JKAU: Eng. Sci., Vol. **28** No. 1, pp: 19 - 30 (1438 A.H./ 2017 A.D.)
Doi: 10.4197/Eng. 28-1.2

A Survey of Access Control and Data Encryption for Database Security

Emad F. Khalaf and **Mustafa M. Kadi**

Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, P. O. Box 80204, Jeddah 21589, Saudi Arabia

ekhalaf@kau.edu.sa

Abstract. With the vast amount of data generated nowadays, organizing and managing of these data are very important to allow the users to access, retrieve, and update their data by using database systems (DBS). Most of the current organizations use DBS to increase the efficiency and the productivity of their organizations, but the security threats are becoming more dangerous to the DB. So, protection of data by keeping it integrated and secured from any undesirable intrusion became the highest priority for these organizations. DB security provides various techniques to protect data from any threats. This paper discusses two techniques used in the DB field to achieve integrity and confidentiality of the data, by using access control policies and data encryption.

Keywords: Database Security, Access Control, Encryption.

1. Introduction

Since the 1960s, the DBS were developed [1] and had become more important for various organizations to keep their data more organized and available for each user. DB refers to a set of data that are related together which act as the facts that can be registered and have implicit meaning [2]. DBs can be classified according to the architectures: centralized DB (CDB) or distributed DB (DDB). The main difference between CDB and DDB is that the CDB stores all data in a single location while the DDB stores the portions of the DB in different physical locations [2]. In a CDB, disruption at a single location makes the whole system unavailable to each user while in a DDB, the users can still be able to access other sites of the DB [2]. DBs can be constructed using various data models such as relational models, hierarchical models, and object-oriented models. These different models of DBs carry within them many of confidential or sensitive information such as credit card numbers, medical records, and student records, which must be protected from any unauthorized users. With increasing threats to DBs, the need to keep the data integrity and confidentiality have emerged against any threats. So many methods emerged to protect the data, and for achieving this target three requirements must be satisfied. These are confidentiality, integrity, and availability. Confidentiality is to prevent an unauthorized user from access to the data by using a set of rules [3]. Integrity is ensuring that the data is not exposed to any modification or destruction [3]. Availability is ensuring that the user can access the information reliably and timely [3]. The loss of any one of these requirements poses a danger to the DB.

The securing of DB systems has received a great attention since the mid-1970s, starting from access control models for DB systems [1], which were considered as one of the earliest security methods to DB security. Access control is a mechanism to guarantee data

integrity and confidentiality by inspecting the user's rights against a set of permissions [4]. Authorization is the process that can specify the database operations that the user can perform and which data that user can access [5]. Another method that can be used to verify the user's identity is the authentication which can be considered the initial phase of gaining access to the DB [6]. Even after authenticating the user inside the database, the database management system (DBMS) also has some methods such as audit trial and view that make the database secure from any unauthorized transactions. Audit trial is a method that can be regarded as the history of all operations that are applied by a certain user into the DB [7]. So, when an illegal operation was executed, the database administrator (DBA) can explore the account number that was used to perform this operation [2]. A view method is a virtual table that can be produced by one or more relational operations with the base table [8]. This method can be used to allow the user to access part of a relation when the user can not access that relation directly. Data confidentiality can also be ensured by using encryption techniques that can be applied to the data. Encrypting data using a cipher will transform it into unreadable form to other users except the one who has the key to decrypt the data [9].

In this paper, the threats of DB security will be presented in Section 2. Presentation of DB security measures takes place in Section 3. In Section 4 access control policies will be discussed. Encryption techniques will be discussed in Section 5. In Section 6 the conclusion will be presented.

2. Threats of Database Security

The threats are any case or event that could adversely affect the DB security, and they can be intentional or accidental [8]. The common threats to DB security are the following:

• **Privilege Abuse**: There are two kinds of Privilege Abuse: Excessive Privilege Abuse (EPA) and Legitimate Privilege Abuse (LPA). EPA is when users have access privileges to the DB that exceed their job tasks; these privileges could lead to misuse for malicious purposes [10]. Misusing the legitimate DB privilege for malicious aims by the authorized user refers to LPA [10].

• **Privilege Elevation**: When the DB has vulnerability, an attacker may be able to exploit this vulnerability for converting the privileges accessibility from ordinary user to an administrator user [10].

• **SQL Injection**: SQL Injection happens when an attacker inserts into a vulnerable SQL data channel a series of unauthorized SQL statements. Attackers can have unlimited access to the entire DB by using SQL injection [11].

