Smart Ticketing For Journey Privacy

Jinguang Han, Liqun Chen, Steve Schneider, Helen Treharne and Stephan Wesemeyer

Surrey Centre for Cyber Security, Department of Computer Science, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom

{j.han,liqun.chen,s.schneider,h.treharne,s.wesemeyer}@surrey.ac.uk

Abstract. Smart ticketing can provide more convenience and flexibility for customers, support seamless connections and reduce ticket queues at entries and exits. Nevertheless, privacy issue has been the primary concern of smart ticketing users. Privacy-preserving smart ticketing schemes have been proposed to protect customers' identities information, but customers' journey information has not been considered extensively. However, journey information is sensitive for customers since it can be used to track and locate customers.

In this paper, we propose a smart ticketing for journey privacy scheme to address the journey privacy issue in smart ticketing. The proposed scheme captures the follow features: (1) For a journey, only one ticket is issued to a user, even if he/she needs multiple transits; (2) Users can purchase tickets from the ticket seller anonymously without releasing anything about their identities information, namely the ticket seller cannot detect whether two journeys are from two different users or a same user; (3) Ticket verifiers can be convinced that whether a user is authorised to pass the stations and cannot profile the user's journey when they collude; (4) A trusted party named police is authorised to trace a user's journey; (5) The journey is fixed to prevent a user from using a cheaper ticket to have a long journey when there are multiple hops between the starting station and the destination station. To the best of our knowledge, it is the *first* smart ticketing scheme which enables users to control release their journey information and is formally treated in term of definition, security model and security poof.

Key words: Smart Ticketing, Security, Privacy, Journey Awareness, Anonymity

1 Introduction

In paper-based ticketing systems, service information must be printed on tickets clearly and customers must go to ticket offices/machines to purchase and take tickets. When validating tickets, manual checking is required. This results in long queues and crowding. Smart ticketing is a new technology which aims to improve customers' experience by simplifying ticket purchase and facilitating ticket validation. Especially, with the advent of smart phones and contactless bank cards to pay for services, smart ticketing is fundamentally changing the methods for which we pay and use public transport services.

Since its convenience and real time information provision, smart ticketing attracts lots of attentions from research community, industries and governments. Nevertheless, privacy issues have been the primary concern of customers. Smart ticketing schemes designed to protect users' personal identity information have been proposed, but journey privacy has not been focused extensively. However, journey information is sensitive for customers since malicious parties can use it to track and infer customers' lifestyles, private businesses, relationships, health condition, *etc.* Hence, it is interesting and important to construct smart ticketing schemes where users' journey information can be protected. Due to one ticket needs to be validated by multiple verifiers in a journey with multiple transmits, how to enable users to control release their journey information remains a challenging and interesting problem.

1.1 Related Work

Public transport is one of the sectors where smart ticketing is adopted. Mut-Puigserver *et al.* [28] surveyed electronic ticketing (e-ticketing) schemes, and summarised the security requirements and functional requirements. The security requirements consist of integrity, authenticity, non-repudiation of origin/receipt, unforgeability, anonymity, transferability, *etc.* The function requirements include expiry date, online/offline verification, portability, reduced size, flexibility, ease of use, efficiency, availability, *etc.*

Considering privacy issues, privacy-preserving e-ticketing (PPET) schemes [25,32,38,39] have been proposed. In [20] , Fan and Lei presented an e-ticketing scheme fore electronic election based on blind signature. In [20], each voter registers to an authority and obtains multiple tokens. For each election, a voter authenticates himself/herself to the authority by submitting a token. Nakanishi *et al.* [29] proposed an electronic coupon (e-coupon) scheme where group signature is applied to protect users' privacy. The e-coupons are unlinkable, while the anonymity of users can be revoked. Song and Korba [36] proposed an e-ticketing scheme for pay-TV systems. In this scheme, each user generates multiple secret-public key pairs and obtains a ticket for each of his/her secret-public key pair. To subscribe a TV channel/program, the user sends one ticket to the provider. Quercia and Hailes [32] proposed an e-ticketing scheme for mobile transactions where both limited-used tickets and unlimited-used tickets were considered. Nevertheless, both the formal definition and security model were not defined in schemes [20,29,36,32]. Arfaoui *et al.* [2] first improved the efficiency of the the set-membership proof scheme [11], and then proposed an m-ticketing scheme where each ticket can be anonymously used up to the maximum number. Both he definition and security model of m-ticketing schemes were formalised, but the security proof was sketchy.

Some PPET schemes considered the location information of customers. Vives-Guasch [37] proposed an automatic fare collection (AFC) system where the group signature scheme [9] was used to provide unlinkability and revocable anonymity. After registration to a trusted third party (TTP), each user gets a group member credential. At the entry, each user proves to the ticket verifier that he/she has obtained a credential from the TTP, and obtains an entrance ticket which is a signature on a timestamp, the identity of the verifier at the entry, *etc.* At the exit, each user sends his/her entrance ticket to the ticket verifier, and obtains an exist ticket which is signature on a timestamp, the identity of the verifier at the exit, *etc.* The fare is calculated according to the identities of verifiers at the entry and exit.

Gudymenko [23] proposed an e-ticketing framework which supports fine-granular billing and local validation. To prevent verifiers tracking tickets and implement fine-granular billing, the ticket seller generates different pseudonyms for different ticket verifiers when issuing tickets to users. This pseudonyms can only be lined by the ticket seller. For each ticket validation, a user uses one pseudonym. When calculating the fare, the ticket seller correlates the different pseudonyms in the ticket. To revoke a location from a ticket, accumulator and anonymous blacklisting techniques were adopted. In [23], both the security model and security proof were not mentioned.

Kerschbaum *et al.* [24] proposed a privacy-preserving billing for e-ticketing scheme in public transport. In [24], the authors first showed that it is easy to obtain a traveler's journey information in Singapore's EZ-Link system [19], and then proposed an encrypted bill processing of travel records where each location is expressed as a bit vector. At the entry, all the identifiers of locations which the user will pass are encrypted by using the Paillier's homomorphic encryption [30]. To validate a ticket, the user proves that his/her identity is included in the ticket. When billing the ticket at the exit, the ticket verifier first executes a linear combination on the encrypted locations, and then decrypt the ciphertext to obtain the total fare which the user should pay. The formal definition and security model were not described and the security proof was sketchy.

