Format Preserving Encryption Technique to Strengthen Data Warehouse Security

[1] Shikha Gupta, [2] Priyanka Bhutani, [3] Ridhi Nim

[1, 2, 3] Department of Information Technology ,University School of Communication and Technology,
Guru Gobind Singh Indraprastha University, Sector 16 C, New Delhi-110078, India

Abstract- Data Warehouses are the most vital asset of an organization which contains sensitive information and data of an organization which is extracted from heterogeneous sources which is used in decision making process. A data warehouse by its nature, creates a security conflict on the other hand, the goal of every data warehouse is to make data accessible. Unfortunately, this also makes them an appearing target for malicious inside and outside attackers. Therefore, security is an important concern which should be defined in order to protect this sensitive information from unauthorized users to ensure integrity and confidentiality. Security should be considered from early stages of designing the data warehouse and hence should be deployed. Encryption is one security technique applied on data to cut out any unauthorized access for maintaining its integrity and delivery performance. In order, to meet these rigorous security challenges posed by protecting different types of information, a new encryption technique is defined known as Format Preserving Encryption mechanism wherein, the encrypted data fits into the existing schema and hence, changes to the database schema and underlying applications would not be required. In this paper, we introduce Format Preserving Encryption (FPE) technique with Advanced Encryption Standard (AES) to encrypt the data before keeping in data warehouse for improving the implementation of security of data warehouse.

Keywords - *Data security, Data warehousing, Format preserving encryption, Advanced Encryption standard (AES).*

1. Introduction

Data warehouse contains the sensitive information and data of an organization which is extracted from heterogeneous sources. Data Warehouses transform data into business knowledge, providing information for adding business value. Therefore, Data warehouses are the heart of enterprise sensitive data. Such sensitive information and data are very important for an enterprise, which is used in decision making process. But, unfortunately this makes them a key target for attackers [1]. We store and transmit such sensitive information through the internet. The internet enables delivery of information worldwide, and the data warehouse provides easy access to organizational data. Therefore, security is an important concern which should be defined in order to protect this sensitive information from unauthorized users. Various security approaches are being used for securing the data at different levels and layers because the data stored in warehouse is extracted from various operational systems.

In a data warehouse, the database is the application. Users interact more directly with the tables and views that a DWH houses through numerous applications, issuing a wide variety of queries which may or may not be known in advance. The techniques that would be applied to secure the database behind a traditional application could severely constrain the users of the data warehouse and undermine its success. Still, security is essential. Without appropriate security, the data warehouse, as the centralized hub of information on the status of the business, becomes just as much a liability as it is an asset[2].

Conceptually, the data warehouse process consists of three steps: Extraction, transformation and loading. Data is at risk during each of these stages. Several factors that render the data warehouse are more vulnerable to attack.

1) Extracted data is rapidly transmitted over insecure communication channels.
2) Extracted data is hoarded on multiple computer systems which may have limited security.
3) The extraction process produces large number of intermediate files and load files having sensitive information, but mayn't be well protected.
4) Retrieving data from data warehouses for creating data marts and which may leads to widely distributed copies of sensitive data.

Encryption is applied on data which can be employed at two places i.e. data at rest and data in transit before entering into warehouse [2]. Enciphering data in transit means encryption is done at the sender side and decryption is done at the receiver side. Enciphering data at rest is used to protect confidential data stored on host from the privileged users. There are various database security policies like authentication, access rights, digital signature etc. Encryption algorithm enciphers the plaintext with the help of key which may varies in size such as 128 bit, 196 bit, 256 bit and converts it into ciphertext.The existing security techniques are not enough for protecting the data. Hence, we need powerful encryption algorithm to achieve powerful security mechanism. Database encryption is the process of converting the plaintext in the database to meaningless cipher text format. Database encryption is implemented using strong encryption such as AES, RSA or SHA256. The major difficulty in the adoption of effective encryption methods is the cost of modifying databases and applications to accommodate encrypted information. In order to overcome these problems a new encryption technique is defined in this paper named Format preserving encryption.