• **Platform Vulnerabilities**: Vulnerabilities in operating systems and any installed additional services on a DB server can result in damage to the DB such as unauthorized access, denial of service, or data corruption [11].

• **Weak Audit Trail**: Audit trails are used to record each user activities in the DB. So, the weakness of an audit trail poses a danger to the organization's DBs [9].

• **Denial of Service**: It is an attack that prevents authorized users from access to the DB. It is a risk threat for any organization [9].

• **Weak Authentication**: Weak authentication may allow attackers to use some methods such as (social engineering and brute force) to steal usernames and passwords for legitimate users and then access the DB [11].

• **Backup Data Exposure**: Many cases of security breaches have included the theft of hard disks and backup tapes, because the backup DB storage media have seldom been protected from any attack. [11].

3. Database Security Measures

DB security is the mechanisms that ensure the protection of the DB from unauthorized users, deliberate threats, data loss, and hackers [8]. It addresses many issues such as legal, ethical, policy, and system-related issues [2]. DB security is a difficult process that any organization should improve to run its activities easily and efficiently [5]. Each organization that is running successfully demand the confidentiality and the integrity of their data are protected against unauthorized access and any malicious or accidental modification [9]. The protection of data is accomplished with different aspects of a database management system (DBMS) [7]. DBMS is the group of applications which manages the data presented in the DB and helps to organize data for better performance [12]. In all DBMSs to reduce the threats, they provide some kinds of security techniques designed for these purposes [13]. There are many security measures that have been created for protecting the DBs. The four main control measures that are implemented to protect DBs from threats are the following; the first one is the access control, the second is inference control, the third is flow control and finally, data encryption. This control measures are shown as in Fig. 1.

The access control (as we mentioned before) is a mechanism to ensure data confidentiality by inspecting the user's rights against a set of permissions [7]. These authorizations are administered by following either a Discretionary Access Control (DAC) policy, Mandatory Access Control (MAC) policy, Role-Based Access Control (RBAC) policy [3], or Attribute-based access control (ABAC). The inference control prevents users, when they have access to only the statistical or summary information, from being able to infer confidential information, which they are not authorized to read. The flow control ensures that the information cannot be flown to reach unauthorized users [2]. The encryption (as we mentioned before) is the process of data transform using a cipher to make these data unreadable to other users except the one who has a key to decrypt the data [8].

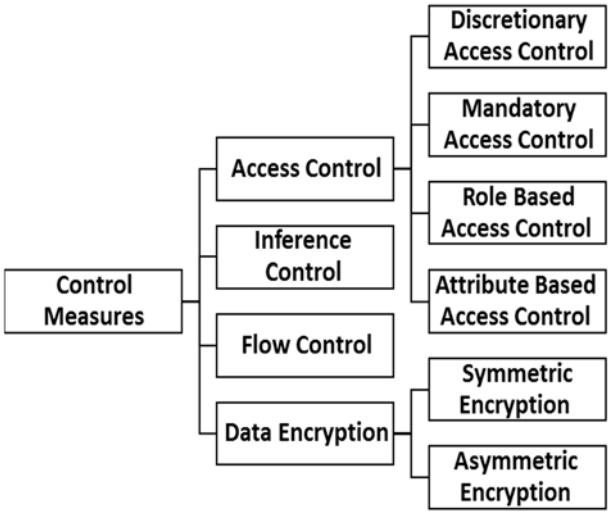

Fig. 1. Control measures.

In this paper, we will only discuss the two techniques that are used to achieve confidentiality. These are access control policies (presented in Section 4) and encryption techniques (presented in Section 5). Because the confidentiality can be high-assurance by using both, access control and encryption techniques, and they are widely discussed in many articles on DB security, and also widely used in many DBMSs such as Oracle.

4. Access Control Policies

Access control is defined as a mechanism that used to maintain data confidentiality by inspecting the users' rights against a set of authorizations [7]. A security administrator or security officer [12] that is responsible for managing a DBS specifies these authorizations. There are many types of access control policies, but the three traditional access control

policies are DAC, MAC, and RBAC policy. In the modern age, ABAC is a promising alternative to traditional policies of access control (DAC, MAC, and RBAC) which attracts the attention in both recent academic literature and industry application [14]. These various policies can be combined to make DB systems more protected [13].

