COMP90050 Advanced Database Systems

Project Report

Security in Data Management

Group 17

Wuang Shen 716090 wuangs@student.unimelb.edu.au

Zhao Peng 899126 zpeng2@student.unimelb.edu.au

Tiange Wang 903588 tiangew2@student.unimelb.edu.au

18th May, 2018

Lecturer: Rao Kotagiri

Contents:

1. Abstract

The database has been widely used by organisations, institutions and companies etc. Thus any kinds of errors such as information leakage or damage would lead to the severe impact on these organisations. So it is significant to ensure the security in database management. Security in database management involves various security issues such as unauthorised activity or users, server crash, data loss, etc. There are different types of security control methods to prevent these kinds of problems which includes data integrity, backup, access control, auditing, encryption and authentication. In the following report, we will introduce and analyse these methods in details.

2. Introduction

Nowadays, most organization use database technology to store and archive data collected from system users. In most case, this data is private and sensitive, and is subject to government regulation and privacy agreements with users. For instance, people may provide their personal information like their address or phone number to service provider, when they register themselves into the system. Without proper security control, this information may get leaked from some malicious cyber activities or operation errors. To protect data from compromises of data confidentiality, integrity and availability, types of security control methods like integrity controls, backups, access control, auditing, authentication and encryption are considered for daily data management. Data integrity control assure the accuracy and consistency of data. Unauthorized data modification can be detected by integrity check methods like RAID parity, mirroring and checksum. Backups provide data recovery from unexpected cases like human error, malicious attacks, and even natural disasters, which results in data loss. Database user can consider types of backup like online, offline, physical, logical, full and incremental backups. Access control restricts users' access to resources by access control policies like mandatory access control, discretionary access control and role-based access control, and only allow privileged users to perform actions on certain resources. Trace-based auditing and transaction logs auditing is used to detect suspicious behaviors happening in the database. To assist with access control, authentication mechanism like password, watermarking and Kerberos authentication is required to verify user's identification. Encryption algorithm like DES and RSA transfer cipher data into encrypted data, making data unreadable for unauthorized users. Apart from detailed explanation for those security controls, different methods for each type of security control will be compared in this report.

3. Data integrity

Data integrity refers to data consistency and accuracy, which supports system's reliability. This is a fundamental element of system or application's security. Without appropriate techniques to protect data integrity, systems that highly relies on data will be vulnerable to security threats and attacks.

There are many different factors may make data corrupted and harm data integrity. Malicious attackers can take advantages of system's vulnerability, and implement intrusion strategies. By doing this, attackers will get access to confidential data and even can make changes or wipe out all data stored in the database system if they gained high privileges from deploying security attacks. Those attackers may even insert dangerous code into the system to help them

to implement further attacks in future. Data integrity issue will also occur if software or hardware errors exist in system's daily operations. In consequence, not only data stored on local storage device but also data transmitted through the network will be damaged. Applications without error handling may run unintended overwrite to the database, then violate data integrity. For example, when two users are trying to read same data concurrently and then overwrite it, the system may overwrite one user's data on another and then result in data loss, if the application doesn't properly manage the data concurrency. Human error is another risk that put data integrity in danger. A user may delete or update data by accident, causing data loss and system malfunction. This normally happens when people send the wrong command to the systems, like rm command, or use the wrong way to run the application against instructions.

3.1Techniques to Assure Data Integrity:

3.1.1 RAID Parity

RAID (redundant array of independent disks) is a data storage technology. It combines multiple physical hard drives into one or more logical drives to share or replicate data among them. By using this technique, data management system can get better data integrity and fault tolerance. RAID generally refers to RAID-5, but RAID-3, RAID-4 and RAID-5 all can use parity to assure data integrity (Patterson et al., 1993). Parity is the result of XOR logical operation across the array (Sivathanu et al., 2005). It is designed for data recovery if a single disk fails, but it is also useful for data integrity check, which can be performed by comparing stored data and recomputed data. For RAID-5, there are four disks in one array. Data will be written into three disk units, and XOR result of those three units will be stored in the space on the 4th drive. Therefore, if one drive in the array broken down, original data can be reconstructed by simply applying XOR operations on remaining drives. For integrity check, integrity violations can be detected if XOR result of 3 units is different from the information stored on 4th disk.

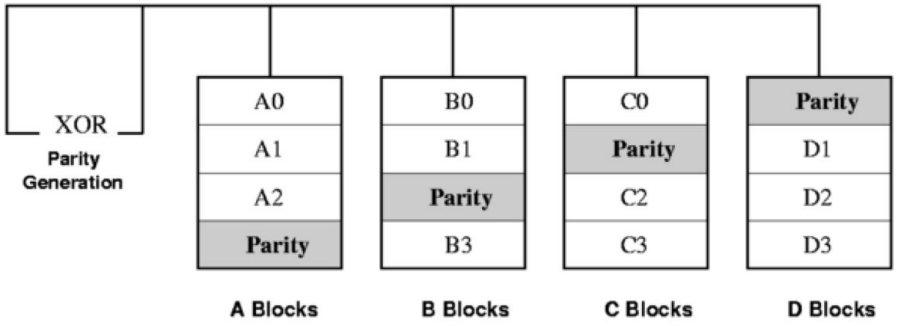

Figure 1: RAID-5: Independent data disks with distributed parity (Sivathanu et al., 2005).