Format preserving encryption means encrypting the data without changing the format and data type. The format and type of cipher text is equivalent to plaintext. This feature makes it have some advantages over traditional block ciphers in some applications like credit card encryption. This technique mainly focuses on enhancing the security of data warehouse against unauthorized access.

The remainder of this paper is organized as follows. In Section 2, we present the existing format preserving encryption methods and discuss the specific issues and requirements for their use in data security environments. In section 3, problem statement is defined. In Section 4, we present the proposed technique for data security to strengthen data warehouse security, followed by section 5, conclusion.

2. Related Work

In the last few years, format-preserving encryption (FPE) has emerged as a useful tool in applied cryptography. The goal of FPE is stated as, under the control of a symmetric key K, deterministically encrypt a plaintext X into a cipher text Y that has the same format as X. For example, encryption of social security numbers (SSNs), credit card numbers (CCNs) of a given length, postal addresses of some particular country[3]. By preserving data formats, sizes and referential integrity, FPE provides an efficient method for "sanitizing" data without need for massive masking or lookup tables and also by maintaining the format of data being encrypted, database schema changes are zero and application changes are minimized. This enables us to secure the data with minimum effort and cost [3].

2.1 Basic Data Type Preservation

The first depiction of FPE goes back to1981. That is when the original US government specification for the Data Encryption Standard (DES) encryption algorithm included a description of how to use DES in a way that preserved the format of data on a character-by-character basis mapping a decimal digit to another decimal digit, for example[3]. Cryptographers John Black and Phil Rogaway in (2002), published a careful description of three approaches to FPE and proved that they were secure. The most important approach described by Black and Rogaway used a feistel network to get the desired properties of both preserving the format of the data as well as showing that the approach was secure. [4]

1) *FPE from a prefix cipher:* In prefix cipher FPE algorithm each integer in a plaintext is assigned by pseudo random weights. The weights are defined by applying an existing block cipher to each integer.
To encrypt data with the Prefix method, we first construct a table which stores a permutation over the full plaintext set, and then simply look up the cipher text value using the plaintext. This means that encryptions and decryptions are very fast, table set up is more expensive and acceptable for small set sizes.

2) *FPE from cycle walking:* The cycle walking construction works by encrypting the plain text with an existing block cipher repeatedly until the cipher becomes in acceptable range. If we have a plaintext X, we can create an FPE algorithms from the block cipher by repeatedly applying the AES or 3DES until the result is satisfying required FPE range as shown in Fig.1below.

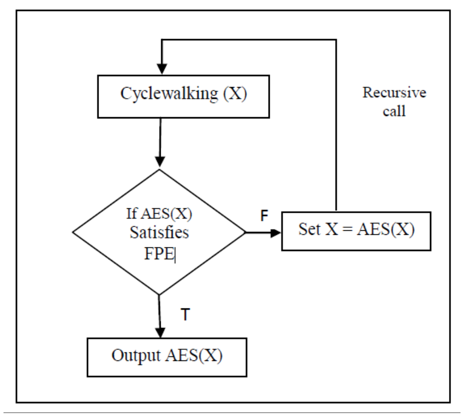


Fig 1. Cycle Walking Method[7]

```
Cycle walking FPE(x)
{
if AES(x) is an element of M
return AES(x)
else
return Cycle Walking FPE(AES(x))
}
```

Cycle walking performance depends on the size of the cipher text and the size of the required FPE output. The larger the difference is between them, it needs too much iteration [3].

3) *FPE from feistel network:* Feistel networks were created by cryptographer Horst Feistel in the 1970s. A feistel network divides its input into two parts which are shuffled and combined with a sequence of keys. It is possible to make a FPE algorithm using feistel

network. A feistel network needs a source of pseudo-random values for the sub keys for each round, and the output of AES algorithm can be used as these pseudo-random values.

In feistel network, we can define an input string being divided into a left part that we write as Ln and a right part that we write as Rn. These two parts are then used to create an output that also has a left part that we write as Ln+1 and a right part that we write as Rn+1.The variable n is used to suggest that this process happens several times, and that is exactly a Feistel network does[12].

