International Journal of Applied Science and Engineering

Designing a cryptosystem for data at rest encryption in mobile payments

Pinki Prakash Vishwakarma[1*], Amiya Kumar Tripathy[2, 3], Srikanth Vemuru[1]

[1] *Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India*
[2] *Department of Computer Engineering, Don Bosco Institute of Technology, Mumbai, India*
[3] *School of Science, Edith Cowan University, Perth, Australia*

ABSTRACT

Since the evolution of m-commerce, security and entrustment of digitized transactions have become of captious concern to financial institutions. Card information hacking has caused money losses around the world, therefore it is imperative for financial institutions to get rid of such losses. Currently, the number of mobile payment schemes have been purposed but primarily the schemes aim attention at transaction security, fraud detection and prevention, not on data at rest encryption in mobile payments. Therefore, this work aims attention to encrypt sensitive static data residing at database server in mobile payments. Data at rest is the static data i.e., card details of the users which resides at the server. It is essential to ensure that the sensitive data of the payment users stay protected so as to prevent the adversaries looking for unauthorized access to the data. The encryption of data at rest is accomplished at the database level in this work. Cryptography is increasingly being used to combat against the security of sensitive data to guarantee data confidentiality and data integrity. In this work a cryptosystem is proposed which describes the management of cryptographic keys of the sensitive data at rest, in a mobile payment system with symmetric cryptographic implementation, the keys involved are identical for both encrypting and decrypting the sensitive data.

Keywords: Data confidentiality, Integrity, Encryption, Key management, Mobile payments.

OPEN ACCESS

Received: March 31, 2020
Revised: August 21, 2020
Accepted: September 26, 2020

Corresponding Author:
Pinki Prakash Vishwakarma
vishwakarmapp@gmail.com

Copyright: The Author(s). This is an open access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), which permits unrestricted distribution provided the original author and source are cited.

Publisher:
Chaoyang University of Technology
ISSN: 1727-2394 (Print)
ISSN: 1727-7841 (Online)

1. INTRODUCTION

Card information hacking has caused money losses around the world. Therefore, it is essential to get rid of such losses to the payment industry or financial institutions. The card holder data consist of card number, also known as primary account number (PAN), card holder name, expiry month and expiry year. It is important to keep the cardholder data storage to a minimal and to ensure the cryptographic keys used for encryption of cardholder data (Kadambi et al., 2009). The three states of digital data are data in use, data at rest and data in transit. Data in use is an active data going through constant changes stored physically in database, data warehouse, etc. Data at rest is an inactive data which is in a form of the repository such as database, archive, data warehouse, mobile device, etc. (Bhatia and Verma, 2017). The data in transit require a secure communication channel which can be accomplished by using a secure communication protocol between the initiator and the target user. Protecting only one type of sensitive data and paying no attention to other types of sensitive data can face disaster. In consequence, it is essential that the organization looks for a comprehensive solution to

deal with all sensitive data.

The card details of the mobile payment users are stored in the database server. The consumers will not use a payment finance service which is insecure. The database security risks are unauthorized access or adversary access to sensitive data in databases (Zaw et al., 2019; Wang et al., 2020). The adversary, attacking the database server should not be able to mishap the protection system. The adversary attacks a payment application, in this application, the adversary is basically looking for the payment credentials (card details) of the users. With the help of payment credentials, the adversary can perform a fraudulent transaction. Data at rest is the static data i.e., card details of the users which resides on the server. The adversary can hack the static data residing on the server. Historical analysis results say that most data vulnerabilities manifest by virtue of unsecured data at rest (Sultan et al., 2016).

In a secure mobile payment system, the transaction flow must have end-to-end encryption. Software-based encryption techniques are most suitable for data at rest in mobile payments. It is essential to protect data at all points, starting from transaction initiation, through the payment network and end-to-end. It is important to ensure that the sensitive data of the payment users stay protected. The proposed approach of mobile payment technology bestows protection to the sensitive card details data of the payment users. However, to bestow protection to the sensitive information of the cardholder, the data at rest should be encrypted.

To defend the sensitive data at rest, encryption and decryption operations are carried out with management of key. Sensitive information in a mobile payment system must be encrypted, whether the data is at rest or in transit. To obviate data breaches and to fortify mobile payment security for data at rest in mobile payments layered security approach is recommended. Tokenization process is one part of a layered security approach. The data at rest, i.e., card data stored on the server is replaced with a secure token so that hackers are not able to access the card data. Therefore, mobile point of sale (mPOS) solutions are Europay, MasterCard, and Visa (EMV) - enabled and follow tokenization process.

The data at rest requires the sensitive data to be stored securely, the payment credentials of the payment users should not reside on the mobile device. The sensitive data of the users is encrypted and stored. It is essential to know how the sensitive data is encrypted and decrypted. The cryptographic keys used for encryption and decryption must be stored securely. It should be relatively difficult for the adversary to know the cryptographic keys and the encryption/decryption process. The algorithms used for implementing encryption and decryption process must elude the adversary, leaving the sensitive data in a secure state. There is an increase in demand that the databases and the file servers which store card details of the payment users should be considered as hazardous (PCI Security Standards Council, 2010).

