Cryptographically-Generated Virtual Credit Card Number for Secure Card-Not-Present Transactions

Chan-Ho Park[1] and Chang-Seop Park[2]
[1] Department of Computer Science, Dankook University
Jukjeon, Yongin, 448-701 – Republic of Korea
[e-mail: chpark6737@gmail.com]
[2] Department of Software, Dankook University
Jukjeon, Yongin, 448-701 – Republic of Korea
[e-mail : csp0@dankook.ac.kr]
*Corresponding author: Chang-Seop Park

Received October 5, 2015; revised January 28, 2016; accepted February 27, 2016; published April 30, 2016

Abstract

Card-Not-Present (CNP) transactions taking place remotely over the Internet are becoming more prevalent. Cardholder authentication should be provided to prevent the CNP fraud resulting from the theft of stored credit card numbers. To address the security problems associated with CNP transactions, the use of a virtual card number derived from the transaction details for the payment has been proposed, instead of the real card number. Since all of the virtual card number schemes proposed so far are based on a password shared between the cardholder and card issuer, transaction disputes due to the malicious behavior of one of the parties involved in the transaction cannot be resolved. In this paper, a new virtual card number scheme is proposed, which is associated with the cardholder's public key for signature verification. It provides strong cardholder authentication and non-repudiation of the transaction without deploying a public-key infrastructure, so that the transaction dispute can be easily resolved. The proposed scheme is analyzed in terms of its security and usability, and compared with the previously proposed schemes.

Keywords: CNP Transaction, CNP Fraud, Cardholder Authentication, Non-Repudiation

This research was supported by Basic Science Research Program through the National Research Foundation of Korea [NRF-2014R1A1A2055074].

http://dx.doi.org/10.3837/tiis.2016.04.022 **ISSN : 1976-7277**

1. Introduction

Card-Present (CP) transactions are ones where the cardholder and the merchant are all physically present during the payment authorization, while Card-Not-Present (CNP) transactions take place remotely over the Internet. The EMV (Europay, MasterCard and Visa) standard [1] for secure CP transactions was introduced to tackle the developing threat of magnetic strip card counterfeiting. According to a recent survey [2], the fraud level of CP transactions in the U.K. was brought down in the last few years and U.S. card companies are beginning to distribute EMV cards. An EMV transaction consists of three stages. The first is card authentication: a chip in the card proves to the merchant terminal that it is authentic. Second, cardholder verication involves the customer either entering a PIN or signing for the transaction. Third, the card produces the message authentication codes for transaction authorization; as these codes use a secret key shared between the card and the issuing bank, they can only be verified if the merchant terminal is online. Several vulnerabilities due to improper implementations have been reported [3, 4, 5].

On the other hand, criminals are moving to CNP transactions which remained beyond the scope of EMV, since using stolen and counterfeit cards has become more difficult. The Secure Electronic Transactions (SET) protocol [6] was proposed for secure CNP transactions, namely to protect credit card numbers from malicious parties, and even from merchants. Unfortunately, SET never succeeded in the marketplace because of its high overhead and additional requirement of public key infrastructure (PKI). Instead, the Secure Socket Layer (SSL) protocol has been widely used to secure the credit card information exchanged. The security services necessary for secure CNP transactions are transaction security (confidentiality and integrity) between the cardholder and merchant, the merchant (server) authentication, and the cardholder (client) authentication. The SSL is a de facto standard to provide the first two security services. Due to the characteristics of CNP transactions, the cardholder authentication is also indispensable to prevent CNP fraud using stolen credit card numbers. For cardholder authentication, several schemes based on the password have been previously proposed, including 3-D secure and variable Vcard schemes. 3-D Secure (branded by Visa as the 'Verified by Visa' and MasterCard as 'MasterCard SecureCode') [7] has been introduced to authenticate the cardholder based on a password. 3-D Secure would pop up a password entry form to a cardholder who attempted an online card payment; the cardholder would enter a password and, if it was correct, would be returned to the merchant website to complete the transaction. While it is currently deployed in some countries, some security weaknesses of 3-D Secure have been pointed out [8, 9].

As a result of CNP transactions, online merchants and credit card processing companies may collect and store the cardholder's credit card information for various purposes, including user convenience. However, the resulting database can be a fascinating target for criminals to steal credit card information, and several attacks against such databases have been reported, including a recent breach of credit card information at the discount retailer, Target [10]. There are two approaches to protect credit card information stored in merchant sites. One of them, called 'Tokenization', is promoted by the payment card industry [11]. The merchant or credit card processing company generates tokens corresponding to the credit card numbers, while the credit card information is encrypted and stored in the 'card vault'. The merchant uses the tokens for internal use only. The other involves the use of a 'virtual' credit card number [12, 13, 14, 15, 16] instead of the 'real' one. The virtual credit card number is derived from the password or secret key as well as transaction-related information, so that it is variable for each transaction.

In this paper, we propose a new virtual card number which is fixed for each transaction. Since it is derived from the cardholder's public key for the purpose of generating a transaction signature, we call it a 'cryptographically-generated' virtual credit card number. However, PKI and PKI-related concepts such as certificates and the certificate revocation list (CRL) are not needed, either to provide the non-repudiation of the transaction or to resolve transaction disputes among the cardholder, merchant and card issuer. On the other hand, the previous virtual card number schemes do not provide such properties, since a password or secret key shared between the cardholder and card issuer is used to generate the virtual card number. Another advantage of our 'fixed' virtual card number is its convenience for the merchant for internal administrative purposes, since it is a fixed value used to identify the corresponding cardholder. In Section 2, related studies on secure CNP transactions are introduced, together with the previous virtual card number schemes. The new virtual card number scheme is proposed in Section 3, and its security is analyzed and compared with the previous schemes in Section 4. Finally, concluding remarks are given in Section 5.

