An Optimized Credit-Payment System for Expressway Toll Collection Systems

Goichiro HANAOKA[1], Tsuyoshi NISHIOKA[2],
Yuliang ZHENG[3] and Hideki IMAI[1]

[1]The 3rd Department, Institute of Industrial Science, the University of Tokyo
Roppongi 7-22-1, Minato-ku, Tokyo 106-8558, JAPAN
Phone & Fax: +81-3-3402-7365
E-Mail: hanaoka@imailab.iis.u-tokyo.ac.jp
imai@iis.u-tokyo.ac.jp
[2]ADVANCE CO., LTD.
Koremasa 10-20-17, Fuchu-shi, Tokyo, 183-0014, JAPAN
E-Mail: nishioka@advance.co.jp
[3] School of Comp. & Info. Tech., Monash University,
McMahons Road, Frankston, VIC 3199, AUSTRALIA
E-Mail: yzheng@fcit.monash.edu.au

Abstract

Credit-based electronic payment systems are considered to play important roles in future automated payment systems. Like most other types of payment systems, however, credit-based systems proposed so far generally involve computationally expensive cryptographic operations. Such relatively heavy computational load is preventing credit-based systems from being used in applications that require very fast processing. A typical example is admission-fee payment at the toll gate of an expressway without stopping a vehicle that travels at a high speed. In this article, we propose a very fast credit-based electronic payment protocol for admission-fee payment. More specifically, we propose a payment system in which communications between a high-speed vehicle and a toll gate consist of only very simple and efficient computations. The proposed system makes use of an optimized Key Pre-distribution System (or KPS) to obtain high resistance against collusion attacks.

Key words: Credit-Based Payment, Expressway Toll Collection System, Key Predistribution System, ID-Based Cryptosystem, Collusion Attack

1 Introduction

In many countries, drivers are charged for using highways with their cars. Up to now, most toll gates are operated manually, slowing down traffic significantly. Increasing traffic density in Japan already caused traffic jams in front of the toll gates. To solve these problems, manual toll gates should be replaced by electronic toll gates. Then payment can be carried out while the car is passing the gate, without stopping. Of course this requires that the whole transaction can be carried out in a very short period of time, typically 100ms[6]. A second aspect of Expressway Toll Collection (ETC) systems is the security of the financial transactions. This can be guaranteed by public-key cryptography (PKC). It is well-known, however, that PKC is computationally very expensive, and this conflicts with the requirement of a short processing time[1]. Therefore, ETC systems developed so far are all based on prepaid cards. With credit cards, toll collection also requires an inquiry at the credit card company, what needs additional computations and consumes time. Since both computational complexity and time are scarce in ETC systems, credit cards could not gain acceptance for this. However, credit cards have many advantages. In particular, they are already very widespread (e.g. 465million VISA cards and 300 million Master Card cards) and they can be used in shops or restaurants as well as on the internet. It is near at hand that such an universal means of payment should be usable for ETC as well. The high availability of credit cards and their uncomplicated use will help to establish ETC systems.

In this article, we propose a light-weight credit-based payment protocol that does not require public-key encryption/decryption during communications. Therefore, in the gates, cheap computers with a low performance are sufficient, and the user's device can be realized as an ordinary IC card. The proposed protocol is based on the *Key Predistribution System* (KPS), since its low computational complexity permits to process the toll collection within the required time. Additionally, the KPS does not need any prior communications. Using information that uniquely identifies a single user, e.g. the car's number plate, directly within the KPS secret algorithm allows to detect a user's illegal behavior and to protect a user's privacy simultaneously. We show also how to optimize the KPS for our payment system to obtain a high resistance against collusion attacks. For a typical security parameter setting, the collusion threshold of the optimized KPS is 32 times higher than that of the conventional KPS while using the same amount of memory as the KPS center. The memory required by the user is even reduced by the optimization.

2 Credit-Based Payments for ETC

2.1 ETC and Credit-based payments

In ETC, users can complete their payment for their use of the road simply by passing the toll gate. Technologies for ETC are regarded to be quite important in terms of efficiency of traffic. Namely, if we can realize the safe communication between toll gates and cars within limited time for communications, traffic on the road can work much more smoothly.

Although such systems have already been realized using prepaid-based payments, their functions are not enough. Usually, prepaid cards are emitted not as general-purpose cards but limited to certain systems. It will be hard to convince users of ETC systems when they have to use separate prepaid cards for each toll gate operation. Further without a

[1] We can deal with this problem by setting up another toll gate before the “real” gate. However, such method requires more cost and restricts flexibility of the structure of the payment system.

general-purpose prepaid card system, procedures to join such systems must specified for each system individually. Then, however, users might be reluctant to join ETC systems. These disadvantages might bring problems in the ETC.