Mobile payments are more secure than credit/debit cards as it uses tokenization and encryption techniques to façade the primary account number at the time of payment. With mobile payments the prime technologies such as user authentication and device fingerprint authentication methods are most convenient and secure to use. Sensitive data examination and determination is performed to defend from attacks. However, this analysis is carried out first by identifying the critical data in mobile payments and how it is being used. Thenceforth, in which data protection technique can be applied to the sensitive data is decided depending on possible attacks, it is imperative to know how captious is the sensitive data and how to protect it. There are two cryptography methods to encrypt and decrypt data. In symmetric key cryptography to encrypt and decrypt the data same encryption key is used. While in asymmetric key cryptography to encrypt and decrypt the data a pair of keys referred to as public key and private key is used. Symmetric key cryptography is best suited for data at rest encryption.

As per Payment Card Industry Data Security Standard (PCI/DSS), the card details of the user are stored in a database, there is no need to encrypt the entire database. However, there is a need to identify the particular fields required by (PCI/DSS) to be protected and enabling encryption at the column level (Storage Networking Industry Association (SNIA) Storage Security Industry Forum, 2009). Protecting data at rest involves methods such as access control, data encryption and data retention policies (Shabtai et al., 2012; Gugelmann et al., 2015). Access control is the selective limit of access to a place or other resource. The user access rights should be defined per user /group, per database object for protection of sensitive data. Data retention policies/guidelines must be set that describes which data will be archived, how long it will be kept. Periodic auditing of sensitive data should be part of the policy and should occur on schedule occurrences (Shabtai et al., 2012; Gugelmann et al., 2015). The proposed approach imparts leading edge technology, thereby protecting sensitive data.

As compared to traditional payment methods securing mobile payments need a variant science of mind. Software based security technologies like tokenization and encryption/decryption play important role in mobile payments as compared to hardware-based security models which are now obsolete (Gugelmann et al., 2015). The payment credentials of the users are not only associated with the primary account number (PAN) which is used in securing payment transaction, but it also deals with the cryptographic keys used in key management process. The tokenization process replaces a sensitive data, i.e., primary account number of the user with a random number known as token.

Before performing an attack, the adversary needs to understand how the mobile payment application works. By doing this the adversary can get the required information to perform an attack. With the help of sophisticated tools, the adversary can understand the operation of mobile payment

application and learn the location of sensitive data stored (Inside Secure, 2009). The adversary will examine and determine the mobile payment application to fetch cryptographic keys as it is the prime target. If the cryptographic keys and the sensitive data of the payment users are not secured, then it is very easy for the adversary to perform attack. The sensitive data at rest must be stored securely therefore, the storage of sensitive data should not reside directly on the server, it should be stored in encrypted form Vishwakarma et al. (2018). The cryptographic keys used to encrypt the data at rest should also be protected. Tokenization process is not same as encryption process. An encryption process uses a mathematical method to make the data illegible. Therefore, tokenization and encryption techniques used in mobile payments bestow end-to-end data security.

Using encryption techniques to protect sensitive data-at-rest or data in transit is a standard practice. Despite of being data in any state the data confidentiality, data integrity, sensitive information stored must be protected against unauthorized access (Bhatia et al., 2017; Moulds, 2007; Setiadi et al., 2019). The basic security goals of confidentiality and integrity are imperative for mobile commerce as they are for electronic commerce (Turban et al., 2015). The defiance associated with encryption mechanism is all about protection of encryption keys. Nosrati and Bidgoli compared various encryption algorithms with respect to mobile banking. A method for mobile banking security, considering authentication and authorization as security layers was discussed. The emphasis was on securing transactions rather than the data which resides on the database server (Nosrati and Bidgoli, 2016). A cryptographic algorithm for the mobile phones rest on call back technique and one-time password focused on transaction security process and not either on the consumer personal information (Javidan and Pirbonyeh, 2010).

The Apache Spark framework makes use of resilient distributed datasets (RDDs) which is not encrypted. Therefore, Shah et al. have used a combination of cryptographic splitting and encryption for securing data-at-rest, which is specific to the Apache Spark framework (Shah et al., 2016). In defiance of the encryption process revealed, considering that the adversaries do not know all the encryption keys, the encrypted text is secure, as it is impervious to the hackers to hack the ciphertext without obtaining all encryption keys (Huang et al., 2012).

Encryption plays an important role in many prospects; moreover, organizations need to harbor data, whether it's on servers, end user devices or storage devices. On a regular basis, the encryption method used should be examined so that their compatibility and competency can be used whenever necessary. A multifactor authentication technique emanates as a prevailing protection scheme to defend digital assets and financial transactions in the internet. (Wang et al., 2020; Vishwakarma et al., 2018; Tabrizchi and Rafsanjani, 2020). The authentication of a mobile payment transaction is an amalgamation of user password and device fingerprints (Vishwakarma et al., 2018).

In credit card environment to encrypt credit card details, encryption within the database is the most widely method used (Stilgherrian, 2015). Nxumalo et al. proposed the tokenization technology to store sensitive data (Nxumalo et al., 2014). Tokenization alone is not competent (Vishwakarma et al., 2016) as the card information of the payment users are stored in the database server. As sensitive data is also stored on smartphones, to mitigate this problem data encryption for smartphones is proposed by Muslukhov et al. (2016). If the adversary attacks or gain access to the database server, the card number and user details can be compromised. Therefore, it is imperative to protect the sensitive data of the payment users. The objective of the proposed approach is to facilitate encryption of sensitive data while nurturing the data confidentiality and integrity consent level.

