A New Credit Card Payment System Based on 3D-Secure™ Using One-time-use Transaction Numbers

Farid Javani1, Shahriar Mohammadi2

1Information Technology Group, International Branch of University of Guilan, Rasht, Iran
farid.djavani@gmail.com

2Information technology group, Industrial Department, Khaje Nasir Toosi University of Technology, Tehran, Iran
smohammadi40@yahoo.com

Abstract: Along with rapid growth in usage of credit card in e-commerce, rough and robust security measure and technologies should be developed. One old problem about the credit card transaction -that also exists in widely used payment system like 3D-Secure™- is the possibility of disclosure of sensitive data which is being stored by merchants during the transaction. A solution to this problem is using provisional information in transactions. In this paper, solving this problem, we propose a one-time-use transaction number generation method using hash functions in 3D-Secure™ payment system.

Keywords: one-time transaction numbers; electronic payment system; 3D-Secure™;

I. Introduction

Credit cards have been default payment system in recent years in electronic commerce [1] and their usage continues to grow in US and many other countries [2]. Notwithstanding the security concerns and frauds about credit card systems, they can be considered as successful electronic payment methods [1]. However further growth in usage of electronic payment systems and correspondingly, credit cards, along with their many benefits like economic efficiency and growth, customer and merchant satisfaction and expansion of exports in countries [2], fundamentally depend on designing and developing more and more secure payment systems [3],[4]. Obviously, security plays an important role in acceptance of electronic payment systems but there are some deterrent issues in developing security systems. Cost, for example, is a vital factor that directly affects developing process of security systems; that is, more security means more cost. Therefore, developing and introducing payment systems with perfect balance of security statements and technical protections will be more helpful for acceptance of credit cards than mere concentration on the later one [4]. New security models will be more easily accepted and applicable if they do not need fundamental changes in the security models and architectures which are currently being used. This is a key aspect we have considered in our model.

During credit card transactions, the need for storage of costumers’ sensitive credit card data is inevitable; service providers, merchants and other organizations dealing with credit cards need to store and process credit card data for reasons like fraud detection, chargebacks and advertising. The problem is that, especially in small businesses, these large amounts of confidential customer information, including costumers’ credit card numbers, are precious targets for attackers and fraudsters. Protection of stored credit card data is an issue that can affect credit card market in both good and bad ways. High protection of card data may increase credit card usage while more successful attacks on stored sensitive credit card data can cause anxiety and fear of losing money and privacy in costumers. With the aim to cover and solve this problem, Payment Card Industry Data Security Standard (PCI DSS) was developed in 2006 with the cooperation of five major payment brands forming PCI Security Standards Council (PCI SSC). PCI DSS is developed to reduce credit card data attacks and fraud by providing best practices about prevention, detection and appropriate reaction to security incidents. PCI DSS includes a set of 12 requirements as a baseline for security measures that should be implemented by any entity that stores, processes and transmits cardholder data [5]. Exact compliance with the standard ensures a solid security architecture against credit card data attacks but there are more issues that should be considered; costs of PCI DSS for merchants, lack of proper rules about dealing with the fraudulent insiders dilemma in the standard, and security packages proposed by third parties promising to provide appropriate security measures are practical hindrances for implementation of PCI DSS [6]. Besides, many entities have difficulties in exactly understanding the scope of the standard and thus many misconceptions of the standard happen [7]. With these encumbrances, exact compliance with the standard is almost impossible for most merchants and therefore more practical solutions should be developed. Fraudulent insiders, as mentioned before, can illegally access to stored credit card data despite of tough and robust security measures for credit card data storage and processing. Merchants or insiders may sell confidential credit card data or use them in other illegal ways and apparently there is not...
practical way to prevent this particular kind of fraud as long as real credit card numbers and other sensitive cardholder information are stored on merchant side.

Threats and possible attacks for stored credit card data also exist in modern authentication and payment systems like 3D-Secure™. In 3D-Secure™ payment system credit card information including credit card numbers is stored in merchant side causing the discussed attacks and threads for the stored sensitive credit card information become applicable. We solve this problem of 3D-Secure™ payment system.