However, credit-based payments are available for many purposes. And, since there are already a lot of credit-card holders in the world, they can easily join the system. Hence, if credit-based payment can be realized in ETC, the system will be more efficient.

Although credit-based payments have a number of advantages over other payment systems in terms of its simplicity, openness and so on, there seems to be a consensus among both researchers and practitioners regarding the relative inefficiency of the protocol. Namely, since messages in credit-based payments consist of simple contents, they must be sent with high authenticity and confidentiality by using cryptographical techniques. Conventional credit-based payment systems (e.g. SET[2, 3, 4], CyberCash[5]) use public-key cryptosystems for this purpose. As well known, public-key cryptosystems require a large amount of computation time. Thus, when a conventional credit-based payment system is applied to ETC straightforwardly, it seems to be difficult to finish the communication between a toll gate and a car while the car passes the gate. Furthermore, toll gates also have to communicate with the credit company during the communications. The total time for communications is estimated to be 100ms. There has already been an attempt to solve this problem by using new cryptographical techniques such as elliptic curve cryptosystems or signcryption[17]. These technologies make the credit-payment systems much more efficient[7]. Nevertheless, their performance is considered to be too low to work effectively in ETC; so still computers with high performance are required even if these technologies are used. In this article, we propose an optimized credit-based payment system for ETC taking these requirements into account.

2.2 Requirements for ETC

In order to carry out credit-based payment system for ETC efficiently, some requirements must be fulfilled. These are shown below:

Requirement 1. A users' computational power is assumed to be low.

Requirement 2. The computational power of the toll gates is also assumed to be low.

Requirement 3. Communications should be limited to a number as small as possible.

Requirement 4. Messages between users and toll gates must be kept secret and authenticated.

Requirement 5. Users' privacy should be protected if possible.

Typically in ETC, the available time for processing a payment is limited to approximately 100 ms in total. Extensive computations take a lot of time and therefore attention must be paid to requirements 1 and 2. This holds in particular for the car, where we can only assume computers with low performance such as IC cards. But also for the toll gates no computers with high computation power are expected because this reduces the costs for the equipment. **Requirement 3.** is also a consequence of the strictly limited time for communications. **Requirement 4.** and **5.** are usual requirements in many payment systems. When the system is constructed to be able to detect users' illegal behavior more easily, users' privacy is also revealed more easily. Such tradeoff can be regarded as the general problem in all of the electronic payment systems. In our system, of course, we have to consider it carefully.

The cryptographical algorithms and protocols at present allow to fulfill these requirements only with prepaid cards. Elliptic curve cryptosystems are 10 times faster than RSA, but still they are to slow to make contactless payments with credit cards feasible. Therefore here a different approach is proposed based on the KPS. This allows to implement credit–based ETC using IC cards.

2.3 Properties of ETC

ETC possesses some useful properties. Our optimization of credit-based payment for ETC is based on them.

Property 1. Payment procedures are executed when car and toll gate meet.

Property 2. The users' cars can be clearly identified by unique information (e.g. number plates, shapes, colors and so on) .

Property 3. All the users that passed an entrance toll gate also have to pass an exit toll gate.

Property 1. and **2.** indicate that toll gates can obtain the unique information of users that want to use the road operated by the toll gates. Since these users' unique information can be regarded as their identifiers, we can apply an ID-Based key cryptosystem to ETC. Assuming that users' personality is not detected by using the users' unique information, the users' privacy can be protected. Besides, **Property 3.** indicates that toll gates have extra time to detect a user's illegal behavior that could not be detected at the entrance toll gate. If the illegal behavior of a user is detected while the user is being on the road, he can be stopped when passing the exit toll gate.

3 Key Predistribution System

3.1 Suitable Cryptosystem for ETC

By using suitable cryptographical primitives, it is possible to fulfill all the requirement stated in 2.2. As mentioned in 2.3, we can apply an ID-Based cryptosystem to ETC and still protect the users' privacy.

The concept of ID-Based key cryptosystems was originally proposed by Shamir[9, 10]. Maurer and Yacobi proposed an ID-Based key distribution scheme with following Shamir's concept [11, 12]. However, their scheme requires a quite huge computational power. Okamoto and Tanaka[13] also proposed a key-distribution scheme based on a user's identifier, but it requires previous communications between a sender and a receiver to share their employed key. Although Tsujii and others proposed several ID-Based key-distribution schemes[14, 15], almost all of them have been broken[16]. These schemes does not seem to fulfill the requirements mentioned in 2.2. Blom's ID-Based key-distribution scheme[8], however, does not have serious problems when applied to ETC. The Key Predistribution System (KPS) proposed by Matsumoto and Imai[1] is known as the generalized version of Blom's scheme. In the following subsections, we give a brief review of the KPS.