4.1 Discretionary Access Control Policy

DAC policy controls the users' access to the data, relying on the authorization rules and the identity of the user [7]. The owner of data or DBA decides who can access the data based on granting and revoking of privileges for users [13]. DAC policy can permit users to grant privileges on the data to the other users [7]. In other words, the DAC policy allows system users to permit or reject the other users access to the data under their control [15]. The authorization administrator is responsible for the function of granting and revoking authorizations [7]. There are two kinds of administration policies: the first one is the Centralized Administration and the other is Ownership Administration. In the first type, only some privileged users can grant and invalidate authorizations [13] while the second one permits the owner of the data to give rights to the other users to grant and invalidate authorizations [7].

The access matrix model is considered the first DAC model which was proposed by Lampson [16]. In this matrix, the users represent in rows and the data files are in columns, and each position represents the types of privileges that the user can execute on the data files [2] as shown in Table 1. The owner of data files can decide which user can access the data and what to do with it. For example, the owner allows user John to read and execute File1 but does not allow him to access to File 2, while David is not authorized to access File1 but can read and write File 2, and so on.

The feature of DAC is flexibility, which makes it suitable for many applications [13], and therefore, the most commercial DBMSs support the DAC policy [7]. The primary limit of DAC is its vulnerability to malicious attacks like Trojan Horse attacks because DAC does not provide any control over the manner of information propagation [2].

Table 1. Example of access matrix.

User	File 1	File 2	File 3
Alice	Read, write, and execute	Read	No access
Bob	Read	Read, write, and execute	Read and execute
David	No access	Read and write	Read and execute
John	Read and execute	No access	Read and write

4.2 Mandatory Access Control Policy

MAC policy needs that every user should follow the access rules, which are set up by the DBA [13]. MAC classifies users and data into labels according to a partially ordered set of security classes. These labels are associated with all users and data in the DB [7]. The security classes can be as follows: Unclassified, Confidential, Secret, and Top Secret. The Unclassified is considered the lowest level, and Top Secret is the highest level (TS > S > C > U) [2]. MAC is mandatory because the labeling of data occurs automatically, and ordinary users cannot alter labels unless an administrator authorizes them [17].

The most common implementation of MAC policy is a Multilevel Security (MLS), which preferred for use in the military, government, and intelligence applications [13]. Multilevel DB systems are attempting to develop DB systems that protect classified data from unauthorized users depends on the data classification and the user's clearances [17]. There are two common models for MLS: Bell-LaPadula (BLP) model for assuring confidentiality of information flows and Biba

model for assuring the integrity of information. The features of MAC policy prevent any illegal flow of information [2], thus making MLS systems immune to Trojan Horse attacks. The drawback of MAC policy is that it is difficult to implement in many applications so it is suitable only for limited environments such as military applications [2].

4.2.1 Bell-LaPadula Model

The first appearance of BLP model by David Elliott Bell and Leonard J. LaPadula is in 1973 [18]. The goal of this model is to protect data confidentiality by preventing information from flowing through the high-level user/data to user/data at the low-level [16]. This model defines two security properties as follows:

- **The Simple Security Property (**also called No read-up) where the user can read a data if and only if the clearance label of the user dominates the classification of the data [16].
- **The Star Property (**also called No write-down) where the user can write a data if and only if the classification label of the data dominates the clearance label of the user [16].

4.2.2 Biba Model

In 1977, the Biba integrity model was developed by Kenneth J. Biba [18]. The motivation for creating this model is because the BLP model does not have the ability to deal with the integrity of data [18]. Because the BLP model addresses access to classified data without the management of that access, so there is no process to modify access rights [19]. By contrast, the Biba integrity model depicts the rules for how the data integrity are protected. The goal of Biba model is to ensure data integrity by preventing users from indirectly modifying data they cannot write [16]. This model prevents the flow of information from low-level data to higher data. The Biba model defines two rules. These are the reverse of the BLP model:

- **The Simple Integrity Axiom** (also called No read-down) where the user at a particular integrity level must only read data at a higher integrity level [18].
- **The Star Integrity Axiom** (also called No write-up) where the user at a particular integrity level must only write to any data at a lower integrity level [18].

4.3 Role-Based Access Control Policy

The RBAC policy is considers an alternative to MAC policy and DAC policy [2]. David Ferraiolo and Rick Kuhn have developed RBAC since 1992 [18]. The motivation of this model is to facilitate authorization administration, and also to represent access control policies directly for the organizations [13]. With RBAC, according to the job functions performed in an organization, the system administrator creates the roles and then grants privileges to those roles according to the jobs [18]. The role is defined as the job functions of organizations and users are assigned to the roles on the basis of their jobs [17]. For example, the student in a university can see his grades of courses, but cannot modify it while the teacher can enter student's grades and change it. So the access is granted to the users based on their jobs or roles in the system. RBAC is better suitable for commercial environments compared to DAC and MAC [16]. The advantage of RBAC is reducing the cost and complexity of security administration in DBs because roles work as a link between access modes for data and users [17].

