Imperial Journal of Interdisciplinary Research (IJIR)
Vol-3, Issue-4, 2017
ISSN: 2454-1362, http://www.onlinejournal.in

Credit Card Fraud Detection Using Hidden Markov Model

Akash Pawar[1], Vishwajit Patil[2], Swapnil Martin[3] **&**
Miss Sangita Chaudhari[4]
1,2,3,4Department of Computer Engineering, Vidyavardhini college of Engineering and Technology, Mumbai, India.

Abstract: *Today's word the use of internet increasing. it provide cashless shopping for customer so we use credit card for online transactions. Credit card is the best way to fast online transaction for shopping . HMM(hidden morkov model) provide appropriate service to detect or prevent user those are not genuine. in HMM transaction will provide security to detect fraud transaction.*

In the existing credit card fraud detection business processing system, fraudulent transaction will be detected after transaction is done. It is difficult to find out fraudulent and regarding loses will be barred by issuing authorities. Thus, during the transaction we generate questions using verification engine. This process of generating questions is done every time for more security. If all the questions are not answered then the transaction fails and a message is generated of the same.

We repeat the same process of generating questions for every transaction just for higher security purpose and minimum frauds. We present detailed experimental results to show the effectiveness of our approach and compare it with other techniques available in the literature

1. Introduction

We are using credit card for cash payment. The payment method are online or offline. For offline payment Cardholder present his credit card to the merchant and merchant swipe card on swipe machine for payment.. In online shopping we want small amount of information to do a transaction. In this method transaction mainly done via cable or wireless on the internet.

So hacker can easily break the system security that's why credit card fraud increasing every year. The cardholder does not known's some one stolen card information to do a fraud transaction.

In day to day life credit cards are used for purchasing goods and services by the help of virtual card for online transaction or physical card for offline transaction. In physical transaction, Credit cards will insert into payment machine at merchant shop to purchase goods. Tracing fraudulent transactions in this mode may not be possible because the attacker already steal the credit card [4]. The credit card company may go in financial loss if loss of credit card is not realized by credit card holder. In online payment mode, attackers need only little information for doing fraudulent transaction. To commit fraud in these types of purchases, a fraudster simply needs to know the card details (secure code, card number, expiration date etc.). Most of the time, the genuine cardholder is not aware that someone else has seen or stolen his card information. In this purchase method, mainly transactions will be done through Internet or telephone. Small transactions are generally undergo less verification, and are less likely to be checked by either the card issuer or the merchant. Card issuers must take more precaution Card issuers must take more precaution against fraud detection and financial losses. Credit card fraud cases are increasing every year. In 2008, number of fraudulent through credit card had increased by 30 percent because of various ambiguities in issuing and managing credit cards. Credit card fraudulent is approximately 1.2% of the total transaction amount.

2. Advance Encryption Standard

AES is used for encrypt data in the form of cipher text. in HMM we are encrypt all our database .because If attacker access the database then he/she can't change or modify data.

AES has a plain text and Secrete key. AES can be perform on 128,192,256 bite plain text and cipher key. Input will generate cipher text while performing AES operations.

10,12,14 rounds for 128,192,256 bit keys.

Regular round are 9,11,13

Final round is different 10^{th},12^{th},14^{th}.

Four Operations are perform on the plain text aur key to produce cipher text.

