E-salary, E-address) when the user requested employee information that has a security level higher than himself as shown in Fig. A-3.


Employ Information

8

salim

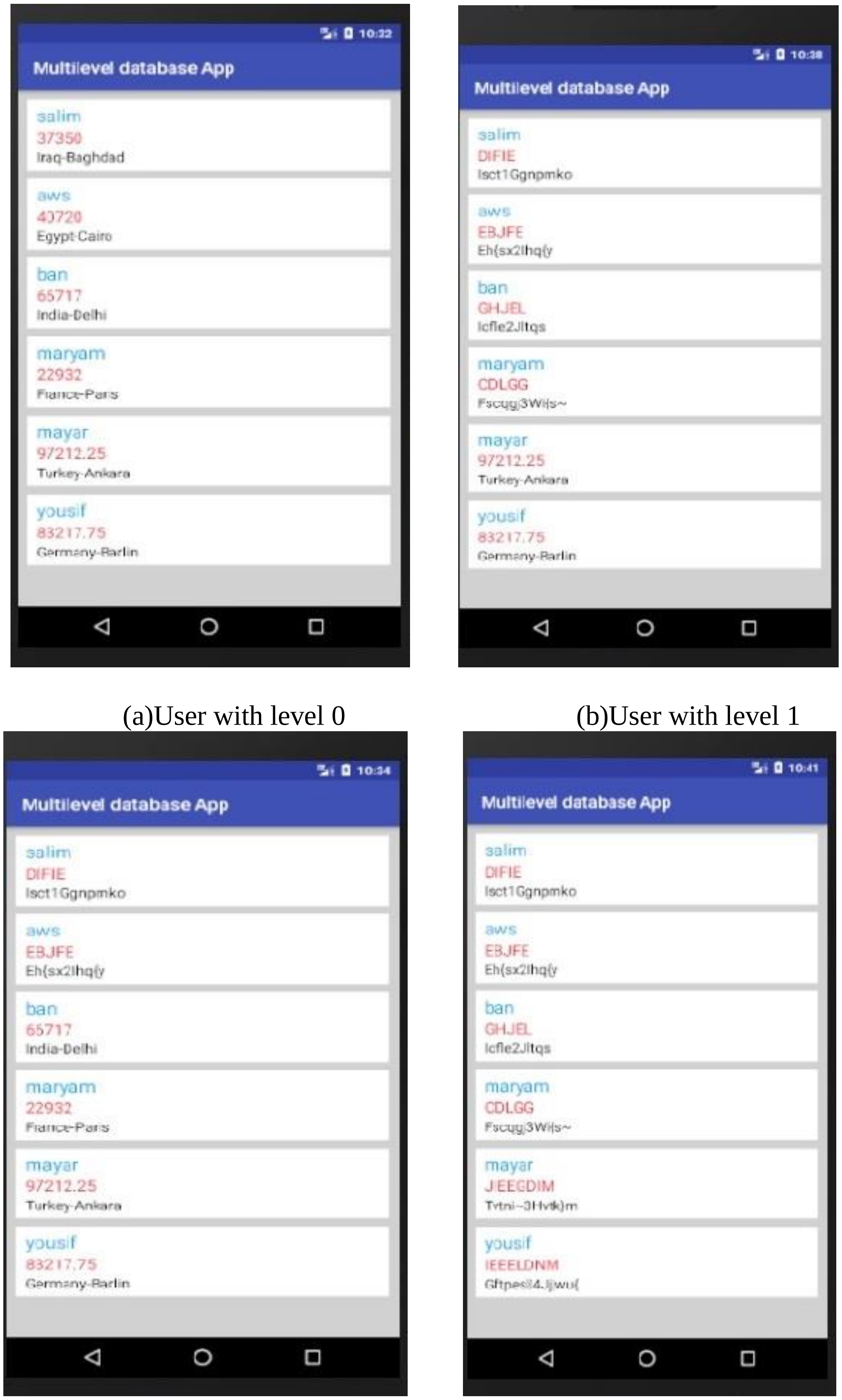
DIFIE

Isct1Ggnpmko

Fig. A-3. Result of search operation

When the report button is pressed in the same interface, the employee information will be displayed according to the security level of the user and the security level of the displayed information (different report states showed in

Fig. A-4. the states are: (a) User with level 0 (all important information will appear clearly), (b)user with level 1 and (c) with level 2 (some of important information will appear clearly which have security level equal or above current user security levels), (d) User with level 3 (all important information will appear in encrypted format).