Rupp *et al.* [33,34] proposed a privacy-preserving pre-payments with refunds scheme which was derived from the e-cash scheme [10] and the signature scheme [8]. In this scheme, users first generate their secret-public key pairs, and register to the transportation authority (TA) by using their public keys. When buying a ticket, each user proves to the ticket seller that he/she is a legal user by proving the knowledge of his/her secret key. If the proof is correct, the vendor generates a ticket for the user. The ticket includes the public key of the user and is an extended coin of the e-cash scheme [10]. When using a ticket, the user shows the ticket to the ticket verifier (reader) and proves the ownership of the ticket by proving the knowledge of his secret key. If the ticket verification is successful, the ticket verifier generates a refund calculation token (RCT) for the user. The RCT consists of an trip authorisation token (TAT) which is a blind signature on his/her public key, the timestampe and the ticket verifier's identity. At the exit, the user proves the ownership of his/her TAT and submits RCT to the ticket verifier (reader). The ticket verifier generates a refunds token (RT) for the user according the identity of the ticket verifier included in the TAT. Finally, the user can use the RT to get refunds. Since the ticket verifier at the exit knows the identity of the ticket verifier at the entrace, it knows the user's journey. The security model of this scheme was fomalised by using the ticket authority security game and users' privacy game. Nevertheless, the security proof of this scheme is sketchy, instead of formal reduction.

Milutinovic *et al.* [27] proposed a public transport ticketing scheme where commitment scheme [31], partial blind signature scheme [1], zero-knowledge proof of knowledge [16] and anonymous credential scheme [14] are used to protect users privacy. In this scheme, each user first generates his/her secret-public key pair, and registers to the ticketing system operator (TSO) to obtain a credential. Then, the user can recharge his/her e-purse or buy travel products from public transport operators (PTOs). After recharging his/her e-purse or buying travel products, the user is issued some e-tokens which are unlinkable and partial blind signatures on the commitment of his/her secret key and specific information including validity period, denomination, product types, *etc.* Notably, both TSO and PTOs cannot link the issued e-tokens to the user's identity information. When beginning a trip, the user proves to the TSO that he/she has valid e-tokens and sends the starting station to the TSO. If the proof is correct, the TSO generates a trip-begin ticket for the user and deletes the e-token from the user's device (app). The ticket is a signature on the start location, timestamp and reduction information. At the end of the trip, the user sends his/her trip-begin ticket to the PTO. The PTO calculates the fare according to the starting station and the destination station, and generates a trip-end ticket which is a signature on the fare, veherical number, nonce, begin time and end time. The user sends his/her trip-end ticket to the TSO who reduces the fare from the his/her e-purse. In the case that there is a random trip inspection, each user should submit this/her trip-begin ticket to the inspection authority and proves that he/she has obtained a valid credential from the TSO. Furthermore, to prevent sharing a ticket, the user needs to show the picture attribute included in the credential. Hence, the inspection authority knows some information of the user's journey, at least the starting location. This scheme provides good features, but it was not formally treated in term of definition, security model and security proof.

1.2 Contributions

Privacy-preserving e-ticketing/smart ticketing schemes have been proposed to protect users' privacy. However, how to control release a user' journey information to different verifiers has not been considered.

In this paper, we propose a smart ticketing for journey privacy scheme. The proposed scheme can provide the following features: (1) For a journey, only one ticket is issued to a user, even if he/she needs multiple transits; (2) Users can purchase tickets from the ticket seller anonymously without releasing anything about their personal identity information, namely the ticket seller cannot detect whether two journeys are from two different users or a same user; (3) Ticket verifiers can be convinced that whether a user is authorised to pass the stations and cannot profile the user's journey even if they collude; (4) For public safety, a trusted party named police is authorised to trace a user's journey if required; (5) The journey in a ticket is fixed to prevent a user from using a cheaper ticket to have a long journey when there are multiple hops between the starting station and the destination station. To the best of our knowledge, it is the *first* smart ticketing scheme which enables users to control release their journey information and is formally treated in term of definition, security model and security poof.

1.3 Paper Organisation

This paper is organised as follows. The preliminaries are introduced in Section 2. In Section 3, a concrete smart ticketing for journey privacy scheme is proposed. We analyse the performance of our scheme in Section 4. In Section 5, the security proof of our scheme is formally presented. Finally, Section 6 concludes this paper.

2 Preliminaries

In this section, we introduce the preliminaries used throughout this paper, including formal definition, security model, bilinear group, complexity assumptions, zero-knowledge proof and BBS+ signature. The syntax is summarised in Table 1.

A smart ticketing for journey privacy scheme consists of five entities: central authority $\mathcal{CA}$, ticket seller $\mathcal{S}$, user $\mathcal{U}$, ticket verifier $\mathcal{V}$ and police $\mathcal{P}$. $\mathcal{CA}$ initialises the scheme and issues credentials to $\mathcal{S}$, $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{P}$, respectively. When buying a ticket, $\mathcal{U}$ sends $\mathcal{S}$ his/her journey information J_U consisting of verifiers' identities . For each verifier $\mathcal{V}$ with identity $ID_V \in J_U$ and the police $\mathcal{P}$, $\mathcal{S}$ generates authentication tags Tag_V and Tag_P, respectively. The ticket for $\mathcal{U}$ is $T_U = ((Tag_V)_{ID_V \in J_U}, Tag_P)$. When being verified by $\mathcal{V}$ with $ID_V \in J_U$, $\mathcal{U}$ sends the authentication tag Tag_V to $\mathcal{V}$. If Tag_V is valid, $\mathcal{U}$ is authorised to pass; otherwise, $\mathcal{U}$ is not authorised to pass. In the case that $\mathcal{U}$'s journey needs to be traced, $\mathcal{P}$ can detect all the verifiers' identities included in T_U by using the authentication tag Tag_P.

Fig. 1 shows the workflow of our smart ticketing for journey privacy scheme.

2.1 Formal Definition

The definition of smart ticketing for journey privacy is formalised by the following five algorithms:

- $\mathsf{Setup}(1^\ell) \rightarrow (MSK, PP)$. $\mathcal{CA}$ inputs a security parameter 1^ℓ, and outputs the master secret key MSK and the public parameters PP.
- Regist: This algorithm consists of the following three sub-algorithems:
  1. $\mathsf{T\text{-}S\text{-}Reg}$ $(\mathcal{S}(ID_S, SK_S, PK_S, PP) \leftrightarrow \mathcal{CA}(MSK, PK_S, PP)) \rightarrow (\sigma_S, (ID_S, PK_S))$. This is an interactive algorithm executed between $\mathcal{CA}$ and $\mathcal{S}$. $\mathcal{S}$ runs the secret-public key pair generation algorithm $\mathcal{KG}(1^\ell) \rightarrow (SK_S, PK_S)$, inputs its identity ID_S, secret-public key pair (SK_S, PK_S) and the public parameters PP, and outputs a credential σ_S. $\mathcal{CA}$ inputs the master secret key MSK, $\mathcal{S}$'s public key PK_S and the public parameters PP, and outputs the identity ID_S and the public key PK_S.