I. *SubBytes*

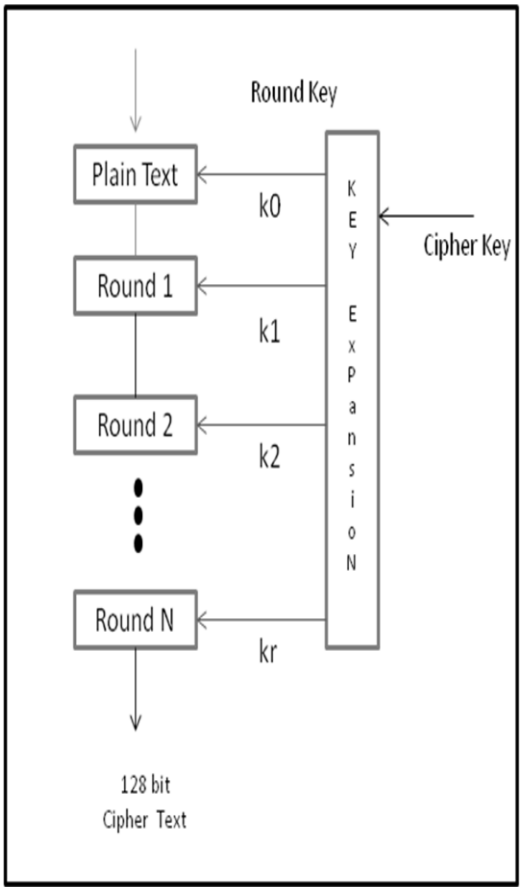
II. *ShiftRows*

III. *MixColumns*

IV. *AddRoundKey*

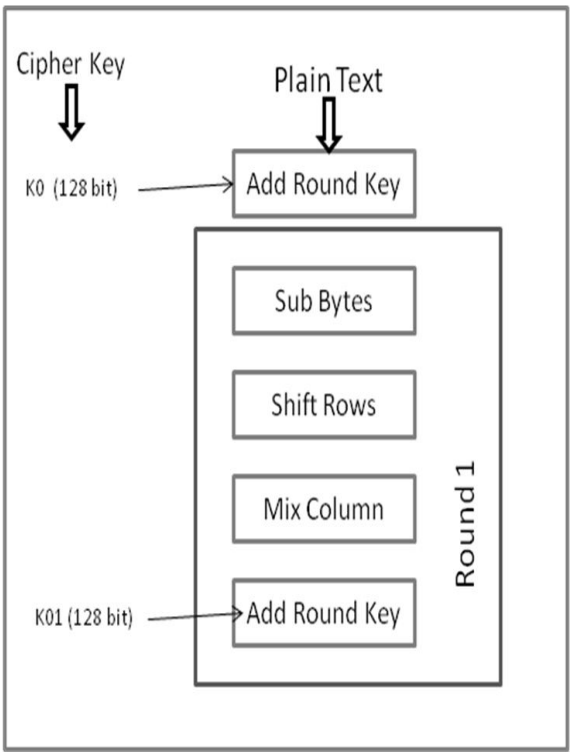
Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

The schematic of AES structure is given in the following illustration −



2.1 Encryption Process

Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes. The first round process is depicted below



2.1.1. Byte Substitution (SubBytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

2.1.2. Shiftrows

Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row. Shift is carried out as follows –

I. First row is not shifted.
II. Second row is shifted one (byte) position to the left.
III. Third row is shifted two positions to the left.
IV. Fourth row is shifted three positions to the left.
V. The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

2.1.3. MixColumns

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

2.1.4. AddRoundkey

The 16 bytes of the matrix are now considered as 128 bits and are XOR to the 128 bits of the round key. If this is the last round then the output is the cipher text. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

]2.2 Decryption Process

The process of decryption of an AES cipher text is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order –

I. Add round key

II. Mix columns

III. Shift rows

IV. Byte substitution

2.3 AES Analysis

In present day cryptography, AES is widely adopted and supported in both hardware and software. Till date, no practical cryptanalytic attacks against AES has been discovered. Additionally, AES has built-in flexibility of key length, which allows a degree of 'future-proofing' against progress in the ability to perform exhaustive key searches.

However, just as for DES, the AES security is assured only if it is correctly implemented and good key management is employed.

3. Captcha

CAPTCHA stands for Completely automated public Turing test to tell computer and human apart. it is a program that tells computer The user is a human being or a robot or computer software. To know that the user is a human or a machine captcha generates some tests that are very easy for a human but not for the machine. Captacha tests contains some desorted texts or images because humans can read and understand distorted text or images, but current computer programs can't. The test should be like for example: a string with combination with characters and numbers, a mathematical equation,disorted images. humans can read distorted text as the one shown below, but current computer programs can't.
Captcha is also used to prevent bots as well as other programs of computer for submitting or accessing Web servers or websites. It can also prevent websites from spammers. Most of the Web Servers and Websites are being benefited from Captcha. As far as Captcha entry work is concerned, it is a simple typing work.

reCAPTCHA is a free service that protects your site from spam and abuse. It uses advanced risk analysis techniques to tell
humans and bots apart. With the new API, a significant number of your valid human users will pass the reCAPTCHA challenge
without having to solve a CAPTCHA.

Conclusion

The work on this project has helped the team in great way in exploring the requirement of an organization. Implementation was in synchronization with the analysis done before hand and after rounds of testing of the software developed. This project has met its objectives to produce: a generic map of the planning functions and processes of project.
We are developing our project in and Microsoft Server to provide the flexibility to application "HMM" to provide full functionality with the finest of details about any customer of credit card.
So we can say that core purpose of designing "CREDIT CARD FRAUD DETECTION SYSTEM USING" is to manage the task related to the credit card purchase and to reduce frauds in online purchases.

HMM is not a product to be sold; it is a more about Customers privacy and Security.

.

References

[1] "Global Consumer Attitude towards On-Line Shopping,"
http://www2.acnielsen.com/reports/documents/2005_cc_onlineshopping.pdf, Mar. 2007.

[2] "Statistics for General and On-Line Card Fraud,"
http://www.epaynews.com/statistics/fraud.html, Mar. 2007.

[3] S. Ghosh and D.L. Reilly, "Credit Card Fraud Detection with a Neural-Network," Proc. 27th Hawaii Int'l Conf. System Sciences: Information Systems: Decision Support and knowledge-Based Systems, vol. 3, pp. 621-630, 1994.

[4] S.J. Stolfo, D.W. Fan, W. Lee, A. Prodromidis, and P.K. Chan, "Cost-Based Modelling for
fraud and Intrusion Detection: Results from the JAM Project," Proc. DARPA Information Survivability Conf.n and Exposition, vol. 2, pp. 130-144, 2000.