One efficient solution to the issues of stored credit card data is using provisional credit card information in online transactions. Provisional credit card information is information that can be used instead of real credit card information in online transaction and are generated using real credit card data. Most important characteristic of provisional credit card information is that they have a close logical or mathematical relation with the real credit card information which cannot be transpirable or detectable. In our proposed electronic payment system, we use provisional credit card numbers to enhance 3D-Secure™ payment system. Rest of this paper is organized as follow; section 2 reviews related works and studies, section 3 provides a description of 3D-Secure™ payment system and introduces our proposed payment system, in section 4 we discuss security features of our model and we conclude in section 5.

II. Related Work

Many studies have been done proposing technics with the main idea of using provisional credit card information or so-called one-time or limited-use transaction numbers as a proxy for real credit card information, specifically, credit card numbers. In these technics, based on characteristics of the transaction which the cardholder intends to do, a transaction number with limitation like special amounts, merchants, and fixed time periods is generated and used instead of real credit card number. With these properties of proxy or transaction numbers, using them instead of real credit card numbers in online transactions makes stored sensitive credit card data and transaction information in merchant side become useless for fraudsters. Furthermore, in the case of one-time use credit card numbers, anew usage of these numbers by fraudsters can raise fraud alarm during authentication process.

Using one-time-use and limited-use credit card numbers has been subject of many studies and several methods have been proposed. In the system proposed by Frenklin et al. in [8] when a costumer decides to make an online transaction, she invokes a software module to send a request to the issuer institution asking for generating a transaction number. The issuer then generates a number, in the same format with real credit card number, as a proxy for real credit card number for a single transaction. Cardholder uses this number in order to make the transaction and the issuer also authorizes the transaction via this proxy number. The merchant does not know (and does not need to know) if he is dealing with the real or a proxy credit card number. In this method the merchant does not need to install any additional software but the cardholder does. Li et al. propose a system in which each transaction number is generated by putting the previous generated transaction number and a secret-embedded on the card-into a hash function. In this method some changes should be done in the hardware; a chip should be installed on the card, and also smart card readers should be enhanced to be able to generate transaction numbers [9]. Of course this method is only applicable when card readers become cheap enough to be widely used in non-commercial environments.

Offline transaction number generation schemes have also been proposed. In all of these methods an application should be installed on cardholder device in order to generate the transaction numbers. This can be any device like PC, laptop, PDA or mobile phone. These schemes have advantages when a connection to the issuing institution is not available, like in telephone shopping. In the offline scheme proposed by Rubin et al. the cardholder, related to her transaction, enters some restrictions to the number generator application and the application encodes these restrictions using a secret key-stored on the cardholder’s devise-to generate a limited-use transaction number [10]. In [11] Molloy et al. also propose a scheme that the cardholder does not need to connect to the issuer to generate the transaction number. The transaction number in this scheme is generated using hash function rather than encryption.

In all of the discussed methods, the generated limited-use transaction numbers have the same format as real credit card numbers; in our scheme, we use transaction codes rather than credit card alike numbers. In the rest of this paper we will use transaction number and transaction code in lieu of each other.

III. Credit Card Payment System Based on 3D-Secure™ Using One-time-use Transaction Numbers

We propose an online transaction number generation scheme that uses a hash function to generate the transaction number. We have two reasons for selecting hash function and not encryption; first reason is the high speed and low cost of number generation process with hash function, and the second reason is that, in fact, there is no need for features provided by encryption because generating the transaction number at the beginning of the transaction and the verification of the generated number during the authentication, are both done in the issuer’s side.

A. Transaction number generation

We suppose that merchant and credit card issuing bank or institution have their unique identification code. We will use following notation:

\[ TN: \text{Generated transaction number} \]
\[ I: \text{Issuer’s identification code} \]
\[ M: \text{Merchant’s identification code} \]
\[ T: \text{Total amount of items} \]
\[ C: \text{Credit card number} \]
\[ D: \text{Expiration date} \]
\[ H(\cdot): \text{Hash function} \]

The cardholder sends credit card number and expiration date and the merchant sends his identification code and the amount of the items to the issuer. The issuer receives this information and forwards them to the number generator application. The number generator application in issuer side
gets \( I, M, T, C \) and \( D \) as input, combines them to create a string and at last generates the transaction number by putting this information into a hash function as follow:

\[
\Psi = H(I, M, T, C, D) \\
TN = I \| \Psi
\]

Figure 1. shows this process. The transaction number, \( TN \) is consisted of 2 parts; \( I \), the issuer code and \( \Psi \), output of the hash function. Attaching the issuer code as a prefix to output of the hash function is to speed up the authentication process; that is, with this prefix, less time will be spent in order to find appropriate issuer. Generated transaction number is send to the merchant by issuer and the cardholder does not need enter it manually; so the length of \( \Psi \) depends only on the used hash function and necessary security factors.

