Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.050461

ARTICLE

AnonymousTollPass: A Blockchain-Based Privacy-Preserving Electronic Toll Payment Model

Jane Kim[1], Soojin Lee[1], Chan Yeob Yeun[2] and Seung-Hyun Seo[1,3,*]

[1]Department of Electronic & Electrical Engineering, Graduate School, Hanyang University, Seoul, 04763, Korea

[2]Department of Electrical Engineering and Computer Science, Khalifa University of Science and Technology, Abu Dhabi, 127788, United Arab Emirates

[3]School of Electrical Engineering, Hanyang University, ERICA, Ansan, 15588, Korea

*Corresponding Author: Seung-Hyun Seo. Email: seosh77@hanyang.ac.kr

Received: 07 February 2024 Accepted: 10 April 2024 Published: 20 June 2024

ABSTRACT

As big data, Artificial Intelligence, and Vehicle-to-Everything (V2X) communication have advanced, Intelligent Transportation Systems (ITS) are being developed to enable efficient and safe transportation systems. Electronic Toll Collection (ETC), which is one of the services included in ITS systems, is an automated system that allows vehicles to pass through toll plazas without stopping for manual payment. The ETC system is widely deployed on highways due to its contribution to stabilizing the overall traffic system flow. To ensure secure and efficient toll payments, designing a distributed model for sharing toll payment information among untrusted toll service providers is necessary. However, the current ETC system operates under a centralized model. Additionally, both toll service providers and toll plazas know the toll usage history of vehicles. It raises concerns about revealing the entire driving routes and patterns of vehicles. To address these issues, blockchain technology, suitable for secure data management and data sharing in distributed systems, is being applied to the ETC system. Blockchain enables efficient and transparent management of ETC information. Nevertheless, the public nature of blockchain poses a challenge where users' usage records are exposed to all participants. To tackle this, we propose a blockchain-based toll ticket model named *AnonymousTollPass* that considers the privacy of vehicles. The proposed model utilizes traceable ring signatures to provide unlinkability between tickets used by a vehicle and prevent the identity of the vehicle using the ticket from being identified among the ring members for the ticket. Furthermore, malicious vehicles' identities can be traced when they attempt to reuse tickets. By conducting simulations, we show the effectiveness of the proposed model and demonstrate that gas fees required for executing the proposed smart contracts are only 10% (when the ring size is 50) of the fees required in previous studies.

KEYWORDS

Blockchain; electronic toll collection; smart contract; traceable ring signature

1 Introduction

The rapid advancement of intelligent vehicle performance is accelerating the establishment of vehicular services and Intelligent Transport Systems (ITS) [1–4]. An Electronic Toll Collection (ETC)

system, which is one of the main pivotal services of the ITS, is a wireless payment system that allows drivers to pass through toll facilities without stopping and paying toll fees [5]. The system facilitates smooth traffic flow and enables efficient traffic management. Due to these advantages, the implementation of ETC Systems is gradually increasing in developing countries such as India FASTag [6], and Pakistan NADRA [7], aimed at providing convenient payment options, improving efficiency, and complying with government regulations on carbon emissions. Accordingly, the market size of the ETC System is expected to increase from $9.2 Billion in 2020 to $17.7 Billion in 2027 [8].

The current ETC System faces two significant challenges. The first challenge is how *TSP*s can securely share toll information in a distributed environment. Currently, in most countries operating ETC Systems, multiple companies and institutions have their own Toll Service Provider (*TSP*)s and manage Toll Plaza (*TP*)s in each region [1]. Thus, ETC Systems have been developed in decentralized environments where each *TSP* has its own database, considering the realistic situation of ETC Systems built in each country. At this point, since vehicles move through multiple regions and pay tolls via several *TSP*s and *TP*s, *TSP*s should be able to share toll usage information for accurate toll payments in a decentralized environment. However, when sharing toll information in a distributed environment, security issues may arise related to data synchronization and validation among *TSP*.

Secondly, the vehicle's travel route is revealed during the electronic toll payment process. electronic toll payments using credit cards are flawed in protecting driver privacy. That is because credit card transaction data can lead to patterns of behavior that can reveal identifying information at any given time. In fact, ETC Systems (e.g., E-ZPass [9], hi-pass [10], and Vietnam Electronic Toll Collection (VETC) [11]) used in many countries make credit card-based payments in a similar way to the use of unique codes. However, this may expose personal information connected to the credit cards themselves. In these systems, *TSP*s can get payment data that can identify the driver, such as tolling information, each time the vehicle makes a toll payment. Especially when drivers commute by vehicle or regularly visit specific places like hospitals or supermarkets, *TSP*s can derive personal information such as the driver's driving patterns and destination based on the collected payment information. Exposing a driver's personal information is highly risky because a privacy disclosure can trigger many safety concerns and even physical attacks on the driver [12,13]. To address these challenges, it is necessary to have a privacy-preserving data-sharing method for the ETC System in decentralized environments to ensure the unlinkability of each toll usage record of vehicles and conceal the entire toll usage history.

In the design of a secure data sharing model in a distributed environment, blockchain technology can be utilized to provide data traceability, integrity, and distributed ledger functionality. So far, several researchers [14–16] have been working on blockchain-based electronic toll payment systems. Ying et al. [14] proposed a blockchain-based highway electronic toll payment framework for efficient authentication using aggregate signature. Deng et al. [15] designed a smart contract-based electronic toll payment scheme for vehicle-to-RSU transactions. However, the blockchain models proposed in these papers still suffer from the issue of exposing users' personal information to Roadside Units (RSUs), toll plazas, or other users, as the users' toll usage records and payment details are directly stored and shared on the blockchain ledger. To preserve a vehicle's privacy, Guo et al. [16] proposed a blockchain-based privacy-preserving electronic toll payment scheme using zero-knowledge Succinct Non-interactive ARgument of Knowledge (zk-SNARK) and group signature. Nevertheless, the group leader can still trace who generated a signature. Furthermore, the requirement for vehicles to execute smart contract operations for each toll payment is both financially and computationally inefficient.

To hide the user's toll usage records from other entities such as *TSP*s and toll plazas, we proposed *AnonymousTollPass*, a blockchain-based anonymous electronic toll ticket payment model.

Vehicles buy a one-time *AnonymousTollPass* ticket for toll payment using smart contracts, and *AnonymousTollPass* tickets are generated based on traceable ring signatures (TRS) [17]. According to TRS's characteristics, the user's identity using the ticket remains concealed. If someone who has already used the ticket attempts to reuse the same ticket, their identity is revealed. Since the used *AnonymousTollPass* ticket information is recorded in the smart contract, the stored data is resistant to forgery and tampering and is safely managed by *TSP*s. We implemented our smart contract system to prove the proposed model's effectiveness. Then, we deployed it on the Ethereum blockchain, measuring the gas fee, processing time of each phase, and key sizes according to the ring sizes. The simulation results show that the proposed model exhibits reduced implementation costs compared to the existing privacy-preserving ETC model, attributed to the reduction in computational load within smart contracts. The main contributions of this paper are as follows:

1. **Blockchain-based ETC Model:** We designed the *AnonymousTollPass* model, a blockchain-based ETC payment system. It eliminates the single point of failure and ensures reliable toll usage data sharing among the participants.
2. **Anonymous Tickets:** We introduced a method to hide users' identities with *AnonymousTollPass* tickets, which is based on the traceable ring signature (TRS) algorithm. When a vehicle uses the ticket, no one, including the *TSP* that sold the ticket, can infer exactly which of the ring members used the ticket except for double payment.
3. **Smart Contracts with Lower Costs:** We designed two smart contracts, *TicketSaleContract* and *TicketUsageContract*, to handle ticket usage history. These smart contracts demonstrate cost efficiency that improves with ring size; as the ring size increases, the operational cost decreases. With a ring size of 50, the proposed smart contract requires only about 10% of the gas fees compared to those used in previous studies.

The rest of this paper is organized as follows: We discuss related works for ETC Systems in Section 2. We describe overall model including security requirements and *AnonymousTollPass* ticket in Section 3. We propose two smart contract systems and a basic protocol of our *AnonymousTollPass* model in Section 4. Then, we provide security analysis in Section 5. Also, we provide the performance evaluation of our model in Section 6. We discuss a case study of South Korea in Section 7. Finally, we conclude this paper in Section 8.

2 Related Works

2.1 Text Layout Electronic Toll Collection for ITS

An ITS is an advanced transportation system that integrates communication, sensors, AI, and other intelligent technologies into legacy transportation systems to provide innovative applications. Accordingly, various studies [18–21] have been conducted to enhance user convenience through data management systems that collect and analyze transportation information. Borges et al. [18] proposed a privacy-preserving ETC scheme that utilizes a protocol called "Priced Oblivious Transfer" (POT) [19]. To use the POT protocol, the service provider needs to prepare tickets for all possible entrances when a vehicle exits the highway. It has a significant time overhead as all vehicles must use the system simultaneously. Randriamasy et al. [20] proposed an ETC model for vehicles equipped with C-ITS (Cooperative intelligent transport system) devices. They identified the vehicle's geolocation at a toll plaza with a barrier environment and automatically opened the barrier for vehicles that successfully paid the toll. In this model, it requires an additional device. Aung et al. [21] proposed a system that reserves planned roads to alleviate congested traffic environments. Instead of collecting tolls, they used the T-coin (Traffic coin) they proposed to provide rewards for using alternative routes based on traffic

congestion. References [20, 21] proposed an efficient ETC System, but they did not address the issue of user privacy exposure. Additionally, since the systems of [19–21] operate with a single central server, they have the issue of a single point of failure.