2. CNP Transaction Security and Related Works

A CNP transaction is defined as a transaction taking place remotely over the Internet, as shown in **Fig. 1** (a). When receiving order information (*OI*) including a transaction amount (*Amount*) and transaction number (*Tno*), the cardholder sends their real credit card number (*Rcard*) together with their name and billing address (ID_C) to the merchant through a *Payment Response* message. Then, the merchant requests the card issuer (ID_{Issuer}) to authorize the current transaction by sending an *Authorization Request* message which contains the merchant's identifier (ID_M). If *Rcard* and *Amount* are valid, the card issuer responds with an *Authorization Response* {*Accept*} message. Otherwise, *Authorization Response* {*Deny*} message is returned. Depending on the result of the authorization, a *Transaction Result* {*Receipt* or *Rejection*} message is sent to the cardholder.

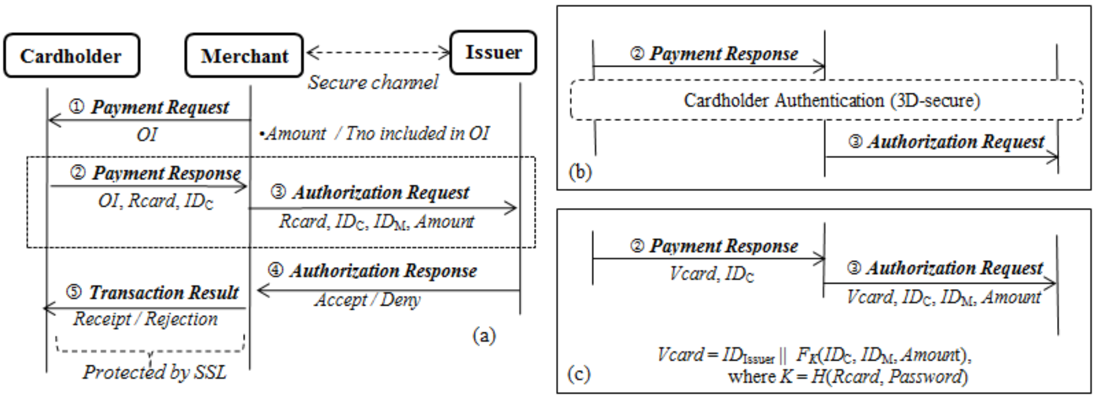

Fig. 1. CNP Transactions based on 3-D Secure and Virtual Card Number

The communication path between the cardholder and merchant is protected by SSL. Namely, the confidentiality and integrity of the transaction are assured, as well as the authentication of the merchant. However, the authentication of the cardholder is missing in **Fig. 1** (a). 3-D Secure [7] has been introduced to authenticate the cardholder based on a password shared between the cardholder and card issuer (**Fig. 1** (b)). After receiving the credit card information through the *Payment Response* message, the merchant redirects the cardholder to the card issuer and the cardholder enters his password. If verified successfully, the card issuer notifies the merchant of the successful authentication of the cardholder. Since the credit card numbers are processed and stored by the merchant, there is a risk of their being exposed to an adversary. The risk of stored credit card information being stolen is an increasing threat to CNP transactions.

Several academic solutions [12, 13, 14, 15, 16] have been proposed to mitigate the damage caused by stolen card numbers, all of which are based on the use of a ‘virtual’ credit card number (Vcard) instead of the ‘real’ credit card number (Rcard). **Fig. 1** (c) shows a CNP transaction based on Vcard from **Construction 1** proposed by [13]. It is assumed that *Password* has been pre-shared between the cardholder and card issuer.

Construction 1. *Vcard Generation* [13]

Given *Rcard* and *Password* of a cardholder (ID_C),

$K := H(Rcard \,||\, Password)$, where $H(.)$ is an one-way hash function

Given ID_{Issuer}, ID_C and ID_M,

$Vcard := ID_{Issuer} \,||\, F_K(ID_C, ID_M, Amount)$, where F_K (.) is a keyed MAC function whose output is adjusted to fit the typical credit card number.

After receiving {$Vcard, ID_C, ID_M, Amount$} via the merchant, the card issuer compares the received *Vcard* with the one computed using the *Rcard* and *Password* which have been retrieved from its database based on ID_C. It the comparison is successful, the transaction is finally authorized. Since the Vcard plays the role of the MAC value for each transaction, it is used only once and, hence, the risk of its being reused by an adversary is avoided. Several weaknesses of the Vcard protocol described in [13] are discussed in Section 4. Other solutions [12, 14, 15, 16] are similar to the one in [13] in terms of generating and using the Vcard. SecureClick proposed in [16] is not efficient, in that the Vcard is generated by the card issuer and returned to the cardholder.

3. A New CNP Transaction Scheme based on Vcard

3.1 Design Principles

The proposed method of generating a Vcard is based on the concept of the CGA (Cryptographically Generated Address) [17], which is originally used to derive an IPv6 address from the public key of a node for the purpose of binding the IPv6 address to its public key. The Vcard proposed herein is also derived from the public key, PK_C, of the cardholder, namely $Vcard = ID_{Issuer} \,||\, H_1(PK_C \,||\, ID_C \,||\, ID_{Issuer})$. Once the Vcard generated by the cardholder is registered at the card issuer, the authenticity of the public key can be verified if the Vcard can be found in the card issuer’s database, without PKI and public-key certificates. The cardholder can prove the ownership of the Vcard through the digital signature using the private key SK_C corresponding to the public key PK_C. Even if the Vcard is compromised, an adversary cannot carry out CNP transactions successfully without knowing the private key of the cardholder.