As diagram is shown above, data "0" is presented by A0, B0, C0. Parity information is calculated by XOR operation on A0, B0 and C0, and stored on D blocks. Similarly, below data "0", data "1", "2" and "3" follow the same mechanism to store information.

3.1.2 Mirroring

Mirroring is another common way to assure data integrity. It is actually a form of RAID-1, and also called data replication. It is done by keeping two or more copies of duplicated data stored on different drives. Data integrity can be assured by comparing those copies, and data corruption will be detected if there exists any difference among replicated data. It is easy to implement Mirroring, but weaknesses of this technique are also quite obvious. Mirroring can not detect integrity violation if same modifications applied to original and duplicated copies,

and then modified data will be considered as part of original data (Premkumar and Shanthi, 2014). Therefore, Mirroring is a method that is more suitable for integrity violations caused by data corruption rather than malicious attack. Besides, this technique is inefficient regarding time and storage space, which is a costly solution for companies to adopt. Different from RAID parity, data recovery is not possible for 2-way mirroring if any data loss occurs, as a simple comparison does not indicate which copy contains original data. Although it's possible for more than three copies mirroring, it requires more storage space and time to implement the integrity check.

3.1.3 Checksum

Checksum is a type of redundancy check for the purpose of detecting integrity violation. Each block of data will be assigned with a checksum value, which is generated by cryptographic hash functions. Integrity violation can be detected by comparing stored and newly computed checksum values. There are different hash functions options available for this process. An appropriate hash function should be collision resistant and preimage resistant. Strong collision resistant means it is computationally infeasible to find two data with the same checksum, while preimage resistant means it is computationally infeasible to calculate data for any given checksum (Cornell, 2007). Those two properties of hash function make checksum hard to be manipulated if anyone tries to modify data.

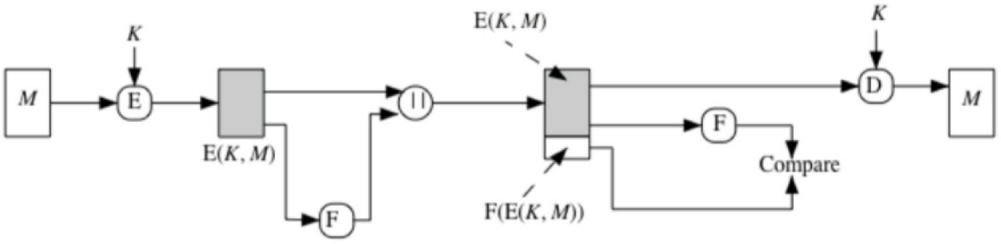

Figure 2: Checksum calculation and verification for encrypted data (Parampalli, 2017)

As you can see from above diagram, the hash value of encrypted data E(K,M) is calculated by F function as F(E(K,M)), and then this value is appended to E(K,M). Integrity verification process is demonstrated on the right side of the diagram. Any modification to encrypted data can be detected by comparing previously and newly computed checksum.

	RAID Parity	Mirroring	Checksum
Pros	· can be used for data recovery · require less time and storage space	· easy to perform	· require less time and storage space · integrity check works for most case

Cons	· only available for single drive failure	· time consuming · require large storage space · may not work for macilious cyber attacks	· performance depends on hash function

Table 1. The comparison of different techniques of Assure data integrity.

4. Data backup

Data backup is a necessary step to ensure system's continuity. With appropriate data backup policy, data loss can be recovered by previously replicated or archived backup, in case of unexpected failure. Things such as malicious attacks, natural disaster, and government regulation, make data backup and recovery a key factor ensuring system's daily function.

4.1 Necessity of data backup

4.1.1 Malicious Attacks:

Same as the problem faced by data integrity, malicious cyber attacks can cause the software or even hardware malfunction, results in data loss. Tools like anti-virus and intrusion detection system can help to reduce this risk to some extent. However, the risk of data loss caused by malicious cyber activities still exists, even if the system equipped with the latest security configuration.

4.1.2 Natural or Human Made Disasters:

Different from malicious attacks, the risk of data loss due to disasters like flood, fire, earthquake or other human-made disasters can be easily overlooked. Those physical threats destroy hardware devices and data warehouse infrastructure. Data recovery will be impossible to process if original copy and replication stored in the same location.

4.1.3 Government Regulation:

Data backup is not only an approach to help company or individual to recover information from data loss, but it is also an important part of government regulation on historical archiving rules abided by most originations (Martin, 2012). For instance, metadata generated from systems or applications are required to be archived for a certain of years, due to potential security auditing.

4.2 Types of backups:

4.2.1 Online vs Offline:

Online backup (also called hot backup) refers to data backup occurred while the database is running. It is mainly designed for the case where database servers are required to be available seven days a week and 24 hours a day without downtime. (Akhtar et al., 2012) During this process, as a user may insert or update data into a database, a locking mechanism is required to ensure data integrity, which is usually handled by database management system. In this case, online backup is supposed to be done when working load is low.