Table 1. Syntax Summary

1^ℓ	A security parameter
$\mathcal{CA}$	Central authority
$\mathcal{S}$	Ticket seller
$\mathcal{V}$	Ticket verifier
$\mathcal{U}$	User
$\mathcal{P}$	Police
ID_S	The identity of $\mathcal{S}$
ID_V	The identity of $\mathcal{V}$
ID_U	The identity of $\mathcal{U}$
ID_P	The identity of $\mathcal{P}$
$\epsilon(\ell)$	A negligible function in ℓ
σ_S	The credential of $\mathcal{S}$
σ_V	The credential of $\mathcal{V}$
σ_U	The credential of $\mathcal{U}$
σ_P	The credential of $\mathcal{P}$
J_U	The set of $\mathcal{U}$'s journey consisting of verifiers
Ps_U	A set of pseudonyms of $\mathcal{U}$
Ps_V	The pseudonym generated for the verifier $\mathcal{V}$
Tag_V	An authentication tag for the verifier $\mathcal{V}$
Tag_P	An authentication tag for $\mathcal{P}$
T_U	A ticket of $\mathcal{U}$
$\|X\|$	The cardinality of the set X
$x \xleftarrow{R} X$	x is randomly selected from the set X
$A(x) \rightarrow y$	y is computed by running the algorithm $A(\cdot)$ with input x
$\mathcal{KG}(1^\ell)$	A secret-public key pair generation algorithm
$\mathcal{BG}(1^\ell)$	A bilinear group generation algorithm

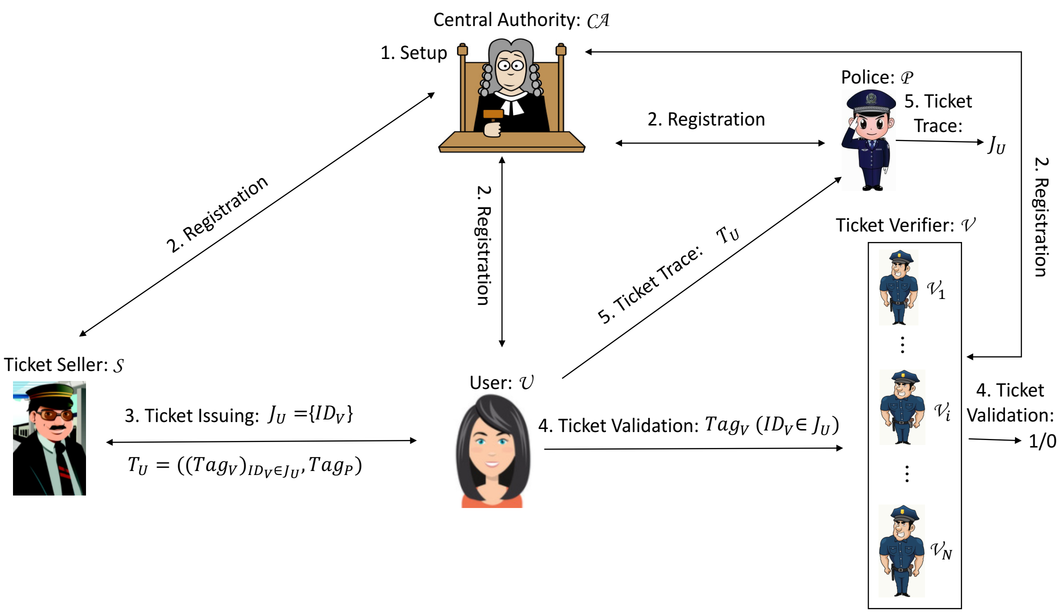

Fig. 1. The Workflow of Our Smart Ticketing For Journey Privacy Scheme

2. $\mathsf{T\text{-}V\text{-}Reg}$ $(\mathcal{V}(ID_V, SK_V, PK_V, PP) \leftrightarrow \mathcal{CA}(MSK, PK_V, PP)) \rightarrow (\sigma_V, (ID_V, PK_V))$. This is an interactive algorithm executed between $\mathcal{CA}$ and $\mathcal{V}$. $\mathcal{V}$ runs $\mathcal{KG}(1^\ell) \rightarrow (SK_V, PK_V)$, inputs its identity ID_V, secret-public key pair (SK_V, PK_V) and the public parameters PP, and outputs a credential σ_V. $\mathcal{CA}$ inputs the master secret key MSK, $\mathcal{V}$'s public key PK_V and the public parameters PP, and outputs the identity ID_V and the public key PK_V.

3. $\mathsf{U\text{-}Reg}$ $(\mathcal{U}(ID_U, SK_U, PK_U, PP) \leftrightarrow \mathcal{CA}(MSK, PK_U, PP)) \rightarrow (\sigma_U, (ID_U, PK_U))$. This is an interactive algorithm executed between $\mathcal{CA}$ and $\mathcal{U}$. $\mathcal{U}$ runs $\mathcal{KG}(1^\ell) \rightarrow (SK_U, PK_U)$, inputs its identity ID_U, secret-public key pair (SK_U, PK_U) and the public parameters PP, and outputs a credential σ_U. $\mathcal{CA}$ inputs the master secret key MSK, $\mathcal{U}$'s public key PK_U and the public parameters PP, and outputs the identity ID_U and the public key PK_U.

4. $\mathsf{P\text{-}Reg}$ $(\mathcal{P}(ID_P, SK_P, PK_P, PP) \leftrightarrow \mathcal{CA}(MSK, PK_P, PP)) \rightarrow (\sigma_P, (ID_P, PK_P))$. This is an interactive algorithm executed between $\mathcal{CA}$ and $\mathcal{P}$. $\mathcal{P}$ runs $\mathcal{KG}(1^\ell) \rightarrow (SK_P, PK_P)$, inputs its identity ID_P, secret-public key pair SK_P, PK_P and the public parameters PP, and outputs a credential σ_P. $\mathcal{CA}$ inputs the master secret key MSK, $\mathcal{P}$'s public key PK_P and the public parameters PP, and outputs the identity ID_P and the public key PK_P.

- $\mathsf{T\text{-}Issuing}(\mathcal{U}(SK_U, PK_U, J_U, \sigma_U, PP) \leftrightarrow \mathcal{S}(SK_S, PK_S, PP)) \rightarrow (T_U, J_U)$. This is an interactive algorithm executed between t $\mathcal{U}$ and $\mathcal{S}$. $\mathcal{U}$ takes input his secret-public key pair (SK_U, PK_U), his journal information J_U consisting of identities of ticket verifiers, his credential σ_U and the public parameters PP, and outputs a ticket $T_U = \big((Tag_V)_{ID_V \in J_U}, Tag_P\big)$ where the authentication tags Tag_V and Tag_P can be validated by the verifier $\mathcal{V}$ with $ID_V \in J_U$ and the police $\mathcal{P}$, respectively. $\mathcal{S}$ takes as input his secret-public key pair (SK_S, PK_S) and the public parameters PP, and outputs the journey information J_U.