2.2 Blockchain-Based Electronic Toll Collection System

Blockchain means a distributed ledger technology in which multiple nodes that do not trust each other store a ledger without a trusted agency [22]. Using a cryptographic hash function and electronic signature, the recorded transactions' order and contents in the ledger cannot be modified, providing data integrity. Blockchain technology plays a significant role in various fields that require safe data sharing and storage, such as vehicular networks [1], healthcare [23], and smart cities [24]. Also, researchers have conducted studies on integrating blockchain technology into the ETC System, which operates as a distributed system involving *TP*s and multiple *TSP*s the ETC System, to safely manage toll payment information. *TP*, *TSP*, and vehicles participate as blockchain nodes to safely perform toll payments through smart contracts distributed on the blockchain and store toll payment information such as toll transaction date and payment amount in the blockchain. A blockchain-based ETC System is generally structured as shown in Fig. 1.

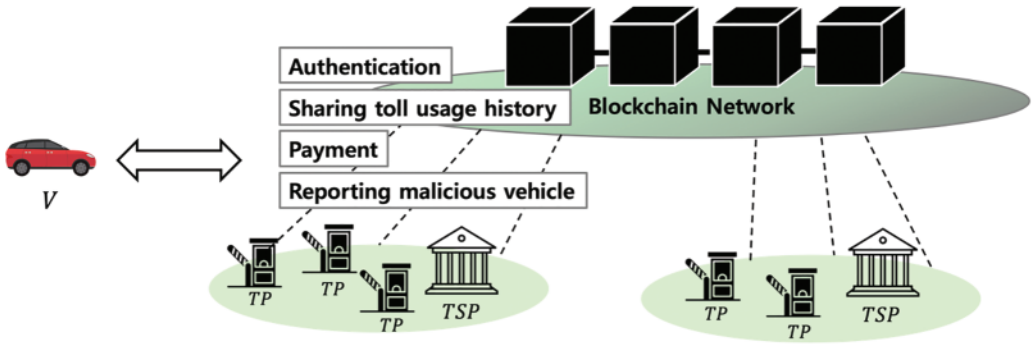

Figure 1: General blockchain-based ETC Systems

Ying et al. [14] suggested a blockchain-based efficient highway toll paradigm considering vehicle platoons. In the model, a vehicle platoon leader pays the toll fee for all platoon members simultaneously by applying an aggregate signature algorithm, enabling efficient vehicle toll payment. Deng et al. [15] proposed two electronic payment schemes, including a vehicle-to-RSU (V-R) transaction and a vehicle-to-RSUs (V-Rs) transaction. Vehicles conduct toll payments through a smart contract and RSUs. Chiu et al. [25] proposed a blockchain-based electronic toll collection system using a practical byzantine fault tolerance (PBFT) algorithm as a consensus algorithm for fast transaction processing. Xiao et al. [26] suggested a blockchain-based toll collection system for public sharing using edge nodes. In order to reduce the overhead of an edge node when there are a large number of toll payment requests, the authors applied a proxy server and greedy algorithm for efficiently matching the edge node and vehicle. Shukla et al. [27] proposed a deep learning-based dynamic toll pricing scheme. The proposed model predicts vehicle traffic and determines the toll payment amount according to lane type and vehicle class. References [14, 15, 25–27] proposed a blockchain-based electronic toll payment model. However, the vehicle's public key, vehicle ID, and location of the toll plaza are stored in the blockchain so attackers can track a driver's private information and driving routes. To overcome the limitation, some studies concern blockchain-based electronic toll systems, considering vehicle privacy. Wang et al. [28] designed a credit electronic toll collection system, including an evidence chain

framework. Vehicle information is stored in a trusted storage center, and the evidence is recorded in the blockchain. However, RSUs and toll stations still know the driver's personal information and payment history. Guo et al. [16] proposed a blockchain-based privacy-preserving payment scheme. The proposed scheme protects the vehicle's location privacy by hiding the toll station information the vehicle has passed through by using zk-SNARK (zero-knowledge succinct non-interactive argument of knowledge) proof and a group signature algorithm. Nevertheless, a group leader can know who generates a signature according to the characteristics of the group signature. Therefore, the vehicle's driving route may be revealed regardless of the vehicle's will. In addition, vehicles should participate as a blockchain node and execute a payment smart contract whenever they start or end the toll service. So, vehicles must update and store the blockchain ledger in real-time, which burdens them due to their limited memory, low computing power, and mobility.

To effectively protect the privacy of vehicles from other vehicles, RSUs, and toll plazas, we propose a blockchain-based toll payment model utilizing a traceable ring signature that can safely hide the vehicle's toll plaza usage information. Also, vehicles with relatively low computational power and memory capacity do not participate as blockchain nodes in our proposed model. So, the smart contract only stores ticket information to lower the execution costs of the smart contract.

3 System Architecture

In this section, we define the entities and threat models in the proposed model. We also present the security requirements and describe how we design *AnonymousTollPass* model.

3.1 Overall Model

The proposed model is based on a private blockchain in which *TSP* and *TP* participate, storing toll payment information. Vehicles with relatively low computational power and memory capacity do not participate as blockchain nodes in our blockchain-based electronic toll ticket system. Instead, *TSP*s and *TP*s, which have sufficient computational power and memory, maintain the blockchain ledger. The blockchain ledger stores information about used toll tickets, public key lists for ticket validation, payment dates and times, and related *TP* information. The recorded toll payment data is used as a statistical indicator for urban infrastructure construction and traffic flow analysis [29]. There are four main entities in our *AnonymousTollPass* model: Registration Authority *RA*, Vehicle *V*, Toll Plaza *TP*, and Toll Service Provider *TSP*. The main entities are described as follows:

- Registration Authority *RA*: A registration authority *RA* is an organization, which manages registration information. The vehicle owner registers his vehicle information with the *RA* to use the ETC service. Only registered vehicles *Vs* can purchase an *AnonymousTollPass* ticket from the *TSP*.
- Vehicle *V*: Vehicles have their identity registered in the government sector through their platoon number in advance. A vehicle *V* buys an *AnonymousTollPass* ticket from the *TSP* to use the toll service. The *V* submits the ticket to TP_x for toll payment after using the highway. A *V* can read the blockchain ledger to check its ticket-related data.
- Toll Service Provider *TSP*: A Toll Service Provider is a service organization that operates toll plazas. It works as a full node in the private blockchain network. The *TSP* sells *AnonymousTollPass* tickets, which can be used in the toll plaza system, to *V*s and deploys a ticket sale contract and a ticket usage contract in the blockchain.
- Toll Plaza *TP*: Toll Plazas are located at every highway entry and exit point. Following a real ETC scenario, the Toll Plaza (*TP*) consists of an entry Toll Plaza TP_n where vehicles enter the

highway and an exit Toll Plaza TP_x where vehicles pay tolls and leave the highway. They have sufficient computing power and memory for *AnonymousTollPass* ticket verification and ledger management. They also maintain the blockchain ledgers to record ticket information.

In the proposed model, a V submits an *AnonymousTollPass* ticket instead of paying a toll directly to TP_x. The *AnonymousTollPass* ticket is created based on a traceable ring signature to hide the ticket user's private information according to the characteristics of the traceable ring signature. Vs buy an *AnonymousTollPass* ticket from the *TSP* and ticket prices vary depending on the travel distance. For instance, expressway toll fees in South Korea range from about $1 to $50, and the toll price is the same even if vehicles drive on different routes [30]. Reflecting it, the *TSP* gathers Vs traveling the same distance and forms a ring. The *TSP* deploys a ticket sale contract for checking ticket sales information and a ticket usage contract for managing used tickets. The Vs, which belong to the ring and purchase the ticket, receive the smart contract addresses from the *TSP*. Each V can check the ring group information to which it belongs and ticket information through these smart contracts. The process of the overall model is shown in Fig. 2.