3.2 Properties of KPS

The KPS has three remarkable properties. First, there is no need to send messages for the key distribution between the entities who will make a cryptographic communication. Second, its key-distribution procedure consists of simple calculations so that its computational cost is quite small. Finally, in order to share the key, a participant should only input its partner's identifier to its KPS secret algorithm. Thus, when the KPS is utilized, the computational performance can be set up to be low and the number of communications between sender and receiver can be limited to a small number. Hence, by applying the KPS efficiently, the requirements for ETC can be met. However, the KPS has a certain collusion threshold; when more users cooperate they can calculate the authority's secret information. Hence, the KPS cannot be applied to ETC in a straightforward manner.

3.3 A Brief Review of KPS

In the KPS, all users are given an individual secret algorithm by the KPS center. Any pair of users can share a common key simply by putting the partner's identifier into their secret algorithms. This subsection introduces how the users' secret algorithms are produced and how users share a common key.

Let the n-dimensional vectors x_A and x_B be the effective IDs of entities A and B, respectively. The $n \times n$ symmetric matrices $G^{(\mu)}$ ($\mu = 1, \cdots, h$) are called the KPS-center algorithm. The $G^{(\mu)}$s are produced by the KPS center and kept secret to all other entities. $G^{(\mu)}$ generates the μ-th bit of the communication keys among users, and h is the length of these keys. $X_A^{(\mu)}$ and $X_B^{(\mu)}$ are the KPS-secret algorithms of A and B, respectively. $X_A^{(\mu)}$ and $X_B^{(\mu)}$ are calculated by the KPS center as follows:

$$X_A^{(\mu)} = x_A \; G^{(\mu)}, \tag{1}$$

$$X_B^{(\mu)} = x_B \; G^{(\mu)}. \tag{2}$$

$X_A^{(\mu)}$ and $X_B^{(\mu)}$ are contained in *tamper-resistant-modules* (TRM) and distributed to A and B, respectively. By using $X_A^{(\mu)}$ and $X_B^{(\mu)}$, A and B share their symmetric key as follows:

$$A: \; k_{AB}^{(\mu)} = X_A^{(\mu)} \; {}^t x_B, \tag{3}$$

$$B: \; k_{AB}^{(\mu)} = X_B^{(\mu)} \; {}^t x_A, \tag{4}$$

where $k_{AB}^{(\mu)}$ indicates the μ-th bit of the shared key k_{AB} between A and B, and ${}^t x$ indicates the transpose of x.

As already mentioned above, $G^{(\mu)}$ is a $n \times n$ matrix. Hence, by using n linearly independent KPS-secret algorithms, the KPS-center algorithm is easily revealed (note that, in order to participate in this collusion attack, each adversary has to break his TRM). In order to avoid such collusion attacks, we need to increase the value of n. However, since the number of $G^{(\mu)}$'s elements is n^2, a large memory size is required for the KPS center to increase the value of n. Hence in a conventional linear scheme, we cannot cope with collusion attacks efficiently.

4 An Optimized Credit-Based Payment System for ETC

The first part of this section shows how to construct a suitable credit-based payment protocol for ETC by applying the KPS. In the second part, an optimization of the KPS for our payment system is described.

4.1 Credit-Based Payment Protocol for ETC

4.1.1 Basic Concepts

Since conventional credit-based payment protocols requires huge computational cost, they cannot be applied to ETC straightforwardly. However, the payment style in ETC has some properties as mentioned in 2.3. ETC's properties allow us to construct the optimized credit-based payment protocol as follows:

- **Property 1.** and **2.** mean that toll gates can obtain the users' unique information and regard them as their identifier. By a suitable ID-Based cryptosystem (e.g., KPS), it is possible to make an authentication of a user and easily establish a cryptographical communication between a toll gate and a user. Furthermore, although the toll gate obtains the unique information of the user, the gate cannot obtain the personal information of the user (e.g., user's name).
- As it is well known, a credit company's authentication procedure of a user's payment requires a huge amount of time in comparison to the time for communications between a toll gate and a user. But such procedure can be done while the user is on the road. In the case that the payment is not authenticated the user can easily be detected at the exit toll gate according to properties 2 and 3.

Table 1 shows the notation which will be used in the following.

4.1.2 Detailed Description of Our Protocol

In this subsection, our credit-based payment protocol for ETC is described in detail. Figure 1 summarizes the flow of messages in our protocol. This protocol has 4 phases; **Phase (a)** is the Precomputation phase by U, **Phase (b)** is the Communication phase between U and T, **Phase (c)** is the Communication phase between T and P, and **Phase**

Table 1: Parameters for our payment system.