- $\mathsf{T\text{-}Validating}(\mathcal{U}(SK_U, PK_U, Tag_V, PP) \leftrightarrow \mathcal{V}((SK_V, PK_V), PK_S, PP)) \rightarrow (\perp, (1, Tag_V)/(0, Tag_V)$. This is an interactive algorithm executed between $\mathcal{U}$ and $\mathcal{V}$ with $ID_V \in J_U$. $\mathcal{U}$ takes as input his secret-public key pair (SK_U, PK_U), the authentication heard Tag_V and the public parameters PP, and outputs $\perp$. $\mathcal{V}$ takes input his secret-public key pair (SK_V, PK_V), $\mathcal{S}$'s public key PK_S and the public parameters PP, and outputs $(1, Tag_V)$ if $ID_V \in J_U$ and the authentication tag Tag_V is valid; otherwise, it outputs $(0, Tag_V)$ to indicate fail.

- $\mathsf{T\text{-}Trace}(\mathcal{U}(T_U) \leftrightarrow \mathcal{P}(SK_P, PK_P, T_U, PP)) \rightarrow (\perp, J_U)$. This is an interactive algorithm executed between $\mathcal{U}$ and $\mathcal{P}$. $\mathcal{U}$ takes as input its ticket T_U, and outputs $\perp$. $\mathcal{P}$ takes as inputs his secret-public key pair (SK_P, PK_P), the ticket T_U and the public parameters PP, and outputs $\mathcal{U}$'s journey J_U.

Definition 1. *A smart ticketing for journey privacy scheme is correct if*

$$\Pr\left[\begin{array}{l|l} & \mathsf{Setup}(1^{\ell}) \rightarrow (MSK, PP)\,; \\ & \mathsf{T\text{-}S\text{-}Reg}(\mathcal{S}(ID_S, SK_S, PK_S, PP) \leftrightarrow \mathcal{CA}(MSK, \\ & PK_S, PP)) \rightarrow (\sigma_S, (ID_S, PK_S)); \\ & \mathsf{T\text{-}V\text{-}Reg}(\mathcal{V}(ID_V, SK_V, PK_V, PP) \leftrightarrow \mathcal{CA}(MSK, \\ \mathsf{T\text{-}Validating}(\mathcal{U}(SK_U, & PK_V, PP)) \rightarrow (\sigma_V, (ID_V, PK_V)); \\ PK_U, Tag_V, PP) \leftrightarrow & \mathsf{U\text{-}Reg}(\mathcal{U}(ID_U, SK_U, PK_U, PP) \leftrightarrow \mathcal{CA}(MSK, \\ \mathcal{V}((SK_V, PK_V), PK_S, & PK_U, PP)) \rightarrow (\sigma_U, (ID_U, PK_U)); \\ PP)) \rightarrow (\perp, (1, Tag_V)) & \mathsf{P\text{-}Reg}(\mathcal{S}(ID_P, SK_P, PK_P, PP) \leftrightarrow \mathcal{CA}(MSK, \\ & PK_P, PP)) \rightarrow (\sigma_P, (ID_P, PK_P)); \\ & \mathsf{T\text{-}Issuing}(\mathcal{U}(SK_U, PK_U, J_U, \sigma_U, PP) \leftrightarrow \mathcal{S}(SK_S, \\ & PK_S, PP)) \rightarrow (T_U, J_U); \\ & ID_V \in J_U \end{array}\right] = 1$$

and

$$\Pr\left[\begin{array}{l|l} & \mathsf{Setup}(1^{\ell}) \rightarrow (MSK, PP)\,; \\ & \mathsf{T\text{-}S\text{-}Reg}(\mathcal{S}(ID_S, SK_S, PK_S, PP) \leftrightarrow \mathcal{CA}(MSK, \\ & PK_S, PP)) \rightarrow (\sigma_S, (ID_S, PK_S)); \\ & \mathsf{T\text{-}V\text{-}Reg}(\mathcal{V}(ID_V, SK_V, PK_V, PP) \leftrightarrow \mathcal{CA}(MSK, \\ \mathsf{T\text{-}Trace}(\mathcal{U}(T_U) \leftrightarrow & PK_V, PP)) \rightarrow (\sigma_V, (ID_V, PK_V)); \\ \mathcal{P}(SK_P, PK_P, T_U, & \mathsf{U\text{-}Reg}(\mathcal{U}(ID_U, SK_U, PK_U, PP) \leftrightarrow \mathcal{CA}(MSK, \\ PP)) \rightarrow (\perp, J_U) & PK_U, PP)) \rightarrow (\sigma_U, (ID_U, PK_U)); \\ & \mathsf{P\text{-}Reg}(\mathcal{S}(ID_P, SK_P, PK_P, PP) \leftrightarrow \mathcal{CA}(MSK, \\ & PK_P, PP)) \rightarrow (\sigma_P, (ID_P, PK_P)); \\ & \mathsf{T\text{-}Issuing}(\mathcal{U}(SK_U, PK_U, J_U, \sigma_U, PP) \leftrightarrow \mathcal{S}(SK_S, \\ & PK_S, PP)) \rightarrow (T_U, J_U). \end{array}\right] = 1.$$

2.2 Security Model

The security model of smart ticketing for journey privacy is defined by the following two games which are executed between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

Ticket Seller Security Game. This game is used to define the ticket seller's security, namely even if users, verifiers and the police collude, they cannot forge a valid ticket. This game is formalised as follows:

Setup. $\mathcal{C}$ runs $\mathsf{Setup}(1^{\ell}) \rightarrow (MSK, PP)$ and sends PP to $\mathcal{A}$.

Registration Query. $\mathcal{A}$ can make the following queries.

1. Ticket Seller Registration Query. $\mathcal{C}$ runs $\mathcal{KG}(1^{\ell}) \rightarrow (SK_S, PK_S)$ and $\mathsf{T\text{-}S\text{-}Reg}(\mathcal{S}(ID_S, SK_S, PK_S, PP) \leftrightarrow \mathcal{CA}(MSK, PK_S, PP)) \rightarrow (\sigma_S, (ID_S, PK_S))$, and sends (PK_S, σ_S) to $\mathcal{A}$.
2. Ticket Verifier Registration Query. Suppose that $Corrupt_V$ be the set consisting of the identities of verifiers corrupted by $\mathcal{A}$. $\mathcal{A}$ submits an identity ID_V. $\mathcal{C}$ runs $\mathcal{KG}(1^{\ell}) \rightarrow (SK_V, PK_V)$ and $\mathsf{T\text{-}V\text{-}Reg}(\mathcal{V}(ID_V, SK_V, PK_V, PP) \leftrightarrow \mathcal{CA}(MSK, PK_V, PP)) \rightarrow (\sigma_V, (ID_V, PK_V))$. If $ID_V \in Corrupt_V$, $\mathcal{C}$ sends (SK_V, PK_K, σ_V) to $\mathcal{A}$. If $ID_V \notin Corrupt_V$, $\mathcal{C}$ sends (PK_K, σ_V) to $\mathcal{A}$.
3. User Registration Query. Suppose that $Corrupt_U$ be the set consisting of the identities of users corrputed by $\mathcal{A}$. $\mathcal{A}$ submits an identity ID_U. $\mathcal{C}$ runs $\mathcal{KG}(1^{\ell}) \rightarrow (SK_U, PK_U)$ and $\mathsf{U\text{-}Reg}(\mathcal{U}(ID_U, SK_U, PK_U, PP) \leftrightarrow \mathcal{CA}(MSK, PK_U, PP)) \rightarrow (\sigma_S, (ID_U, PK_U))$. If $ID_U \in Corrupt_U$, $\mathcal{B}$ sends (SK_U, PK_U, σ_U) to $\mathcal{A}$. If $ID_U \notin Corrupt_U$, $\mathcal{B}$ sends (PK_U, σ_U) to $\mathcal{A}$.