Offline backup (also called cold backup) is on the opposite of online backup. The database is shut down when the offline backup occurs, and users can not do any operations on the

database (Akhtar et al., 2012). Thus, this backup is normally taken at non-working day, to reduce the effect of unavailability for users. Furthermore, to provide availability to users, an offline backup can also be processed from replication server instead of the running server.

	Online Backup	Offline Backup
Pros	Database can still be available	Simple to perform
Cons	Need locking mechanism	Database will be down

Table 2. The comparison of online and offline backup methods.

4.2.2 Physical vs Logical:

Physical backup is a direct raw copy of physical database file. It simply moves data from one device to another. The whole process is simple to perform, and no special software is required to support the backup. Data will not be interpreted during backup and recovery. Without extra data processing, the backup and recovery will be much faster than a logical backup. However, the backup is not portable to machines with a different configuration, because of the lack of data interpretation (Usenix, 2018).

Logical backup processed through SQL queries. Database file like tables, schemas will be exported through database management system. The output of backup also includes some information about how those logical data structured, so it is normally larger than a physical backup. As data can be interpreted by database management system, the backup can be machine independent (Usenix, 2018).

	Physical Backup	Logical Backup
Pros	·fast to perform ·simple to perform	· portable to different machines
Cons	· non-portable, require machine with same configuration	· slower to perform · require more storage space

Table 3. The comparison of physical and logical backup methods.

4.2.3 Full vs Incremental:

Full backup is a complete copy of the original dataset. Although this backup is easy to perform and contain full information of the original dataset, it is very inefficient regarding time and storage space. Therefore, this backup method is normally used on a periodic basis for data centres with a large amount of data. Also, in most cases, full backups are chosen as initial backups, which will be followed by other backup methods later.

Incremental backup is designed for overcoming the inefficiency of full backup method. Compared with full backup approach, incremental backups consume less time and storage space, by only backing up the changes since the last backup. For most case, this method only backup log files. However, incremental backup takes longer time to restore, as complete data

stored in multiple backups. Data recovery process may fail if one of those backup files lost or damaged, because of the dependency among those backup files.

	Full Backup	Incremental Backup
Pros	·Fast and easy recovery ·Easy version control ·Complete dataset backup	·Fast backup process ·Less storage space required
Cons	·More storage space is required ·Backup process is time consuming	·Slower recovery ·Initial full back is required ·High dependency among backups

Table 4. The comparison of full and incremental backup methods.

5. Access control

5.1 Introduction

Besides data integrity and backup, there are also many other techniques which help to secure the database. Access control is one of them which restricts the access for a selective users to certain places, resources or database (En.wikipedia.org, 2018). Without access control, the credential data will be in danger of exposing to unauthorized users or parties. For example, when your company runs a website and if you do not control the access of who can access to the back end of the website, then everyone can manipulate it where your opponent might hack into your website and change everything to ruin your business. Thus with access control, we can eliminate the danger that data revealed to unauthorized parties. There are many access control policies which are generally used now such as Mandatory Access Control, Discretionary Access Control and Role Based Access Control etc. We will discuss them in the following sections of this report.

5.2 Access Control Policies

5.2.1 Mandatory Access Control

Mandatory Access Control (MAC) constrains the access of users where the objects or elements are classified by levels. The users and elements in the system are each assigned with a security level with the sensitivity with such information. The security levels include TopSecret (TS), Secret (S), Confidential (C) and Unclassified (U) where TS > S > C >U. Each of the security levels is the dominant of itself and the ones below it are their hierarchy (Akshay and Meshram 2012).

There are two principles in Access Control -- Read down and Write up. Read down is that a subject can only read from the objects which the access level is within or below it. For example, if I am a user whose security level is assigned to S, then I can only read the objects which have the security level of S, C or U. Write up principle means a subject can only write the objects which dominate the access level of the subject. For instance, I' m still the user with security level S, with the write-up principle, I can write a TS object, but I cannot write C or U object. It might seem a little bit odd that we could not write levels blow us but this is for security issues that we do not leak important secrets downwards. In the same word which

helps to understand, since I cannot read the secrets in TS, then I won't leak data from TS to U (Sandhu and Samarati, 1994).

5.2.2 Discretionary Access Control

Discretionary Access Control (DAC) governs the access based on the user's privileges or access rights. DAC rules define the access nodes for the users who object is allowed access or denied access. When a user requests an operation or access for an object in the system, the system will first check whether this user has the privileges of access on this object or not; if yes, then the user is allowed for accessing; if no, then a denial message will be sent to the user (Sandhu and Samarati, 1994).

DAC has been widely used such as commercial or industrial environments because of its flexibility. However, it also has some drawbacks that one can easily bypass the access nodes through the authorisations. (Sandhu and Samarati, 1994) For instance, I can share the information which I can access to another friend who is not authorised to access.