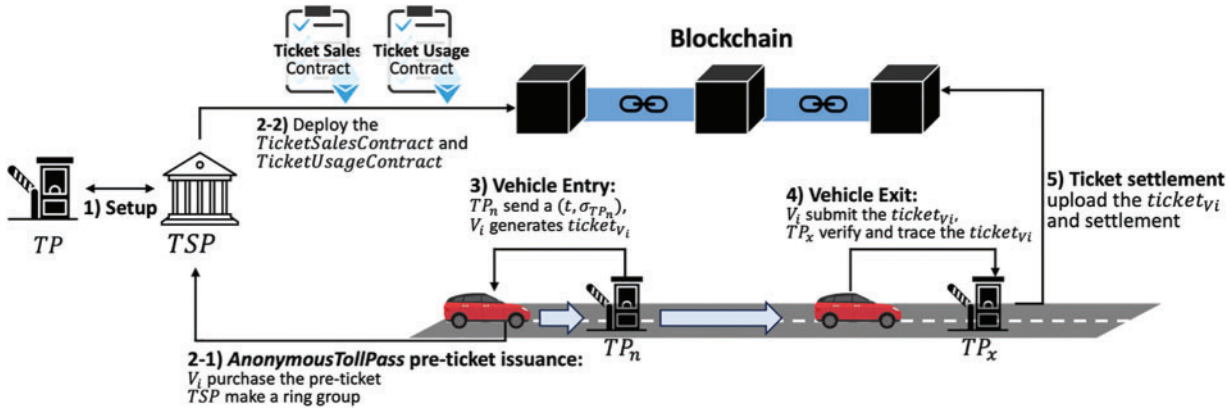

Figure 2: Overall model

3.2 Threat Model & Security Requirements

In the proposed model, we classified the threat models into three categories: A malicious V, a curious *TP*, and a curious *TSP*. The malicious V attempts to evade toll payment by reusing *AnonymousTollPass* tickets that others have used. It can also try to forge the location of the highway entry point, attempting to travel a greater distance than the purchased ticket would allow. The curious *TP* and the curious *TSP* can attempt to reveal V's identity by analyzing V's ticket usage histories and figuring out V's driving route through recorded transactions. In order to operate an electronic toll collection model secure against attackers, the proposed model should satisfy the following security requirements:

1. **Identity Privacy:** When a V uses the electronic toll collection model, the model should hide the V's toll plaza usage information to protect the privacy of the V. All *TP*s and *TSP*s should not be able to identify the V through the *AnonymousTollPass* ticket received from the V. However, if the V behaves maliciously, the model should be able to reveal his identity.
2. **Ticket Unlinkability:** An attacker may attempt to infer the V's driving route through the used ticket information stored in the blockchain. Therefore, the proposed model should not be able to link *AnonymousTollPass* tickets generated by the same vehicle. In addition, even if the public

key of the V for a specific ticket is exposed, other participants should not be able to infer other tickets of Vs from the exposed information.

3. **Resistance to distance forgery:** The *AnonymousTollPass* ticket price is proportional to the distance traveled by the V from the entry toll plaza to the exit toll plaza. Therefore, the proposed model requires the toll plaza location information when a V uses a ticket. However, a malicious V may attempt to forge information about a TP_n or a TP_x to pay a lower fee. To ensure a secure toll payment model, the proposed model should enable TP_x to verify which TP_n the V used to enter the highway when verifying the V's ticket, and the V must not be able to modify their usage history.
4. **Abuse of tickets:** A malicious V may attempt to reuse an *AnonymousTollPass* ticket that has already been used, or it may try to steal the information of a ticket purchased by other vehicles and use it for toll payment. To prevent a malicious V from abusing tickets, the proposed model should enable TP_x to verify whether an *AnonymousTollPass* ticket has been properly paid for.

3.3 AnonymousTollPass Ticket

In this paper, we propose an *AnonymousTollPass* ticket to preserve vehicles' privacy in the blockchain-based electronic toll payment model. The *AnonymousTollPass* ticket is a one-time toll ticket generated based on the traceable ring signature (TRS) algorithm [17]. The identity of each V using tickets is concealed within a ring group, making it difficult for TPs and other vehicles Vs to determine which ring group member utilized the ticket. In addition, if a malicious vehicle attempts to use a spent ticket, its identity is revoked through the TRS's characteristics. The *AnonymousTollPass* ticket $ticket_{V_i}\ (m, \sigma)$ consists of a message m and its corresponding TRS value σ. The *AnonymousTollPass* tickets are composed of four algorithms: Key pair generation (GEN), *AnonymousTollPass* ticket generation (*ticket*GEN), *AnonymousTollPass* ticket verification (*ticket*VER), and *AnonymousTollPass* ticket trace (*ticket*TRACE).

- $(pk_{V_i}, sk_{V_i}) \leftarrow GEN(1^{\lambda})$. The GEN algorithm takes a security parameter 1^{λ} as input and generates the public key pk_{V_i} and the private key sk_{V_i}. Let G be a multiplicative group of order q with g as the generator of G. Each V_i selects a random x_i from $\mathbb{Z}_q$, and calculates $y_i = g^{x_i}$. Each V_i uses $sk_{V_i} = \{pk_{V_i}, x_i\}$ as its private key and $pk_{V_i} = \{g, y_i, G\}$ as its public key.
- $ticket_{V_i}(m, \sigma) \leftarrow ticketGEN\left(sk_{V_i}, PK_N, m\right)$. When V_i generates a *AnonymousTollPass* ticket, the *ticketGEN* algorithm takes a private key sk_{V_i}, public key list of the ring group members PK_N, and message m as inputs, then outputs $ticket_{V_i}$. The detailed process is shown in Algorithm 1.

Algorithm 1: *ticket*GEN.

Input: $m \in \{0,1\}^*, PK_N, sk_{V_i}$
Output: $ticket_{V_i}(m, \sigma) = (m, A_1, c_N, z_N)$

$h = H(PK_N), \sigma_i = h^{x_i}\left(x_i \in \mathbb{Z}_q\right)$
$A_0 = H'(PK_N, m), A_1 = (\sigma_i / A_0)^{1/i}$
For all j ($j = (1, \ldots, N), j \neq i$), compute $\sigma_j = A_0 A_1^j \in G$
Pick random $\omega_i \leftarrow \mathbb{Z}_q$, and set $a_i = g^{\omega_i}, b_i = h^{\omega_i} \in G$
For all j ($j = (1, \ldots, N), j \neq i$), pick random $z_j, c_j \leftarrow \mathbb{Z}_q$ and set $a_i = g^{z_i} y_i^{c_j}$, $b_j = h^{z_j} \sigma_j^{c_j} \in G$

(Continued)

Algorithm 1 (continued)

Set $c = H''(PK_N, A_0, A_1, a_N, b_N)$, where $a_N = (a_1, \ldots, a_N), b_N = (b_1, \ldots, b_N)$
$c = c - \Sigma_{j \neq i} c_j (modq)$ and $z_i = \omega_i - c_i x_i (modq)$
Where $c_N = (c_1, \ldots, c_N), z_N = (z_1, \ldots, z_N)$
$\sigma = (A_1, c_N, z_N)$
return (m, σ)

- $\{0,1\} \leftarrow$ *ticket*VER $\left(ticket_{V_i}, PK_N\right)$. The *ticket*VER algorithm verifies whether the ticket submitted by V_i is generated by a member belonging to the PK_N. It takes the *AnonymousTollPass* ticket $ticket_{V_i}(m, \sigma)$, and public key set PK_N as inputs. The output is 1 if the verification succeeds and 0 if it fails. The detailed process is shown in Algorithm 2.

Algorithm 2: *ticket*VER.

Input: $ticket_{V_i}(m, \sigma), PK_N$
Output: $\{0,1\}$
For all $i \in N$, Check $g, A_1 \in G, c_i, z_i \in \mathbb{Z}_q, y_i \in G$
If not, **return** 0
Then compute $a_i = g^{z_i} y_i^{c_i}, b_i = h^{z_i}$ for all $i \in N$
Return $H''(PK_N, A_0, A_1, a_N, b_N) == \Sigma_{i \in N} c_i (modq)$

- $\{indep, pk_i\} \leftarrow$ *ticket*TRACE $(ticket_{V_i}, ticket_{V_j}, \mathrm{PK}_N)$. When V_i submits a ticket, the *ticket*TRACE algorithm checks whether the ticket has already been used. It takes the $ticket_{V_i}$, a previously used $ticket_{V_j}$, and the public key list of the ring group PK_N as inputs. If the ticket was generated by a different V, it outputs '*indep*'. If the ticket was generated by the same signer, it outputs 'pk_i'. The detailed process is shown in Algorithm 3.

Algorithm 3: *ticket*TRACE.

Input: $ticket_{V_i}(m, \sigma), ticket_{V_j}(m', \sigma'), PK_N$
Output: $\{indep, pk_{V_i}\}$
$h = H(PK_N)$
$A_0 = H'(PK_N, m), A_0' = H'(PK_N, m')$
for all $i \in N$
Compute $\sigma_i = A_0 A_1^{\ i} \in G, \sigma_i^{'} = A_0' A_1'^{\ i} \in G$
if $\sigma_i == \sigma_i^{'}$ then
    Store pk_i in **TList** //**TList** is a temporal list
    end if
end for
if #**TList** $== 1$ then
    return pk_i
else
return indep
end if

4 AnonymousTollPass Model

In this section, we first propose our smart contract system for managing *AnonymousTollPass* tickets. Then we present a basic protocol for our *AnonymousTollPass* model in detail.

4.1 Smart Contract System

The smart contract system for *AnonymousTollPass* model consists of *TicketSaleContract* and *TicketUsageContract*. A *TSP* deploys both contracts after *AnonymousTollPass* ticket sales.