4.4 Attribute-Based Access Control Policy

In traditional access control policies (DAC, MAC, and RBAC) the authorizations should be assigned directly to users, or indirectly through predefined attribute types

that assigned to the user such as roles or groups, so the management of these policies is complex and time-consuming [20]. The promising alternative is ABAC, which is a logical access control policy based on the evaluation of environment conditions, data attributes, user attributes, and the set of policies that are specified regarding those conditions and attributes [20] which are shown in Fig. 2. ABAC allows the owners of data or administrators to accept or reject user requests without further specific information about the user and for a numerous number of users that might request access to the data [21]. So, when new users join into the system, the user is assigned the attributes that are necessary to access the required data without the need for adjustment to the predefined rules or data attributes. The access decisions can change among requests by basically altering attribute values without modifying the user/data relationships. This feature leads to a more dynamic access control management capability and limits long-term maintenance requirements of data protections [20].

There are two methods for specifying authorization policies: Logical formula authorization policy (LAP) and enumeration authorization policy (EAP) [23]. LAP refers to a Boolean expression comprising of subexpressions associated with logical operators (*e.g.*, AND, OR, ≥, ≠). These subexpressions compare the attribute values with constant values or another attribute [24]. An example of this method is Extensible Access Control Markup Language (XACML). While EAP includes a set of tuples where each of them represented as (user attribute values, data attribute values), grants privileges to a set of users to practice an action on a set of data defined by the user and data attribute values that are mentioned in the tuple [24]. An example of this method is Next Generation Access Control (NGAC).

The advantages of ABAC have dynamic capabilities offer higher efficiency, flexibility, scalability and security than traditional access control policies [25]. The disadvantage of ABAC is very complexity due to the specification and maintenance of the policies [26]. After our review of the access control policies, we concluded that the ABAC is the best policy to implement in most organizations, according to the forecast mentioned in Gartner "by 2020, 70% of organizations will use attribute-based access control… as the dominant mechanism to protect critical assets, up from less than 5% today" [27]. Table 2 summarizes the earlier comparison between DAC, MAC, RBAC, and ABAC.

5. Data Encryption

Data encrypting using a cipher will transform information into unreadable form to all users except the one who has the key to decrypt the data [9]. Even if attackers bypass the access control policies, they still need the encryption keys to decrypt the data [3]. Process of encryption depends on the algorithm and the key used to encrypt the data. Encryption can be divided into two kinds: Symmetric encryption and Asymmetric encryption [28]. The encryption can be implemented on three different levels as follows (they are shown in Fig. 3):

- **Storage-Level Encryption:** Used to encrypt information in the storage subsystem. It is suitable for encrypting files, entire directories, or tape media [3]. The encryption strategy can not be related to data sensitivity or user privileges, because the storage subsystem has no knowledge of the DB scheme [3]. So, selective encryption cannot be done at some portions of the DB like row, column, or table. It is limited to the file granularity. The advantage of storage-level encryption is avoiding any changes to existing applications [3].

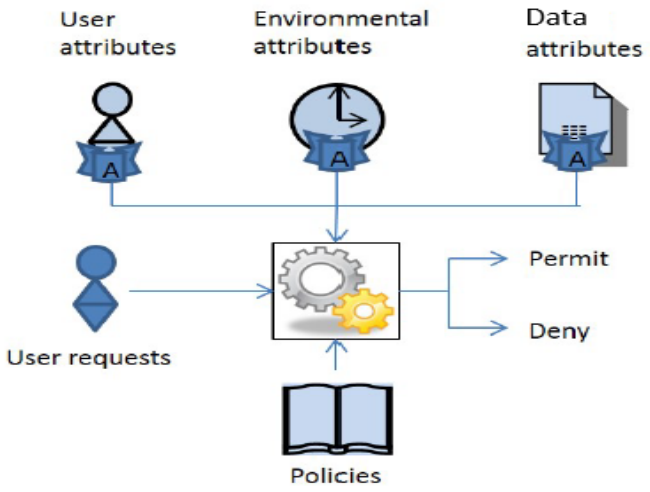

Fig. 2. ABAC model [22].

Table 2. Comparison between DAC, MAC, RBAC, and ABAC.

