Database Security with AES Encryption, Elliptic Curve Encryption and Signature

Than Myo Zaw, Min Thant
ITMO University
Saint-Petersburg, Russia
thanmyozaw7@gmail.com; thanmyo@media-publisher.ru

S. V. Bezzateev, *Member, IEEE*
Saint-Petersburg State University of Aerospace Instrumentation
Saint-Petersburg, Russia
bsv@aanet.ru

***Abstract*— A database is an organized collection of data. Though a number of techniques, such as encryption and electronic signatures, are currently available for the protection of data when transmitted across sites. Database security refers to the collective measures used to protect and secure a database or database management software from illegitimate use and malicious threats and attacks. In this paper, we create 6 types of method for more secure ways to store and retrieve database information that is both convenient and efficient. Confidentiality, integrity, and availability, also known as the CIA triad, is a model designed to guide policies for information security within the database. There are many cryptography techniques available among them, ECC is one of the most powerful techniques. A user wants to the data stores or request, the user needs to authenticate. When a user who is authenticated, he will get key from a key generator and then he must be data encrypt or decrypt within the database. Every keys store in a key generator and retrieve from the key generator. We use 256 bits of AES encryption for rows level encryption, columns level encryption, and elements level encryption for the database. Next two method is encrypted AES 256 bits random key by using 521 bits of ECC encryption and signature for rows level encryption and column level encryption. Last method is most secure method in this paper, which method is element level encryption with AES and ECC encryption for confidentiality and ECC signature use for every element within the database for integrity. As well as encrypting data at rest, it's also important to ensure confidential data are encrypted in motion over our network to protect against database signature security. The advantages of elements level are difficult for attack because the attacker gets a key that is lose only one element. The disadvantages need to thousands or millions of keys to manage.**

Keywords— AES Encryption; ECC Encryption; ECC Signature; PQC; Confidentiality; Integrity

I. INTRODUCTION

A database is a collection of data items that provides an organizational structure for information storage [1]. Information stored in databases is often considered as a valuable and important corporate resource. Database also provides a mechanism for querying, creating, modifying and deleting data. Today, there exist many different types of databases, not only the traditional relational databases but several other architectures designed to handle different types of data. A database can store relationships and data that are more complicated than a simple list with lesser or no redundancy. A relational database stores data in tables. Normally a table is based on one information theme. For example, an IOV list can be divided into name table, car type table, and address table. A table is a two dimensions grid of data that contains columns and rows. The convention in relational database world is that columns represent different attributes of an entity and each row represents the instance of the entity. The organizations moved into non-relational databases in the late 90˝s. It does not support Relational database. Database security is concerned with ensuring the secrecy, integrity, and availability of data stored in a database.

To provide good, strong, and more efficient method that eliminate unauthorized users for accessing database and secret data by clear easy steps: manage computer risk, prevent weakness in security management, and additional measures of security by develop method for encrypt sensitive data before storage in the database.

In this paper, we create a more secure way to store and retrieve database information with ECC encryption and signature that is confidentiality, integrity and availability. This paper is organized as follows: Section II. describes Background of Database, Section III describes Security Challenges in Database, Section IV describes Database Security with AES Encryption, ECC Encryption and Signature. A conclusion is drawn in Section V.

II. BACKGROUND ON DATABASE

A. Relational Database

A relational database is a set of formally described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables. A relational database is a collection of data items with pre-defined relationships between them. These items are organized as a set of tables with columns and rows. Tables are used to hold information about the objects to be represented in the database. Each column in a table holds a certain kind of data and a field stores the actual value of an attribute. The rows in the table represent a collection of related values of one object or entity. Each row in a table could be marked with a unique identifier called a primary key, and rows among multiple tables can be made related using foreign keys. This data can be accessed in many different ways without reorganizing the database tables themselves.

978-1-7281-2288-5/19/$31.00 ©2019 IEEE

The relational database was invented in 1970 by E. F. Codd [2], then a young programmer at IBM. "A Relational Model of Data for Large Shared Data Banks," Codd proposed shifting from storing data in hierarchical or navigational structures to organizing data in tables containing rows and columns. Each table, which is sometimes called a relation, in a relational database contains one or more data categories in columns, also called attributes. Each row, also called a record or tuple, contains a unique instance of data, or key, for the categories defined by the columns. Each table has a unique primary key, which identifies the information in a table. The relationship between tables can then be set via the use of foreign keys -- a field in a table that links to the primary key of another table.