4.1.1 TicketSaleContract

The *TSP* that sells an *AnonymousTollPass* ticket deploys *TicketSaleContract* on the blockchain network. This contract records ticket-related data on the blockchain and allows participants to access it. The ticket-related data consists of *Ring size*, *PK list*, and *Ticket price* as follows:

- *ring_size*: It is the number of *V* that purchased the same price ticket.
- *pk_list*: It is the public key list of the ring members.
- *ticket_price*: It is the price of the *AnonymousTollPass* ticket.

These ticket-related data are stored in the blockchain through the *Register_ticket_information( )* function and can be checked through the *Read_ticket_information( )* function.

- *Register_ticket_information( )*: This is a function to record *AnonymousTollPass* ticket information on the *TicketSaleContract*. Since only the *TSP* can sell toll tickets, we use a modifier to ensure that only *TSP* can execute this function. After the *TSP* sells *AnonymousTollPass* tickets to *V*s that purchased the ticket at the same price, the *TSP* executes this function with ring size, a *PK* list of *V*s, and the ticket price as inputs.
- *Read_ticket_information( )*: This is a function to check the information of the *AnonymousTollPass* ticket. The *V* can check the information on the purchased ticket. A *TP* verifies the ticket submitted by the *V* through the function.

Algorithm 4: Ticket sale contract.

```
struct ticket_information{
    string ring_size;
    string pk_list;
    int ticket_price;
}
ticket_information TicketInfo;
Function Register_ticket_information(_ring_size, _pk_list, _ticket_price) is TSP{
    TicketInfo.ring_size ← _ring_size;
TicketInfo.pk_list ← _pk_list;
 TicketInfo.ticket_price ← _ticket_price;
 }
 Funtion Read_ticket_information{
     return TicketInfo;
 }
```

4.1.2 TicketUsageContract

The *TP* records the used ticket-related data that it received from each *V* through the *TicketUsageContract*. The ticket-related data is stored in the form of a ticket_state structure which includes timestamp, ticket and a verification result. The ticket_state structure has the following three components:

- *timestamp*: It is the timestamp of when a *V* enters the TP_n.
- *signature*: It is the signature of the *AnonymousTollPass* ticket that the *V* generated.
- *verification_result*: If the ticket verification is successful, 'True' is stored. Otherwise, 'False' is stored.

The variables used in the *TicketUsageContract* are as follows:

- **address**[] TP_address: This is an array that stores the addresses of *TP*s who requested *AnonymousTollPass* ticket exchange.
- **ticket_state**[] request_ticket: This is an array that stores the ticket_state of the tickets that have been requested for exchange from the TP_x.
- **ticket_state**[] used_ticket: This is an array that stores the ticket_state of the used tickets.
- **mapping**(address=>ticket[]) submitted_ticket: This is a variable that maps the ticket_state of the ticket submitted by a *TP* as a key to the *TP*'s address as the corresponding value.
- **string**[] trace_pubkey: This is an array that stores the public key of *V*s revealed through the reuse of tickets.

The description of the five functions of the *TicketUsageContract* is as follows:

- *Exchange_request()*: The *TP* upload *AnonymousTollPass* ticket received from a *V* in the request_ticket array. The *TP* uses *timestamp*, *signature* and *TP*'s address as inputs. Since the verification of the ticket has not been confirmed yet, the initial verification value of the ticket is false.
- *Verify_ticket()*: The *TSP* updates the verification and trace result of the tickets in the request_ticket list. Only the *TSP* should be able to update the ticket verification result. The *TSP* inputs the _verification array and the _trace_pubkey array into this function. The _verification array stores the verification result of the request_ticket, and the _trace_pubkey array stores the traced public key value of the ticket if the verification result is 'false', otherwise it stores Null value. Only tickets with a verification result that is 'true' are stored in the used_ticket list.
- *Read_request_ticket()*: This is the function to check the request_ticket list. The *TSP* executes this function to verify the requested tickets for toll payment.
- *Read_used_ticket()*: This function checks already used ticket sets. The *TP* executes this function when authenticating a ticket.
- *withdraw()*: This function allows *TP* to withdraw an amount of Ether corresponding to the ticket price for the tickets in the used_ticket list.

Algorithm 5: Ticket usgae contract.

```
1:      struct ticket_state{
2:          string timestamp;
3:          string signature;
4:          string verification;
5:       }
```

(Continued)

Algorithm 5 (continued)

```
address[]  TP_address;
ticket_state[] request_ticket;
ticket_state[] used_ticket;
mapping(address=>ticket[])submitted_ticket;
 string[] trace_pubkey;

 Function Read_request_ticket(){
      return request_ticket;
 }
 Function Read_used_ticket(){
      return used_ticket;
 }
 Function  Exchanged_request(_timestamp, _signature, _TP_address){
 register ticket_state[_timestamp,_signature, false] to request_ticket;
      register _TP_address to  TP_address;
 }
 Funtion Verify_ticket(_verification[], _trace_pubkey[]) is TSP{
      for(i = 0, i <requet_ticket.length; i + +)
           register _verifiacation[i] to request_ticket[i].verification;
           if _verification is  TRUE
                register requet_ticket[i] to used_ticket;
           else if
                register _trace_pubkey to trace_pubkey;
           end if
           submitted_ticket[TP_addr[i]]=>request_ticket[i];
      end for
 }
 Funtion withdraw(){
      int total_exchange_ticket = 0;
 for (i = 0; i <submitted_ticket[msg.sender].length; i + +)
      if  (submitted_ticket[msg.sender][i].verification==TRUE)
           total_exchage_ticket += 1 ether
      end if
 end for
 payable(msg.sender).transfer(total_exchange_ticket)
 }
```

4.2 A Basic Protocol

The basic protocol for *AnonymousTollPass* model consists of five steps: (1) Registration, (2) *AnonymousTollPass* pre-ticket issuance, (3) Vehicle Entry, (4) Vehicle Exit, (5) Ticket settlement. The list of notations is shown in Table 1. We assume that all participants, such as a toll service provider *TSP*, toll plazas *TP*s, and vehicles *V*s have each own public-private key pair for performing signature algorithms. In this paper, we use the Elliptic Curve Digital Signature Algorithm (ECDSA) [31].

Table 1: List of notations

Notation	Description
V_i	Vehicle i
TSP	Toll service provider
TP_n	The entry point of Toll Plaza
TP_x	The exit point of Toll Plaza
$ticekt_{V_i}$	A ticket generated by V_i
pk_A	Public key of entity A
sk_A	Private key of entity A
PK_N	Public key list for set N vehicles
σ_A	Signature generated by entity A
t	Timestamp
H ()	Cryptographic hash function (e.g., Keccak 256)
$Sign\ (sk_A, m)$	Signature generation algorithm for message m
$Verify\ (pk_A\sigma_A, m)$	Signature verification algorithm for σ_A

4.2.1 Registration

The Toll Service Provider *TSP* registers the identity information of the new participating Toll Plaza TP_n in this registration phase. TP_n submits its public key pk_{TP_n} to the *TSP* for registration. The *TSP* checks whether the submitted pk_{TP_n} is not registered before and adds the public key information to a registered *TP* list. The *TSP* shares the updated list to TP_n and other entities of the *AnonymousTollPass* model.

4.2.2 AnonymousTollPass Pre-Ticket Issuance

The *TSP* issues *AnonymousTollPass* pre-tickets to vehicles Vs who want to purchase tickets. Vs initiate a purchase request to the *TSP* for the desired ticket price. The *TSP* groups Vs who are purchasing *AnonymousTollPass* tickets of the same amount into a N-member ring group. Each $V_i\ (i = 1, 2, \cdots, N)$ makes the payment for the requested ticket and generates a pair of private and public keys.

$$\left(pk_{V_i}, sk_{V_i}\right) \leftarrow GEN\left(1^{\lambda}\right) \tag{1}$$

Each V_i submits the generated pk_{V_i} to the *TSP*. The *TSP* collects pk_{V_i} submitted by each V_i belonging to the ring group.

$$PK_N = \{pk_{V_1}, pk_{V_2}, \ldots, pk_{V_N}\} \tag{2}$$

The *TSP* creates and deploys two smart contracts which are *TicketSaleContract* and *TicketUsage-Contract* on the blockchain networks. It registers *AnonymousTollPass* pre-ticket information of the ring group by running *register_ticket_information( )* function of the *TicketSaleContract*. The *TSP* shares the two smart contract addresses with each ring group member V_i. Each V_i can read PK_N through *Register_ticket_information( )* of the *TicketSaleContract*, which is used for generating a valid *AnonymousTollPass* ticket.