Table 1. Table of Notations

OI	Order information including *Amount* and *Tno*
$Amount$	Transaction amount
Tno	Transaction number maintained by the merchant
ID_C	Cardholder name and Billing address
ID_M, ID_{Issuer}	Merchant's and Card issuer's identifiers, respectively
$Rcard$	Real credit card number
$Vcard$	Virtual credit card number
$H_1(.)$, $H_2(.)$	Second-preimage and collision resistant hash functions
PK_C, SK_C	Public and Private keys of cardholder for signature
$Sig(SK_C, \sigma_h)$	Signature of σ_h using SK_C, where $\sigma_h = H_2(\sigma)$
$Vrf(PK_C, \sigma, Sig(SK_C, \sigma_h))$	Verification of Signature using PK_C (0 or 1)

3.2 The Proposed Scheme

Before performing CNP transactions based on the Vcard, the cardholder should generate and register it with the card issuer. Our Vcard can be obtained from **Construction 2**. The cardholder sends it securely to the card issuer. If the *Vcard* is unique in the database of the card issuer, it is registered and stored with the corresponding *Rcard*. Otherwise, the above step is repeated after choosing another pair of public and private keys. However, the probability of collision (generating a Vcard owned by another cardholder) is negligible, since the whole output of the hash function $H_1(.)$ is used as the Vcard without truncation (e.g. 160 bits in the case of SHA-1).

Construction 2. *Our Vcard Generation*

- $(PK_C, SK_C) \leftarrow \boldsymbol{Gen}_S(1^n)$, where $\boldsymbol{Gen}_S$ is a probabilistic algorithm which takes as input a security parameter 1^n and outputs a pair of public and private keys for a cardholder.

Given ID_{Issuer} and ID_C,

- $Vcard := ID_{Issuer} \,||\, H_1(PK_C \,||\, ID_C \,||\, ID_{Issuer})$, where $H_1(.)$ is a second-preimage resistant hash function.

The pair of public and private keys, PK_C and SK_C, of the cardholder is stored in his PC or smartphone, and the private key is encrypted with a password solely known to the cardholder. The cardholder can initiate a CNP transaction with a merchant, as shown in **Fig. 2**. For cardholder authentication, a signature scheme of **Construction 3** is employed. When receiving a *Payment Request* $\{OI\}$ message, the cardholder computes $\boldsymbol{Sig}(SK_C, \sigma_h)$, where $\sigma_h = H_2(\sigma)$ and $\sigma = OI \,||\, ID_C \,||\, ID_{Issuer}$, and sends a *Payment Response* $\{\sigma, \boldsymbol{Sig}(SK_C, \sigma_h), PK_C\}$ message to the merchant. *Tno* included in *OI* is a transaction number maintained by the merchant for all of the transactions associated with the merchant.

Construction 3. *Our Signature Scheme* $\boldsymbol{S} = (Gen, Sig, Vrf)$

- Gen : $(PK_C, SK_C) \leftarrow \boldsymbol{Gen}_S(1^n)$
- Sig : On input (SK_C) and a message σ, output $\boldsymbol{Sig}(SK_C, H_2(\sigma))$, where $H_2(.)$ is a collision-resistant hash function.
- Vrf : $b := \boldsymbol{Vrf}(PK_C, \sigma, \boldsymbol{Sig}(SK_C, H_2(\sigma)))$.
  If $b = 1$, the signature is valid. Otherwise, it is invalid.

The merchant first verifies $\boldsymbol{Sig}(SK_C,\ \sigma_h)$ using the public key PK_C of the cardholder. However, the merchant cannot verify the authenticity of PK_C at this point. It will be checked through the card issuer during the authorization stage. If the verification of $\boldsymbol{Sig}(SK_C,\ \sigma_h)$ is NOT successful, namely $\boldsymbol{Vrf}(PK_C, \sigma, \boldsymbol{Sig}(SK_C, \sigma_h)) = 0$, the transaction fails and a *Transaction Result* {*Rejection*} message is sent to the cardholder. Otherwise, the merchant requests the card issuer to authorize the current transaction. For this purpose, the merchant computes $H_1(PK_C \,||\, ID_C \,||\, ID_{Issuer})$ based on the information provided by the cardholder, and generates $Vcard = ID_{Issuer} \,||\, H_1(PK_C \,||\, ID_C \,||\, ID_{Issuer})$. Then, an *Authorization Request* {$Vcard$, ID_C, ID_M, $Amount$, σ_h} message is sent to the card issuer. At this stage, the card issuer verifies the authenticity of PK_C through the *Vcard*. If the *Vcard* is found in the card issuer's database, its authenticity is confirmed and then the remaining procedure, such as authorizing *Amount*, is processed as usual.