Conceptually, database is a component of database system. Besides database, database system consists of database users, database applications and Database Management Systems (DBMS) [3]. Database users need not to be always human. It is possible, for example, for other software programs to be users of the database. Users interact with database application and application further depends on the DBMS to extract and store data in the database. The DBMS acts as a gatekeeper. All the information owing in or out of database must pass through the DBMS. It is a critical mechanism for maintaining quality of data and database. Users and database applications are not allowed directly to interact with database. A Database Management System is an intermediary between database applications and database. The DBMS creates and manages the database. DBMS can be categorized based on its data model. Database is divided into tables and they are connected through a “key field”. RDBMS is the most famous and used database model [3].

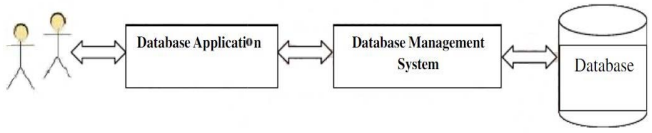

Fig. 1. A Database System

RDBMS remain a key technology to store structured data. But with growing size of data, companies do need modern technologies to maintain and process data. RDBMS are not that good for large data volumes with varying datatypes. They also have scalability problem and often result into failure while performing distributed sharing. Oracle Real Application Clusters (RAC) [4] is a relational database cluster that provides high availability, reliability and performance. Also, MySQL cluster [5] is another example where relational database scale on large cluster. RDBMS satisfy ACID (Atomicity, Consistency, Isolation and Durability) [6] properties defined by Jim Gray in the late 1970s. Consistency is bottleneck for scalability of relational databases. RDBMS follow strict data model and can not violate ACID properties. That is why NoSQL data store were developed to address the challenges of traditional databases.

III. Security Challenges in Database

Database security assures the security of databases against threats. It is concerned within information security control that involves the data protection, the database applications or stored functions protection, the database systems protection, the database servers and the associated network links protection. For data protection enforcement of access control policies based on data contents, subject qualifications and characteristics, and other relevant contextual information, such as time mechanisms are used. The main database security risks are unauthorized or unintended activity or misuse by authorized database users, database administrators, or network or system managers, or by unauthorized users or hacker inappropriate access to sensitive data, metadata or functions within databases, or inappropriate changes to the database programs, structures or security configurations. Also, malware infections causing incidents such as unauthorized access, leakage or disclosure of personal or proprietary data, deletion of or damage to the data or programs, interruption or denial of authorized access to the database, attacks on other systems and the unanticipated failure of database services may occur in database.

IV. Database Security with AES Encryption, ECC Encryption and Signature

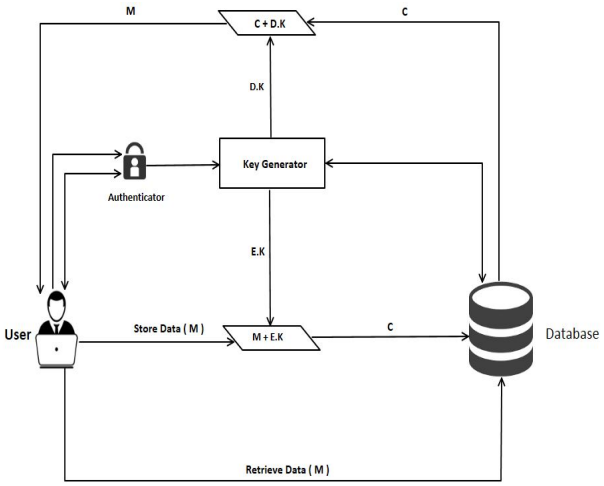

Fig. 2. Database store and retrieve schema

A database is an organized collection of data, generally stored huge amount data petabyte or zettabyte or much more and accessed electronically from a computer system. In database data is organized rows and columns format in a series of tables such that tables $(T_{1,\ldots,n})$. We can perform insert, update, delete operation on database. Database is used by various areas like hospital, defense, school, college, government office, social media etc. to store the sensitive information. As databases contains the sensitive information, security of such databases is a primary concern. cryptography is used to secure data. A user wants to the data stores or request, the user needs to authenticate. When a user who is

authenticated, he will get key from a key generator and then he must be data encrypt or decrypt within the database.