4.2.3 Vehicle Entry

In this phase, V_i receives a proof of entry from the toll plaza TP_n located at the entry zone when it enters the highway to use its *AnonymousTollPass* ticket. When V_i enters the highway, TP_n generates its signature σ_{TP_n} for a timestamp t and sends it to V_i.

$$\sigma_{TP_n} = Sign\left(sk_{TP_n}, t\right) \tag{3}$$

Then, V_i verifies σ_{TP_n} and generates a valid *AnonymousTollPass* ticket by executing the algorithm *ticket* GEN with private key sk_{V_i}, public key set PK_N, and timestamp t as inputs.

$$\{0,1\} = Verify\left(pk_{TP_n}, \sigma_{TP_n}, t\right) \tag{4}$$

$$ticket_{V_i} = ticket\text{GEN}\left(sk_{V_i}, PK_N, t\right) \tag{5}$$

4.2.4 Vehicle Exit

In the Vehicle Exit phase, V_i, who exits the highway, submits an *AnonymousTollPass* ticket to TP_x, and the TP_x verifies it. Firstly, V_i sends $ticket_{V_i}$, σ_{TP_n} and the addresses of the *TicketSaleContract* and *TicketUsageContract* to TP_x. After receiving, TP_x executes the *Read_ticket_information( )* function of the *TicketSaleContract* to confirm the ticket-related data. TP_x verifies the value of σ_{TP_n} and checks that the value of the toll ticket is appropriate for the distance traveled by V_i.

$$\{0,1\} = Verify\left(pk_{TP_n}, \sigma_{TP_n}, t\right) \tag{6}$$

Also, TP_x verifies the $ticket_{V_i}$ by checking if it was generated from the ring group PK_N through Algorithm 2.

$$\{0,1\} = \text{ticketVER}\left(PK_N, ticket_{V_i}\right) \tag{7}$$

Then, TP_x checks information about the used tickets through the *read_used_ticket( )* function. TP_x traces $ticket_{V_i}$ and used tickets to verify whether $ticket_{V_i}$ is a reused ticket using Algorithm 3. If the ticket has not been used before, the output is 'indep'. When the ticket is reused by V_i, the output is pk_{V_i}.

$$\{indep, pk_i\} = \text{ticketTRACE}\left(ticket_{V_i}, ticket_{V_j}\right) \tag{8}$$

All results must be 'indep' to pass the verification. After verification, TP_x shares the information that $ticket_{V_i}$ has been used with other *TP*s by running the *exchange_request( )* function. The transaction of *exchange_request( )* is shown in Fig. 3.

The size of $ticket_{V_i}$ varies depending on the ring size, and it can consume a significant amount of gas fee to store the ticket information in the ledger. To reduce the gas fee, *TP*s share ticket information off-chain and store the hash value of $ticket_{V_i}$, which is $H\left(ticket_{V_i}\right)$, instead of storing $ticket_{V_i}$ on-chain.

4.2.5 Ticket Settlement

In the Ticket Settlement phase, the *TSP* verifies the tickets stored in the request_ticket[] list and processes the settlement for valid tickets. The *TSP* checks the request_ticket[] list through the *read_request_ticket( )* function and validates each ticket in the request_ticket[]. Table 2 shows the results of *read_exchange_request( )*.

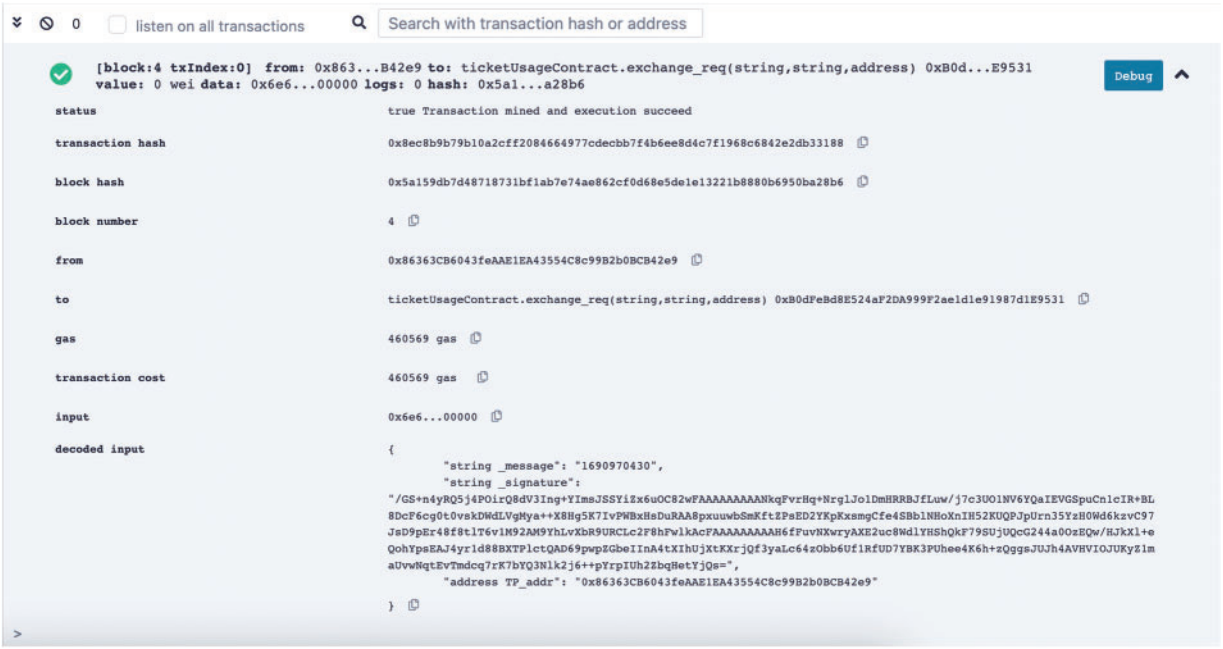

Figure 3: The transaction of *exchange_request( )* on remix IDE

Table 2: Example of read *exchange_request( )* execution result

#	Request_ticket			TP_address
	Timestamp	Signature	Verification	
1.	1690970430	/GS+n4yRQ5j4POirQ8dV3 Ing+YImsJ...	false	0x86363CB6043feAAE1EA4 3554C . . .
2.	169070472	2M8nakcvmF/PUzzbHiL NSviPhb2um...	false	0x75aAcAa07Ab0CcF4bA09 DE98 . . .
3.	1691023804	+vM11PAPaypnv5jn7WS XT7cRGnFV...	false	0x36cEDa5Ea288fcAE4E85d 276098 . . .

For tickets that pass the verification, the *TSP* validates whether the tickets have been reused. Then, the *TSP* executes *verify_ticket( )* function to store the verification result of the toll tickets in the used_ticket[] list. Fig. 4 shows the transaction of *verify_ticket( )*. The updated results are recorded in the blockchain as shown in Table 3. The *TP* can convert the verified tickets into ETH through *withdraw( )* function.

5 Security and Privacy Analysis

5.1 Identity Privacy

The proposed protocol provides privacy-preserving authentication through *AnonymousTollPass* ticket. A curious *TP* can attempt to infer pk_{V_i} for the corresponding $ticket_{V_i}$ submitted by V_i. Also, a curious participant exposes the identity of a vehicle from the list of used tickets recorded on the

blockchain. As for the first case, The *AnonymousTollPass* ticket is generated using a traceable ring signature, and due to the properties of this signature, the identity of the signer (i.e., vehicle) protected during the verification process. Even if the *TSP* knows the pk_{V_i} when purchasing a ticket, it cannot infer pk_{V_i} through the ticket when using the ticket. Furthermore, even members of the same ring group cannot infer which member created each other's ticket. As for the second case, the vehicle pre-pays the ticket price to the *TSP*, which is managed through a smart contract between the *TSP* and *TP*. The vehicle is not a direct participant in the smart contract transactions, so it remains undisclosed regardless of the ticket usage records.

Figure 4: The transaction of *verify_ticket( )* on Remix IDE

Table 3: Example of *verify_ticket( )* execution result

#	Submitted_ticket				Trace_pubkey
	Request_ticket (value)			TP_address (key)	
	Timestamp	Signature	Verification		
1.	1690970430	/GS+n4yRQ5j4POirQ 8dV...	True	0x86363CB6043f eAAE1EA . . .	NULL
2.	169070472	2M8nakcvmF/PUzzb HiLN...	True	0x75aAcAa07Ab0 CcF4bA09 . . .	NULL
3.	1691023804	+vM11PAPaypnv5jn7 WS...	False	0x36cEDa5Ea288f cAE4E85d2 . . .	KFIMLVlu 9HW...

5.2 Ticket Unlinkability

When a vehicle uses multiple *AnonymousTollPass* tickets, if the tickets used by the vehicle are linked, the vehicle's route could be exposed. Therefore, the tickets used by a single vehicle must be unlinkable to each other. A curious participant can try to infer if a V_i has created the same ticket by looking at multiple used ticket lists recorded on the blockchain. Let us assume that V_i used two tickets, $ticket_{V_i}$ and $ticket'_{V_i}$, which are included in the same PK_N. A curious participant can try to infer pk_{V_i} for both tickets using Algorithm 3. When a vehicle purchases a ticket from the *TSP*, a new key pair (pk_{V_i}, sk_{V_i}) is generated and used for each ticket. Therefore, they compute $ticket_{V_i}(m, A_1, c_N, z_n)$ and $ticket_{V_i}\prime\,(m', A'_1, c'_N, z'_n)$ with different sk_{V_i} values during their creation using Algorithm 1. Even if two tickets are traced, comparing $A_0A_1^i$ and $A'_0A'^i_1$ computed through the two tickets, pk_{V_i} is not included in the **TList** which is a temporal list for storing pk_{V_i}s using in Algorithm 3. Additionally, Algorithm 3 is based on tracing tickets within the same ring group, so it cannot be used if two tickets were created based on different ring groups.