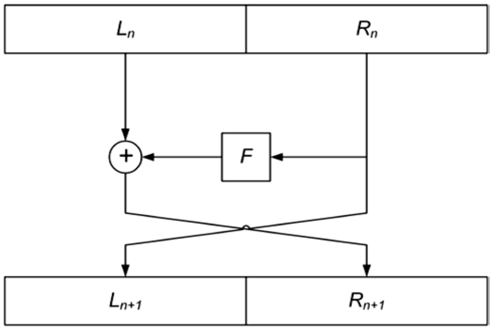


Fig 2. A Single Round of Feistel Network[6]

In 2009, RtE (Rank-then-encipher) is a general FPE mode proposed by Bellare [7]. He demonstrates how to convert FPE problem on a complex domain into that in an integer set through rank and unrank algorithm. RtE can solve FPE problem on arbitrary regular language. All strings of various encodings can be expressed as regular expression, so RtE can be used to encrypt character data.

In 2010, Bellare [8] proposed FFX mode which is combined with tweak feature . FFX can solve FPE problem on by establishing a bijective mapping between and an integer set that converts FPE on into integer FPE. Both RtE and FFX use transformation to simplify FPE on an original complex domain into FPE on an integer set.

3. Problem Statement

Encryption is not that prevalent in data warehouse because of the complicated encryption and decryption algorithms. Encryption and decryption processes also degrade performance considerably. However, with the frequent use of the internet as an access and delivery mechanism, encryption should be seriously considered to protect the organization from costly security breaches. When we decided to develop a cryptographic solution to data warehouse security which could be applied in the complex heterogeneous environment tracked in the business world, then we recognized certain objectives like, it must encipher and decipher data on machines with different data types or character sets. Encryption should occur as early as possible in the extraction process and decryption should occur at the end point. It must function on various hardware platforms and operating systems. These requirements based on our organization necessities represent a difficult challenge.

During encryption and decryption there is a need for changing the database to store the enciphered data. The main disadvantage in traditional encryption method is the cost of modifying the existing databases and applications to process encrypted information.

4. Proposed Solution

In this paper we have defined a way to meet the rigorous security challenges posed by protecting diverse types of information, by using a new data security technique known as format preserving encryption or data type preservation. Format preservation provides several distinct benefits that build on solid strong-encryption practices. The main aim of FPE is to encipher the data without the need to modify all of the systems that use that data; such as database field, queries and all the application program .

4.1. FPE as A Mode of AES Algorithm

Traditional encryption methods have huge impact on data structures, schemas and applications. Format-preserving encryption, a mode of AES is used to overcome this challenge by enciphering data, while preserving its original format and without sacrificing encryption strengths [1]. Structured data, such as social security numbers, Tax ID, Credit card, account, can be encrypted in position. Traditional encryption methods change the original format of data[13]. For example, a 9 digit social security number encrypted with AES produces a long alphanumeric string. As a result, database schema changes are required to facilitate the original format. The Format preserving encryption technique defined in this paper, results in the preservation of data type of the plaintext. Fig.3 shows the enhanced encryption technique by using a combination of AES and FPE.

Format preserving encryption (FPE) refers to a set of techniques for encrypting data such that the cipher text has the same format as the plaintext. A format-preserving encryption scheme is applicable for many real-life applications. FPE is a high-quality encryption scheme that allows encryption with minimal modifications to the original plaintext.

The different names for FPE techniques are Data type Preserving Encryption (DPE) and Feistel Finite Set Encryption Mode (FFSEM). The main aim of all the techniques is to get back the same size, and data type as the original plain text is being encrypted as well as transmitting sensitive data securely over the multi-system environments with minimum changes[11].

4.2. Enhanced Encryption Technique (AES + FPE)

Format-Preserving Encryption (FPE) is a new approach for enciphering structured data, such as credit card or Social Security numbers. FPE makes it possible to integrate data-level encryption into legacy business application frameworks that were previously difficult or impossible to address. It uses a published encryption method with an existing, proven algorithm to encrypt data in a way that does not alter the data format. The result is a strong encryption scheme that allows for encryption with minimal modifications to the way that existing applications work.