$E_k(t)$	to encrypt t by using a key k.
$D_k(t)$	to decrypt t by using a key k.
$E'_k(t)$	$\{E_k(\text{sessionkey}),E_{\text{sessionkey}}(t)\}$.
$D'_k(E'_k(t))$	$D_{\text{sessionkey}}(t)$, where sessionkey is obtained as $D_k(\text{sessionkey})$.
$H(t)$	to hash t.
$Sig_k(t)$	$E_k(H(t))$.
$MAC_k(t)$	a message authentication code of t using k.
Pv_e	participant e's private key.
Pb_e	participant e's public key.
$PIData$	payment instruction data, which indicates user's secret information for credit payment.
$OIData$	order information data, which indicates the entrace and the exit toll gate that user applies.
$AuthReqData$	Authorization request data.
$Chall_e$	participants e's challenge.
$Ack_{PRes}, Ack_{PReRess}$	acknowledgments.
U	a user.
T	an entrance toll gate.
E	an esit toll gate.
P	a payment gateway, which authorizes users' payments.
S	a server, which manages unauthorized users' unique information.
x_e	prticipant e's identifier (especilally, x_U is computed as H(user's unique information).
x_{e_v}	participant e's identifier for message verification (x_{U_v} is computed in the same way as x_U by using another hash function).
$X_e(\cdot)$	participant e's secret algorithm. $X_e(\cdot)$ provides the function of key sharing; $X_{e_1}(x_{e_2}) = X_{e_2}(x_{e_1}) = k_{e_1e_2}$.
$X_{e_v}(\cdot)$	prticipant e's secret aloforithm for message verification. $X_{e_v}(\cdot)$ provides the function of key sharing; $X_{e_{1v}}(x_{e_2}) = X_{e_2}(x_{e_{1v}}) = k_{e_{1v}e_2}$.

(d) is the Communication phase between U and E. Only in phases **Phase (b)** and **(d)**, the tiem for communications is strictly limited. **Phase (a)** and **(c)** allow to make a complex calculations.

Phase (a): U has to do the procedure described below as a preparation in advance.

U obtains x_T, x_{T_v} in a certain way (e.g., broadcasting) and computes the keys k_{UT}, k_{UT_v} and k_{U_vT} as follows:

$$k_{UT} = X_U(x_T), \qquad k_{UT_v} = X_U(x_{T_v}), \qquad k_{U_vT} = X_{U_v}(x_T).$$

Then, U produces $PReq$ as follows:

$$PReq = \{E'_{k_{UT}}(OI, PI, Chall_U)\},$$

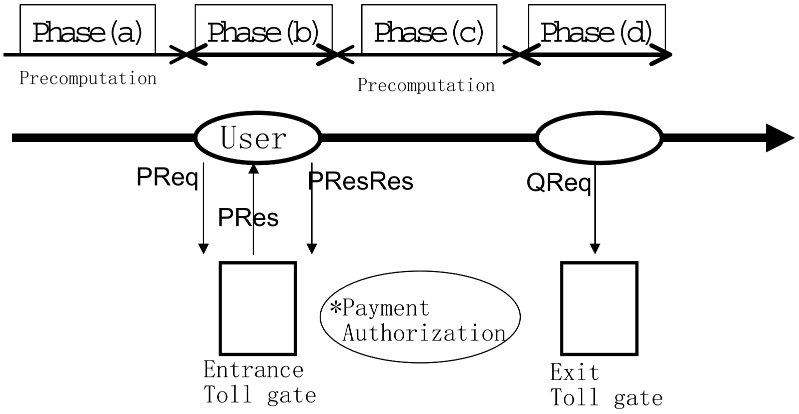

Figure 1: Optimized credit-based payment protocol for Electronic Toll Collection system.

where $OI = \{OIData, H(PIData), MAC_{k_{UT_v}}(H(PIData), H(OIData))\}$,
$PI = \{E'_{Pb_P}(PIData, H(OIData)), Sig_{Pv_U}(H(PIData), H(OIData))\}$.

$PReq$ will be sent at the start of the communication between U and T. Note that the ture recipient of PI is P. Since $PIData$ is encrypted with P's public key, T cannot read it. However, P can be confident of the hash value of $PIData$ by verifying OI and $Sig_{Pv_U}(H(PIData), H(OIData))$. This property is similar to dual signature[2] in SET.