Factors	DAC	MAC	RBAC	ABAC
Access Control to Information	Through owner of data	Through fixed rules	Through roles	Through attributes
Access Control Based on	Discretion of owner of data	Classification of users and data	Classification of roles	Evaluation of attributes
Flexibility for Accessing Information	High	Low	High	Very high
Access Revocation Complexity	Very complex	Very easy	Very easy	Very easy
Support for Multilevel Database System	No	Yes	Yes	Yes
Used in	Initial Unix system	The U.S. department of defense	ATLAS experiment in CERN	The Federal government

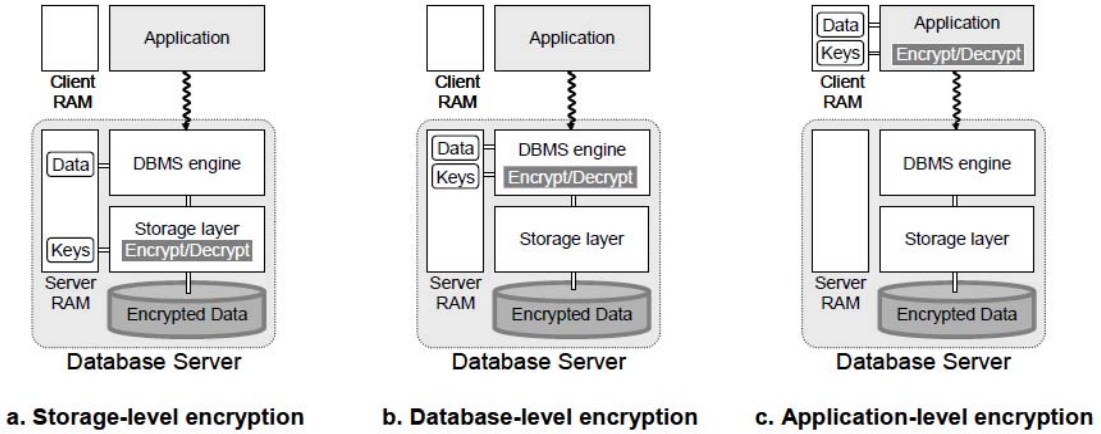

Fig. 3. The three different levels of encryption [3].

• **Database-Level Encryption (DLE):** Used to secure data inserted to or retrieved from the DB. The encryption strategy can be related to data sensitivity and user privileges, because the encryption can be part of the DB design [3]. It can use selective encryption for encrypting some portions of the DB like row, column, or table. DLE can cause low performance of DBMS because the encryption forbids the use of an index on encrypted data [3]. The encryption process may do some changes to applications.

• **Application-Level Encryption:** The encryption and decryption processes are performed within the applications. The encryption is performed within the application that introduces the data into the DB; the data is sent encrypted, thus stored and retrieved encrypted, to be finally decrypted within the application [3]. The benefit of application-level encryption is reduced excessive loads on the DB server due to the encryption and decryption operations by separating the encryption keys from the encrypted data stored in the DB. However, applications must be modified to support encryption and decryption capabilities.

5.1 Symmetric Encryption (also called secret key encryption)

Symmetric encryption uses the same key to the process of encryption and decryption of data. It can use either the stream cipher or block cipher, where a stream cipher is encrypting a stream of data one bit or one byte at a time, while a block cipher is encrypting a block of data as a whole and producing a block of equal length [34]. The advantage of symmetric encryption is having algorithms that are faster and less complex than asymmetric algorithms. The disadvantage is the problem of key distribution since two parties must agree to use the same secret key before they start encrypting and decrypting data. There are several algorithms for symmetric encryption such as Blowfish. The popular symmetric encryption algorithms which are accredited by National Institute of Standards and Technology (NIST), are Data Encryption Standard (DES), Triple Data Encryption Standard (TDES), and Advanced Encryption Standard (AES) [29]. DES is considered the first encryption standard that has been recommended by NIST in 1977 [28]. It is a block cipher which has a block size of 64-bits and uses a key size of 56-bits. The DES algorithm includes two basic building blocks of encryption: Substitution (diffusion), and permutation (confusion) that are repeated for a total of 16 cycles [2, 30]. DES became an inadequate security because many attacks proved the weaknesses of DES [28]. To improve the DES without designing a complete new cryptosystem TDES was developed [31]. TDES algorithm is similar to the original DES, but in order to increase the level of encryption it applied three times [28]. It has block size of 64-bits and two different key sizes of 112 or 168 bits. TDES has not been broken yet, but it is not preferred to use because it is slower than other block cipher methods [28]. In 2001, NIST recommended that the AES is the new encryption standard to be the replacement for DES [32]. AES has a block size of 128 bits and different key lengths of 128, 192, or 256 bits. AES includes different numbers of rounds for substitution and permutation processes depending on the key size [30]. The advantages of AES are that it is very fast and more secure compared with DES, and TDES. Table 3 illustrates the difference between DES, TDES, and AES.