5.2.3 Role Based Access Control

The basic idea of Role Based Access Control (RBAC) is the notion of role. Users are assigned to different positions which they can perform certain operations based on their roles. In RBAC model, the system grants the access to specific services on roles rather than allow it directly to the users. Thus the user plays a role authorised to do all the activities that the role is granted to do. Generally, users can take different roles based on different situations decided by the administrator. And also, different users can play the same roles. This model has many advantages, and we will introduce some of them in this report.

Frist of all, RBAC assign roles to users make it easier for management. If a user had promotion and his or her responsibility has changed in the system, we can remove specific roles from the user and apply other roles to him or her. We do not need to go into each object and remove the relation between the objects and the user which would cause much more time and inconvenience. Secondly, the hierarchy is very common in assigning roles. So when a user is assigned to the specific role, he or she will simultaneously inherit all the privileges and authorisations for that role. Thirdly, applying for roles on users separates the duties that they will not mess with the system. For example, in the real world, a person who is responsible for paying a debt should not be the one who authorises the paycheck (Sandhu and Samarati, 1994).

5.3 Comparisions

	MAC	DAC	RBAC
Access methods	through administrator	classification of users	roles of users
Where used	initial unix system	U.S Department of Defence	widely used

Table 5. The comparison of different access methods.

In MAC, the administrator assigns security levels to the users for the access of subjects and objects in the system. It is a quite simple policy, but it does provide excellent security on the access of the information. RBAC models assign roles to users to grant access based on the roles which are smooth and precise for management. These models are all beneficial and are being widely used in companies nowadays.

6. Auditing

Aside from the technology above, we have another security control method in the database which is auditing. Auditing will discover and observe any actions performed by database users. Hence it will ensure that there is no suspicious behaviour happening within the database. For example, since people or any other organisations and companies use computers daily and we will put our personal information on the different website to register or do online shopping. So it is possible that hackers will hack into the database of these kinds of website and stole our personal information for other illegal use. Hence with auditing, the system will be watching every operation performed within these databases such as transactions which will reduce the risks that the information could be modified or retrieved illegally.

6.1 Auditing Methods

When having the appropriate choice of auditing method, we will achieve better result an improve the accuracy when applying database audit. And the method of auditing mainly focused in two ways; The first way is based on Mis-usage, and the other methods are based on the anomaly.

Auditing and detection based on misusage will first analyse the different attack method and learn the different characteristics of these attacks, then we will compare our data to these characteristics to see whether an action is legal or not. There are two methods: pattern matching and expert system based on rules. By pattern matching, we will compare the models of our data in the library which is simple and high efficient. And for the Expert system based on rules, we will compare the operating characteristics we retrieved by users to the rules in the system to see whether a user behaviour is illegal or not (Shao 2015).

Auditing method based on anomaly is that by establishing a set of normal behaviours of a subject, we will find any other operations as abnormal. These methods will use the concept of data mining such that we retrieve a large set of data and train them to extract a model of user characteristics. Hence when we see any actions on the database system that does not match the normal user behaviour, we will consider it as an abnormal behaviour (Shao 2015).

6.2 Oracle example

Many databases perform auditing for security issues such as SQL, Oracle, etc. We will introduce some auditing types that Oracle used in the following section. Oracle audit both successful and unsuccessful statement executions and also different activities of users.

The first auditing type used in Oracle is Statement Auditing which allows us to track statements by its form rather than the schema objects. For instance, “AUDIT TABLE”, audit DDL statement, ignoring the table which was issued. Privilege auditing enables us to track the use of powerful system privileges which is more useful comparing to statement audit.

Another type is schema Object Auditing which we can audit certain statements on a specific schema object such as AUDIT SELECTION ON employees (Docs.oracle.com. 2018).

6.3 Auditing techniques

There are several auditing techniques we will introduce. The first one is Trace-based auditing which is usually built into the native capacity of database management system directly. These techniques can audit actions like both successful and unsuccessful login and logoff attempts or restart of the database server. However, this technique may result in performance degradation when commands are set to enable auditing (Craig 2007).

The second auditing technique is scan and pares the database transaction logs. The database will use the transaction logs to audit every modification made within the database and will capture the actions that what data has been changed when it was altered by which user. The disadvantage of this technique is that modification will be lost when stop logging and it is difficult to maintain the logs over a long time (Craig 2007).

	Trace-based auditing	Scan and parse
pros	easy to use and set up	can capture modifications within database
cons	insufficient granularity	do not capture reads on logs difficult to maintain

Table 6. The comparison of different auditing methods

7. Authentication

Shown in the diagram below is the process of authentication, which is a verification mechanism which checks the user's credentials and confirms the identities of users through some methods, such as the secret key, encryption algorithms and biological features (En.wikipedia.org, 2018). It is the essential security requirement that identifying the users in order to verify their operations on data before providing the permissions to them (Anon, 2003). For the database security aspect, the administrator of the database could offer authorities to different users, and each user connects to the database with its password to process different levels and files of modification and deletion. The authority should correspond to the role of the user. In other words, it is impossible that the product database operators have access to modify and delete any records of the stakeholder database, which owes to the effect of authentication. Once the database system is without the authentication, it may lead to some such disordered and negative issues that low-authority users gain access to alter high-level data and records.