The generated transaction code is added to cardholder data at issuer database for authentication and other necessary processes of the transaction. After all the processes related to a transaction are done, the related code will be deleted from cardholder data unless the case that transaction may have been reported as fraudulent. The merchant will also save the generated transaction number for further verifications.

**B. 3D-Secureᵀᴹ**

In Card Not Present (CNP) transaction, correct authentication of cardholder is the first and main security concern. With accurate and solid authentication systems fraudulent use of credit card and chargebacks can be avoided to the most extent. 3D-Secureᵀᴹ or three-Domain Secure is an authentication system developed by Visa which provides cardholder authentication within its three domains, Issuer Domain, Acquirer Domain and Interoperability Domain. It is also being used by major credit card companies; Visa uses 3D-Secureᵀᴹ as Verified by Visa and MasterCard uses it as MasterCard SecureCode. The cardholder needs to enroll in 3D-Secureᵀᴹ in order to be able to use its services. The model divides payment systems as follows:

- **Issuer Domain**
  - This domain is responsible for determining the cardholder’s eligibility to participate in the 3D-Secureᵀᴹ service, managing the enrollment of their cardholders in the service and authenticating cardholders during online purchases. It includes Cardholder, Issuer and Access Control Server (ACS).
- **Acquire Domain**
  - This domain is responsible for determining the merchant’s eligibility to participate in the 3D-

Secureᵀᴹ service, defining the procedures to ensure that merchants participating in Internet transactions are operating under a merchant agreement with the Acquirer and providing the transaction processing for authenticated transactions. It includes Merchant, Merchant Server Plug-in (MPI) and Acquirer.

- **Interoperability Domain**
  - This domain is responsible for generating selected certificates for the use of 3D-Secureᵀᴹ entities and facilitating the transaction exchange between the other two domains with a common protocol and shared services. It includes Visa Directory Server (DS), Certificate Authorities and Authentication History Server (AHS)

Issuer, Merchant and Acquirer are Visa member financial institutions. Messages sent between the systems and functions in model are as follows:

- **VREQ**: Verify Enrollment Request is a message from Merchant Server Plug-in to the Visa Directory Server or from Visa Directory Server to the ACS, asking whether authentication is available for a particular card number.
- **VERes**: Verify Enrollment Response is a message from the ACS to Merchant Server Plug-in telling whether authentication is available.
- **PAReq**: Payer Authentication Request is a message sent from the Merchant Server Plug-in to the Access Control Server requesting the issuer to authenticate its cardholder.
- **PARes**: Payer Authentication Response is a message formatted, digitally signed, and sent from the Access Control Server to the Merchant Server Plug-in providing the results of the issuer’s 3D-Secureᵀᴹ cardholder authentication.

1) **Purchase transaction flow in 3D-Secureᵀᴹ**

1. Cardholder decides to shop online, selects items and enters necessary information.
2. MPI send VEReq to DS to ask whether authentication is available for the card number.
3. DS send VEReq to ACS to ask whether authentication is available for the card number.
   - If authentication is not available the process continues with step 5.
4. ACS sends VERes to DS.
5. DS forwards the response to MPI.
   - If authentication is not available the process continues with traditional systems.
6. MPI sends PAReq to ACS via cardholder’s device.
7. ACS receives the PAReq.
8. ACS authenticates the cardholder by its own authentication method and then formats PARes and signs it.
9. ACS sends PARes to MPI via cardholder’s device and sends selected data to AHS.
10. MPI receives PARes.
11. MPI checks PARes signature.
12. Merchant continues with the authorization exchange with its acquirer [12], [13].

Figure 2. shows this information flow. Credit card data is being stored in merchants’ systems in every transaction done within 3D-Secureᵀᴹ model, and thus the threat of being compromised exists for cardholder data. In our model we enhance 3D-Secureᵀᴹ payment model by using one-time-use transaction numbers, which are generated from cardholder and transaction data. Number generation process is done using hash function on issuer side.