2. MATERIAL AND METHODS

Data at rest encryption could be one of the solutions for guarding the data. Encryption and management of keys go hand in hand, is a minimum level security ensuring protection for card details of the consumers. The mobile payment system user's card information data is stored on the database server. Therefore, the card information is encrypted with the data at rest encryption process along with the key management server where the respective keys to the encryption/decryption process are located. Data encryption keys must be updated on a regular basis (e.g., half yearly or quarterly). Fig. 1 shows the data at rest encryption process. The card details of the consumers are stored on the database server; the key management server takes care of the key generation and its management. Hence, with the help of key management sever the card details of the user can be encrypted and stored on the database server.

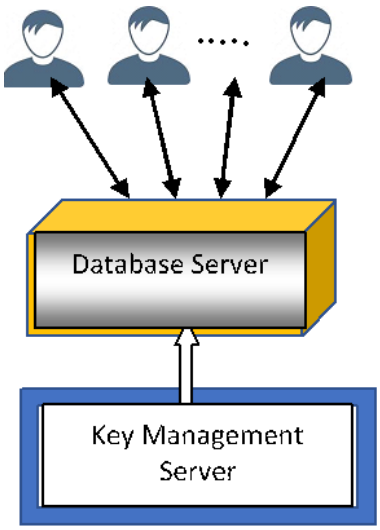

Fig. 1. Data at rest encryption

The terminologies used in the key management process are as follows:

- *Sensitive Data:* Any data which need to be secure is the sensitive data, i.e., card number, account number, etc.
- *Key:* A static key is used for encryption/decryption of the sensitive data. A key is a piece of information that arbitrates the functional output of a cryptographic algorithm. In encryption algorithm, a key specifies the transmutation of plain text into cipher text, and vice versa for decryption algorithms. A key is usually easier to protect than an encryption algorithm, also easier to break if compromised. Therefore, the security of the encryption system reckons on the key being kept secret. There are two types of keys used Data Encryption Key (DEK) and Key Encryption Key (KEK).
- *Salt:* A salt is a random data used as an additional input to a hash function that hashes a passphrase or password. It is similar to the nonce. The primary function of a salt is to defend against dictionary attacks versus a list of password hashes and against pre-computed rainbow table attacks. For each password, a new salt is andomly generated.
- *Encryption algorithm:* Keys are generated to be used with a given suite of algorithms, called a cryptosystem. A symmetric key algorithm is used; identical key is used for both encryption and decryption.
- *Passphrase:* A password can also be used as a key. The practical difference between keys is that the latter is intended to be generated, read, remembered and reproduced by a human user.
- *Custodian:* A custodian is a person who has the responsibility of taking care or protecting the key. In the proposed approach we have two key custodians; the passphrase selected by them is used by the system.

2.1 Proposed Methodology

From consumer perspective as well as development perspective, it is important to assure that the card detail data is safe whereas the security professionals also consider encrypting data at rest is valuable. The selection of key management methods and strong encryption algorithms is discerning for consummation of any encryption scheme. The proposed methodology describes the cryptanalysis of managing the cryptographic keys of the sensitive data at rest