Phase (b): Just at the start of the communication between U and T, U has to sent $PReq$ to T. While, T has to detect U's unique information and, by using U's unique information, T calculate x_U, x_{U_v} and compute k_{UT},k_{UT_v} as follows:

$$k_{UT} = X_T(x_U), \qquad k_{UT_v} = X_{T_v}(x_U).$$

By using these the keys, T decrypts and verifies $PReq$ as follows:

$$D'_{k_{UT}}(E'_{k_{UT}}(OI, PI, Chall_U)) = \{OI, PI, Chall_U\}$$

and, if $MAC_{k_{UT_v}}(H(PIData), H(OIData))$ is valid, T accepts $OIData$.

Following these procedures, T produces $PRes$ as the answer for $PReq$. In order to calculate $PRes$, T has to obtain the exit toll gate's identifier x_E, x_{E_v} from $OIData$ and computes the keys $k_{U_vM}, k_{TE_v}, k_{TE}$ as follows:

$$k_{U_vM} = X_T(x_{U_v}), \qquad k_{TE_v} = X_T(x_{E_v}), \qquad k_{TE} = X_T(x_E).$$

By using these employed keys, T encrypts and signs $PRes$ as follows:

$$PRes = E'_{k_{UT}}(Ack_{PRes}, Log, Chall_U, Chall_T, MAC_{k_{U_vT}}(Ack_{PRes}, Log, Chall_U, Chall_T)),$$

where

$$Log = E'_{k_{TE}}(LogData, MAC_{k_{TE_v}}(LogData)).$$

$PRes$ is sent to U as soon as these procedures are finished.

On receiving $PRes$, U decrypts and verifies as follows:

$$D'_{k_{UT}}(E'_{k_{UT}}(Ack_{PRes}, Log, Chall_U, Chall_T, MAC_{k_{U_vT}}(Ack_{PRes}, Log, Chall_U, Chall_T)))$$
$$= \{Ack_{PRes}, Log, Chall_U, Chall_T, MAC_{k_{U_vT}}(Ack_{PRes}, Log, Chall_U, Chall_T)\}$$

and, if $MAC_{k_{U_vT}}(Ack_{PRes}, Log, Chall_U, Chall_T)$ and $Chall_U$ are valid, U accepts Log.

Following these procedures, U sends $PResRes$ to T as the acknowledgment for $PRes$. $PResRes$ is computed as follows:

$$PResRes = E'_{k_{UT}}(Ack_{PResRess}, Chall_T, MAC_{k_{UT_v}}(Ack_{PResRes}, Chall_T)),$$

On receiving $PResRes$, T decrypts and verifies it as follows:

$$D'_{k_{UT}}(E'_{k_{UT}}(Ack_{PResRes}, Chall_T, MAC_{k_{UT_v}}(Ack_{PResRes}, Chall_T)))$$
$$= \{Ack_{PResPRes}, Chall_T, MAC_{k_{UT_v}}(Ack_{PResRes}, Chall_T)\}$$

and, if $MAC_{PResRes}$ and $Chall_T$ are valid, T allows U to enter the road.

Phase (c): While U is on the road, both T and U have enough time to make heavy computations. T produces $AuthReq$ as shown below.

$$AuthReq = \{PI, E'_{Pb_P}(AuthReqData, H(PI)), Sig_{Pv_T}(AuthReqData, H(PI))\}$$

The structure of $AuthReq$ is almost same as $AuthReq$ in SET[3, 4] and P decrypts and verifies in the same way as in SET's procedure. Following this procedure, P sends $AuthReq$ to T (this procedure is also same as SET's). $AuthReq$ gives the permission of the payment. If U's payment is not authenticated, T sends x_U to server S. The server distributes the identifier to all toll gates. By using this identifier, the toll gates can detect the user whose payment is not authenticated and request the user to pay by cash. This user's identifier is preserved in each toll gate's disk and he will be stopped in **Phase (b)** at the next time. The procedure, how an unauthenticated user can get his name removed from the toll gate' disk must be specified by the toll gates' operator.

In this phase, U obtains x_E, x_{E_v} in advance and computes the keys k_{UE},k_{UE_v} as follows:

$$k_{UE} = X_U(x_E), \qquad k_{UE_v} = X_U(x_{E_v}).$$

Afterwards, U produces $QReq$ as follows:

$$QReq = E'_{k_{UE}}(Log, x_T, MAC_{k_{UE_v}}(Log, x_T))\}.$$

Phase (d): At the start of the communication between U and E, U sends $QReq$ to E. So E obtains U's unique information. Then, by using them, E computes k_{UE} and k_{UE_v} as follows:

$$k_{UE} = X_E(x_U), \qquad k_{UE_v} = X_{E_v}(x_U).$$

Then, E decrypts and verifies $QReq$ as follows:

$$D'_{k_{UE}}(Log, x_T, MAC_{QReq})) = \{Log, x_T, MAC_{k_{UE_v}}(Log, x_T)\},$$

and if $MAC_{k_{UE_v}}(Log, x_T)$ is valid, E accepts Log and x_T. Next, E computes k_{TE} and k_{TE_v} according to:

$$k_{TE} = X_E(x_T), \qquad k_{TE_v} = X_{E_v}(x_T).$$

By using these employed keys, E verifies Log as follows:

$$D'_{k_{TE}}(Log) = \{LogData, MAC_{TE_v}(LogData)\},$$

and if $MAC_{TE_v}(LogData)$ is valid, E accepts $LogData$ and check the content of $LogData$.