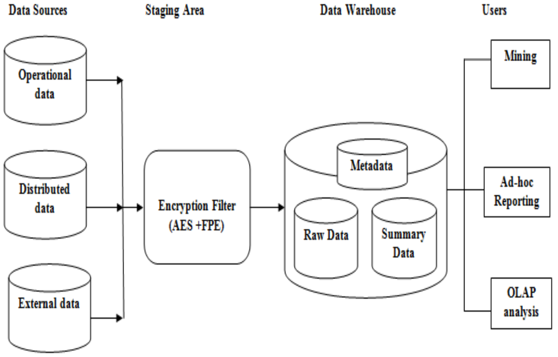


Fig 3. Proposed Encryption technique (AES +FPE) to strengthen data warehouse security

4.2.1 Encryption

Cryptography defines encryption as a process of enciphering messages or information in such a way that only authorized parties can interpret it. In an encryption scheme, the message or information, referred to as, plaintext is encrypted using an encryption algorithm, turning it into an unreadable cipher text.

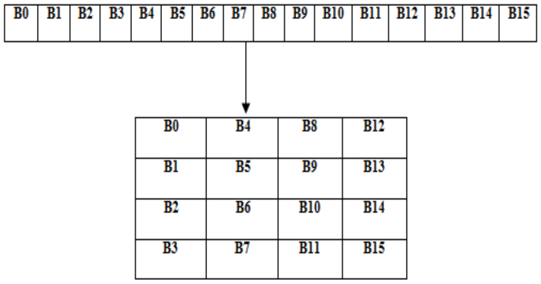
An encryption key is used, which specifies how the message is to be encoded. Any intruder that can see the ciphertext should not be able to determine anything about the original message. An authorized party, however, is able to decipher the ciphertext using a decryption algorithm that usually requires a secret decryption key, so that intruder does not have access to it.

DES, Triple DES, AES, and Feistel network are the examples of modern block ciphers. The computational functions of these ciphers are much complex and cannot be simply busted. The drawback in modern ciphers is the length of the cipher text. So, if we wanted to encipher something at one end point, and decipher it at the other end point, it's actually not that easy because if we encrypt it using AES, the output of the encrypted number or data would be 128 bit block. Sixteen bytes that would need to be sent from one system to the next, until it reaches its destination. In order to store these 16 bytes, 128 bits are required. The cost of changing the database structure is very expensive and the queries linked to database will also be altered. A graphical interface would not demonstrate it [10].

4.2.3 AES Block Cipher

In this paper the projected technique is based on AES encryption algorithm. AES-128 uses, 128 bit plaintext and 128 bit key to produce 128 bit cipher text. The numbers of possible keys are 2^{128} =3.4x$10^{38.}$ A system that tries 2^{55} keys per second requires 149 billion years to break the cipher text [11]. Structure of Round function in AES algorithm: A round for AES consist of 4 operations:-Sub bytes, Shift rows, Mix column, Add round key:

1) *Sub byte Operation*: Sub byte operation substitutes bytes independently in a black box fashion, using a nonlinear substitution table called as S-Box.



B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15

B0	B4	B8	B12
B1	B5	B9	B13
B2	B6	B10	B14
B3	B7	B11	B15

Fig 4.Sub byte operation

2) *Shift row operation:* Shifting rows is used to permute the bytes. In the encryption, the permutation is called Shift Rows. The first row is never shifted. Each byte of the second row is shifted one to the left. The third and fourth rows are shifted by offsets of two and three respectively. Row n is shifted circular left by n-1 times [11].

$$\begin{bmatrix} 00 & 11 & 22 & 33 \\ 44 & 55 & 66 & 77 \\ 88 & 99 & AA & BB \\ CC & DD & EE & FF \end{bmatrix} \rightarrow \text{ShiftRow} \rightarrow \begin{bmatrix} 00 & 11 & 22 & 33 \\ 55 & 66 & 77 & 44 \\ AA & BB & 88 & 99 \\ FF & CC & DD & EE \end{bmatrix}$$

Fig 5. Shift row operation

3) *Mix column operation:* The mix column function takes four bytes as input and produces four bytes as output. It operates on each column of the state matrix. Each column a = (a_0, a_1, a_2, a_3) is substituted with (b_0, b_1, b_2, b_3) as shown below in Eq.(1)

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Fig 6. Mix column operation

$$\begin{aligned} b_0 &= 2a_0 + 3a_1 + 1a_2 + 1a_3 \\ b_1 &= 1a_0 + 2a_1 + 3a_2 + 1a_3 \\ b_2 &= 1a_0 + 1a_1 + 2a_2 + 3a_3 \\ b_3 &= 3a_0 + 1a_1 + 1a_2 + 2a_3 \end{aligned} \qquad (1)$$

The mix column transformation operates at the column level and transforms each column of the state to a new column.

