**Abstract**

As we know, data is the important elements for database because database is collection of data and programs to perform operations on that data. With the development of database, data security is becoming one of the most urgent challenges. While protecting the privacy of individual recorders, there are more and more interests in securing aggregate data. Since data encryption is the basic technique used for guarding specific sensitive data-items or objects, it's easy to think that this technique is applied to databases to replace building walls around servers or hard drivers. This can prevent not only outside attacks which come from hackers, but also inside attacks which are from employees who accidentally delete sensitive information. In this paper, we will overview basic encryption terminology in database management system (DBMS) and several factors to affect encryption. Then, we will focus on different encryption applications which include three popular database manufacturers, Oracle, MySQL, Microsoft SQL, and some third party vendors, Protegrity, RSA BSAFE, SafeNet. Through comparing these applications, the final goal is to offer different database encryption options — showing how each variant affects application performance, manageability, and security, then to help a user to develop an own database encryption strategy that will meet its individual needs.

**1. Introduction: Importance for Encryption**

Nowadays, data seems to be kept on everything – from the websites we visit, to how many inventories for one product, to what kind of clothing they are buying, and how many customers this company has. Most important thing is data helps us to extract information and make various decisions. Hence, we stored data in database so that retrieving and maintaining it becomes easy and manageable. However, the effect for a database’s large-scale leakage is much worse than a leaked document. For example, in 2012, the famous social networking website *LinkedIn* was hacked by Russian hackers, *Russian cybercriminals*. More than 6 million users could not log in their accounts1. In the same year, *Dangdang.com*, one of China's biggest e-commerce websites declared their database was hacked too. Form October 2011 to March 2012, More than 12 million users’ information was leaked. Some users deposited some e-money in their accounts and hackers had taken the money out2. These cases happened every day and they tell us how important to secure data in database.

To achieve a safe database environment, Database security emphasizes three main properties: confidentiality, integrity and availability3. Roughly speaking, the confidentiality property prevents unauthorized persons to access the protected data. The integrity property guarantees that the data cannot be corrupted in an invisible way. The third property, availability ensures timely and reliable access to the database.

Over the last two decades, database security generates two main methods, access controls and encryption to protect data. Usually, access controls to gate that should or should not be allowed access to the database, and encryption to protect data at rest. They can meet the requirements for those aforementioned properties. Confidentiality connects with access controls. Integrity points to encryption. Availability means these two methods should not hinder users’ operation. In the real world, these two methods are complementary each other. Sometimes, access controls could not block some unauthorized users. For example, an intruder can infiltrate the information system and try to mine the database footprint on disk. Access controls breaks down. Then, it is the turn for encryption. The purpose of database encryption is to ensure the database opacity by keeping the information hidden to these unauthorized users. Even if attackers get though the firewall and bypass access control policies, they still need the encryption keys to decrypt data.

**2. Database Encryption Basics: What needs to be known?**

* **Definition**
* **Classification**

**3. Different Factors for Database Encryption**

* **Encryption granularity**
* **Encryption Algorithm**
* **Encryption Key**
* **Key Management**

1. **Cryptographic Access Control**
2. **Secure Key Storage**

**4. Applications**

* **Data Encryption within Database**

1. **Oracle**
2. **MySQL**
3. **Microsoft SQL**

* **Data Encryption outside Database**

1. **Protegrity**
2. **RSA BSAFE**
3. **SafeNet**

**5. Conclusions**