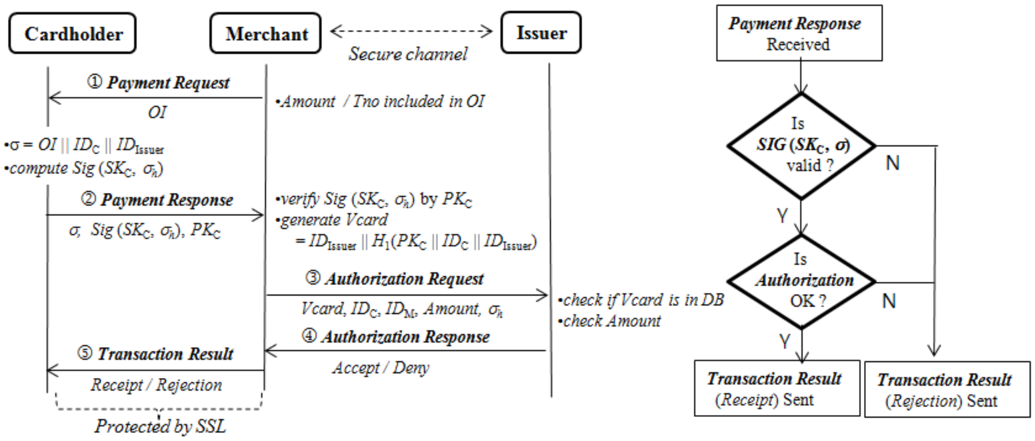
Cardholder
Merchant
Secure channel
Issuer
① Payment Request
OI
•Amount / Tno included in OI
•σ = OI || ID_C || ID_{Issuer}
•compute Sig (SK_C, σ_h)
② Payment Response
σ, Sig (SK_C, σ_h), PK_C
•verify Sig (SK_C, σ_h) by PK_C
•generate Vcard
= ID_{Issuer} || H_1(PK_C || ID_C || ID_{Issuer})
③ Authorization Request
Vcard, ID_C, ID_M, Amount, σ_h
•check if Vcard is in DB
•check Amount
④ Authorization Response
Accept / Deny
⑤ Transaction Result
Receipt / Rejection
Protected by SSL
Payment Response Received
Is SIG (SK_C, σ) valid ?
N
Y
Is Authorization OK ?
N
Y
Transaction Result (Receipt) Sent
Transaction Result (Rejection) Sent

Fig. 2. Proposed CNP Transaction based on Vcard

If the verification is successful, the card issuer replies with an *Authorization Response* {*Accept*} message. Subsequently, a *Transaction Result* {*Receipt*} message is sent to the cardholder, to notify them that the current transaction was successful and provide them with a receipt for it. Both the merchant and the card issuer keep this transaction in their transaction log database. On the other hand, if the *Vcard* received does not exist in the card issuer's database, which means that it is not valid or registered, the card issuer rejects the transaction by sending an *Authorization Response* {*Deny*} message.

4. Analysis and Comparisons

Unlike the previously proposed Vcards [12, 13, 14, 15, 16] which are variable at each transaction, the proposed Vcard is a fixed virtual credit card number which is transaction-independent, since the transaction details such as the transaction amount and merchant's ID are not used to generate the Vcard. So, from now on, the previously proposed Vcards are denoted as "variable Vcards", while the proposed Vcard is denoted as the "fixed Vcard.

4.1 Security Services for Secure CNP Transaction

The security services necessary for secure CNP transactions are *transaction security* (confidentiality and integrity) between the cardholder and merchant, the *merchant* (*server*) *authentication*, and the *cardholder* (*client*) *authentication*. The SSL is a de facto standard to provide the first two security services. However, for cardholder authentication, several schemes based on the password have been previously proposed, including 3-D secure and variable Vcard schemes. On the other hand, our proposed fixed Vcard scheme is based on the digital signature for the cardholder authentication. Nevertheless, the PKI is not needed for the issuance and verification of the public key certificate, since the authenticity of the public key can be provided by the fixed Vcard registered at the card issuer. The "registered" fixed Vcard plays the role of public key certificate for the cardholder without deploying an additional PKI, so that it provides strong cardholder authentication as well as non-repudiation.

4.2 Forgery Attack against Our Proposed Fixed Vcard

The security for the cardholder authentication of our proposed fixed Vcard scheme is based on the infeasibility of forging any registered Vcard in the card issuer's database. Forging a registered Vcard means that an adversary finds any pair of public and private keys (PK_C', SK_C') such that $H_1(PK_C' \| ID_C \| ID_{\text{Issuer}}) = H_1(PK_C \| ID_C \| ID_{\text{Issuer}})$ and $PK_C' \neq PK_C$, where $H_1(PK_C \| ID_C \| ID_{\text{Issuer}})$ is a part of a target Vcard. If such a forgery is successful, the adversary can disguise the victim cardholder. The success possibility of such a forgery attack is related to the length of the fixed Vcard. If $|H_1(.)| = l(n)$ for a polynomial l and a security parameter n, the adversary would have to perform almost $2^{l(n)}$ computations to find a public key whose corresponding fixed Vcard matches the target fixed Vcard. Anybody can generate an arbitrary fixed Vcard from any pair of public and private keys. However it cannot be used unless it is registered at the card issuer in advance.

For a formal proof, a hash function is defined as a pair of probabilistic polynomial-time algorithms $\mathbf{\Pi_1} = (\boldsymbol{Gen}_{\mathbf{H}}, H_1^S)$, where $\boldsymbol{Gen}_{\mathbf{H}}$ is a probabilistic algorithm which takes as input a security parameter 1^n and outputs a key s. $H_1^S : \{0, 1\}^* \rightarrow \{0, 1\}^{l(n)}$ is a keyed hash function, where s is a *known* key. The following experiment is defined for $\mathbf{\Pi_1}$, an adversary $\mathcal{A}$, and a security parameter n.