5.3 Resistance to Distance Forgery

A malicious V_i can attempt to travel a longer distance than the toll amount they paid by forging their passed TP_n. Also, a malicious *TP* can forge the vehicle's passing of TP_n to claim more ether for a settlement. As for the first case, when the V_i enters the highway, they use the signature σ_{TP_n} for the timestamp from TP_n. The V_i can try to forge σ_{TP_n} when calculating the *AnonymousTollPass* ticket $ticket_{V_i} = ticket\ \text{GEN}\left(sk_{V_i}, PK_N, \sigma_{TP_n}\right)$. However, TP_x can verify σ_{TP_n} of the ticket submitted by V_i, and confirm the location of TP_n that V_i has passed. It allows TP_x to verify if the distance driven by V_i matches the toll amount on the ticket. Consequently, in order for V_i to forge the distance, they would need to forge σ_{TP_n}. Since V_i does not know the private key of TP_n, V_i cannot generate σ_{TP_n}. TP_n generates the σ_{TP_n} based on ECDSA using its private key, which is not disclosed to others. Also, the ECDSA [29], which is proven to be difficult to derive sk_{TP_n} from its pk_{TP_n} based on the hardness of the ECDLP (Elliptic Curve Discrete Logarithm Problem). Since V_i does not know the private key of TP_n, V_i cannot generate σ_{TP_n}. As for the second case, even if *TP* forges σ_{TP_n}, they cannot create the ticket because they are not a group member of the ticket submitted by V_i and thus do not have the private keys held by the group members.

5.4 Abuse of Tickets

The used *AnonymousTollPass* tickets cannot be reused, and it should be possible to reveal the identity of a vehicle that maliciously reused a ticket. Malicious V_i can attempt to reuse a ticket already used. After using $ticket_{V_i}(m, \sigma)$, V_i can generate a new ticket $ticket_{V_i}(m', \sigma')$ using the new σ'_{TP_n} received from TP_n in order to reuse the ticket. However, during ticket validation, the $ticket_{V_i}(m, \sigma)$ used and stored in the used ticket list is compared with the new $ticket_{V_i}(m', \sigma')$ submitted by V_i through the tracing process using Algorithm 3. In this process, tickets generated with the same private key have the same results of $A_0A_1^i = \mathrm{A}_0 * \dfrac{h^{x_i}}{\mathrm{A}_0}$ and $A'_0A'^{i'}_1 = A'_0 * \dfrac{h^{x_i}}{A'_0}$ revealing pk_{V_i}. Since the information about the used tickets is recorded on the blockchain and cannot be altered, reusing them is impossible.

6 Performance Evaluation

Previous blockchain-based ETC Systems [14–16,25–28] exposed vehicles' toll usage information directly on the blockchain. In [16], they proposed the first ETC System that provides privacy for vehicles. Related studies of blockchain-based ETC Systems are outlined in Table 4. In this section,

we demonstrate that our proposed smart contract systems for the *AnonymousTollPass* model have more efficient performance compared to [16].

Table 4: A comparative study of blockchain-based ETC Systems

Ref.	Application	Blockchain platform	Applied algorithm	Smart contract	Privacy
[14]	A vehicle platoon group authentication scheme	Ethereum	Aggregate signature [32]	O	X
[15]	A toll payment scheme	Ethereum	–	O	X
[25]	A PBFT based ETC System	Hyperledger fabric [33]	–	X	X
[26]	A heterogeneous public edge sharing scheme	Ethereum	Greedy algorithm [34]	O	X
[27]	A dynamic toll pricing scheme	Ethereum	Jeffry's prior [35] & LSTM	O	X
[28]	A vehicle behavior management mechanism	Hyperledger fabric	–	X	X
[16]	A privacy-preserving payment scheme	Blockmaze [36]	zk-SNARK [37], & Group signature [38]	O	O
Ours	An anonymous toll ticket model	Ethereum	Traceable ring signature [17]	O	O

We implemented our smart contract systems by using Remix v0.8.7 [39], and then deployed the smart contract using Ganache v2.5.4 [40] and a virtual Ethereum test blockchain. The deployed smart contract was executed using Python v3.9.1 [41]. We utilized a laptop equipped with Mac OS Monterey v12.5.1, an Intel Core i7 1.2 GHz CPU and 16 GB of RAM for the experiments. Our experimental scenario refers to the actual ETC System in South Korea called Hipass to realistically set the maximum allowable time parameters for *AnonymousTollPass* ticket generation and verification. In South Korea's ETC System, there is a speed limit of 30 km/h near the tollgate and DSRC (Dedicated Short Range Communication) based on IEEE 802.11 standard with a communication range of up to 90 m. So, we assume that the vehicle communicates with the *TP* from a distance of approximately 30 m, traveling at a speed of 30 km/h, allowing for communication for 3.5 s. Therefore, to ensure smooth ETC System operation without delays, the time required for *AnonymousTollPass* ticket generation and verification must be within 3.5 s.

Fig. 5 shows the execution time of the *ticket*TRACE phase according to the number of used tickets when the ring size is 50. As the number of used tickets increases, the time also increases because more

tickets must be checked for reuse. The time it took to check the reusability of the last ticket when there were 49 used tickets was approximately 2.5 s.

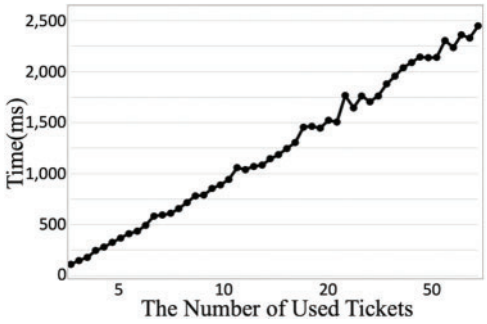

Figure 5: The ticket verification time according to the number of used tickets

Fig. 6 shows the execution time of each phase according to the ring size. As the ring size increases, the key size used in the *ticket*GEN and *ticket*VER phases increases, resulting in increased computation time. When the ring size is 50, the *ticket*GEN phase takes 81 ms, the *ticket*VER phase takes 79 ms, and the *ticket*TRACE phase takes 1737 ms. The total time required is approximately 3.3 s, which is less than the previously assumed 3.5 s requirement. Therefore, our *AnonymousTollPass* system can be applied effectively in a real-world situation.

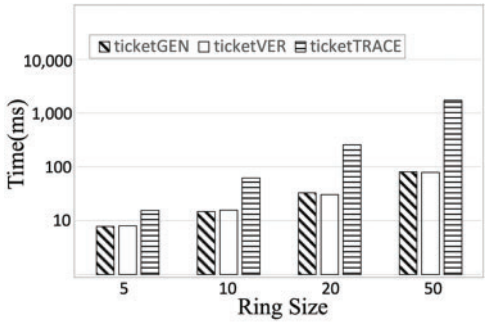

Figure 6: The performance of each *AnonymousTollPass* algorithm according to the ring size

We also measured the key size for each phase of the *AnonymousTollPass* model. The result is shown in Fig. 7. The key size increases linearly as the ring size increases. When the ring size is 50, the key size in the *ticket*GEN phase is about 8.6 MB, and in the *ticket*VER phase, it is about 13 MB, which is a reasonable size.

Table 5 shows the gas fees consumed by the three main functions of the *TicketSalesContract* and *TicketUsageContract* in our smart contract system. For *register_ticket_information( )*, the gas fee increases as the ring size increases, with 387,824 for a ring size of 5 and 2,056,796 for a ring size of 50, as more information is registered with larger ring sizes. For *exchange_request( )*, the gas fee is the same for each ticket, with 17,260 gas consumed when the function is executed. For *verify_ticket( )*, the gas fee increases slightly as the ring size increases, with 5,467,598 gas consumed for a ring size of 5 and 5,524,174 for a ring size of 50.

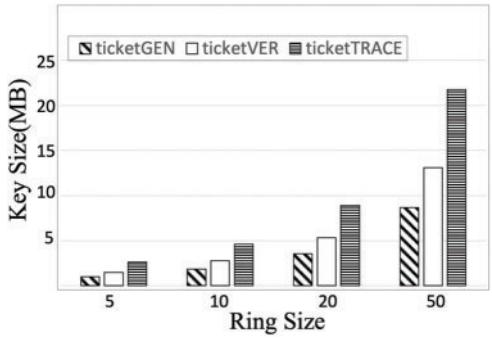

Figure 7: The key size of each *AnonymousTollPass* algorithm according to the ring size

Table 5: The gas consumption of each function according to the ring size

Ring size				
Function	5	10	20	50
register_ticket_information( )	387,824	402,124	733,043	2,056,796
exchange_request( )	17,260			
verify_request( )	5,467,598	5,473,886	5,486,447	5,524,174

Table 6 compares the total gas fees and ether costs consumed by the functions for each number of vehicles with Vehiclock [16]. The gas fees consumed in the proposed smart contract system were calculated based on the average gas fee consumed when executing functions of the *TicketSalesContract* and *TicketUsageContract*, with ether cost calculated by converting 1 gas to 1 gwei. The total gas fee is calculated through *register_ticket_information( )* + (*exchange_request( )* ∗ ring size) + *verify_ticket( )*. As the ring size increases, more information about the ring members is stored in the blockchain and the number of used tickets increases, so the total gas fee increases as well. However, setting the ring size to 50 rather than 10 for creating 5 rings can reduce the total gas fee. In [16], all signature generation and verification processes are performed on the smart contract and recorded in the blockchain.

