ISSN 2278 – 3091

International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Vol. 4 , No.4 Pages : 49 - 50 (2015)
Special Issue of ICEEC 2015 - Held on August 24, 2015 in The Dunes, Cochin, India
http://warse.org/IJATCSE/static/pdf/Issue/iceec2015sp10.pdf *(ICEEC 2015)*

A Hybridized Framework For Safer Storage Of User's Billing Information On Merchant's Site

Shadkam Islam[1], Hima Bindu Maringanti[2]
Booking.com, Amsterdam, shadkam.islam@gmail.com
[2]North Orissa University, Odisha, India, profhbnou2012@gmail.com

Abstract ***: The present paper is a short and a theoretical proposal of implementing a secured user billing information detail , that is least (not at all) prone to leakage. This objective is being achieved by the use of a private key, which encrypts the Credit card / Debit card that is dynamic in nature, as it is changed for every merchant per transaction. The proposal presented in short paper is a self-validated and a practically implementable solution to online transaction security, with no burden on the user of the internet.***

Key words : Public key, Private Key, Encrypt and Decrypt, Merchant-id, User-id .

INTRODUCTION

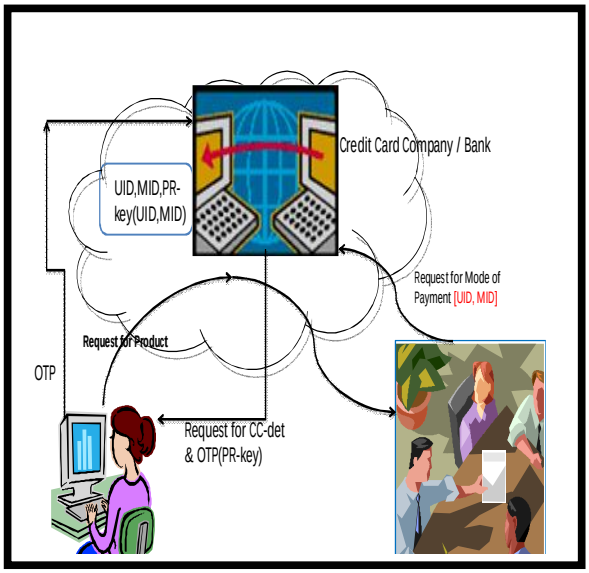
Many a times, companies / merchants have to store user billing information (Credit Card / debit card related data). Though they store it in an encrypted form, in any unfortunate incident of data leakage to unauthorized persons, the following repercussions are observed:

- at best, user confidence is gone, with most of them deciding to block that card,
- at worst, the information to decrypt the billing information may also have got compromised, and
- before all the users could be informed about blocking the cards, the hackers may do some transactions.

Also, users who would like to avail some services, hesitate in doing so, if the company / merchant stores billing information (which is generally the case for subscriptions). This is because
once the Credit Card details are out, there is no way to take them back / erase it, unless one gets the cards blocked. The downside is, once a card is blocked, it is blocked everywhere, and new card details (as and when they become available) will have to be re-entered by the user for all the services they had used the earlier card for. The control is there with the merchant ; so the user always has this risk at the back of their head that even if they discontinue the service / they still would have to check their bills to see that they haven't been charged.

PROPOSED APPROACH

A novel approach for keeping the user billing information safe from unauthorized persons has been proposed in this section. The merchant storing the Credit Card details does not need to know the details. He / She only needs to be able to charge the user making transactions using the Credit Card details. This is possible if the Credit Card issuing company (bank) provides the option to the user to set a private key (password) [1] on a per merchant basis, analogous to the traditional One Time Pad substitution Cipher [3]. This process tends to create a unique key based upon the user id (UID) and the merchant id (MID), probably also time-stamped and which is discarded after use or once the session expires, whichever is earlier [2]. Whenever the user is entering his/her Credit Card details on a merchant's website, the user also enters this private key along with this info. The merchant site automatically sends the credit card details (CC-details), user id (UID) and the merchant id (MID) to the credit card company (CCC) or the Bank. The details which the merchant stores would be encrypted based on this private key and the private key would only be stored with Credit Card company / bank; not with the merchant. Whenever the merchant has to receive / take some money from the user, they would send the Encrypted Credit Card details to the Credit Card company and the user's id. It can be user's back login id, or his / her email id / any primary key using which, the bank can identify the user and the merchant's id. In turn, the Credit Card company, based on the user-id and merchant-id, would get the private key from its records, decrypt the Credit Card details, and accept / reject the transaction. The above functioning of the proposed system is depicted in the figure 1 below:



ISSN 2278 – 3091

International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Vol. 4 , No.4 Pages : 49 - 50 (2015)
Special Issue of ICEEC 2015 - Held on August 24, 2015 in The Dunes, Cochin, India
http://warse.org/IJATCSE/static/pdf/Issue/iceec2015sp10.pdf *(ICEEC 2015)*

Credit Card information storage by the merchant:
At present, Credit Card details are stored by the merchants. In the proposed approach, encrypted Credit Card details (without the key needed to decrypt it) would be stored by the merchants. In the form which merchant provides to the user for entering Credit Card details, it would also provide a password-field to enter the private-key for this merchant. There are multiple ways to use it (with varying level of security):

a) The Mobile App / WebPage takes care of encrypting all the Credit Card details with the private key, and then only sending it to merchants' web server. So, not only merchant's database, but even merchant's web server would never have access to the private key.

b) The above involves, particularly in case of WebPage, some processing with JavaScript – if we want to avoid that, all these details could be submitted to the Credit Card company's server, which would encrypt Credit Card details with the private key, and send it to merchant's web server.

c) The Mobile App / Web Page sends it directly to merchant's web server (this is the flow closest to how things work generally at the moment) [4], and the merchant guarantees that the private key (and non-encrypted Credit Card details) would be stored nowhere; private key would simply be used to encrypt Credit Card details (that too, this one time only), and only the encrypted Credit Card details would be stored.

d) Apart from storage of encrypted Credit Card details on the merchant's site, user would have to enter the private key for this merchant on the Credit Card Company's site as well.

This may be done at the same time, or before / after Credit Card details have been entered on the merchants' site.

Credit Card information usage by the merchant:
At present, merchant sends Credit Card details and merchant's id to the Credit Card company / bank to request the transaction.
In the proposed approach, merchant would send encrypted Credit Card details, along with user-id and merchant-id to the Credit Card company (as noted earlier too) to request the transaction.
Credit Card company, based on the user-id and merchant-id, would get the private key from its records, decrypt the Credit Card details, and make (or not) the transaction.

CONCLUSION AND DISCUSSIONS

In the event of real or perceived information leakage from the merchant's database, users can either change their private key for this merchant on the Credit Card company website or disable this merchant temporarily. Since the unauthorized persons have access only to the encrypted details, even if they send requests for transaction, since the key is changed / merchant id is disabled (for this user); so no transaction can happen. However, user's Credit Card can continue to work at other places / with other merchants.

The proposed system enables storage of the Credit card details in an encrypted form; wherein the private key used for encryption is available only with the provider , the Credit card company or the Bank, who operated here as an intermediary authenticator of the (UID, MID) pair , in addition to being time-stamped to the time of initiation of the transaction between the User and the Merchant.

In case the users want to end a subscription, apart from doing it on the merchant's site, they can also disable merchant / change private key for that merchant, at the Credit Card company site. They would have peace of mind and they would not have to keep checking their Credit Card bills for any transactions, still being made by that merchant.

REFERENCES

[1] S. Fumy, and P. Landrock, "Principles of Key Management", *IEEE Journal on Selected Areas in Communication* , June 1993.
[2] T. Ritter , "The Efficient Generation of Cryptographic Confusion Sequences", *Crptologia*, vol. 15, no. 2, 1991.
[3] W. Stallings, *Cryptography and Network Security*, 4th ed. Prentice Hall, 2007.
[4] P. Cheng et. al., "A Security Architecture for the Internet Protocol", *IBM Systems Journal*, no.1, 1998.