The second-preimage finding experiment $\mathbf{2PR}_{\Pi_1,\mathcal{A}}(n)$

- $s \leftarrow \boldsymbol{Gen}_{\mathbf{H}}(1^n)$.
- The adversary $\mathcal{A}$ is given (s, x) and outputs x', where $x, x' \in \{0, 1\}^*$.
- $\mathbf{2PR}_{\Pi_1,\mathcal{A}}(n) = 1$ if and only if $x \neq x'$and $H_1^S(x) = H_1^S(x')$.

Definition 1. A hash function $\mathbf{\Pi_1} = (\boldsymbol{Gen}_{\mathbf{H}}, H_1^S)$ is a second-preimage resistant if for all probabilistic polynomial-time adversaries $\mathcal{A}$ there is a negligible function *negl* (.) such that $\Pr[\mathbf{2PR}_{\Pi_1,\mathcal{A}}(n) = 1] \leq negl(n)$.

Theorem 1. If $\mathbf{\Pi_1} = (\boldsymbol{Gen}_{\mathbf{H}}, H_1^S)$ is a second-preimage resistant hash function, then Construction 2 is secure against a forgery attack.

Proof. Let $\mathbf{\Pi_1}'$ denote Construction 2, and let $\mathcal{B}$ be a PPT adversary attacking $\mathbf{\Pi_1}'$ for the purpose of forging a Vcard of a particular cardholder ID_C. So, we define the following experiment:

The Vcard forging experiment $\mathbf{Vforge}_{\Pi_1',\mathsf{B}}(n)$

- The adversary B is given $(s, PK_C, ID_C, ID_{\text{Issuer}})$ and outputs PK_C'.
- $\mathbf{Vforge}_{\Pi_1',\mathsf{B}}(n) = 1$ if and only if $PK_C' \neq PK_C$ and
  $H_1^S(PK_C' \,\|\, ID_C \,\|\, ID_{\text{Issuer}}) = H_1^S(PK_C \,\|\, ID_C \,\|\, ID_{\text{Issuer}})$.

Consider the following adversary A attacking $\mathbf{\Pi_1}$ in $\mathbf{2PR}_{\Pi_1,\mathsf{A}}(n)$ using adversary B as a subroutine.

Adversary A:

- On input (s, x), where $x = PK_C \,\|\, ID_C \,\|\, ID_{\text{Issuer}}$.
- $z := x$.
- Run B (s, z).
  B returns $z' = PK_C' \,\|\, ID_C \,\|\, ID_{\text{Issuer}}$, where $PK_C'(\neq PK_C)$
- $x' := z'$.
- A outputs x'.

It is evident that $\Pr[\mathbf{2PR}_{\Pi_1,\mathsf{A}}(n) = 1] \geq \Pr[\mathbf{Vforge}_{\Pi_1',\mathsf{B}}(n)]$. Since $H_1^S(.)$ is a second-preimage resistant hash function, $\Pr[\mathbf{2PR}_{\Pi_1,\mathsf{A}}(n) = 1] \leq negl(n)$. We conclude that $\Pr[\mathbf{Forge}_{\Pi_1',\mathsf{B}}(n)] \leq \Pr[\mathbf{2PR}_{\Pi_1,\mathsf{A}}(n) = 1] \leq negl(n)$, namely Construction 2 is secure against a forgery attack. ■

ID_C is mandatory to compute $H_1(PK_C \,\|\, ID_C \,\|\, ID_{\text{Issuer}})$ for security reason. Without it, an adversary's attack strategy can be extended to target any cardholder instead of a particular cardholder. Namely, the adversary tries to find any pair of public and private keys (PK_C', SK_C') such that $H_1(PK_C' \,\|\, ID_{\text{Issuer}}) = H_1(PK_C \,\|\, ID_{\text{Issuer}})$ and $PK_C' \neq PK_C$. Its role is similar to salting password to increase the difficulty of a password-guessing attack.

4.3 Signature Forgery Attack

As mentioned in the previous section, if an adversary can find (PK_C', SK_C') which can be used to generate the target Vcard, the adversary can disguise the victim cardholder. On the other hand, if the adversary can forge a signature for the *Payment Response* message with respect to the signature scheme $\boldsymbol{S}$ of **Construction 3**, the impersonation attack can also be successful. Let $\boldsymbol{S}' = (Gen', Sig', Vrf')$ denote a signature scheme such that:

- Gen' : $(PK_C, SK_C) \leftarrow \boldsymbol{Gen}_S(1^n)$
- Sig' : On input (SK_C) and a message σ, output $\boldsymbol{Sig}(SK_C, \sigma)$,
- Vrf': $b := \boldsymbol{Vrf}(PK_C, \sigma, \boldsymbol{Sig}(SK_C, \sigma))$. If $b = 1$, the signature is valid.

As in **Definition 1**, a collision-resistant hash function of **Construction 3** is defined as $\mathbf{\Pi_2} = (\boldsymbol{Gen}_{\mathbf{H}}, H_2^S)$, where $H_2^S : \{0, 1\}^* \rightarrow \{0, 1\}^{l(n)}$.

Theorem 2. If $\boldsymbol{S}'$ is *existentially unforgeable under an adaptive chosen-message attack* (or *secure*), then $\boldsymbol{S}$ of **Construction 3** is also *secure*.