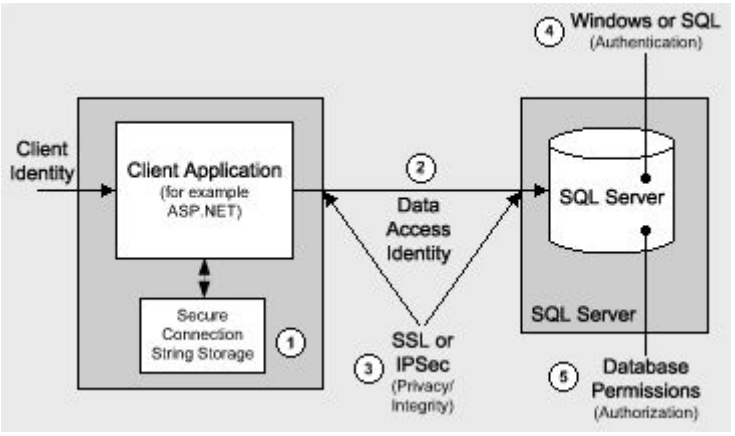

Figure 3. The sketch of the authentication process (Meier et al., 2006).

The password is the essential form of authentication. When a user establishes the connection to the database, the set of ID and password must be provided to avoid unauthorised use of the database. Meanwhile, the administrator of the database is in charge of encrypting the passwords, storing the ciphertext in the data dictionary and allowing users to change the passwords at any time to enhance the essential password features. Hence, users who attempt to connect to the database could process the authentication through the pair of ID and password saved in the database. However, the password method is vulnerable to theft and misuse and not sufficient to defend all the attacks. Some weak passwords even become the target of attacks, including Brute Force, Man-in-the-middle attack and replay attack. Therefore, to achieve the password system, the various requirements are necessary to different roles in the process of the implement. Besides the requirement of strong passwords for users, for the data dictionary, the security system must guarantee the passwords to be confidential. In addition, there are some simple steps for the administrator of the database to reinforce the control of database security. For instance, it constrains users and provides them with identifying code and three login attempts to avoid hackers maliciously occupied the resources. The server could also lock the account automatically after several failed login attempts in an hour. And the password will expire for a period, and the system asks the user to update the password periodically.

For the integrity and non-variability of the database, the database watermarking is a standard method to authenticate. It refers to a widely used technique that embedding additional but not visible information (watermarking) into the original content of the database and not distorting the raw data (Shelar et al., 2016). The watermarking technique could be divided into two steps: watermark embedding and watermark verification (K.Rathva and G.J, 2013). In the first step, the administrator of the database utilises the private key K to insert the watermark into the database and display the database with the watermark publicly. Then in the step of the verification, use the suspicious database as the input parameter with the private key K to calculate the new watermark information and compare the new and initial watermark to determine whether the database is modified.

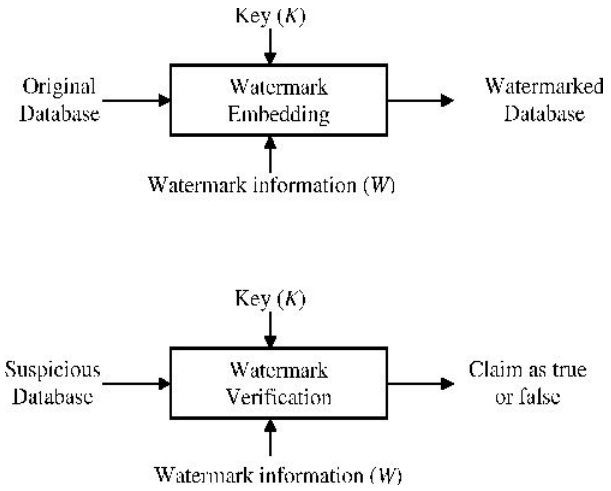

Figure 4. The watermarking algorithm for database (Alfagi et al., 2016).

The watermarking has many features. The primary attributes are the detectability, robustness and updatability (K.Rathva and G.J, 2013). For example, the administrator could have methods to check the watermark through the private key, and the suspicious database and the watermark could maintain the features after the basic operations like insert, modify and delete, and even some database attacks. Moreover, the watermark is not able to change after the insertion or deletion of the tuple of the database. However, the watermarking technique is not available when the information cannot be modified, and the database is without the primary key (Shelar et al., 2016). Meanwhile, since the updatability is challenging to achieve, it is easy to attract attacks, such as the insertion attack, which inserts new tuples to the database to impact on the watermark to lead to the synchronisation errors (K.Rathva and G.J, 2013).

In addition, there is an authentication serves for protecting the communication between the client and server named Kerberos authentication. It enables both the client and server could authenticate each other to prevent the eavesdropping, replay attack and maintain the integrity of the data (El-Emam et al., 2011). The process of Kerberos authentication for database contains three roles: Client (User), Server (Database) and key distribution centre (KDC).