4. Police Registration Query. $\mathcal{A}$ submits a polic's identity ID_P. $\mathcal{C}$ runs $\mathcal{KG} \rightarrow (SK_P, PK_P)$ and P-Reg$(\mathcal{P}(ID_P, SK_P, PK_P, PP) \leftrightarrow \mathcal{CA}(MSK, PK_P, PP)) \rightarrow (\sigma_P, (ID_P, PK_P))$. $\mathcal{C}$ sends (PK_P, σ_P) to $\mathcal{A}$.

Ticket Issuing Query. $\mathcal{A}$ adaptively submits a journey J_U. $\mathcal{C}$ runs T-Issuing$\big(\mathcal{U}(SK_U, PK_U, J_U, \sigma_U, PP) \leftrightarrow \mathcal{S}(SK_S, PK_S, \sigma_S, PP)\big) \rightarrow (T_U, J_U)$ and sends T_U to $\mathcal{A}$. Let QT be the set which consists of the ticket information queried by $\mathcal{A}$ and initially empty. $\mathcal{C}$ adds (T_U, J_U) into QT.

Output. $\mathcal{A}$ outputs a ticket $T_{U^*} = ((Tag_{V^*})_{ID_{V^*} \in J_{U^*}})$ for a user $\mathcal{U}^*$ with a journey J_{U^*}. $\mathcal{A}$ wins the game if T-Validating$(\mathcal{U}(SK_{U^*}, PK_{U^*}, Tag_{V^*}, PP) \leftrightarrow \mathcal{V}((SK_{V^*}, PK_{V^*}), PK_S, PP)) \rightarrow (\perp, (1, Tag_{V^*}))$ for all $ID_{V^* \in J_{U^*}}$ and $(T_{U^*}, J_{U^*}) \notin QT$.

Definition 2. *A smart-ticketing for journey privacy scheme is* $(\varrho, \epsilon(\ell), T)$ *ticket-seller secure if for all probabilistic polynomial-time (PPT) adversary* $\mathcal{A}$ *who makes* ϱ *ticket issuing queries can win the above game with negligible advantage, namely*

$$Adv_{\mathcal{A}} = \Pr\left[\begin{array}{c}\textsf{T-Validating}(\mathcal{U}(SK_{U^*}, PK_{U^*}, Tag_{V^*}, PP) \leftrightarrow \\ \mathcal{V}((SK_{V^*}, PK_{V^*}), PK_S, PP)) \rightarrow (\perp, (1, Tag_{V^*}))\end{array}\right] \leq \epsilon(\ell)$$

for all $ID_{V^*} \in J_{U^*}$.

User Privay Game. This game is used to define the user' security, namely even if some verifiers collude with potential users, they cannot profile the journeys of other users. This game is formalised as follows:

Setup. $\mathcal{C}$ runs Setup$(1^\ell) \rightarrow (MSK, PP)$ and sends PP to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ can make the following queries.

Registration Query. $\mathcal{A}$ can make the following registration queries.

1. Ticket Seller Registration Query. $\mathcal{C}$ runs $\mathcal{KG}(1^\ell) \rightarrow (SK_S, PK_S)$ and T-S-Reg$(\mathcal{S}(ID_S, SK_S, PK_S, PP) \leftrightarrow \mathcal{CA}(MSK, PK_S, PP)) \rightarrow (\sigma_S, (ID_S, PK_S))$, and sends (PK_S, σ_S) to $\mathcal{A}$.
2. Ticket Verifier Registration Query. Let $Corrupt_V$ be the set consisting of the identities of verifiers corrupted by $\mathcal{A}$. $\mathcal{A}$ adaptively submits a verifier' identity ID_V. $\mathcal{C}$ runs $\mathcal{KG}(1^\ell) \rightarrow (SK_V, PK_V)$ and T-V-Reg$(\mathcal{V}(ID_V, SK_V, PK_V, PP) \leftrightarrow \mathcal{CA}(MSK, PK_V, PP)) \rightarrow (\sigma_V, (ID_V, PK_V))$. If $ID_V \in Corrupt_V$, $\mathcal{C}$ sends (SK_V, PK_K, σ_V) to $\mathcal{A}$. If $ID_V \notin Corrupt_V$, $\mathcal{C}$ sends (PK_K, σ_V) to $\mathcal{A}$.
3. User Registration Query. Let $Corrupt_U$ be the set consisting of the identities of users corrupted by $\mathcal{A}$. $\mathcal{A}$ adaptively submits a user' identity ID_U. $\mathcal{C}$ runs $\mathcal{KG}(1^\ell) \rightarrow (SK_U, PK_U)$ and U-Reg$(\mathcal{U}(ID_U, SK_U, PK_U, PP) \leftrightarrow \mathcal{CA}(MSK, PK_U, PP)) \rightarrow (\sigma_S, (ID_U, PK_U))$. If $ID_U \in Corrupt_U$, $\mathcal{B}$ sends (SK_U, PK_U, σ_U) to $\mathcal{A}$. If $ID_U \notin Corrupt_U$, $\mathcal{B}$ sends (PK_U, σ_U) to $\mathcal{A}$.
4. Police Registration Query. $\mathcal{C}$ runs $\mathcal{KG} \rightarrow (SK_P, PK_P)$ and $(\mathcal{P}(ID_P, SK_P, PK_P, PP) \leftrightarrow \mathcal{CA}(MSK, PK_P, PP)) \rightarrow (\sigma_P, (ID_P, PK_P))$. $\mathcal{C}$ sends (PK_P, σ_P) to $\mathcal{A}$.

Ticket Issuing Query. $\mathcal{A}$ adaptively submits a journey J_U to $\mathcal{C}$. $\mathcal{C}$ runs T-Issuing$(\mathcal{U}(SK_U, PK_U, J_U, \sigma_U, PP) \leftrightarrow \mathcal{S}(SK_S, PK_S, \sigma_S, PP)) \rightarrow (T_U, J_U)$ and sends T_U to $\mathcal{A}$.