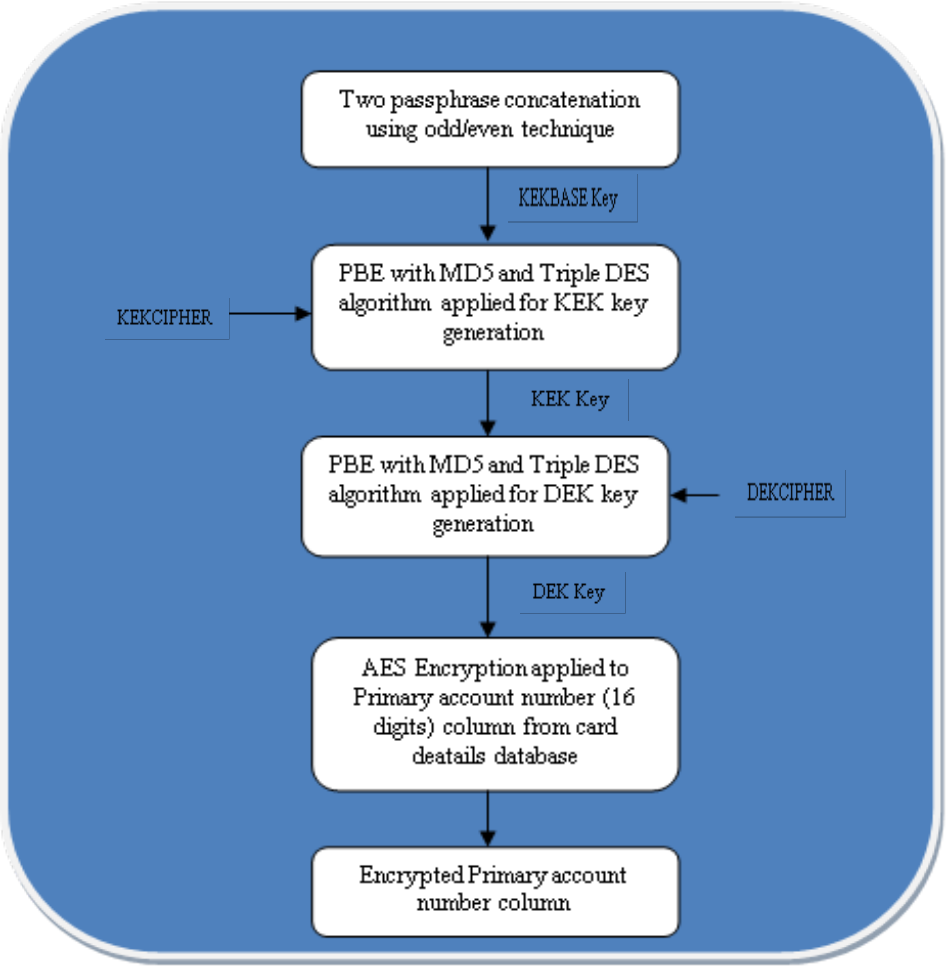

Fig. 2. Flow of Encryption process

in mobile payment system. It includes dealing with the generation, storage and use of keys and further describe how the keys are chosen, distributed and stored securely.

The sensitive data is the primary account number, i.e., card number of the user is encrypted and stored in the card user database with the help of DEK key. To generate the DEK key there is a process of cryptographic keys followed, which is depicted in the Fig. 2. Initially the two passphrases rovided by two custodians are concatenated with the help of odd/even technique resulting in formation of KEKBASE key.

The password based encryption (PBE) is a symmetric cryptography, which performs the encryption process with the help of password-like key. PBE is using message digest 5 (MD5) a hash algorithm and triple Data Encryption Standard (3DES) a symmetric encryption algorithm to generate KEK and DEK key. Now the generation of KEK key is performed with the help of KEKBASE key and KEKCIPHER to which PBE with MD5 and triple DES algorithm is applied. KEKCIPHER is formed by concatenating two KEKSALTS stored in the database. Further DEK key is generated with the help of KEK key and DEKCIPHER, to which again PBE with MD5 and triple DES algorithm is applied. DEKCIPHER is formed by concatenating two DEKSALTS stored in the database.

The advanced encryption standard (AES) algorithm is a symmetric block cipher cryptosystem which encrypts the primary account number of the payment user with the help of DEK key. Finally, the card number of the payment user is stored in the database in encrypted form.

The step break down of the encryption/decryption process of a cryptosystem is shown in the Fig. 3. The KEKBASE key is generated with two half passphrase provided by custodian-1 and custodian-2 which is concatenated by using the odd/even formula. Further for generation of key encryption key (KEK), KEKBASE key and KEKCIPHER are the inputs required. Then for generation of the data encryption key the inputs required are the KEK and DEKCIPHER. Once, after DEK is obtained, the card data can be encrypted/decrypted with the help of symmetric key algorithm. The symmetric key algorithm uses the same key (DEK) for encryption and decryption of the sensitive data. In pursuance of securing the system, the keys and salts are stored in different schemas. The proposed approach has a powerful cryptosystem to deal with the current security requirement in payment industry.

In contemplation of achieving paramount security all the keys, passphrases can be stored in different database schema. KEKsalts and DEKsalts are for storing the salts for the appropriate key and finally the custodian-1 and custodian-2 for storing the passphrases. The databases used for encryption and decryption process in this each data base contain one table. The DEKsalt database has a table which stores salt used for DEK and KEKsalt database has a table which stores salt for KEK. The custodian-1 database has a table, which contains half password set by first custodian, whereas custodian-2 database has table, which contains half password set by second custodian.

2.1.1 Generating KEKBASE Key

A KEKBASE key is one of the input parameters required for generation of Key Encryption key (KEK). The two custodians are used to increase the security level in the cryptosystem. The custodian 1 will store half passphrase 1 in database and custodian 2 will store half passphrase 2 in another database. Now KEKBASE key is generated by concatenation of both half passphrase 1 and half passphrase 2 by using Odd/Even formula. Concatenation of half passphrase 1 and half passphrase 2 is considered as a passphrase.

- Assume half passphrase 1 is “Hard to guess”
- Assume half passphrase 2 is “and long enough”
- Half passphrase 1 ‖ Half passphrase 2

Now the passphrase is “hard to guess and long enough” is also the plaintext applied as input to the odd-even technique.

Now to the passphrase odd-even technique is applied, which is as follows:

Step 1: Give number to each word in the plaintext in sequence.
hard: h = 1, a = 2, r = 3, d = 4
to: t = 5, o = 6
guess: g = 7, u = 8, e = 9, s = 10, s = 11
and: a = 12, n = 13, d = 14
long: l = 15, o = 16, n = 17, g = 18
enough: e = 19, n = 20, o = 21, u = 22, g = 23, h = 24

Step 2: Now begin with the numbered words, first write the characters from the top of the table in left to right sequence in first row then second row continue writing from left to right and so on.

h	a	r	d	t	o
g	u	e	s	s	a
n	d	l	o	n	g
e	n	o	u	g	h

Step 3: Start from the first column, write down the words starting from the first row of the first column downwards, then, repeat the same for second column first row until the last column reaches. So, you will get six words as there are six columns.
hgne/audn/relo/dsou/tsng/oagh