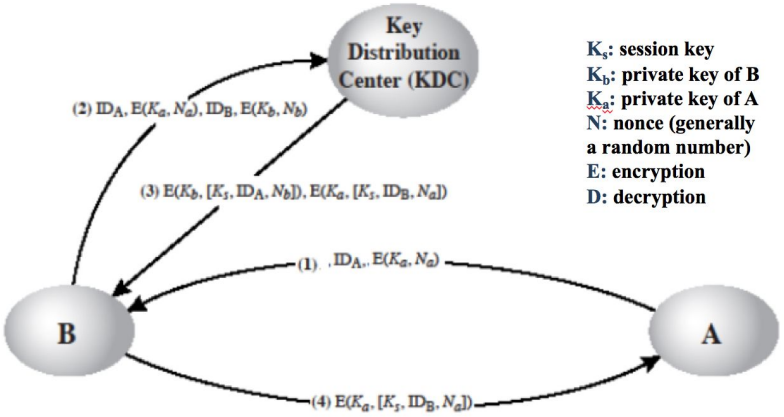

Figure 5. The key distribution centre (KDC) (Slideplayer.com, 2018).

The KDC is the crucial component to realise the Kerberos authentication. The methodology is to use the trusted third party called key distribution centre (KDC) which is the secure

machine on the Internet (Al-Janabi and Rasheed, 2011). When processing the communication, both sides entrust it to certify both identities and share the secret key with the KDC. As shown above, suppose that Alice tries to communicate with Bob. First, she sends a message to the KDC, which is protected by the shared key Ka from the KDC, and the communication request to Bob. Then the KDC generates a new encryption key Ks for communications between Alice and Bob and sends it in a message called a 'ticket'. One 'ticket' message consists of two messages. The first is for Alice with a new key. The second one is for Bob which is encrypted with the private key of Bob, and also with the new key. Alice sends the second message which belongs to Bob to Bob. Lastly, she shares the session key Ks with Bob.

In Kerberos authentication, there are two major disadvantages. The KDC must always be online and connect to the parties of communications (Al-Janabi and Rasheed, 2011). Once it goes offline, the communication initialisation will be failed. Moreover, if the KDC is under attack, the communications between any pairs of Client and Server will be eavesdropped and leaked.

	Password	Watermarking	Kerberos authentication (KDC)
Pros	● Avoid low-authority users modify the database. ● Accomplish the essential authentication with low cost.	● Hard to remove and provide long-time protection. ● Not modify any items in database.	● Reinforce the reliability of the communications.
Cons	● Require the data dictionary to guarantee the passwords to be confidential. ● Easy to be stolen and misuse. ● Need the maintenance mechanism.	● Not available when the information is not able to be modified. ● Attract the insertion attack. ● Require the primary key.	● The KDC must always be online. ● Once be attacked, the information will be eavesdropped and leaked.

Table 7. The comparison of the authentication methods.

As the table above, the password, watermarking and Kerberos authentication respectively achieve the authentications for user/authority, data in database and communications between user and database. These three methods of authentication do not conflict with various features but also some disadvantages.

8. Encryption

There are always some sensitive data stored in the database or the server which are required to be protected from attackers. In order to improve the security level of the database,

encryption technique was developed and became the standard method for various databases (Basharat, Azam and Muzaffar, 2012). The encryption technique stands for the process of concealing or transforming data in the database into ciphertext. Excluding the administrator and developers with the access key, the enciphered information is not available for others to protect the data from being accessed by individuals unauthorised access. In the simple database structure, the encryption technique primarily performs on the storage side for data and client side for users (personal information & password).

The security of database is often a considerable challenge for database admins and developers. Once the database is attacked or broken, the data and password which stored in the database as the form of plaintext are easily captured by the attackers directly. Sometimes the database works well, but attackers pretend like the system admin or database admin, access to the console and utilise the 'SELECT' command to steal the plaintext of data and passwords. In other words, without the encryption, the data and password would be exposed to attackers directly.

The database encryption could be divided into symmetric and asymmetric methods. As the display below, the symmetric encryption algorithm requires the sender (Alice) and receiver (Bob) to select a unified secret key for encryption and decryption before processing communications (Shmueli et al., 2010). Then Alice encrypts the plaintext with the secret key into the ciphertext through the encryption algorithm and sends it to the Bob. After Bob receives the ciphertext, if he wants to interpret the message, he must know the secret key and use the same key with the inverse encryption algorithm to decrypt the ciphertext to make it readable.

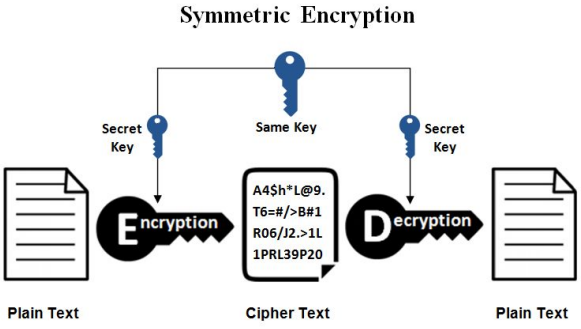

Figure 6. The display of the process of symmetric encryption (SSL2BUY.com, 2017).