$LogData$ gives date and time when U passed the entrance toll gate, as well as the exit toll gate that U mentioned at the entrance toll gate and so on. Hence, if U's behavior is different from the statements made at the entrance toll gate for, it can be detected easily.

4.1.3 Properties of Our Protocol

In our protocol, toll gates can obtain the users' unique information but not their personal data such as names. Besides, all the procedures that require huge computational cost are done while users are on the road. Hence, the computational performance required for the system can be quite low. Namely, our system can be realized only by using ordinary IC-cards for users and low-performance computers for toll gates.

4.2 Optimization of KPS

As mentioned in 3.3, although the KPS requires only moderate computational cost and no prior communications, there exists a serious problem that more than a threshold number of colluders can break the whole system. By increasing the collusion threshold to be high enough, this problem can be solved. However, since the memory size that is required for the center algorithm is proportional to the square of the collusion threshold, the collusion threshold cannot be increased easily. However, for the application of ETC not all of the possible communication links supported by the KPS are required. Omitting unnecessary links allows to increase the collusion threshold.

Table 2: Required and unrequired communications in our payment protocol, where ○ and × indicate required and unrequired, respectively.

	user	toll gate	server
user	×	○	×
toll gate	○	○	○
server	×	○	○

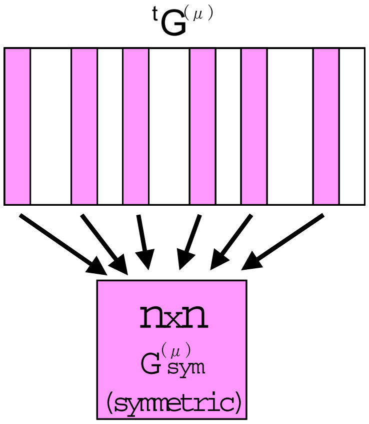

Figure 2: Embedding $G^{(\mu)}_{sym}$ into $G^{(\mu)}$.

4.2.1 Required and Unnecessary Communication Links in Our Protocol

Participants in our protocol can be classified into 3 classes; users, toll gates and servers. In the conventional KPS, any pair of entities in the system can share a common key. However in our payment system, there are a lot of pairs of entities that will have no communication. Table2 which communication links are required and which are not in our payment protocol.

4.2.2 Optimization of KPS

There exists already an optimizing method that increases the collusion threshold by removing functions for specified communications. This method was named a *Hierarchical KPS*[18]. In a Hierarchical KPS, participants are classified into 2 classes; providers and consumers, and by removing the function for the communications among consumers, the collusion threshold can be increased significantly. By applying this method efficiently, the KPS can be adapted to our protocol as well.

In our optimized KPS, the KPS-center algorithms $G^{(\mu)}$ $(\mu = 1, \cdots, h)$ are $m \times n$ asymmetric matrices $(m \gg n)$. In each $G^{(\mu)}$, a $n \times n$ symmetric matrix $G^{(\mu)}_{sym}$ is embedded as shown in Figure 2. The selection of the rows of $G^{(\mu)}$ that construct $G^{(\mu)}_{sym}$ is described as $k^{(\mu)}_{sel}$. The m-dimensional vector x_U is the effective ID of user U, the n-dimensional vector y_T is the effective ID of toll gate T and the n-dimensional vector z_S is the effective ID of server S. Their secret algorithms are calculated as follows:

Table 3: Calculations for key sharing, where $\overline{Y_T}$ is a n-dimensional vector whose elements are selected from Y_T according to $k_{sel}^{(\mu)}$.

secret algorithm \ partner's ID	user x_U	toll gate $y_T(y_{T'})$	center $z_S(z_{S'})$
user X_U	–	$k_{UT} = X_U^{(\mu)}\ {}^t y_T$	–
toll gate Y_T	$k_{UT} = Y_T^{(\mu)}\ {}^t x_U$	$k_{TT'} = \overline{Y_T^{(\mu)}}\ {}^t y_{T'}$	$k_{TS} = \overline{Y_T^{(\mu)}}\ {}^t z_S$
center Z_S	–	$k_{TS} = Z_S^{(\mu)}\ {}^t y_T$	$k_{SS'} = Z_S^{(\mu)}\ {}^t z_{S'}$

$$X_U^{(\mu)} = x_U\ G^{(\mu)}, \quad Y_T^{(\mu)} = y_T\ {}^tG^{(\mu)}, \quad Z_S^{(\mu)} = z_S\ {}^tG_{sym}^{(\mu)},$$

where $X_U^{(\mu)}$,$Y_T^{(\mu)}$ and $Z_S^{(\mu)}$ are U's, T's and S's secret algorithms, respectively.