Step 4: Now for every word of the column, reverse the text for all six columns.
engh/ndua/oler/uosd/gnst/hgao

Step 5: Then write the whole text with continuation by eliminating the spaces in between the six words.
enghnduaoleruosdgnsthgao

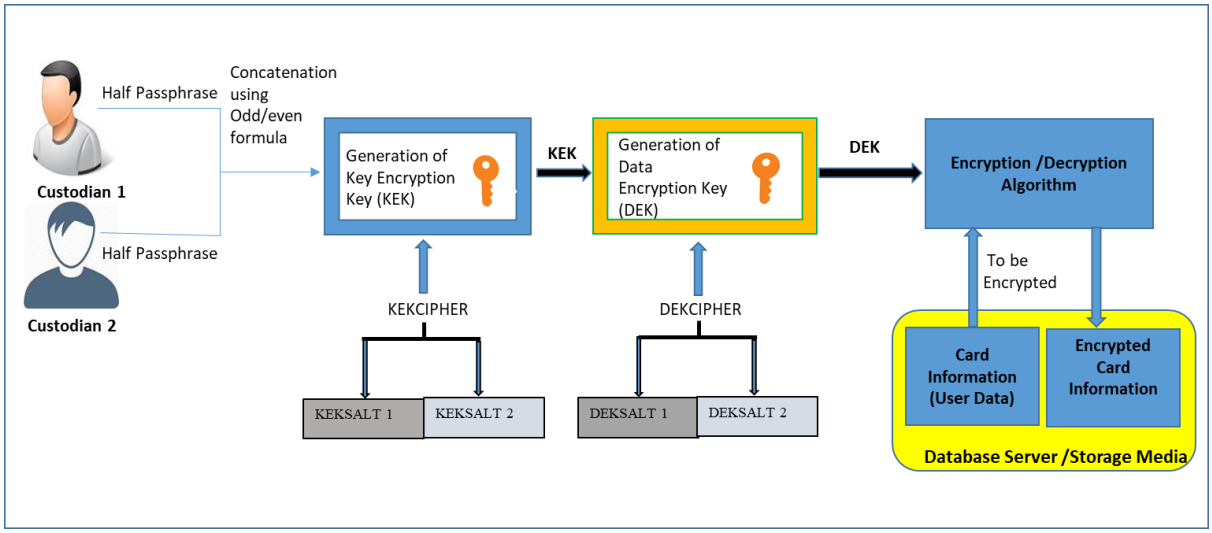

Fig. 3. Step break down of the encryption/decryption process of a cryptosystem

Step 6: Assign number to all the characters starting from 1 to so on.
e = 1, n = 2, g = 3, h = 4, n = 5, d = 6, u = 7, a = 8, o = 9, l = 10, e = 11, r = 12, u = 13, o = 14, s = 15, d = 16, g = 17, n = 18, s = 19, t = 20, h = 21, g = 22, a = 23, o = 24

Step 7: First write all the odd numbered characters, and then write the even numbered characters in sequence.
egnuoeusgshanhdalrodntgo

Step 8: Now reverse the whole word starting from the last character of the word to the first character of the word.
ogtndorladhnahsgsueounge

Step 9: For the above step apply SHA 256 technique and a cipher will occur which is known as KEKBASE key.

2.1.2 Generating KEK and DEK

The encryption process in payment application uses "PBE with MD5 and Triple DES" algorithm to encrypt and decrypt the sensitive data of the payment application. In this application we are using the Java cryptographic extension with the unlimited strength jurisdiction policy as per jdk specification. For encrypting the card details and storing it following process has been followed:

2.1.2.1 Retrieving the Key Encryption Key (KEK)

i) System retrieves password from both the custodian databases.

ii) The System does the concatenation of both custodian passwords in odd and even fashion to form KEKBASE key.

iii) System retrieves the two KEKSALTS from KEKsalt databases to form KEKCIPHER .

iv) Presently system has 2 information that are KEKBASE key and the KEKCIPHER.

vi) At this moment using Java cryptographic extension and using algorithm "PBE with MD5 and Triple DES" along with 100 iterations, the system generates the key which we name as Key Encryption Key (KEK).

2.1.2.2 Retrieving the Data Encryption Key (DEK)

i) The System retrieves DEKSALTS from DEKsalt databases to form DEKCIPHER.

ii) Now system cast DEKCIPHER by concatenating DEKSALT 1 and DEKSALT 2.

iii) System retrieves the KEK generated.

iv) At present, the system has two information KEK and DEKCIPHER, so it is time to generate DEK.

v) After accumulating all the required information, java cryptographic extension and using "PBE with MD5 and Triple DES" algorithm with 100 iteration system generates key at run time which is named as DEK.

2.1.3 Encryption Process

Advanced encryption standard (AES) is a secured encryption standard algorithm. To encrypt the card information system, it uses the Java cryptographic extension and "AES/ECB/PKCS5 Padding" algorithm. For encryption this algorithm requires the secret key, i.e., DEK and the sensitive data to be encrypted, i.e., primary account number of the user i.e., card number. Formerly we apply the process, we receive encrypted card information which is stored in the database.