5.2 Asymmetric Encryption (also called public-key encryption)

Asymmetric encryption utilizes two keys; one is called public key used for encryption, and the other that used for decryption is called private key [30]. The public keys are distributed to each user, but the private keys are kept hiden.

The private key can not be deduced from the public key [33]. The use of asymmetric algorithms can be classified into three categories: Encryption/Decryption, key exchange, and digital signatures [34]. The advantage of asymmetric encryption is that it is easy to use for key exchange and more secure than symmetric encryption. The disadvantage is that it is slower and more complex than symmetric encryption. Some of the examples of asymmetric encryption algorithms are Rivest Shamir Adleman (RSA), ELGAMAL, and Elliptic curve cryptography (ECC) [29]. The RSA is one of the first public key encryption was introduced in 1977 [34]. It is widely used for secure data transmission [33] and it uses mathematical processes to transform a message (represented as a number or a series of numbers) into a ciphertext [35]. The security strength of RSA comes from the difficulty of factoring the product of two large prime numbers [33]. ElGamal encryption was described by Taher Elgamal in 1984 [34]. It can be used for both digital signature and encryption [36] and is based on the Diffie–Hellman key exchange [33]. The main disadvantage of El-Gamal is the message expansion by a factor of two that takes place during encryption [36]. The security strength of ElGamal comes from the difficulty of computing discrete logarithms in a finite field [33]. ECC was described independently by Neal Koblitz and Victor S.Miller in 1985, and NIST approved it in 2006 [33]. It is becoming widely implemented in smaller devices like cell phones. ECC depends on elliptic curves algebraic structure over finite fields to define the public/private key pair [33]. The main advantage of ECC over other public key algorithms like RSA and El-Gamal is requiring smaller key lengths for ensuring the same level of security [37]. The security strength of ECC comes from the difficulty of computing elliptic curve discrete logarithms [37]. Table 4 illustrates the differences between RSA, ELGAMAL, and ECC.

Table 3. Comparison between DES, TDES, and AES.

Factors	DES	TDES	AES
Developed	1977	1978	2001
Block Size	64 bits	64 bits	128 bits
Key Size	56 bits	112, 168 bits	128, 192, 256 bits
Number of Rounds	16	48	10, 12, 14
Performance	Slow	Slower	Faster
Security	Proven Inadequate	In case of 112-bit key is weak	Secured

Table 4. Comparison between RSA, ELGAMAL, and ECC.

Factors	RSA	ELGAMAL	ECC
Developed	1977	1984	1985
Performance	Fast	Fast	Fastest
Power Consumption	High	Low	Lowest
Security Attacks	Timing	Adaptive chosen-ciphertext attacks	Pollard's rho algorithm
Hardware and Software Implementation	Not very efficient	Fast and efficient	Fast and very efficient
Security Strength	The difficulty of factoring the product of two large prime numbers	The difficulty of computing discrete logarithms in a finite field	The difficulty of computing elliptic curve discrete logarithm problem

After our discussion of the data encryption, we concluded that the data encryption provides confidentiality, but doesn't give assurance of integrity unless used some digital signature or Hash function [9]. The better algorithm in the symmetric key encryption is the AES algorithm in terms of cost, speed, security, and implementation, while in asymmetric key encryption, the ECC algorithm is the better in terms of speed and security [32].

6. Conclusion and Future Work

This paper is a survey to present two techniques used to protecting the data and achieving confidentiality and integrity on the DB. It was mentioned that the threats that

affect the security of databases and discussed the different policies used to control access to the data, such as: DAC, MAC, RBAC, and ABAC. The three different encryption algorithms for each of the symmetric encryption such as (DES, TDES, and AES) and asymmetric encryption (such as: RSA, ELGAMAL, and ECC) were discussed. From this survey, It can be recognize that the importance of protecting DB by using a suitable access control policy and the appropriate encryption algorithm to grant a very high level of confidentiality and integrity on the data.

In the future, ABAC is a promising alternative for currently applied polices, it has many advantages over them, however it still needs many studies on specific topics, such as attribute meaning, attribute storage, attribute confidentiality, and supporting model independent architectures [14]. For Data encryption, always there will be a need for more effective and efficient algorithms that insure the security and privacy of data, even if the database is hacked, data decryption, will be a great challenge for the hackers.