Ticket Validation Query. $\mathcal{A}$ adaptively submits Tag_V to $\mathcal{C}$. $\mathcal{C}$ runs T-Validating$(\mathcal{U}(SK_U, PK_U, Tag_V, PP) \leftrightarrow \mathcal{V}(SK_V, PK_V, PK_S, PP)) \rightarrow (\perp, (1, Tag_V)/(0, Tag_V))$ and returns $(1, Tag_V)$ to $\mathcal{A}$ if Tag_V is valid and $ID_V \in J_U$; otherwise, $(0, Tag_V)$ is returned to indicate $ID_V \notin J_U$. Let QV

be the set which consists of the ticket validation information queried by $\mathcal{A}$ and initially empty. $\mathcal{C}$ adds (T_U, J_U) into QV

Ticket Trace Query. $\mathcal{A}$ adaptively submits a ticket T_U. $\mathcal{C}$ runsT-Trace($\mathcal{U}(T_U) \leftrightarrow \mathcal{P}(SK_P, PK_P, T_U, PP)) \rightarrow (\perp, J_U)$, and returns J_U to $\mathcal{A}$ if T_U is a valid ticket. Let QT be the set which consists of the ticket trace information queried by $\mathcal{A}$ and initially empty. $\mathcal{C}$ adds (T_U, J_U) into QT.

Challenge. $\mathcal{A}$ submits two verifiers' identities $ID_{V_0^*}$ and $ID_{V_1^*}$. $\mathcal{C}$ flips an unbiased coin with $\{0,1\}$ and obtains a bit $b \in \{0,1\}$. $\mathcal{C}$ sets $J_{U^*} = \{ID_{V_b^*}\}$ and runs T-Issuing($\mathcal{U}(SK_{U^*}, PK_{U^*}, J_{U^*}, \sigma_{U^*}, PP) \leftrightarrow \mathcal{S}(SK_S, PK_S, \sigma_S, PP)) \rightarrow (T_{U^*}, J_{U^*})$ where $T_{U^*} = (Tag_b^*)$ and $Tag_b^* \notin T_U$ for all $(T_U, J_U) \in QV$ and $(T_U, J_U) \in TT$. $\mathcal{C}$ sends T_{U^*} to $\mathcal{A}$.

Phase 2. It is the same as in Phase 1 with the limitation that $ID_{V_0^*} \notin Corrupt_V$, $ID_{v_1^*} \notin Corrupt_V$, $T_{U^*} \notin QV$ and $T_U^* \notin QT$.

Output. $\mathcal{A}$ outputs his guess b' on b. $\mathcal{A}$ wins the game if $b' = b$.

Definition 3. *A smart-ticketing for journey privacy scheme is $(\epsilon(\ell), T)$ user secure if for all probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ can win the above game with negligible advantage, namely*

$$Adv_{\mathcal{A}} = \left| \Pr\left[b' = b\right] - \frac{1}{2} \right| \leq \epsilon(\ell).$$

We say that a smart ticketing for journey privacy scheme is selectively user secure if an initialisation phase Initialisation is added before the the Setup phase.

2.3 Bilinear Group

Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_\tau$ be three cyclic groups with prime order p. A pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_\tau$ and satisfies the following three properties [7]:

1. Bilinearity. For all $x, y \in \mathbb{Z}_p$, $g \in \mathbb{G}_1$ and $\mathfrak{g} \in \mathbb{G}_2$, $e(g^x, \mathfrak{g}^y) = e(g^y, \mathfrak{g}^x) = e(g, \mathfrak{g})^{xy}$;
2. Non-degeneration. For all $g \in \mathbb{G}_1$ and $\mathfrak{g} \in \mathbb{G}_2$, $e(g, \mathfrak{g}) \neq 1_\tau$, where 1_τ is the identity element of $\mathbb{G}_\tau$;
3. Computability. For all $g \in \mathbb{G}_1$ and $\mathfrak{g} \in \mathbb{G}_2$, there exists an polynomial-time efficient algorithm to compute $e(g, \mathfrak{g}) \in \mathbb{G}_\tau$.

Galbraith, Paterson and Smart [21] classified parings into the following three basic types:

1. Type-I: $\mathbb{G}_1 = \mathbb{G}_2$;
2. Type-II: $\mathbb{G}_1 \neq \mathbb{G}_2$ but there is an efficiently computable isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$;
3. Type-III: $\mathbb{G}_1 \neq \mathbb{G}_2$ and there are no efficiently computable isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$.

In this paper, we use Type-III pairing since the elements on $\mathbb{G}_1$ is short (160 bits).

2.4 Complexity Assumptions

Definition 4. (q-Strong Diffie-Hellman (q-SDH) Assumption [4]) *Let* $\mathcal{BG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$. *Suppose that* g *and* $\mathfrak{g}$ *are generators of* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, respectively. Given a* $(q+2)$*-tuple* $(g, g^x, g^{x^2}, \cdots, g^{x^q}, \mathfrak{g}) \in \mathbb{G}_1^{q+1} \times \mathbb{G}_2$*, we say that q-strong Diffie-Hellman assumption holds on* $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ *if for all probabilistic polynomial-time (PPT) adversary* $\mathcal{A}$ *can output* $(c, g^{\frac{1}{x+c}}) \in \mathbb{Z}_p \times \mathbb{G}_1$ *with negligible advantage, namely*

$$Adv_{\mathcal{A}}^{q-SDH} = \Pr\left[\mathcal{A}(\mathfrak{g}, g, g^x, g^{x^2}, \cdots, g^{x^q}) \rightarrow (c, g^{\frac{1}{x+c}})\right] \leq \epsilon(\ell),$$

where $c \in \mathbb{Z}_p - \{-x\}$.

Definition 5. ((JOC Version) q-Strong Diffie-Hellman (JOC-q-SDH) Assumption [5]) *Let* $\mathcal{BG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$. *Given a* $(q+3)$*-tuple* $(g, g^x, \cdots, g^{x^q}, \mathfrak{g}, \mathfrak{g}^x) \in \mathbb{G}_1^{q+1} \times \mathbb{G}_2^2$*, we say that the q-SDH assumption holds on the bilinear group* $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ *if for all probabilistic polynomial-time (PPT) adversaries* $\mathcal{A}$ *can outputs* $(c, g^{\frac{1}{x+c}}) \in \mathbb{Z}_p \times \mathbb{G}_1$ *with negligible advantage, namely*

$$Adv_{\mathcal{A}}^{JOC-q-SDH} = \Pr\left[(c, g^{\frac{1}{x+c}}) \leftarrow \mathcal{A}(g, g^x, \cdots, g^{x^q}, \mathfrak{g}, \mathfrak{g}^x)\right] < \epsilon(\ell),$$

where $c \in \mathbb{Z}_p - \{-x\}$.