2.1.4 Decryption Process

In pursuance of retrieving the original string system performs the same process as in the encryption process except the algorithm used is in decryption mode.

3. RESULTS

The dataset represents actual mobile payment transactions consummated by various consumers of mobile payment system (Vishwakarma et al., 2016). Data was simulated from the repository of mobile payment data in which the consumer's card details are stored for performing financial transactions. The card detail of the consumer is the sensitive data at rest subject to protection which is stored in the card user database. The consumer data set contains the card user identification number (ucid), user identification number (userid), primary account number (PAN), card expiry month and card expiry year in the card user database is as shown in Fig. 4. Encryption at the database level is performed i.e., column level encryption is performed therefore; the primary account number of the user is encrypted.

Encryption key management is about managing the cryptographic keys used in the encryption process of sensitive data. The symmetric key cryptography is used to encrypt as well as decrypt the data. That is primarily same encryption key is used to defend data at rest. The sensitive data to be encrypted is stored in a user card details database.

In this work a cryptosystem is proposed in which a data encryption key is used to encrypt and decrypt the user's primary account number. Nevertheless, the data encryption key (DEK) is generated using the key encryption key (KEK) and DEKCIPHER while, key encryption key (KEK) is generated using KEKBASE key and KEKCIPHER. Therefore, the simulation process of sensitive data encryption begins with first generating KEKBASE key. The KEKBASE key is generated by applying odd/even technique to concatenated two half passphrases is as shown in Fig. 5.

After acquiring KEKBASE key it is the time to generate a key encryption key (KEK). The KEK key generation is performed using KEKBASE key and KEKCIPHER is as shown in the Fig. 6.

←T→	ucid	userid	pan	expmonth	expyear
Edit Copy Delete	1	1	4478258936982587	3	25
Edit Copy Delete	2	2	4678789654123654	11	21
Edit Copy Delete	3	3	5134789056783456	2	22
Edit Copy Delete	4	4	4456875634129807	3	22
Edit Copy Delete	5	5	4456565855461480	3	22
Edit Copy Delete	6	6	4478523698745632	4	22
Edit Copy Delete	7	7	5333329397316448	9	25
Edit Copy Delete	8	8	5142568921285785	6	22
Edit Copy Delete	9	9	4310647843537406	12	25
Edit Copy Delete	10	10	4611133627964308	9	24

Fig. 4. Snapshot of card user detail database before encryption

```
1st Half Passphrase Fetched!!!
Passphrase 1 -> hard to guess


2nd Half Passphrase Fetched!!!
Passphrase 2 -> and long enough

Concatenation of Halfpassphrase 1 and Halfpassphrase 2 performed!

Concatenated Passphrase: hard to guess and long enough

After appyling Odd-Even Technique: t#ednggnaoolsedhugrauohsn

SHA-256 cipher text: 5219d7f6aa2d86a080cfb0a31135c57f506e5a8c394d297ee7370de4c6029cc9
```

Fig. 5. Output of KEKBASE key generation

```
Keksalts fetched from database
KEKCIPHER Generated -> javax.crypto.Cipher@3d8c7aca
KEK key generated -> nyQloSX/HafNZNAFTbVU99KvOIJlbnnZcKz5NEqJZgdetMB22rQkVNOKLXm48dlzUAAH5XId3xbW9Z04vp/Xaqt+UPbW62NO
```

Fig. 6. Output of Key encryption key (KEK) generation

```
Deksalts fetched from database
DEKCIPHER Generated -> javax.crypto.Cipher@5ebec15
DEK key generated -> KtWYlF3WBeZysPn7CtM4sg5At5tDaefi
```

Fig. 7. Output of Data encryption key (DEK) generation

	ucid	userid	pan	expmonth	expyear
Edit Copy Delete	1	1	/blDs+0OPBatMwfRBsLoLmnHIBStVgtnHEaqpGVrh8U=	3	25
Edit Copy Delete	2	2	AVXeLCd8nLBcy/7N4PlOV2nHIBStVgtnHEaqpGVrh8U=	11	21
Edit Copy Delete	3	3	YaWOq5yzK5SkM3IKLfyzDWnHIBStVgtnHEaqpGVrh8U=	2	22
Edit Copy Delete	4	4	09JbOF6F2+Cn5MHPCVDth2nHIBStVgtnHEaqpGVrh8U=	3	22
Edit Copy Delete	5	5	HE1HOGhaHqlOtWa6lto58GnHIBStVgtnHEaqpGVrh8U=	3	22
Edit Copy Delete	6	6	nRKWwFnpY10uWO+tyDEryGnHIBStVgtnHEaqpGVrh8U=	4	22
Edit Copy Delete	7	7	/kmVVAu058kfKvKfhgnwd2nHIBStVgtnHEaqpGVrh8U=	9	25
Edit Copy Delete	8	8	ZOnVJ5zolEkRUABbONM4qmnHIBStVgtnHEaqpGVrh8U=	6	22
Edit Copy Delete	9	9	8WDbtgIUI7/CjuO3URpTGmnHIBStVgtnHEaqpGVrh8U=	12	25
Edit Copy Delete	10	10	7lYMg9k68jCLebJP+MsmWnHIBStVgtnHEaqpGVrh8U=	9	24

Fig. 8. Snapshot of card user detail database after encryption

Finally the data encryption key (DEK) key is generated with the help of key encryption key (KEK) and DEKCIPHER is as shown in the Fig. 7.