A. Database Confidentiality and Integrity

The basic characteristics of information that must be available in the database system is confidentiality of data which means the quality or state of preventing discloser or exposure to unauthorized individual or system, in other meaning confidentiality is ensuring that only those with right and privileges to access a particular set of information are able to do so and that who are not authorized are preventing from obtaining access[7]. In general information integrity it means different things to different people, and probably continues to do so for some time, the definition of integrity as being concerned with improper modification of information "Our definition of security refers to the protection of data against unauthorized disclosure, or destruction, integrity refers to the accuracy or validity of data [7]. The view of security indicates that integrity is on component of security and accuracy and validity is one component of integrity. In this database, we use AES symmetric encryption [8] and ECC encryption for database Confidentiality and ECC signature for database Integrity. A 256 bits key in ECC offers about the same security as 3072 bits key using RSA [9].

	C_1	C_2	C_3	C_4	C_5	C_n
R_1	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	d_{1n}
R_2	d_{21}	d_{22}	d_{23}	d_{24}	d_{25}	d_{2n}
R_3	d_{31}	d_{32}	d_{33}	d_{34}	d_{35}	d_{3n}
R_4	d_{41}	d_{42}	d_{43}	d_{44}	d_{45}	d_{4n}
R_5	d_{51}	d_{52}	d_{53}	d_{54}	d_{55}	d_{5n}
R_m	d_{m1}	d_{m2}	d_{m1}	d_{m4}	d_{m5}	d_{mn}

Fig. 3. Database structure

We denote rows in Fig. 3 as

- $R_i = (d_{i1} \| d_{i2} \| ... \| d_{im}),\ i = 1,2,...,m$

We denote columns in Fig.3 as

- $C_j = (d_{1j} \| d_{2j} \| ... \| d_{nj}),\ j = 1,2,...,n$

B. Type – 1. Rows-level encryption with AES

We create to encrypt with AES encryption for each row with each key in the database. E is the AES encryption function and D is the AES decryption function. Now we use AES encryption with 256 bits random key.

Encryption with AES:

- $e_i = E(R_i, k_i)\ ,\quad i = 1,2,...,m$

Decryption with AES

- $R_i = D(e_i, k_i)\ ,\quad i = 1,2,...,m$

C. Type – 2. Columns- level encryption with AES

We create to encrypt with AES encryption for each column with each key in the database. E is the AES encryption function and D is the AES decryption function. Now we use AES encryption with 256 bits random key.

Encryption with AES:

- $e_j = E(C_j, k_j)\ ,\quad j = 1,2,...,n$

Decryption with AES:

- $C_j = D(e_j, k_j)\ ,\quad j = 1,2,...,n$

D. Type – 3. Elements level encryption with AES

We create to encrypt with AES encryption for each element with each key in the database. E is the AES encryption function and D is the AES decryption function. Now we use AES encryption with 256 bits random key.

Encryption with AES

- $e_{ij} = E(d_{ij}, k_{ij})\ , i = 1,2,...,m\ , j = 1,2,...,n$

Decryption with AES

- $d_{ij} = D(e_{ij}, k_{ij})\ , i = 1,2,...,m\ , j = 1,2,...,n$

Type – 4. Rows-level encryption with AES and ECC

We create to encrypt with AES encryption algorithm for each row, and then encrypt AES 256 bits random key by using ECC [10,11] encryption algorithm. We use ECC asymmetric key algorithm for each row. Elliptic curves are well-understood: they offer smaller key sizes [12] and more efficient implementations [13] at the same security level as other widely deployed schemes such as RSA [14]. We denote that E is the AES encryption function, D is the AES decryption function, $\mathcal{E}$ is the ECC encryption function, $\mathcal{D}$ is the ECC decryption function, PK_i is the public key of ECC algorithm, and SK_i is the private key of ECC algorithm. Now we use AES encryption with 256 bits random key.