4) *Add round key:* The Add Round Key operation is a simple XOR operation among the State and the Round Key. The Round Key is derived from the Cipher key by means of the key schedule. The State and Round Key are of the same size and to obtain the next State an XOR operation is done per element as shown in fig 7.

$$\begin{bmatrix} a_0 & a_4 & a_8 & a_{12} \\ a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \end{bmatrix} \oplus \begin{bmatrix} k_0 & k_4 & k_8 & k_{12} \\ k_1 & k_5 & k_9 & k_{13} \\ k_2 & k_6 & k_{10} & k_{14} \\ k_3 & k_7 & k_{11} & k_{15} \end{bmatrix} = \begin{bmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{bmatrix}$$

Fig 7. Expanded Key Matrix Is Combined With the State by a Bitwise XOR

5. Conclusion

In this paper a better encryption technique is defined in order to achieve strong data encryption during data at rest as well as in transit. This implies that we can preserve the format of the data by combining it with a strong encryption algorithm and make it as secure as, an AES algorithm.

An individual technique alone is not secured therefore, for better security we use combination of more than one techniques and also increase the number of permutations at the time of encryption. This combination of advance encryption algorithm (AES) and format preservation will increase the attacker's burden. The basic idea is to use a strong block cipher such as AES.

Using Format preserving encryption the database schema and applications will never changed. The cost and time for modifying the data base is reduced. In addition to data warehouse security, there may be other areas in which this technique may bear out useful, such as by providing an extra check on data integrity. The FPE technique defined in this paper is very useful for real time applications such as for encrypting social security number, credit card, personal health information records. It retains the original data type and format of plaintext after encryption.

References

[1] N. Yuhanna, "Your Enterprise Database Security Strategy 2010", Forrester Research, September 2009.

[2] Paulraj pooniah, "*Data Warehousing Fundamentals-A comprehensive guide for IT professionals*", John Wiley and sons, 2001.

[3] Terence Spies *"Format Preserving Encryption"*, Voltage Security, Inc.

[4] H. E. Smith and M. Brightwell, "*Using Data type-Preserving Encryption to Enhance Data Warehouse Security*", NIST 20th National Information Systems Security Conference, pp.141, 1997.

[5] V. Hoang and P. Rogaway, "*On generalized Feistel networks", Conference version of this paper,* CRYPTO 2010, Springer, 2010.

[6] Kurra Mallaiah, S. Ramachandram, *"Performance analysis of Format preserving encryption over block ciphers for numeric data",* 2013 4th International conference on computer and communication technology.

[7] *M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers, "Format-preserving encryption", SAC 2009.* LNCS 5867, Springer, 2009.

[8] M. Bellare, P. Rogaway, and T. Spies, "*The FFX mode of operation for format-preserving encryption"* (Draft1.1).February, 2010, Manuscript (standards proposal) submitted to NIST.

[9] J. Black and P. Rogaway*, "Ciphers with Arbitrary Finite Domains".* RSA Data Security Conference, Cryptographer's Track (RSA CT '02), Lecture Notes in Computer Science, vol. 2271, pp. 114-130, Springer, 2002.

[10] AES, "*Advanced Encryption Standard*", National Inst. of Standards and Technology (NIST), FIPS-197, 2001.

[11] Jia, Z Liu, J Li, Z Dong *"A new integer FPE scheme based on Feistel Network*", Advances in Electric and Electronic, 2012, Springer.

[12] www.*verifone.com*/sites/verishield-protect.aspx

[13] National Institute of Standards and Technology. NIST Special Publication 800-38A: *Recommendation for Block Cipher Modes of Operation—Methods and Techniques*, December, 2001.