References

[1] **Thuraisingham, B.,** Database Security: Past, Present, and Future, *IEEE International Congress on Big Data*, 772–774 (2015).

[3] **Elmasri, R.** and **Navathe, S. B.,** *Fundamentals of Database Systems, Pearson Education*, USA, Sixth Edition (2011).

[5] **Bouganim, L.** and **Guo, Y.,** *Database Encryption, Encyclopedia of Cryptography and Security*, Springer US, Second Edition, pp:307-312 (2011).

[2] **Bertino, E.** and **Sandhu, R.,** Database Security – Concepts, Approaches, and Challenges, IEEE Computer Society, **2**(1): 2–20 (2005).

[4] **Gaikwad, T. R.** and **Raut, A. B.,** A Review on Database Security, *International Journal of Science and Research*, **3**(4): 372–374 (2014).

[1] **Pourzargham, H.,** Importance of Security in Database, *International Journal of Computer Science and Network Security*, **15**(5): 29–32 (2015).

[3] **Singh, S. K.,** *Database System: Concepts, Design and Applications*, Pearson Education, USA, (2009).

[8] **Kaur, R., Kiranpreet** and **Verma, P.,** Survey on Database Security, *International Journal of Computer Applications*, **105**(10): 27–31 (2014).

[9] **Basharat, I., Azam, F.** and **Muzaffar, A.,** Database Security and Encryption: A Survey Study, *International Journal of Computer Applications*, **47**(12): 28-34 (2012).

[10] **Rohilla, S.** and **Mittal, P. K.,** Database Security: Threats and Challenges, *International Journal of Advanced Research in Computer Science and Software Engineering*, **3**(5): 810–813 (2013).

[11] **Singh, S.** and **Rai, R. K.,** A Review Report on Security Threats on Database, *International Journal of Computer Science and Information Technologies*, **5**(3): 3215–3219 (2014).

[12] **Kulkarni, S.** and **Urolagin, S.,** Review of Attacks on Databases and Database Security Techniques, *International Journal of Emerging Technology and Advanced Engineering*, **2**(11): 253-263 (2012).

[13] **Patil, A.** and **Meshram, B. B.,** Database Access Control Policies, *International Journal of Engineering Research and Applications*, **2**(3): 3150–3154 (2012).

[14] **Servos, D.** and **Osborn, S.L.,** Current Research and Open Problems in Attribute-Based Bccess Control, *ACM Computing Surveys*, **49**(4): 1-45 (2017).

[15] **Ahn, G. J.,** *Discretionary Access Control, Encyclopedia of Database Systems*, Springer, New York,**1**: 864–866 (2009).

[16] **Gertz, M.** and **Jajodia, S.,** *Handbook of Database Security: Applications and trends*, Springer, New York (2008).

[17] **Sahafizadeh, E.** and **Parsa, S.,** Survey on Access Control Models, *2nd International Conference on Future Computer and Communication*, **1**: 1-3 (2010).

[18] **Yadav, A.** and **Shah, R.,** Review on Database Access Control Mechanisms and Models, *International Journal of Computer Applications*, **120**(18): 21–25 (2015).

[19] **Coombs, W. T.,** *PSI Handbook of Business Security, Praeger Security International*, Westport, Connecticut, USA, (2008).

[20] **Hu, V. C., Hu, V. C., Ferraiolo, D. Kuhn, D. R., Schnitzer, A. Sandlin, K., Miller, R.** and **Scarfone, K. A.,** *Guide to Attribute Based Access Control (ABAC) Definition and Considerations*, NIST Special Publication 800-162, Gaithersburg, Maryland, USA, (2014)

[21] **Hu, V. C., Kuhn, D. R., Ferraiolo, D. F.** and **Voas, J.,** Attribute-Based Access Control, *Computer*, **48** (2): 85-88 (2015).

[22] **Ferraiolo, D., Hu, V., Chandramouli, R**. and **Kuhn**, **R.,** *A Comparison of Attribute Based Access Control Standards for Data Service Applications*, NIST Special Publication 800-178, Gaithersburg, Maryland, USA, (2016)

[23] **Biswas, P., Sandhu, R.** and **Krishnan, R.,** Label-Based Access Control: An ABAC Model with Enumerated Authorization Policy, *International Workshop - New Orleans, Louisiana, USA* (2016)

[24] **Biswas, P., Sandhu, R.** and **Krishnan, R**., A *Comparison of Logical-Formula and Enumerated Authorization Policy ABAC Models*, Springer International Publishing Switzerland, pp: 122–129 (2016).