Encryption with AES:

- $e_i = E(R_i, k_i)\ ,\quad i = 1,2,...,m$

Encrypt this AES random key by using public key PK_i of ECC encryption algorithm for each row:

- $\varepsilon_i = \mathcal{E}\ (k_i, PK_i)\quad , i = 1,2,...,m$

In this statement, we obtain the encrypt ε_i for each row. When we want to get back AES random key, we need to decrypt AES random key by using the secret key SK_i of ECC decryption algorithm for each row:

- $k_i = \mathcal{D}(\varepsilon_i, SK_i)$, $i = 1,2,...,m$

In this statement, we obtain AES random key k_i. When we want to get back each row R_i, we need to decrypt each column by using AES random key k_i:

- $R_i = D(e_i, k_i)$, $i = 1,2,...,m$

Elliptic Curve Digital Signature Algorithm

We need to trust each other for data integrity which is only be changed by authorized people or processes and prevent altered data from unauthorized people. The Elliptic Curve Digital Signature Algorithm (ECDSA) was standardized in FIPS 1864 [15]. We denote that S is the signing by ECC signature function, and V is the verification by ECC signature function.

- Signing: $\varepsilon_i = S(R_i, SK_i)$, $i = 1,2,...,m$
- Verify a signature: $R_i = V(\varepsilon_i, PK_i), i = 1,2,...,m$

E. Type – 5. Column level encryption with AES and ECC

We create to encrypt with AES encryption algorithm for each column, and then encrypt AES 256 bits random key by using ECC [10,11] encryption algorithm. We use ECC asymmetric key algorithm for each column. We denote that E is the AES encryption function, D is the AES decryption function, $\mathcal{E}$ is the ECC encryption function, $\mathcal{D}$ is the ECC decryption function, PK_i is the public key of ECC algorithm, and SK_i is the private key of ECC algorithm. Now we use AES encryption with 256 bits random key.

Encryption with AES:

- $e_j = E(C_j, k_j)$, $j = 1,2,...,n$

Encrypt this AES random key by using public key PK_i of ECC encryption algorithm for each column:

- $\varepsilon_j = \mathcal{E}(k_j, PK_j), j = 1,2,...,n$

In this statement, we obtain the encrypt ε_j for each column. When we want to get back AES random key, we need to decrypt AES random key by using the secret key SK_j of ECC decryption algorithm for each column:

- $k_j = \mathcal{D}(\varepsilon_j, SK_j)$, $j = 1,2,...,n$

In this statement, we obtain AES random key k_j. When we want to get back each column C_j, we need to decrypt each column by using AES random key k_j:

- $C_j = D(e_j, k_j)$, $j = 1,2,...,n$

Elliptic Curve Digital Signature Algorithm

We need to trust each other for data integrity which is only be changed by authorized people or processes and prevent altered data from unauthorized people. We denote that S is the singing of ECC signature function, and V is the verification of ECC signature function.

- Signing: $\varepsilon_j = S(C_j, SK_j), j = 1,2,...,n$
- Verify a signature: $C_j = V(\varepsilon_j, PK_j), j = 1,2,...,n$

F. Type – 6. Elements level encryption with AES and ECC

We create to encrypt with AES encryption algorithm for each element, and then encrypt AES 256 bits random key by using ECC [10,11] encryption algorithm. We use ECC asymmetric key algorithm for each element. We denote that E is the AES encryption function, D is the AES decryption function, $\mathcal{E}$ is the ECC encryption function, $\mathcal{D}$ is the ECC decryption function, PK_{ij} is the public key of ECC algorithm, and SK_{ij} is the private key of ECC algorithm. Now we use AES encryption with 256 bits random key.

Encryption with AES:

- $e_{ij} = E(d_{ij}, k_{ij})$, $i = 1,2,...,m$, $j = 1,2,...,n$

Encrypt this AES random key by using public key PK_{ij} of ECC encryption algorithm for individual elements are encrypted separately. We use 521 bits of ECC encryption algorithms.

- $\varepsilon_{ij} = \mathcal{E}(k_{ij}, PK_{ij}), i = 1,2,...,m$, $j = 1,2,...,n$

In this statement, we obtain the encrypt ε_{ij} for each element. When we want to get back AES random key, we need to decrypt AES random key by using the secret key SK_{ij} of ECC decryption algorithm for each element.