By using these secret algorithms and $k_{sel}^{(\mu)}$, all participants can compute their common keys as illustrated in Table 3.

Evaluation and security of this optimized KPS are discussed in Section 5.

5 Evaluation and Security Discussion

5.1 Credit-Based Payment Protocol

Since the required computation time and other parameters strongly depend on the implementation of the system it is difficult to estimate them. Here we evaluate our protocol by comparing it to SET which is the most common credit–based payment protocol at the moment. It turns out that in our protocol all the procedures during the time for communications consist of quite simple calculations and that the number of communications is low enough. Table 4 summarizes the comparison in terms of computation costs and the number of communications.

Note that SET applied in straight-forward manner cannot provide all the functions required for ETC, e.g. messages for acknowledgments and other purposes. To implement these in SET would additionally require a huge amount of extra computational cost and communications. Furthermore it is well–known that public–key encryption / decryption and public–key signature generation / verification are computationally much more expensive than symmetric–key encryption and calculation of an inner product. Besides the maximum number of communications in our protocol is less than that in SET. In consequence, the protocol proposed here is much faster than SET.

Regarding security, attacks based on illegal forgery of a user's unique identification must be considered. However for such an attack to be successful, the legal user's IC card is also required. So the proposed ETC protocol is just as secure as normal credit–card payments. If a user loses his IC card, such kinds of attack can be prevented by terminating the card's validity.

When we implement this payment system, we need to consider how to deal with problems when a car tries to pass through a toll gate without paying (either by not

Table 4: Comparison of our protocol with SET, where *ske*/*skd* indicates symmetric-key encryption/decryption, *pke*/*pkd* indicates public-key encryption/decryption, *psg*/*psv* indicates public-key signature generation/verification and *ipc* indicates inner products calculation (note that straightforwardly-applied SET requires huge amount of extra computational cost and communications).

	our protocol	SET
a user's computational cost at an entrance toll gate	$3ske$, $3skd$	$1psv$
a user's computational cost at an exit toll gate	0	0
a toll gate's computational cost at an entrance toll gate	$5ipc$, $6ske$, $5skd$	$1pke$, $1pkd$, $2psg$, $1psv$, $1ske$, (+ *purchase authorization* by payment gateway)
a toll gate's computational cost at an exit toll gate	$4ipc$, $6skd$	0
the number of communications at an entrance toll gate	3	4
the number of communications at an exit toll gate	1	0

Table 5: Collusion thresholds to calculate $G^{(\mu)}$ and $G^{(\mu)}_{sym}$

colluders	$G^{(\mu)}$	$G^{(\mu)}_{sym}$
users	m	$m - n + \log_2 n$
toll gates	n	n
servers	*impossible*	n
toll gates + servers	n toll gates	n

identifying itself, by using an invalid number plate). However, this is a general problem of ETC and has been considered in other researches. Thus, we do not investigate it in this article.

5.2 Optimized KPS for Our Protocol

Collusion thresholds of our optimized KPS are described in Table 5.

Considering these collusion thresholds, m and n are determined mainly according to the numbers of users and toll gates, respectively. Namely, required memory size for the center algorithm is determined to be proportional to n times m, while, in the conventional KPS the required memory size for the center algorithm is determined to be proportional to $(n+m)^2$. Furthermore, the memory size for the user's secret algorithm is proportional to m. Since in the conventional KPS this is proportional to $(n + m)$, the memory size

Table 6: Required memory size for each type of entities, assuming that the required memory size is same in both our KPS and the conventional KPS (note that n $\ll$ m).

	KPS center	user	toll gate	server
optimized KPS	hnm	hn	hm	hn
conventional KPS	hnm	$h\sqrt{nm}$	$h\sqrt{nm}$	$h\sqrt{nm}$

for the user's secret algorithm can be reduced considerably. The number of users will be much higher than the number of toll gates. Thus, these reductions of memory size are quite significant. Table 6 shows required memory sizes for each type of entity. Assuming that m and n are determined to be 262144 and 256, respectively, our KPS's collusion threshold of users' collusion attack is 262144. This value is approximately 32 times that of the conventional KPS, assuming that 8192 × 8192 symmetric matrices are used in it. Furthermore, the required memory size for user's secret algorithm is one thirty-second of that of the conventional KPS. Although the difference between these two thresholds is quite significant, their required memory sizes in the KPS center are same. Only the memory size in the toll gates is larger for the optimized KPS than for the conventional KPS. But this is not a serious problem since in the toll gates a large amount of memory can be installed easily.