Table 6: The gas consumption and ether cost according to the ring size

		[16]	Proposed protocol
5	gas fee	9,504,775	5,941,722
	ether cost	0.009504	0.005941
10	gas fee	19,009,550	6,048,620
	ether cost	0.019009	0.006048
20	gas fee	38,019,100	6,564,690
	ether cost	0.038019	0.006564
50	gas fee	95,047,750	8,443,970
	ether cost	0.095047	0.008443

However, in our proposed system, only the hash value of the ticket signature and the verification result are recorded on the blockchain, and the ticket generation and verification are performed in the local environment, reducing the computational load of running the smart contract. As a result, when the ring size is 50, our *AnonymousTollPass* smart contract system consumes approximately 10% of the gas required in [16]. This result demonstrates the efficiency of our smart contract system.

7 Case Study Discussion

In this section, we discuss the security and privacy levels by analyzing a case study in South Korea utilizing our *AnonymousTollPass* model. *TP*, *TSP*, and V, which are participants in the ETC System, cannot identify the V_i from a particular *AnonymousTollPass* ticket. But a curious participant can infer the route from the V_i based on ticket usage history recorded on the blockchain.

When a V uses a ticket, the probability that the curious participant can correctly identify the route of a V_i is $1/N$ when the ring size is N. Since the *AnonymousTollPass* ticket is for one-time use, even if a V's driving route is inferred with a probability of $1/N$, it is difficult to link the V 's driving route to personal information. However, for users who drive a consistent route, such as commuting to work or making regular hospital visits, if their driving route is exposed, it can be linked to personal information such as their lifestyle patterns, workplace, and residence. The V generates a new key pair for each ticket, and the ring members change as well. Therefore, for participants except the *TSP* that sells the tickets, it is difficult to infer the public keys used by the same V from the list of used tickets. Even for the *TSP*, it becomes more difficult to infer the V's driving route as the number of tickets used by the V increases. The probability of determining the route when a V uses the ticket r times can be expressed as $(1/N)^r$. As N and r increase, the probability becomes smaller, making it hard to link all the tickets used by a vehicle. Based on this, we calculated the value of r satisfying the security level requirement for each ring size, as shown in Table 7.

Table 7: The number of *AnonymousTollPass* ticket uses satisfying the security level according to the ring size

Ring size	10	20	50
Security level 112	34	26	20
Security level 128	39	30	23
Security level 192	58	45	35

For instance, if a user commute using a vehicle five days a week, he uses 10 tickets per week and 40 tickets per month. For a ring size of 50, this satisfies 192-bit security, which is the highest security level. Thus, when considering real-world driving scenarios for regularly using the ETC System, it is difficult for the participants to determine a single vehicle's route through the used ticket list recorded in the blockchain.

$$(1/2)^{192} > (1/50)^{35} \tag{9}$$

There is no restriction on applying the proposed *AnonymousTollPass* model when launching a new ETC System for toll plazas where none currently exist. However, for countries that already provide ETC Systems, to determine whether different routes taken by the Vs can be grouped together for the same ticket, we examined the number of routes with the same toll fee.

To verify this, we used the ETC toll data of South Korea as a test case. We represented the toll fees for each vehicle type based on the weekday toll data provided by *Korea Expressway Corporation* in 2022 [10] in Fig. 8. When counting all toll routes for each vehicle type within the same price range, we found that the number of routes varied. In the price range of \$6 ∼ \$7 and \$7 ∼ \$8, we found that the maximum number of routes was approximately 80,000. Even in the \$0 ∼ \$1 section with the fewest routes, there are 4,122 routes, which is reasonable enough to create a ring group for vehicles passing through that section.

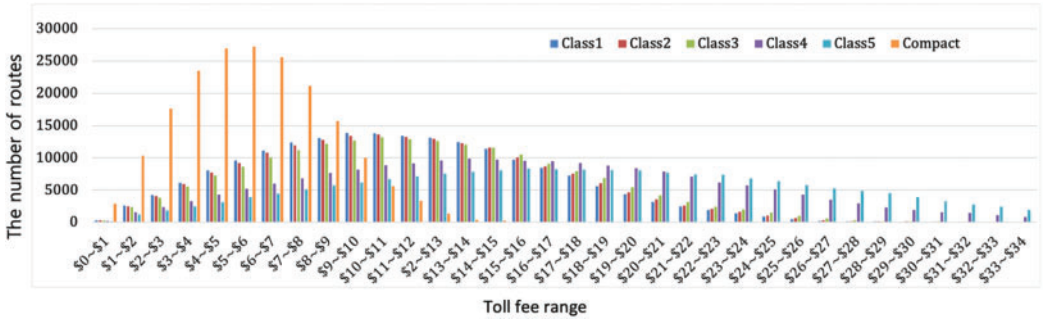

Figure 8: The number of routes by toll fee range

Additionally, we compared popular routes that start in Seoul, and travel to nearby cities. Fig. 9 represents the routes marked as D1∼ D8, which correspond to eight cities (Siheung-si, Gwangmyeong-si, Gunpo-si, Hwaseong-si, Pyeongtaek-si, Yongin-si, AnSeong-si, Gwangju-si), displayed on a map. Each destination is located approximately 20 to 35 km away from the origin. Then, we compared the actual toll fees for each of the 8 destinations and visually indicated the grouping of tickets using colors in Table 8. Through this, we confirmed that the destinations are different, and therefore it is possible to purchase the same ticket.

Figure 9: Routes from Seoul to surrounding cities

Table 8: The toll fees of each route depicted in Fig. 9 according to vehicle classes

Destination	Class 1	Class 2	Class 3	Class 4	Class 5	Compact
D1	1.792	1.794	1.794	2.184	2.496	0.897
D2	1.482	1.56	1.56	1.872	2.028	0.741
D3	2.028	2.028	2.106	2.574	2.964	1.014
D4	2.73	2.808	2.886	3.588	4.134	1.365
D5	3.51	3.588	3.666	4.524	5.07	1.755
D6	3.588	3.666	3.744	4.758	5.538	1.794
D7	5.6	5.7	5.9	7.7	8.9	2.8
D8	3.432	3.432	3.588	4.542	5.226	1.7

8 Conclusion

In this paper, we proposed a privacy-preserving blockchain-based toll payment model for ETC Systems, called *AnonymousTollPass* model. Vehicles use disposable *AnonymousTollPass* tickets generated based on TRS, which provide anonymity to the vehicles but can reveal the identities of malicious vehicles. Moreover, the usage history of *AnonymousTollPass* tickets and information about malicious vehicles are shared between the ETC service provider and toll plazas through smart contracts to ensure the integrity of the ticket usage history. To demonstrate the effectiveness of the proposed model, we calculated the gas fee the cost of executing the smart contract and checked that it consumes less gas compared to previous work. Also, we utilized the ETC System in South Korea as a test case and demonstrated the practicality of the proposed model in real-world situations. Additionally, *AnonymousTollPass* model can be utilized in ETC Systems and various payment systems that require anonymity in the future.

Acknowledgement: We are grateful to the editors and reviewers for their insightful feedback on this work. We would also like to thank the National Research Foundation of Korea (NRF) and the Korea government (MSIT) for their support.

Funding Statement: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C1095591).

Author Contributions: Authors confirm contribution to the paper as follows: Study conception and design: Jane Kim, Seung-Hyun Seo; data collection: Jane Kim, Soojin Lee; analysis and interpretation of results: Jane Kim, Soojin Lee; draft manuscript preparation: Jane Kim, Chan Yeob Yeun, Seung-Hyun Seo. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used in the experiments should be clearly cited in the article.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Lee and S. H. Seo, "Design of a two layered blockchain-based reputation system in vehicular networks," *IEEE Trans. Vehicular Technol.*, vol. 71, no. 2, pp. 1209–1223, Nov. 2021. doi: 10.1109/TVT.2021.3131388.

[2] N. Zhao, X. Zhao, M. Chen, G. Zong, and H. Zhang, "Resilient distributed event-triggered platooning control of connected vehicles under denial-of-service attacks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 6, pp. 6191–6202, Jun. 2023. doi: 10.1109/TITS.2023.3250402.

[3] D. Das, S. Banerjee, P. Chatterjee, U. Ghosh, and U. Biswas, "A secure blockchain enabled v2v communication system using smart contracts," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 4, pp. 4651–4660, Dec. 2023. doi: 10.1109/TITS.2022.3226626.