[25] *Practice Guide: Attribute Based Access Control: Executive Summary*, National Institute of Standards and Technology Special Publication 1800-3a, Gaithersburg, Maryland, USA, (2015).

[26] **Verma, S., Singh, M.** and **Kumar, S.,** Comparative analysis of Role Base and Attribute Base Access Control Model in Semantic Web, International *Journal of Computer Applications*, **46**(18): 1–7 (2012).

[27] *Market Trends: Cloud-Based Security Services Market, Worldwide*, 2014, https://www.gartner.com/doc/2607617 [accessed August 21, 2015].

[28] **Kumar, Y., Munjal, R.** and **Sharma, H.,** Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures, *International Journal of Computer Science and Management Studies*, **11**(03): 60–63 (2011).

[29] **Sasi, S. B., Dixon, D.** and **Wilson, J.,** A General Comparison of Symmetric and Asymmetric Cryptosystems for WSNs and an Overview of Location Based Encryption Technique for Improving Security, *International Organization of Scientific Research Journal of Engineering*, **4**(3): 1–4 (2014).

[30] **Pushpa, M.** and **Sujitha, M.,** A Survey on Some of the Symmetric Key Encryption Algorithms used for Database Security, *International Journal of Electronics Communication and Computer Engineering*, **6**(5): 583–587 (2015).

[31] **Alanazi, H. O., Zaidan, B. B., Zaidan, A. A., Jalab, H. A., Shabbir, M.** and **Al-Nabhani, Y.,** New Comparative Study Between DES, 3DES and AES within Nine Factors, *Journal of Computing*, **2**(3): 152–157 (2010).

[32] **Singh, G.** and **Supriya,** A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security, *International Journal of Computer Applications*, **67**(19): 33-38 (2013).

[33] **Arya, P. K., Aswal, M.** and **Kumar, V.,** Comparative Study of Asymmetric Key Cryptographic Algorithms, *International Journal of Computer Science and Communication Networks*, **5**(1): 17-21 (2015).

[34] **William, S.,** *Cryptography and Network Security: Principles and Practice*, Pearson Education, USA, Fifth Edition (2011).

[35] **Mathur M.** and **Kesarwani, A.,** Comparison Between DES, 3DES, RC2, RC6, BLOWFISH and AES, *Proceedings of National Conference on New Horizons in IT- NCNHIT*, 143–148 (2013).

[36] **Shetty, A., Shetty, S.** and **K.,** A Review on Asymmetric Cryptography – RSA and ElGamal Algorithm, *International Journal of Innovative Research in Computer and Communication Engineering*, **2**(5): 98-105 (2014).

[37] **Nimbhorkar, S. U.** and **Malik, L. G.,** A Survey on Elliptic Curve Cryptography (ECC), *International Journal of Advanced Studies in Computers Science and Engineering*, **1**(1): 1-5 (2012).

نظرة عامة لاثنتين من التقنيات المستخدمة في أمن قاعدة البيانات: سياسات التحكم في الوصول إلى قاعدة البيانات وتعمية البيانات

عماد فرح خلف و **مصطفى محمد قاضي**

قسم الهندسة الكهربائية وهندسة الحاسبات، كلية الهندسة، جامعة الملك عبدالعزيز، ص.ب ٨٠٢٠٤، جدة ٢١٥٨٩، المملكة العربية السعودية

المستخلص. مع الكم الهائل من البيانات المولدة في الوقت الحاضر، يجب تنظيم البيانات وإدارتها، بحيث يمكن للمستخدمين الوصول والاسترداد والتحديث للبيانات حسب الحاجة باستخدام نظم قواعد البيانات. أصبح العديد من المنظمات يعتمد على قواعد البيانات للحصول على مزايا الإنتاجية والكفاءة، ولكن قواعد البيانات أصبحت أكثر عرضة للتهديدات الأمنية. لذلك أصبحت حماية البيانات بواسطة إبقائها سليمة وآمنة من أي تطفل غير مرغوب فيه ذات أولوية عالية لتلك المنظمات. يوفر أمن قواعد البيانات تقنيات مختلفة لحماية البيانات من أية تهديدات. تناقش هذه الورقة اثنتين من التقنيات المستخدمة في مجال قاعدة البيانات لتحقيق سلامة وسرية البيانات، وذلك باستخدام سياسات التحكم في الوصول وتعمية البيانات.

كلمات مفتاحية: أمن قاعدة البيانات، التحكم في الوصول، تعمية البيانات.