The output of the cryptosystem is an encrypted sensitive data at rest, i.e., primary account number of the user of mobile payment system. The encrypted primary account number (PAN) is as shown in the Fig. 8.

4. DISCUSSION

For financial institutions securing their customer's data is highest priority. Moreover, encryption imparts the optimum mobile payment security feasible. For adversaries, sensitive data at rest is the data in file systems or in databases and it is apparently more alluring than data in transit. Data at rest in mobile payments is the financial information, i.e., card details of the payment users residing on database servers. For a secured mobile payment system the payment transaction must be secured with end-to-end encryption similarly, sensitive data at rest must be encrypted. The financial institutions should assure the security and confidentiality of consumer's payment data by mitigating the risk of sensitive data exposure or modification.

Encryption of sensitive data at database level provides an opportunity to integrate encryption key management into the encryption implementation. In mobile payment system the card user details such as primary account number is the sensitive data at rest to be secured. Therefore, to secure primary account number a cryptosystem is proposed which uses AES algorithm to encrypt the sensitive card data. Howbeit, an important decision has been made to protect the sensitive card data of consumers in a mobile payment system. It is imperative to fathom the decision about the security controls applied to confidentiality and integrity security principles of sensitive data at rest.

4.1 Attack Facial on Data-At-Rest

The data is susceptible despite of wherever it is populated. It is pivotal to learn where the vulnerabilities lie and taking appropriate decisions for securing the sensitive data. Data at rest is like a legendary pot of gold for the adversaries. The payment industries maintain detailed databases of their users like card information, user identity information, etc., and attacks can begin at database server, client interface, operating system and so on.

- Card information stored on the database server is not encrypted. This can be lessened by encrypting the data using the proposed approach for encryption as it is a supplementary security to build data futile.
- If an adversary is able to access sensitive data, then the data confidentiality and integrity objective are not perpetuated. Hence, to defend data confidentiality and integrity, encryption and decryption process along with SHA algorithm are can be carried out with proper

management of keys.

- Data leakage in database server can manifest the personal and credit card details of the users. By deploying data encryption method, one can protect against accidental data leak. Therefore, encryption is a simple solution to data leakage prevention.
- A server is located in a centralized location and is accessible by a single sign-on process. It can be easy for an intruder to discover state-of-the-art for stealing data. Therefore, Multi-factor authentication (MFA) can be used as one of the elite ways of keeping data safe.

Protecting the sensitive data of payment users is imperative as adversaries discover state-of-the-art for stealing data. Therefore, Encryption plays an important role in protecting data at rest. Solution to attack on data at rest is encrypting the card information of the users which resides on the database server. Encryption is a way ahead to prevent data breaches.

4.2 Security Analysis

The security analysis in mobile payments is procured by using confidentiality and integrity. As the sensitive data of consumers are kept confidential it is imperative to have a decision strategy based on the protection rules applied to defend confidentiality and integrity of sensitive data along with the safety of cryptographic keys.

- Confidentiality: Confidentiality is protecting the information from exposure to illegitimate parties. Confidentiality can be preserved through the encryption/decryption process in mobile payments. In proposed symmetric key cryptography, the data encryption key (DEK) is used to encrypt/decrypt the sensitive data. The consumer card details are encrypted with the DEK and stored in the database server.
- Integrity: Integrity is the consistent information that is not modified or devoured by an unauthorized user. Hash algorithms like message digest version 5 (MD5) and secure hash algorithms (SHA) prevents integrity threat. In the generation of KEK and DEK keys password based encryption with MD5 and triple DES is used whereas SHA is used in generation of KEKBASE key. Message digest 5 (MD5) algorithm or SHA algorithm generates a checksum therefore, if there is any modification of data then the checksum will change. However, this protection works individually from the encryption process so data integrity can be facilitated.

5. CONCLUSION AND FURTHER WORK

In this work a key management of sensitive data of the user is proposed to secure data at rest in mobile payment system. Protecting the sensitive data in databases is critical, as it can safeguard the financial institutions from unauthorized access to the data. Cryptography is a traditional method of preserving secrecy of data in databases. To defend the sensitive data at rest, encryption and decryption operations are carried out by management of key. Data-at-rest encryption safeguards the protection of sensitive information stored in databases and also helps to elate information security.

This work also highlights the security principles that adhere to prevent sensitive information are confidentiality and integrity. Thusly, an optimum solution to secure sensitive data at rest in mobile payment at the database level is imparted with symmetric key cryptographic management. For further work a crypto period can be defined for the encryption key. A crypto period is the time duration of a key defined through which the key is authorized for use. This will be useful in terms of data exposure or when the keys are lost or exposed to an adversary.

ACKNOWLEDGEMENT

The authors thank our colleagues from K L E Foundation who provided knowledge and encouragement that eminently assisted the research carried by us.

REFERENCES

Bhatia, T., Verma, A.K. 2017. Data security in mobile cloud computing paradigm: a survey, taxonomy and open research issues, The Journal of Supercomputing, 73, 2558–2631. https://doi.org/10.1007/s11227-016-1945-y

Gugelmann, D., Studerus, P., Lenders, V., Ager, B. 2015, July-Aug. Can Content-Based data loss prevention solutions prevent data leakage in Web Traffic?, in IEEE Security & Privacy, 13, 52–59.