Proof. Suppose an adversary has forged a signature on a message σ^* such that $\sigma^* \notin \Omega$, where Ω is a set of messages the adversary has used for quering H_2^S as well as $\boldsymbol{Sig}$ oracle. There are two cases. First, there is a message $\sigma \in \Omega$ such that $H_2^S(\sigma) = H_2^S(\sigma^*)$. Namely, the adversary has found a collision in H_2^S, which contradicts the collision resistance of $\mathbf{\Pi_2}$. Second, for every $\sigma \in \Omega$, $H_2^S(\sigma) \neq H_2^S(\sigma^*)$. In this case, the adversary has forged a valid signature on the new message $H_2^S(\sigma^*)$ with respect to the signature scheme $\boldsymbol{S}'$. This also contradicts the assumption that $\boldsymbol{S'}$ is secure. ■

4.4 Insider Attacks and Resolution of Transaction Disputes

When a transaction dispute occurs among the cardholder, merchant and card issuer, each party has to provide evidence demonstrating their non-liability. In our proposed Vcard scheme, since cardholder's signature is used for cardholder authentication and non-repudiation of the transaction, the transaction dispute can be easily resolved. Suppose a cardholder disputes a transaction log in the card issuer's database. There are three cases: first, the cardholder may be dishonest. However, since the signature is generated and stored during the transaction, the signature is clear evidence that he has initiated the transaction.

Second, suppose a malicious employee $\boldsymbol{E}$ of the merchant requests the card issuer to authorize a transaction $t^* = \{Vcard, ID_{\mathrm{C}}, ID_{\mathrm{M}}, Amount^*, \sigma_h^*\}$ which has not been initiated by a legitimate cardholder. If such misbehavior occurs, the cardholder and card issuer request the merchant to adduce the signature $\boldsymbol{Sig}(SK_{\mathrm{C}}, \sigma_h^*)$ corresponding to t^* as proof of the legitimate transaction, where $\sigma_h^* = H_2^S(\sigma^*)$ and $\sigma^* = OI^* \| ID_{\mathrm{C}} \| ID_{\mathrm{Issuer}}$. Let $T = \{ (\sigma^{(i)}, \boldsymbol{Sig}(SK_{\mathrm{C}}, \sigma_h^{(i)})) \mid \sigma^{(i)} = OI^{(i)} \| ID_{\mathrm{C}} \| ID_{\mathrm{Issuer}}$ and $i = 1, 2, 3, \dots \}$ denote a set of previous transactions initiated by a cardholder. If $\boldsymbol{E}$ can find σ^* such that $\sigma_h^* = \sigma_h^{(i)}$ and $(\sigma^{(i)}, \boldsymbol{Sig}(SK_{\mathrm{C}}, \sigma_h^{(i)})) \in T$ for some i, $\boldsymbol{E}$ can adduce $\boldsymbol{Sig}(SK_{\mathrm{C}}, \sigma_h^{(i)})$ as proof for $\sigma^* = OI^* \| ID_{\mathrm{C}} \| ID_{\mathrm{Issuer}}$. However, since $H_2^S(.)$ is a collision resistant hash function which is also second-preimage resistant, the success probability of $\boldsymbol{E}$ is negligible. On the other hand, suppose $\boldsymbol{E}$ replays the previously authorized transaction again. In this case, $(\sigma^{(i)}, \boldsymbol{Sig}(SK_{\mathrm{C}}, \sigma_h^{(i)})) = (\sigma^{(j)}, \boldsymbol{Sig}(SK_{\mathrm{C}}, \sigma_h^{(j)}))$ for $i \neq j$. However, this replay attack cannot be successful since $Tno^{(i)}$ in $\sigma^{(i)}$ is the same as $Tno^{(j)}$ in $\sigma^{(j)}$.

Third, suppose a transaction has been created accidentally or maliciously by an employee of the card issuer. In this case, in order for the merchant to be paid, the merchant should provide the corresponding signature. However, since the signature cannot be forged as shown in Section 4.3, the dispute can be resolved.

4.5 Security and Usability Comparisons

In this section, we compare the variable Vcard [13] and our proposed fixed Vcard in terms of their security and usability. First, it is assumed in the variable Vcard scheme that a password has been shared between the cardholder and card issuer. As shown in **Fig. 1** (c), the variable Vcard, $Vcard = ID_{\mathrm{Issuer}} \| F_K(ID_{\mathrm{C}}, ID_{\mathrm{M}}, Amount)$, does not have to be kept secret as long as the *Rcard* and *Password* used to compute $K = H(Rcard \| Password)$ are not exposed. However, suppose a malicious employee of the card issuer illegally accesses the cardholder's *Rcard* and *Password* as well as ID_{C}. If a transaction is illegally created by the employee, the cardholder cannot prove that he is not associated with it. On the other hand, the fixed Vcard is secure against such an insider attack, as discussed in Section 4.4, since there is no secret shared between the cardholder and card issuer. Furthermore, an *off-line* password-guessing attack can be mounted against the variable Vcard. Even though *Rcard* is proposed as a secret input when computing $K = H(Rcard \| Password)$, *Rcard* is not usually considered as a secret value. Hence, given $(Rcard, Vcard, ID_{\mathrm{Issuer}}, ID_{\mathrm{C}}, ID_{\mathrm{M}}, Amount)$, the following attack can be tried. Most of cardholders choose 8 characters with a combination of numbers and lower-case letters. Then, there are $(36)^8 \approx 2.10^{12}$ possible passwords, which corresponds to about 40 bits of entropy. After computing $K^* = H(Rcard \| Password^*)$ and $Vcard^* = ID_{\mathrm{Issuer}} \| F_{K^*}(ID_{\mathrm{C}}, ID_{\mathrm{M}}, Amount)$ for each candidate $Password^*$, it can be checked if $Vcard^* = Vcard$.