Definition 6. (Decisional Diffie-Hellman (DDH) Assumption [18]) *Let* $\mathcal{BG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$. *Give a 3-tuple* $(\xi, \xi^\alpha, \xi^\beta, T) \in \mathbb{G}_1^3$*, we say that the decisional Deffie-Hellman assumption holds on* $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ *if for all probabilistic polynomial-time (PPT) adversaries* $\mathcal{A}$ *can distinguish* $T = \xi^{\alpha\beta}$ *or* $T = R$ *with negligible advantage, namely*

$$Adv_{\mathcal{A}}^{DaDH} = \left|\Pr\left[\mathcal{A}(\xi, \xi^\alpha, \xi^\beta, T = \xi^{\alpha\beta}) = 1\right] - \Pr\left[\mathcal{A}(\xi, \xi^\alpha, \xi^\beta, T = R) = 1\right]\right| < \epsilon(\ell)$$

where $R \xleftarrow{R} \mathbb{G}_1$.

Definition 7. (Symmetric External Diffie-Hellman (SXDH) Assumption [22]) *Let* $\mathcal{BG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$. *We say that the symmetric external Diffie-Hellman assumption holds on* $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ *if the decisional Diffie-Hellman (DDH) assumption holds on both* $\mathbb{G}_1$ *and* $\mathbb{G}_2$.

Notably, the DDH assumption is believed to be hard in both $\mathbb{G}_1$ and $\mathbb{G}_2$ in the Type-III pairing [22].

2.5 Zero-Knowledge Proof

In this paper, we use zero-knowledge proof of knowledge protocols to prove knowledge and statements about various discrete logarithms including: (1) proof of knowledge of a discrete logarithm modular a prime number [35]; (2) proof of knowledge of equality of representation [17]; (3) proof of knowledge of a commitment opens to the product of two other commitments [15]. We follow the definition introduced by Camenish and Stadler in [16] and formalised by Camenish, Kiayias and Yung in [13]. By

$$\text{PoK:}\{(\alpha, \beta, \gamma) : \Upsilon = g^\alpha h^\beta \ \wedge \ \tilde{\Upsilon} = \tilde{g}^\alpha \tilde{h}^\gamma\},$$

we denote a zero knowledge proof on knowledge of integers α β and γ such that $\Upsilon = g^\alpha h^\beta$ and $\tilde{\Upsilon} = \tilde{g}^\alpha \tilde{h}^\beta$ hold on the groups $\mathbb{G} = \langle g \rangle = \langle h \rangle$ and $\tilde{\mathbb{G}} = \langle \tilde{g} \rangle = \langle \tilde{h} \rangle$, respectively. The convention is that the letters in the parenthesis (α, β, γ) stand for the knowledge which is being proven, while other parameters are known by the verifier.

2.6 BBS+ Signature

Based on the group signature scheme [6], Au, Susilo and Mu [3] proposed a signature and named BBS+ signature. This signature scheme works as follows:

- Setup. Let $\mathcal{BG}(1^{\ell}) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$, h be a generator of $\mathbb{G}_1$ and $g, g_0, g_1, \cdots, g_n$ be generators of $\mathbb{G}_2$.
- KeyGen. The signer selects $x \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computes $Y = h^x$. The secret-public key pair is (x, Y).
- Signing. To sign a block message $(m_1, m_2, \cdots, m_n) \in \mathbb{Z}_p^n$, the signer selects $w, e \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and computes $\sigma = (g_0 g^w \prod_{i=1}^n g_i^{m_i})^{\frac{1}{x+e}}$. This signature on $(m_1, m_2, \cdots, m_n)$ is (w, e, σ).
- Verification. Given a signature (w, e, σ) and $(m_1, m_2, \cdots, m_n)$, the verifier checks $e(Yh^e, \sigma) \stackrel{?}{=} e(h, g_0 g^w \prod_{i=1}^n g_i^{m_i})$. If so, the signature is valid; otherwise, it is invalid.

Au, Susilo and Mu [3] reduced the security of the above siganture to q-strong Diffie-Hellman assumption in Type-II paring. Recently, Camenisch, Drijvers and Lehmann [12] reduced its security to JOC version q-strong Diffie-Hellman assumption in Type-III pairing.

Theorem 1. *BBS+ signature is existentially unforgeable against adaptive chosen message attacks (EU-CMA) if the JOC version q-strong Diffie-Hellman assumption holds on the bilinear group $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ where no efficient isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ exists [12].*

In [12], an efficient proof protocol for BBS+ signature was proposed. Let (σ, w, e) be a signature on a block messages $(m_1, m_2, \cdots, m_n)$, the protocol works as follows. The prover selects $r_1, r_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and computes

$$r_3 = \frac{1}{r_1}, \bar{\sigma} = \sigma^{r_1}, \tilde{\sigma} = \bar{\sigma}^{-e}(g_0 g^w \prod_{i=1}^{n} g_i^{m_i})^{r_1} = \bar{\sigma}^x, \varOmega = (g_0 g^w \prod_{i=1}^{n} g_i^{m_i})^{r_1} g^{-r_2}, w' = w - r_2 r_3$$

$$\text{and PoK} \left\{ (w, e, \sigma, r_1, r_2, r_3, w, m_1, m_2, \cdots, m_n) : \frac{\tilde{\sigma}}{\varOmega} = \bar{\sigma}^{-e} g^{r_2} \ \wedge \ g_0 = \varOmega^{r_3} g^{-w'} \prod_{i=1}^{m_i} g_i^{-m_i} \right\}.$$

Theorem 2. *The above efficient proof protocol for BBS+ signature is zero knowledge proof of knowledge $(\sigma, w, e, m_1, m_2, \cdots, m_n)$ [12].*

3 Smart Ticketing For Journey Privacy

In this section, we first explain the idea of our smart ticketing for journey privacy in high-level overview, and then descirbe the formal construction.

3.1 High-Level Overview

Our smart ticketing for journey privacy scheme works as follows. A central authority $\mathcal{CA}$ initialises the system, and generates a master secret key and public parameters. The master secret key is used to generate credential for the entities: ticket seller $\mathcal{S}$, ticket verifier $\mathcal{V}$, user $\mathcal{U}$ and police $\mathcal{P}$.

When registering to $\mathcal{CA}$, $\mathcal{S}$, $\mathcal{V}$, $\mathcal{U}$ and $\mathcal{P}$ first generates their secret-public key pairs, and then sends their identities and the corresponding public keys to $\mathcal{CA}$. After receiving a registration request from an entity, $\mathcal{CA}$ generates a credential for the entity by using his master secret key.