- $k_{ij} = \mathcal{D}(\varepsilon_{ij}, SK_{ij})$, $i = 1,2,...,m$, $j = 1,2,...,n$

In this statement, we obtain AES random key k_{ij}. When we want to get back each element d_{ij}, we need to decrypt each element by using AES random key k_{ij}:

- $d_{ij} = D(e_{ij}, k_{ij})$, $i = 1, 2, ..., m$, $j = 1, 2, ..., n$

Signing with Elliptic Curve Digital Signature Algorithm

One type of security attack is to intercept some important data and make changes to it before sending it on to the intended receiver. We need to trust each other for data integrity which is only be changed by authorized people or processes and prevent altered data from unauthorized people. We denote that S is the singing by ECC signature function, and V is the verification by ECC signature function.

- Signing: $\varepsilon_{ij} = S(d_{ij}, SK_{ij})$, $i = 1, 2, ..., m$, $j = 1, 2, ..., n$
- Verify a signature: $d_{ij} = V(\varepsilon_{ij}, PK_{ij})$, $i = 1, 2, ..., m$, $j = 1, 2, ..., n$

TABLE I. ADVANTAGES AND DISADVANTAGES OF DATABASE ENCRYPTION AND SIGNATURE

No	Advantages	Disadvantages
Type-1	Implement restrictions on data row access. Access to row-level data in a table.	When an attacker gets one key for a row, we will be losing the hole of the row within the database. Do not support data integrity.
Type-2	Individual columns within a database to be encrypted. Column -level encryption to be significantly more flexible when compare to entire database encryption system.	When an attacker gets one key for a column, we will be losing the hole of the column within the database. Don't support data integrity.
Type-3	Individual elements within a database to be encrypted. When an attacker gets one key which provides for one element and he will be getting only one element. Better audit trails.	Database performance to decrease. Don't support data integrity. Thousands or millions of keys to manage.
Type-4	Implement restrictions on data row access. Security system more reliable and robust. Support data integrity.	Database performance to decrease. Orders of size of keys to manage. Encryption time longer than only AES encryption.
Type-5	Individual columns within a database to be encrypted. More effective for access control. Support data integrity.	Database performance to decrease. Orders of size of keys to manage. Encryption time longer than only AES encryption.
Type-6	More effective for access control system. Support data integrity. In this method, even if an attacker gets one key for an element, he will be gets only one element. Better audit trails. Most secure than all of the above methods.	Thousands or millions of keys to manage. Database performance to decrease.

G. Post Quantum Cryptography

On the future, our database security system will be used Post-Quantum Cryptography (PQC) which is a relatively new cryptologic trend [16, 17] that acquired a NIST status [18] and which aims to be resistant to quantum computers attacks like Shor algorithm [19]. Most quantum algorithms are based on the standard quantum circuit model [20] and are designed to solve problems which are essentially number theoretic such as the Shor's algorithm [21] (see [22] for a general review on the basics of quantum algorithms). Five popular families known to build post-quantum asymmetric cryptography are Hash-based, Multivariate Quadratic-based, Code-based, Lattice-based, and Isogeny-based.

V. CONCLUSION

ECC encryption system is an asymmetric key encryption algorithm for public-key cryptography. It simply generates a public and private key and allows two parties to communicate securely. There is one major advantage however that ECC offers over RSA. A 256 bits key in ECC offers about the same security as 3072 bits key using RSA. In this paper, we create a more secure way to store and retrieve database information that is both convenient and efficient. We use 256 bits of AES Encryption for rows level encryption, column level encryption, and elements level encryption for each element with each key in the database. And then again encrypt AES 256 bits random key by using 521 bits ECC encryption for rows level encryption and column-level encryption. From Table 1 we obtain that the most secure method is type - 6 of element level encryption with AES and ECC encryption for confidentiality. Confidentiality is the ability to hide information from those people unauthorized to view it. Encryption methods are an example of an attempt to ensure confidentiality of data transferred from one computer to another. ECC encryption will be read by only the right people (confidentiality). The ability to ensure that data is an accurate and unchanged representation of the original secure information. One type of security attack is to intercept some important data and make changes to it before sending it on to the intended receiver. ECC signature will be changed by authorized people or processes (integrity). We need to use more to secure for data integrity which is only be changed by authorized people or processes and prevent altered data from unauthorized people. We use ECC signature for every element within the whole database for integrity.