Second, the variable Vcard scheme is not secure against replay attacks, since no nonce values are employed to compute $Vcard = ID_{\mathrm{Issuer}} \| F_K(ID_C, ID_M, Amount)$, where *Amount* is not a nonce value. Hence, an adversary can replay the *Payment Response* {*Vcard*, ID_C} message in **Fig. 1** (c), which will be successfully processed. Even though the transaction between the cardholder and merchant is protected by SSL, a malicious employee of the merchant can perform such an insider attack. Consider *Tno* (in *OI*) to be used to compute Vcard as $Vcard' = ID_{\mathrm{Issuer}} \| F_K(ID_C, ID_M, Amount, Tno)$. However, since *Tno* is generated and maintained not by the card issuer, but by the merchant, the security of the variable Vcard scheme [13] cannot be guaranteed just as it is.

Third, in the case of the variable Vcard scheme, there is no difference in the card number format between the Rcard and Vcard, which means that there is a possibility for the Vcard generated by the cardholder to match either the Rcard or Vcard in the card issuer's database. In this case, the card issuer should request the cardholder to generate a new Vcard, which induces an additional transaction delay and causes unnecessary inconvenience to the cardholder. On the other hand, in the case of the proposed fixed Vcard scheme, a pair of Rcard and Vcard is assigned to each cardholder in advance, so that the problem of collisions does not occur when the transaction is being processed.

Fourth, the variable Vcard scheme uses ID_C, namely cardholder's name and billing address, to identify a particular cardholder, instead of Vcard or Rcard. However, the billing address is not fixed, and the billing address the cardholder gives a merchant could differ from what the card issuer has on its database. Reasons range from recent moves or smaller card issuing banks that use third-party services who can't keep the records up to date, to frequently traveling cardholders who use a family member's address as the address of record. A main reason the variable Vcard scheme uses ID_C to identify a particular cardholder is that the Vcard generated at each transaction is distinct and the Rcard cannot be used for security reasons. This places a big burden on the merchant as well as the card issuer who have to maintain the Vcards based on the cardholder's name and billing address, ID_C. On the other hand, the Vcard in the proposed scheme is a fixed value, so the merchant can maintain and process it as usual. **Table 2** shows a security comparison between the Variable Vcard scheme [13] and proposed fixed Vcard scheme.

Table 2. Security Comparison

	Variable Vcard Scheme	***Fixed Vcard Scheme***
Basic Transaction Security	*provided by SSL*	*provided by SSL*
Cardholder Authentication	*Password*	*Signature*
Insider Attacks (Type 1 and Type 2)	*Insecure due to shared Password and replay Attack*	*Secure with Signature*
Cardholder Identifier	*Name and Billing address*	*Fixed Vcard*
Cryptographic Operations	*1 hash and 1 MAC*	*1 hash and 1 signature generation (verification)*
Other Security Issues	*Off-line password-guessing attack is feasible for most cardholder's password*	*PKI is not needed for signature generation and verification*

- *Basic Transaction Security: confidentiality, integrity and merchant authentication*
- *Insider Attack Type 1: Insider attacks from the merchant*
- *Insider Attack Type 2: Insider attacks from the card issuer*

4.6 Theft of Stored Card Number and Tokenization

Besides the risk of exposure of card numbers during transit, there is also the risk of their exposure while they are stored on a merchant's website. There are several reasons why the merchant stores credit card numbers. Besides the cardholder’s convenience, they can be used for chargeback or analyzing the cardholder’s purchase patterns. As in the case of a one-time password, the variable Vcard was designed to prevent the exposure of card numbers, which is why the variable Vcard is also called a one-time card number [14], transaction number [15] or disposable card number [16]. On the other hand, the proposed fixed Vcard is an invariable card number which can be reused at each transaction. Nonetheless, even if the fixed Vcard is stolen while stored on the merchant’s website, the adversary cannot use it successfully without knowing the cardholder’s private key. The main purpose of ‘Tokenization’ is for the merchant to use a single token corresponding to the cardholder’s Rcard to identify each cardholder, while the Rcard is encrypted and stored in a safe place. In this sense, the concept of ‘Tokenization’ is intrinsic to the proposed fixed Vcard scheme, so that the merchant using the fixed Vcard scheme does not have to expand their system or make a business contract with the third-party for the purpose of implementing ‘Tokenization’.

4.7 Deployment Issue

Our proposed fixed Vcard scheme can be deployed within the current card processing infrastructure without modifying the card issuer’s processing module. For each card issuer, a special 16-digit Rcard, viz. “XXXX-XX00-0000-0000” is reserved for the proposed fixed Vcard scheme, where “XXXX-XX” is the card issuer’s identifier. The merchant provides an option for the cardholder to choose a fixed Vcard scheme as their payment method. When the cardholder decides to purchase something and presses the payment button, a plug-in on the merchant’s web page is executed and the cardholder enters his password to activate the private key for the purpose of signing the current transaction. When the merchant sends an *Authorization Request* message to the card issuer, no additional fields are needed in the message. The fixed Vcard can be packed into the Billing Address field. This field is optionally used to request the AVS (Address Verification Service) from the card issuer, for the purpose of verifying the cardholder’s address. Since the fixed Vcard scheme is proposed for strong cardholder authentication, the AVS is not needed with the fixed Vcard scheme.