![Figure 2. 3D-Secureᵀᴹ information flow](image)

**C. One-time transaction number generation in 3D-Secure**

In our scheme, as stated before, we suppose that merchant and credit card issuing bank or institution have their unique identification code and the cardholder knows the issuer code during the transaction. When cardholder decides to shop online and selects the item she wishes to buy, instead of entering her credit card information at merchant’s page, she enters the issuer’s code. At merchant side, MPI sends the transaction information to the issuer and also redirects the cardholder to the issuer via the issuer code entered by cardholder. Cardholder enters the needed information in the issuer page and issuer generates a transaction code with
transaction information received from the MPI and credit card data. Cardholder does not need to know the transaction code; the issuer sends the transaction code via cardholder’s device to the MPI and using this code, MPI checks if the cardholder has been registered to the 3D-Secureᵀᴹ and after this, the transaction continues with its normal flow.

1) **Purchase transaction flow in 3D-Secureᵀᴹ enhanced with one-time-use Transaction Numbers**

1. Cardholder decides to shop online, selects items and enters the issuer code.
2. MPI redirects the cardholder to issuer and also send transaction’s information including the merchant’s code.
3. Cardholder enters necessary information including credit card number in issuer’s page.
   - Issuer now has all the necessary data, including card number and user device information.
4. Issuer generates a one-time-use transaction code with transaction and cardholder information and adds it to cardholder information for authentication.
5. Issuer sends generated code to MPI via cardholder’s device.
6. MPI send VEReq (using generated code) to Directory Server to ask whether authentication is available for the card number.
   - If authentication is not available, issuer sends real credit card information to MPI and the process continues with step 9.
9. Directory Server forwards VERes to MPI.
   - If authentication is not available the process continues with traditional systems.
10. MPI sends PAReq (identified with the generated transaction code) to ACS via cardholder’s device.
11. ACS receives the PAReq.
12. ACS authenticates the cardholder by its own authentication method and then formats PARes and signs it.
13. ACS sends PARes to MPI via cardholder’s device and sends selected data to AHS.
14. MPI receives PARes.
15. MPI checks PARes signature.
16. Merchant continues with the authorization exchange with its acquirer.

Figure 3. shows this information flow.

---

**Figure 3.** Information flow in 3D-Secureᵀᴹ enhanced with transaction numbers
IV. Security Features

In our proposed method, real credit card data is not stored on merchants’ systems during an online transaction and instead, provisional credit card information is stored. Merchant can perform any necessary process with this provisional data. Neither cardholder nor merchant need to install any additional application or software on their systems; this system provides its security features, enhancing 3D-Secure™ payment system, with four additional steps for generating transaction number during the transaction. An application installed on the issuer system, which generates transaction codes based on purchase and cardholder data. We enhance 3D-Secure™ without any change on existing infrastructure and thus without significant additional cost. Cardholder, as in a normal 3D-Secure™ transaction, only needs to enter her credit card information one time; the authentication process is 3D-Secure™ alike as well. Our system promises more security for stored credit card data, especially for small merchants. In our system stored data on merchant side actually become useless for fraudsters as there is no effective way of finding relative credit card number from a transaction code. The hash function used in generating transaction numbers can be selected considering possible threats facing each banking system and may differ in different systems.

V. Conclusion

We have proposed a new payment system based on 3D-Secure™ payment system which is enhanced with one-time-use transaction numbers; with this, we reduced the threats facing the stored sensitive credit card data on merchant side without any changes on the infrastructure. We used hash function for generating transaction codes which do not have the limitations of being in the same format with real credit card numbers, as in previously proposed methods. In our proposed system, along with security measures provided by 3D-Secure™, real cardholder data is not being stored in merchant side and therefore the threat of being compromised for this data is removed.

References


Author Biographies

F. Javani is a M.Sc. student of Information Technology at the University of Guilan, Iran. He received his bachelor’s degree in computer science in 2009. His research interests include Electronic Payment Systems, Cryptography, Data Security and e-Commerce. He may be reached at farid.djavani@gmail.com

S. Mohammadi is a former senior lecturer at the University of Derby, UK. He also used to be a Network consultant in the UK for more than fifteen years. He is currently a lecturer in the University Of Khajeh, Nasir, Iran. His main research interests and lectures are in the fields of Networking, Data Security, Network Security and e-commerce. He may be reached at mohammadi@kntu.ac.ir or smohammadi40@yahoo.com.