Huang, Y.L., Leu, F.Y., Dai, C.R. 2012. A secure data encryption method by employing a feedback encryption mechanism and Three-Dimensional operation. In: Quirchmayr G., Basl J., You I., Xu L., Weippl E. (eds) Multidisciplinary Research and Practice for Information Systems. CD-ARES 2012. Lecture Notes in Computer Science, 7465. Springer, Berlin, Heidelberg

Inside Secure. 2009 May. Securing mobile payments, white paper. Available at https://www.insidesecure.com/content/download/1133/13650/file/Securing%20Mobile-Payments.pdf

Javidan, R., Pirbonyeh, M.A. Nov, 2010. A new security algorithm for electronic payment via mobile phones, 2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2010), Rome, 7–10, 1–5.

Kadambi, K.S., Li, J., Alan, H. Karp, 2009, August. Near-field communication-based secure mobile payment service, ICEC '09 Taipei, Taiwan, 12–15, 142–151

Moulds, R. 2007, July. The key to widespread data encryption, Computer Fraud and Security, 2007, 18–20.

Muslukhov, I., Sun, S.-T., Wijesekera, P., Boshmaf, Y., Beznosov, K. Oct, 2016. Decoupling data-at-rest encryption and smartphone locking with wearable devices, Pervasive and Mobile Computing, 32, 26–34. net.2020.107118.

Nosrati, L., Bidgoli, A.M. May, 2016. A review of mobile banking security, 2016 IEEE Canadian Conference on Electrical and Computer Engineeri,ng (CCECE), Vancouver, BC, 15–18, 1–5.

Nxumalo, Z.C., Tarwireyi, P., Adigun, M.O. Oct, 2014. Towards privacy with tokenization as a service, 2014 IEEE 6th International Conference on Adaptive Science & Technology (ICAST), Ota, 29–31, 1–6.

PCI Security Standards Council, 2010, October. Understanding the payment card industry data security standard version 2.0, PCI DSS quick reference guide.

Setiadi, D.R.I.M., Faishal Najib, A., Rachmawanto, E.H., Atika Sari, C., Sarker, K., Rijati, N. 2019. A comparative study MD5 and SHA1 algorithms to encrypt REST API authentication on Mobile-based application, 2019 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 206–211. doi: 10.1109/ ICOIACT46704. 2019. 8938570.

Shabtai, A., Elovici, Y., Rokach, L. 2012. A survey of data leakage detection and prevention solutions, Springer-Verlag New York Incorporated.

Shah, S.Y., Paulovicks, B., Zerfos, P. Dec, 2016. Data-at-rest security for spark, 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, 5–8, 1464–1473.

Stilgherrian, June 18, 2015. Encrypting data at rest is vital but it's just not happening, ZDNet.

Storage Networking Industry Association (SNIA) Storage Security Industry Forum, 2009. Solutions guide for data-at-rest.

Sultan, A., Elankayer, S., Vallipuram, M. 2016. A survey on data leakage prevention systems. Journal of Network and Computer Applications, 62, 137–152. https://doi.org/ 10.1016/j.jnca.2016.01.008.

Tabrizchi, H., Rafsanjani, M.K. February, 2020. A survey on security challenges in cloud computing: issues, threats, and solutions. The journal of supercomputing (2020). https://doi.org/10.1007/s11227-020-03213-1.

Turban, E., King, D., Lee, J.K., Liang, TP., Turban, D.C. 2015. Electronic Commerce Payment Systems. In: Electronic Commerce. Springer Texts in Business and Economics. Springer, Cham, 519–557, https://doi.org/10.1007/978-3-319-10091-3_11.

Vishwakarma P.P., Tripathy A.K., Vemuru S. 2018. The Fact-Finding security examination in NFC-enabled mobile payment system. International Journal of Electrical and Computer Engineering (IJECE). 8, 1774 – 1780. DOI: 10.11591/ijece. v8i3.pp1774–1780.

Vishwakarma, P., Tripathy, A.K., Vemuru, S. Dec, 2016. A hybrid security framework for near field communication driven mobile payment model, International Journal of Computer Science and Information Security, USA, 14, 337–348.

Vishwakarma, P.P., Tripathy, A.K., Vemuru, S. 2018. A layered approach to fraud analytics for NFC-Enabled mobile payment system. In: Negi A., Bhatnagar R., Parida L. (eds) Distributed Computing and Internet Technology. ICDCIT 2018. Lecture Notes in Computer Science, 10722. Springer, Cham.

Wang, C., Wang, Y., Chen, Yingying, Liu, Hongbo, Liu, J. 2020. User authentication on mobile devices: Approaches, threats and trends, Computer Networks, Volume 170, 107118, ISSN 1389–1286, https://doi.org/10.1016/j.com

Zaw, T.M., Thant, M., Bezzateev, S.V. 2019. Database security with AES encryption, Elliptic Curve Encryption and Signature, 2019 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF), Saint-Petersburg, Russia, 1–6. doi: 10.1109/ WECONF.2019.8840125.