After receiving the message, the card issuer checks if the card number is “XXXX-XX00-0000-0000”. If the propose fixed Vcard scheme is used as a payment option, the corresponding transaction is processed as described in Section 3.2. Otherwise, the transaction is processed as usual. Therefore, the card issuer does not have to modify the existing processing module, but adds a new module to process the transaction based on the fixed Vcard, together with a branching point to differentiate between Rcard and Vcard. The card issuer’s DB table for the cardholders is also altered to add a column for the fixed Vcard.

5. Conclusion

CNP transactions are becoming more prevalent and, therefore, cardholders are becoming more concerned about credit card fraud, due to the lack of cardholder authentication in the current CNP transactions. Besides ‘3-D Secure’ for cardholder authentication and ‘Tokenization’ for securing the stored card number, several CNP transaction schemes based on a virtual card number have been proposed so far. In this paper, we proposed a new virtual card number for secure CNP transactions. Unlike in the previous virtual card number schemes which are based on a password shared between the cardholder and card issuer, the virtual card number in the proposed scheme is cryptographically generated in that it is derived from the cardholder's public key for signature verification. Hence, non-repudiation and dispute resolution for transactions already processed, which were not included in the previous schemes, can be provided. Furthermore, since the concept of 'Tokenization' is intrinsic to the proposed virtual card number, no extra facility is needed to implement 'Tokenization' for the merchant. We believe that the proposed scheme can be a promising candidate to supplement the current CNP transactions.

References

[1] EMVCo, "EMV - Integrated Circuit Card specifications for payment systems," ver. 4.2, 2008. Article (CrossRef Link)

[2] R. Anderson and S. J. Murdoch, "EMV: Why payment systems fail," *Communications of the ACM*, vol. 57, no. 6, pp. 24-28, June 2014. Article (CrossRef Link)

[3] M. Bond, M. O. Choudary, S. J. Murdoch, S. Skorobogatov and R. Anderson, "Be Prepared: The EMV Preplay Attack," *IEEE Security & Privacy*, vol.13, no. 2, pp. 56-64, Mar.-Apr. 2015. Article (CrossRef Link)

[4] M. Bond, O. Choudary, S. J. Murdoch, S. Skorobogatov and R. Anderson, "Chip and Skim: Cloning EMV Cards with the Pre-play Attack," in *Proc. of 2014 IEEE Symposium on Security and Privacy*, pp. 49-64, May 2014. Article (CrossRef Link)

[5] S. J. Murdoch and R. Anderson, "Security protocols and evidence: Where many payment systems fail," *Financial Cryptography and Data Security*, LNCS 8437, pp. 21-32, Mar. 2014. Article (CrossRef Link)

[6] Mastercard and Visa, "SET: Secure Electronic Transaction specification," ver. 1.0, 1997.

[7] Visa International Service Association, "3-D Secure protocol specification: core functions," ver. 1.0.2, July 2002.

[8] M. Assora and A. Shirvani, "Enhancing the security and efficiency of 3-D Secure," *Information Security*, LNCS, vol. 4176, pp. 489-501, 2006. Article (CrossRef Link)

[9] S. J. Murdoch and R. Anderson, "Verified by Visa and Master Card Secure Code: or, How not to design authentication," *Financial Cryptography*, LNCS, vol. 6052, pp. 336–342, 2010. Article (CrossRef Link)

[10] G. Wallace, J. Pepitone, J. O'Toole, C. Isidore, J. Pagliery and J. Johns, "Target: 40 million credit cards compromised," *CNN Money*, Dec. 19. 2013.

[11] PCI Security Standard Council, "Information Supplement: PCI DSS Tokenization Guidelines," *PCI Data Security Standard*, 2011. Article (CrossRef Link)

[12] Y. Li and X. Zhang, "Securing credit card transactions with one-time payment scheme," *Electronic Commerce Research and Applications*, vol. 4, pp. 413-426, 2005. Article (CrossRef Link)

[13] I. Molloy, J. Li and N. Li, "Dynamic virtual credit card numbers," *Financial Cryptography and Data Security*, LNCS, vol. 4886, pp 208-223, 2007. Article (CrossRef Link)

[14] F. Buccafurri and G. Lax, "A light number-generation scheme for feasible and secure credit-card-payment solutions," *E-Commerce and Web Technologies*, LNCS, vol. 5183, pp 11-20, 2008. Article (CrossRef Link)

[15] F. Javani and S. Mohammadi, "A new credit card payment system based on 3D-Secure using one-time-use transaction numbers," *Information Assurance and Security Letters*, vol. 1, pp. 60-65, 2010. Article (CrossRef Link)

[16] A. Shamir, "SecureClick: A Web payment system with disposable credit card numbers," *Financial Cryptography*, LNCS, vol. 2339, pp. 196-209, 2002. Article (CrossRef Link)

[17] T. Aura, "Cryptographically Generated Addresses (CGA)", *RFC 3972*, Mar. 2005 Article (CrossRef Link)

Chan-Ho Kim has received his B.Sc. in Multimedia Engineering from Dankook University in 2013 and has received his M.Sc. in computer science in 2016. His research interests include network security, financial security, and information security.

Chang-Seop Park has been with the Department of Computer Science at Dankook University, Republic of Korea, since 1990. He has a Ph.D. and a M.Sc. from Lehigh University (1990 and 1987), as well as a B.A. from Yonsei University (1983). He has been working on the wireless mobile network security during the last 5 years. His research interests include network security, cryptographic protocols, and coding theory.