The DES is a widely used symmetric encryption algorithm which utilises a 64-bit key to substitute the 64-bit plaintext blocks into the 64-bit ciphertext blocks bit by bit. However, in the actual use, only 56 bits of the key participate in the DES operation, and the other 8 bits are used for verification to improve security level (Shmueli et al., 2010).

The symmetric encryption algorithm has a small amount of calculation, fast encryption speed and high encryption efficiency. Nonetheless, the lifecycle of the secret key is short. Meanwhile, the security level of symmetric algorithms relies on the secret key. Leaking the secret key will result in that attackers are able to decrypt the messages Alice and Bob send or received. Therefore, the secrecy of the secret key is crucial to the security of communications.

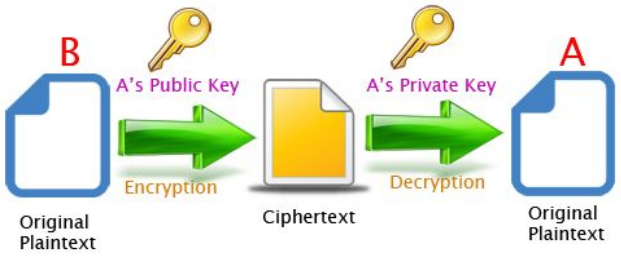

Figure 7. The display of the process of asymmetric encryption (giuseppeurso.eu, 2015).

Asymmetric encryption, also called public-key encryption, requires and generates one pair of keys: one public key and one private key (En.wikipedia.org., 2018). If the data is encrypted with the public key, it can only be decrypted with the corresponding private key; conversely, the public key could also decrypt the data which encrypted by the private key. Since the process of encryption and decryption use two different keys, theoretically, anyone could establish the secure communications with each other, which enhances the reliability compared to the transference of the secret key in symmetric encryption.

The RSA is the most commonly used asymmetric encryption algorithm. Suppose Alice wants to receive Bob's message through an unreliable media. She can generate a pair of 1024-bit keys using the RSA algorithm and publish one of them as a public key to other parties. Then Bob uses the public key to encrypt confidential information and sends it back to Alice. Finally, Alice decrypts the encrypted information with her private key.

However, the RSA algorithm alone cannot resist the man-in-the-middle attack (MITM). Assuming that Eve sent Bob her public key and convinced Bob that it was Alice's public key, then she could become the middle-man attacker between Alice and Bob (En.wikipedia.org., 2018). Eve can intercept all Bob's message to Alice, decrypt the message with her private key, read the message, and then encrypt the message with Alice's public key and pass it to Alice (En.wikipedia.org., 2018). In theory, neither Alice nor Bob will find Eve eavesdropping and intercept their communications. Thus, the RSA algorithm requires co-operating with digital signature technique to increase encryption strength. In addition, as the strength of the algorithm is complicated and the decryption speed is slow, the RSA algorithm is more appropriate for the encryption and decryption of small amounts of data. Above all, the difficulty of factoring two large prime numbers determines the reliability of the RSA algorithm, and the selection of large prime numbers is also challengeable (Basharat, Azam and Muzaffar, 2012).

	Symmetric encryption algorithm (DES)	Asymmetric encryption algorithm (RSA)
Pros	• A small amount of calculation. • High efficiency with fast encryption and decryption speed.	• Suitable for small amounts of data. • Enhances the reliability of transference of encryption keys.
Cons	• Difficult to transfer and	• Complex algorithm strength

	maintain the confidentiality the secret key. ● Short key lifecycle.	with slow decryption speed. ● Challengeable to select large prime numbers. ● Cannot resist the MITM

Table 8. The comparison of the encryption methods.

The table above demonstrates that in order to increase the security level, it is necessary to pay more attention to the encryption algorithm and the size of the key. Nevertheless, the database encryption is not sufficient to resist all the attacks. In addition, some advanced algorithms require high deployment costs and high-performance hardware to support. Hence, the application of the database encryption still have some limitations.

9. Conclusion and future directions

To sum up, database security is a massive challenge for any developers and administrators. In today's world, databases are vulnerable to attract various attacks. In this report, it identified the major risks and problems which the database may encounter. Moreover, it discussed preliminary solutions from six aspects, data integrity, data backup, access control, auditing, authentication and encryption, to decrease the risk of attacks and protect the confidentiality, integrity and availability of data. These preliminary methods have advantages and disadvantages, and some of them still have limitations on implements. When engaging in data management, these aspects should be concerned. The future direction of development and work should focus on these techniques and enhance the effectiveness and efficiency of them.

10. Reference

Akshay Patil and Prof. B. B. Meshram. (2012) Database Access Control Policies. Database Access Control Policies. *International Journal of Engineering Research and Applications* (IJERA), pp. 3150-3154.

Alfagi, A. , Manaf, A. , Hamida, B. , Khan, S. , Elrowayati, A. (2016). Survey on relational database watermarking techniques. *Journal of Engineering and Applied Sciences,* p.11.

Al-Janabi, S. and Rasheed, M. (2011). Public-Key Cryptography Enabled Kerberos Authentication. *2011 Developments in E-systems Engineering*.