Note that it is also possible to install several servers to improve the efficiency of the whole ETC system. In principle, a subset of toll gates and servers could collude. However there is no serious interest in this, so that a relatively low collusion threshold for the toll gates is not a real problem.

6 Conclusion

In this article, a new credit–based payment system for ETC has been proposed. Unlike the payment systems proposed up to now, it is based on an optimized version of the KPS. Because of this during the communications only trivial computations are made and therefore the system can be realized using only IC cards in the cars and simple, low–cost computers in the toll gates. Nevertheless, our payment protocol preserves the user's privacy, since it uses unique information on his car, but not his name or so on.

It has been taken into account that certain collusion attacks can be effective against the KPS. Therefore it has been shown how the KPS can be optimized in our application to increase the resistance against collusion attacks. Our optimization obtains a quite high collusion threshold using just the same amount of memory as the conventional KPS. In a situation that uses a typical security parameter setting, the obtained collusion threshold by our optimization is 32 times as high as that of the conventional KPS.

References

[1] T. Matsumoto and H. Imai, "On the KEY PREDISTRIBUTION SYSTEM: A Practical Solution to the Key Distribution Problem," Advances in Cryptology: Proc. of CRYPTO'87, Springer LNCS 293, pp.185-193, 1987.

[2] MasterCard and Visa, "Secure electronic transaction (SET) specification book 1: Business Decryption," May, 1997.

[3] MasterCard and Visa, "Secure electronic transaction (SET) specification book 2: Programmer's Guide," May, 1997.

[4] MasterCard and Visa, "Secure electronic transaction (SET) specification book 3: Formal Protocol Definition," May, 1997.

[5] CyberCash, "CyberCash Web Server," Reston, VA, 1996, http://www.cybercash.com/

[6] M. David and K. Sakurai, "Security Issues for Contactless Smart Cards," Proc. of PKC'98, LNCS 1431, Springer-Verlag, pp.247-252, 1998.

[7] G. Hanaoka, Y. Zheng and H. Imai, "LITESET: a Light-Weight Secure Electronic Transaction," Proc. of ACISP '98, LNSC 1438, Springer-Verlag, pp.215-226, July, 1998.

[8] R. Blom, "Non-public Key Distribution," Proc. of CRYPTO'82, Plenum Press, pp.231-236, 1983.

[9] A. Fiat and A. Shamir, "How to Prove Yourself: Practical Solutions to Identification and Signature Problems," Proc. of CRYPTO'86, LNCS 263, Springer-Verlag, pp.186-194, 1986

[10] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," Proc. of CRYPTO'84, LNCS 196, Springer-Verlag, pp.47-53, 1985.

[11] U. Maurer and Y. Yacobi, "Non-interactive Public-Key Cryptography," Proc. of Eurocrypt'91, LNCS, Springer-Verlag, pp.498-407, 1992.

[12] U. Maurer and Y. Yacobi, "A Remark on a Non-interactive Public-Key Distribution System," Proc. of Eurocrypt'92, LNCS, Springer-Verlag, pp.458-460, 1993.

[13] E. Okamoto and K. Tanaka, "Identity-Based Information Security management System for Personal Comuputer Networka," IEEE J. on Selected Areas in Commun., 7, 2, pp.290-294, 1989.

[14] H. Tanaka, "A realization Scheme of the Identity-Based Cryptosystems," Proc. of CRYPTO'87, LNCS 293, Springer-Verlag, pp.340-349, 1988.

[15] S. Tsujii and J. Chao, "A New ID-Based Key Sharing System," Proc. of CRYPTO'91, LNCS 576, Springer-Verlag, pp.288-299, 1992.

[16] D. Coppersmith, "Attack on the Cryptographica Scheme NIKS-TAS," Proc. of CRYPTO'94, LNCS 839, Springer-Varlag, pp.40-49, 1994.

[17] Y. Zheng, "Digital signcryption or how to achieve cost(signature&encryption) $\ll$ cost(signature)+cost(encryption)," Proc. of CRYPTO'97, LNCS 1294, Springer-Verlag, pp.165-179, 1997.

[18] G. Hanaoka, T. Nishioka, K. Matsuura, Y. Zheng and H. Imai, "On Hierarchical KPS: An Optimized KPS against Collusion Attacks," Proc. of ISITA'98, pp.247-250, 1998.