[4] A. Badshah *et al.*, "AAKE-BIVT: Anonymous authenticated key exchange scheme for blockchain-enabled internet of vehicles in smart transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 1739–1755, Feb. 2023. doi: 10.1109/TITS.2022.3220624.

[5] F. Don, "Electronic toll collection: An introduction and brief look at potential vulnerabilities," Rockville, MD, USA: SANS Institute, 2004. Accessed: Apr. 04, 2024. [Online]. Available: https://www.giac.org/paper/gsec/3854/electronic-toll-collection-introduction-potential-vulnerabilities/106183

[6] National Electronic Toll Collection (NETC) Fastag. Accessed: Jun. 29, 2023. [Online]. Available: https://www.npci.org.in/what-we-do/netc-fastag/product-overview

[7] E-toll Collection System. Accessed: Jun. 29, 2023. [Online]. Available: https://www.nadra.gov.pk/e-toll-collection-system/

[8] ReportLinker. Accessed: Apr. 01, 2024. [Online]. Available: https://www.reportlinker.com/p05069898/Global-Electronic-Toll-Collection-ETC-Systems-Industry.html?utm+source=GNW

[9] E-ZPass GROUP. Accessed: Mar. 19, 2023. [Online]. Available: https://www.e-zpassiag.com/

[10] Korea Expressway Corporation. Accessed: Jul. 11, 2023. [Online]. Available: https://www.hipass.co.kr/main.do#FamilySite

[11] VETC: Vietnam Electronic Toll Collection. Accessed: Mar. 19, 2023. [Online]. Available: https://vetc.com.vn/

[12] C. Hu, X. Cheng, Z. Tian, J. Yu, and W. Lv, "Achieving privacy preservation and billing via delayed information release," *IEEE/ACM Trans. Netw.*, vol. 29, no. 3, pp. 1376–1390, Mar. 2021. doi: 10.1109/TNET.2021.3063102.

[13] THE DRIVE. Accessed: Jul. 18, 2023. [Online]. Available: https://www.thedrive.com/news/ai-traffic-surveillance-can-link-your-driving-patterns-to-criminal-behavior/

[14] Z. Ying, L. Yi, and M. Ma, "BEHT: Blockchain-based efficient highway toll paradigm for opportunistic autonomous vehicle platoon," *Wirel. Commun. Mob. Comput.*, vol. 2020, pp. 1–13, Sept. 2020. doi: 10.1155/2020/8868656.

[15] X. Deng and T. Gao, "Electronic payment schemes based on blockchain in VANETs," *IEEE Access*, vol. 8, pp. 38296–38303, Feb. 2020. doi: 10.1109/ACCESS.2020.2974964.

[16] Y. Guo, Z. Wan, H. Cui, X. Cheng, and F. Dressler, "Vehicloak: A blockchain-enabled privacy-preserving payment scheme for location-based vehicular services," *IEEE Trans. Mob. Comput.*, Nov. 2022. doi: 10.1109/TMC.2022.3193165.

[17] E. Fujisaki and K. Suzuki, "Traceable ring signature," in *Public Key Cryptography–PKC 2007: 10th Int. Conf. on Practice and Theory*, Beijing, China, Springer, 2007, pp. 181–200.

[18] R. Borges, F. Sebé, M. Valls, "An anonymous and unlinkable electronic toll collection system," *Int. J. Inf. Secur.*, vol. 21, no. 5, pp. 1151–1162, Aug. 2022. doi: 10.1007/s10207-022-00604-8.

[19] B. Aiello, Y. Ishai, and O. Reingold, "Priced oblivious transfer: How to sell digital goods," in *Advances in Cryptology—EUROCRYPT 2001: Int. Conf. on the Theory and Application of Cryptographic Techniques*, Innsbruck, Austria, Springer, 2001, pp. 119–135.

[20] M. Randriamasy, A. Cabani, H. Chafouk, and G. Fremont, "Geolocation process to perform the electronic toll collection using the ITS-G5 technology," *IEEE Trans. Vehicular Technol.*, vol. 68, no. 9, pp. 8570–8582, Jul. 2019. doi: 10.1109/TVT.2019.2931883.

[21] N. Aung, W. Zhang, S. Dhelim, and Y. Ai, "T-coin: Dynamic traffic congestion pricing system for the internet of vehicles in smart cities," *Information*, vol. 11, no. 3, pp. 149, Mar. 2020. doi: 10.3390/info11030149.
[22] R. Böhme, N. Christin, B. Edelman, and T. Moore, "Bitcoin: Economics, technology, and governance," *J. Econ. Perspect.*, vol. 29, no. 2, pp. 213–238, 2015. doi: 10.1257/jep.29.2.213.
[23] T. McGhin, K. R. Choo, C. Z. Liu, and D. He, "Blockchain in healthcare applications: Research challenges and opportunities," *J. Netw. Comput. Appl.*, vol. 135, pp. 62–75, Jun. 2019. doi: 10.1016/j.jnca.2019.02.027.
[24] Y. Bai, Q. Hu, S. H. Seo, K. Kang, and J. J. Lee, "Public participation consortium blockchain for smart city governance," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 2094–2108, Jun. 2021. doi: 10.1109/JIOT.2021.3091151.
[25] W. Y. Chiu and W. Meng, "EdgeTC–A PBFT blockchain-based ETC scheme for smart cities," *Peer-to-Peer Netw. Appl.*, vol. 14, pp. 2874–2886, Mar. 2021. doi: 10.1007/s12083-021-01119-0.
[26] B. Xiao, X. Fan, S. Gao, and W. Cai, "EdgeToll: A blockchain-based toll collection system for public sharing of heterogeneous edges," presented in Conf. IEEE INFOCOM WKSHPS., Paris, France, Apr. 29–May 02, 2019, pp. 1–6.
[27] A. Shukla, P. Bhattacharya, S. Tanwar, N. Kumar, and M. Guizani, "DwaRa: A deep learning-based dynamic toll pricing scheme for intelligent transportation systems," *IEEE Trans. Vehicular Technol.*, vol. 69, no. 11, pp. 12510–12520, Sep. 2020. doi: 10.1109/TVT.2020.3022168.
[28] J. Wang, R. Zhu, T. Li, F. Gao, Q. Wang, and Q. Xiao, "ETC-oriented efficient and secure blockchain: Credit-based mechanism and evidence framework for vehicle management," *IEEE Trans. Vehicular Technol.*, vol. 70, no. 11, pp. 11324–11337, Sep. 2021. doi: 10.1109/TVT.2021.3116237.
[29] Patch. Accessed: Nov. 16, 2023. [Online]. Available: https://patch.com/virginia/oldtownalexandria/here-s-when-expect-most-thanksgiving-traffic-northern-va-dc
[30] Public Data Portal DATA.GO.KR. Accessed: Aug. 02, 2023. [Online]. Available: https://www.data.go.kr/en/%20data/15043728/fileData.do
[31] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, pp. 36–63, Jan. 2001. doi: 10.1007/s102070100002.
[32] L. Cheng, Q. Wen, Z. Jin, H. Zhang, and L. Zhou, "Cryptanalysis and improvement of a certificateless aggregate signature scheme," *Inf. Sci.*, vol. 295, pp. 337–346, Feb. 2015. doi: 10.1016/j.ins.2014.09.065.
[33] E. Androulaki *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. Thirteenth EuroSys Conf.*, Porto, Portugal, 2018, pp. 1–15.
[34] J. Edmonds, "Matroids and the greedy algorithm," *Math. Program.*, vol. 1, pp. 127–136, Dec. 1971. doi: 10.1007/BF01584082.
[35] J. F. Shortle, J. M. Thompson, D. Gross, and C. M. Harris, "General models and theoretical topics," in *Fundamentals of Queueing Theory*, 5th ed. Hoboken, NJ, USA: John Wiley & Sons, pp. 313–361, 2018, vol. 399, pp. 313–361.
[36] Z. Guan, Z. Wan, Y. Yang, Y. Zhou, and B. Huang, "BlockMaze: An efficient privacy-preserving account-model blockchain based on zk-SNARKs," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 3, pp. 1446–1463, Sep. 2020. doi: 10.1109/TDSC.2020.3025129.
[37] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," *Commun. ACM*, vol. 59, no. 2, pp. 103–112, Jan. 2016. doi: 10.1145/2856449.
[38] D. Chaum and E. Van Heyst, "Group signatures," in *EUROCRYPT'91*. Brighton, UK, Apr. 8–11, 1991, pp. 257–265.
[39] Remix. Accessed: Jun. 12, 2023. [Online]. Available: https://remix.ethereum.org/
[40] Trufflesuite. Accessed: Jun. 12, 2023. [Online]. Available: https://trufflesuite.com/ganache/
[41] G. V. Rossum and F. L. Drake, Top-level components. in *Python Reference Manual*, 1st ed. Amsterdam, Netherlands: Centrum voor Wiskunde en Informatica, 1995, vol. 111, pp. 1–52.