REFERENCES

[1] Mrs. Yasmeen, "NOSQL Database Engines for Big Data Management", International Journal of Trend in Scientific Research and Development (IJTSRD) International Open Access Jounal. Vol. 2, Issue 6, Sep – Oct 2018.

[2] E. F. Codd, "A Relational Model of Data for Large Shared Data Banks", Communications of the ACM 13:377-387, January 1970. "Real Application Clusters", http://www.orafaq.com/wiki/Real_ Application _Clusters , 16 November 2017.

[3] M. Rouse, “RDBMS (relational database management system)”, https://searchdatamanagement.techtarget.com/definition/RDBMS-relational-database-management-system, April 2018.

[4] “MySQL Cluster”, https://blogs.oracle.com/mysql/mysql-cluster-76-is-now-generally-available, MySQL. Retrieved on 2013-09-18.

[5] D. Pritchett, “BASE: An Acid Alternative. ACM Queue”, 6(3): 48-55, 2008.

[6] J.D.Cook, “ACID versus BASE for database transactions”, http://www.johndcook.com/blog/ 2009/07/06/brewer-cap-theorem-base/, 2009.

[7] A. R. Pathak, B. Padmavathi, “Survey of Confidentiality and Integrity in Outsourced Databases”, International Journal of Scientific Engineering and Technology, ISSN: 2277-1581. Vol. No.2, Issue No.3, pp. 122-128.

[8] A. M. Abdullah, ”Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data”, Cryptography and Network Security, June 16, 2017.

[9] S. J. Aboud, M. A. AL-Fayoumi, M. Al-Fayoumi, and H.S. Jabbar, " An Efficient RSA Public Key Encryption Scheme ", IEEE, DOI 10.1109/ITNG.2008.199, 2008.

[10] N. Koblitz, “Elliptic curve cryptosystems”. Mathematics of Computation, 48(177), pp. 203-209, 1987.

[11] V. S. Miller, "Use of elliptic curves in cryptography". In H. C. Williams, editor, CRYPTO, volume 218 of LNCS, pages 417-426. Springer, 1986.

[12] A. K. Lenstra and E. R. Verheul, "Selecting cryptographic key sizes". Journal of Cryptology, 14(4), pp. 255-293, 2001.

[13] D. J. Bernstein and T. Lange (editors), "eBACS: ECRYPT Benchmarking of Cryptographic Systems". http://bench.cr.yp.to, October 2013.

[14] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems". Communications of the ACM, 21, pp. 120-126, 1978.

[15] C.F. Kerry and P. D. Gallagher, "Digital Signature Standard (DSS)". FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION -186-4, pp:26-30, 2013. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf.

[16] D. Bernstein, J. Buchmann, and E. Dahmen, "PostQuantum Cryptography", Springer Verlag, pp. 1-14, 2009.

[17] Ç. Kaya, and Koç, "Open Problems in Mathematics and Computational Science", Springer Verlag, pp. 1-4, 2014.

[18] Matthew Scholl, "Information Security and Privacy Advisory Board", http://csrc.nist.gov/groups/SMA/ispab/ (consulted Feb 25, 2019).

[19] P. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”, SIAM J. Comput., no. 5, pp. 1484-1509, 1997.

[20] M. A. Nielsen, and I. L.Chuang, " Quantum Computation and Quantum Information ", Cambridge University Press, 10th Anniversary edition, pp. 202-204, 2010.

[21] P.W. Shor, " Algorithms for quantum computation: discrete logarithms and factoring ", Proceedings 35th Annual Symposium on Foundations of Computer Science, IEEE, DOI: 10.1109/SFCS.1994.365700, 1994.

[22] R.Cleve, A. Ekert, L. Henderson, C. Macchiavello, and M. Mosca, " On Quantum Algorithms ", Complexity 4 (1998) 33, https://www.cs.auckland.ac.nz/~cristian/leahcomplexity.pdf, 1998.