Anon, (2003). *4 Authenticating Users to the Database*. [online] Available at: https://docs.oracle.com/cd/B14117_01/network.101/b10777/authuser.htm.

Basharat, I., Azam, F., and Muzaffar, A. W. (2012). Database security and encryption: A survey study. *International Journal of Computer Applications*, pp. 28-34.

Chaudhari, M. R. R., and Bakal, J. W. (2015). Overview of Database Auditing for Oracle Database. *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, *4*(7), pp. 189-195.

Cornell, D. (2007). *Properties of Secure Hash Functions | Denim Group*. [online] Denimgroup.com. Available at: https://www.denimgroup.com/resources/blog/2007/11/properties-of-1/ [Accessed 23 May 2018].

Craig S. Mullins. (2007) Data Access Auditing: A Compliance Requirement. [online] Available at: http://www.craigsmullins.com/dbta_075.htm [Accessed 23 May 2018]

Docs.oracle.com. (2018). *Database Auditing: Security Considerations*. [online] Available at: https://docs.oracle.com/cd/B19306_01/network.102/b14266/auditing.htm#CHDJBDHJ [Accessed 23 May 2018]

El-Emam, E., Koutb, M., Kelash, H. M., & Faragallah, O. S. (2011). An Authentication Protocol Based on Kerberos 5. *IJ Network Security*, *12*(3), pp.159-170.

En.wikipedia.org. (2018). *Public-key cryptography*. [online] Available at: https://en.wikipedia.org/wiki/Public-key_cryptography [Accessed 21 May 2018].

En.wikipedia.org. (2018). *Access control*. [online] Available at: https://en.wikipedia.org/wiki/Access_control [Accessed 23 May 2018].

Giuseppe Urso Blog. (2015). *Asymmetric RSA encryption in Java*. [online] Available at: http://www.giuseppeurso.eu/en/asymmetric-rsa-encryption-in-java/ [Accessed 21 May 2018].

K.Rathva, M. and G.J, S. (2013). Watermarking Relational Databases. *International Journal of Computer Science, Engineering and Applications*, 3(1), pp.71-79.

Martin, B. (2012). *Data Backup Government Regulations*. [online] Greatlakescomputer.com. Available at: https://www.greatlakescomputer.com/blog/bid/132039/Data-Backup-Government-Regulations [Accessed 23 May 2018].

Meier, J.D. , Mackman, A. , Dunner, M., and Vasireddy, S. , (2006), *Key data access security issues* [ONLINE]. Available at: https://msdn.microsoft.com/en-us/library/ff649357.aspx [Accessed 10 May 2018].

Parampalli, U. (2017). COMP90043 Cryptography and Security: Message Authentication Code[Lecture slides]. The University of Melbourne, 2017.

Patterson, D., Gibson, G., and Katz, R. (1993). A case for redundant arrays of inexpensive disks (RAID). In *Proceedings of the ACM SIGMOD*, pp.109–116.

Premkumar, P. , and Shanthi, D. (2014) An Efficient Dynamic Data Violation Check Technique For Data Integrity Assurance In Cloud Computing. *International Journal of Innovative Research in Science, Engineering and Technology* (IJRSET), 3(3) pp. 2649-2654.

Sandhu, R. S., and Samarati, P. (1994). Access control: principle and practice. *IEEE communications magazine*, *32*(9), pp. 40-48.

Shao, Z., Li, Y., Zhang, K., Zeng, G., and Zhao, S. (2015, August). An Audit Method Based on Mathematical Statistics Detection in Database Audit System. In *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2015 7th International Conference on* (Vol. 2, pp. 203-206). IEEE.

Shelar, P., Patil, P., Khamkar, A., Patil, A., Malavi, T. and Parabkar, N. (2016). Database Security using Watermarking Technique. *International Research Journal of Engineering and Technology (IRJET)*, 03(05).

Shmueli, E., Vaisenberg, R., Elovici, Y. and Glezer, C. (2010). Database encryption. *ACM SIGMOD Record*, 38(3), p.29.

Sivathanu, G., Wright, C.P. and Zadok, E. (2005) Ensuring data integrity in storage: techniques and applications, Proceedings of the 2005 ACM workshop on Storage security and survivability. pp.26-36

Slideplayer.com. (2018). *Key Management and Distribution - ppt video online download*. [online] Available at: http://slideplayer.com/slide/12176296/ [Accessed 20 May 2018].

SSL2BUY Wiki - Get Solution for SSL Certificate Queries. (2017). *Symmetric vs. Asymmetric Encryption – What are differences?*. [online] Available at: https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences [Accessed 21 May 2018].

Usenix.org. (2018). *Physical Backup*. [online] Available at: https://www.usenix.org/legacy/publications/library/proceedings/osdi99/full_papers/hutchinson/hutchinson_html/node7.html [Accessed 22 May 2018].

Usenix.org. (2018). *Logical Backup*. [online] Available at: https://www.usenix.org/legacy/publications/library/proceedings/osdi99/full_papers/hutchinson/hutchinson_html/node6.html [Accessed 22 May 2018].