10:22

Multilevel database App

salim
37350
Iraq-Baghdad

aws
40720
Egypt-Cairo

ban
65717
India-Delhi

maryam
22932
Fiance-Paris

mayar
97212.25
Turkey-Ankara

yousif
83217.75
Germany-Barlin

10:38

Multilevel database App

salim
DIFIE
Isct1Ggnpmko

aws
EBJFE
Eh(sx2Ihq(y

ban
GHJEL
Icfle2Jltqs

maryam
CDLGG
Fscqg|3Wi|s~

mayar
97212.25
Turkey-Ankara

yousif
83217.75
Germany-Barlin

(a)User with level 0

(b)User with level 1

10:34

Multilevel database App

salim
DIFIE
Isct1Ggnpmko

aws
EBJFE
Eh(sx2Ihq(y

ban
65717
India-Delhi

maryam
22932
Fiance-Paris

mayar
97212.25
Turkey-Ankara

yousif
83217.75
Germany-Barlin

10:41

Multilevel database App

salim
DIFIE
Isct1Ggnpmko

aws
EBJFE
Eh(sx2Ihq(y

ban
GHJEL
Icfle2Jltqs

maryam
CDLGG
Fscqg|3Wi|s~

mayar
JIEEGDIM
Tvtni~3Hvk)m

yousif
IEEELDNM
Gftpes!4.Ijwu(

(c)User with level 2

(d) User with level 3

Fig. A-4. Different of displayed report states

امنية قواعد البيانات متعددة المستويات للاندرويد باستخدام طرق تشفير سريعة

نجلاء بديع الدباغ **محمود صبحي محمود**

najlabadie@uomosul.edu.iq mahmoodsubhy1981@gmail.com

كلية علوم الحاسوب والرياضيات **كلية العلوم**

جامعة الموصل ، الموصل ،العراق

تاريخ استلام البحث 2021/12/21 **تاريخ قبول البحث: 2022/2/13**

الخلاصة:

تعد الامنية متعدد المستويات (MLS) أحد الطرق التي تحمي المعلومات المخزنة في الكمبيوتر والأجهزة المحمولة . يصنف المستخدمين والمعلومات إلى مستويات من الامنية؛ وبالتالي، يمكن للمستخدم الوصول إلى المعلومات ضمن مستواه أو أقل امنية.

يستخدم الهاتف الذكي في إدارة بعض الأعمال، والتحكم في أجهزة المنزل والسيارة داخل بيئة المدن الذكية وذلك باستخدام مجموعة من البيانات المخزنة في قاعدة البيانات .يتم استخدام قاعدة البيانات من قبل أكثر من مستخدم مصرح له، وقد تكون بعض هذه البيانات سرية ومهمة تتطلب الحماية من المستخدمين غير المصرح لهم.

في هذا البحث تم تقديم نظام مقترح لتنفيذ مبدأ MLS ضمن ثلاثة مستويات من الامنية .المستوى الأول يمنح المستخدم مستوى الامنية الخاص به .اما المستوى الثاني من الامنية ينقل المستخدمين بين أجزاء النظام وفقًا لمستوى الامنية الخاص بهم (مسؤول النظام أو المستخدم العادي) .المستوى الثالث يسمح للمستخدمين بمعالجة البيانات المشفرة المخزنة في قاعدة بيانات SQLiteباستخدام خوارزمية تشفير بسيطة وسريعة.

تم تنفيذ النظام المقترح في الأجهزة المحمولة الذكية التي يدعمها نظام التشغيل .Android أظهرت النتائج التجريبية أن النظام المقترح لديه القدرة على حماية البيانات في قاعدة البيانات ويمنع المستخدمين من عرض البيانات في المستويات العليا .كما أن عدم قدرة المستخدمين على تغيير مستوى امنية البيانات والذي بدوره يمنع تسرب البيانات من مستويات الأمنية العليا إلى المستوى الأدنى .علاوة على ذلك، يعمل النظام المقترح بسرعة ويحتاج إلى مساحة تخزين صغيرة.

الكلمات المفتاحية: الأمنية متعددة المستويات، امنية قواعد البيانات متعددة المستويات، امنية قواعد البيانات، امنية المعلومات، قواعد البيانات SQLite.