Let J_U be $\mathcal{U}$'s journey information consisting of the identities of verifiers. When buying a ticket from $\mathcal{S}$, $\mathcal{U}$ generates a set of pseudonyms and a proof of his credential to convince $\mathcal{S}$ that he is a registered user. $\mathcal{U}$ submits the pseudonyms, the proof and his journey information to $\mathcal{S}$. For each $ID_V \in J_U$ and $\mathcal{P}$, $\mathcal{S}$ generates an authentication tags Tag_V and Tag_P which includes the public key of $\mathcal{V}$ and $\mathcal{P}$ and can only be validated by $\mathcal{V}$ and $\mathcal{P}$, respectively. The ticket is $T_U = ((Tag_V)_{ID_V \in J_U}, Tag_P)$

When being checked by the verifier $\mathcal{V}$ with $ID_V \in J_U$, $\mathcal{U}$ send the authentication tag Tag_V to $\mathcal{V}$. If Tag_V is valid, $\mathcal{U}$ is authorised to pass this station; otherwise, $\mathcal{U}$ is unauthorised to pass.

In the case that $\mathcal{U}$'s journey needs to be traced, $\mathcal{P}$ requests $\mathcal{U}$ to submit his/her ticket T_U. By using his secret key and T_U, $\mathcal{P}$ can determine $\mathcal{U}$'s journey information $J_U = \{ID_V\}_{ID_V \in J_U}$.

3.2 Construction

Our smart ticketing for journey privacy scheme is formally described in Fig. 2, Fig. 3, Fig. 4, Fig. 5 and Fig. 6.

Setup. $\mathcal{CA}$ runs $\mathcal{BG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ with $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_\tau$. Let $g, h, \xi, \tilde{h}$ be generators of the group $\mathbb{G}_1$ and $\mathfrak{g}$ be generators of $\mathbb{G}_2$. Suppose that $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p$, $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_p$ and $H_3 : \{0,1\}^* \rightarrow \mathbb{G}_1$ are three cryptographic hash functions. CA selects $x_a \xleftarrow{R} \mathbb{Z}_p$ and computes $Y_A = \mathfrak{g}^{x_a}$. The master secret key is $MSK = x_a$ and the public parameters are $PP = (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau, g, h, \xi, \tilde{h}, \mathfrak{g}, Y_A, H_1, H_2, H_3)$.

Fig. 2. Setup Algorithm

Setup. $\mathcal{CA}$ generates a bilinear group $\mathcal{BG}(1^\ell) \rightarrow (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ and selects generators $g, h, \xi \in \mathbb{G}_1$ and $\mathfrak{g} \in \mathbb{G}_2$. $\mathcal{CA}$ chooses a master secret key $x_a \xleftarrow{R} \mathbb{Z}_p$ and computes the public key $Y_A = \mathfrak{g}^{x_a}$. $\mathcal{CA}$ selects three cryptographic hash functions: $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p$, $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_p$ and $H_3 : \{0,1\}^* \rightarrow \mathbb{G}_1$. The public parameters are $PP = (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau, g, h, \xi, \mathfrak{g}, Y_A, H_1, H_2, H_3)$.

Registration. $\mathcal{S}$, $\mathcal{V}$, $\mathcal{U}$ and $\mathcal{P}$ register to $\mathcal{CA}$. $\mathcal{S}$ generates his secret-public key pair $(x_s, (Y_S, \tilde{Y}_S))$ where $x_s \xleftarrow{R} \mathbb{Z}_p$, $Y_S = \xi^{x_s}$ and $\tilde{Y}_S = \mathfrak{g}^{x_s}$, and sends his identity ID_S and public key $(Y_S, \tilde{Y}_S)$ to $\mathcal{CA}$. $\mathcal{CA}$ selects $r_s, e_s \xleftarrow{R} \mathbb{Z}_p$ and computes $\sigma_S = (gh^{r_s} Y_S)^{\frac{1}{x_a + e_s}}$. $\mathcal{CA}$ sends the credential (r_s, e_s, σ_S) to $\mathcal{S}$, and stores the registration information of $\mathcal{S}$ as $(ID_S, Y_S, \tilde{Y}_S, (r_s, e_s, \sigma_S))$. Actually, (r_s, e_s, σ_S) is a BBS+ signature on $\mathcal{S}$'s public key Y_S.

$\mathcal{V}$ generates his secret-public key pair (x_v, Y_v) where $x_v \xleftarrow{R} \mathbb{Z}_p$ and $Y_V = \xi^{x_v}$, and sends his identity ID_V and public key Y_V to $\mathcal{CA}$. $\mathcal{CA}$ selects $r_v, e_v \xleftarrow{R} \mathbb{Z}_p$ and computes $\sigma_V = (gh^{r_v} Y_V)^{\frac{1}{x_a + e_v}}$. $\mathcal{CA}$ sends the credential (r_v, e_v, σ_v) to $\mathcal{V}$, and stores the registration information of $\mathcal{V}$ as $(ID_V, Y_V, (r_v, e_v, \sigma_V))$. (r_v, e_v, σ_v) is a BBS+ signature on $\mathcal{V}$' public key Y_V.

$\mathcal{U}$ generates his secret-public key pair (x_u, Y_U) where $x_u \xleftarrow{R} \mathbb{Z}_p$ and $Y_U = \xi^{x_u}$, and sends his identity ID_U and public key Y_U to $\mathcal{CA}$. $\mathcal{CA}$ selects where $r_u, e_u \xleftarrow{R} \mathbb{Z}_p$ and computes $\sigma_U = (gh^{r_u} Y_U)^{\frac{1}{x_a + e_u}}$. $\mathcal{CA}$ sends the credential (r_u, e_u, σ_U) to $\mathcal{U}$, and stores the registration information of $\mathcal{U}$ as $(ID_U, Y_U, (r_u, e_u, \sigma_U))$. (r_u, e_u, σ_U) is a BBS+ signature on $\mathcal{U}$' Y_U.

$$\mathsf{Ticket\text{-}Seller\text{-}Reg}\Big(\mathcal{S}(x_s, Y_S, \tilde{Y}_S, ID_S PP) \leftrightarrow \mathcal{CA}(MSK, PP)\Big)$$

Ticket Seller: $\mathcal{S}$		Central Authority: $\mathcal{CA}$
Selects $x_s \xleftarrow{R} \mathbb{Z}_p$, and computes $Y_S = \xi^{x_s}$ and $\tilde{Y}_S = \mathfrak{g}^{x_s}$. The secret-public key pair is $(x_s, Y_S, \tilde{Y}_S)$		
	$\xrightarrow{ID_S, Y_S, \tilde{Y}_S}$	Select $e_s, r_s \xleftarrow{R} \mathbb{Z}_p$ and computes
Verifies: $e(\sigma_S, Y_A \mathfrak{g}^{e_s}) \stackrel{?}{=} e(gh^{r_s} Y_S, \mathfrak{g})$.	$\xleftarrow{\sigma_S, r_s, e_s}$	$\sigma_S = (gh^{r_s} Y_S)^{\frac{1}{x_a + e_s}}$.
Keeps the credential as $Cred_S = (e_s, r_s, \sigma_S)$.		Stors $(ID_S, Y_S, \tilde{Y}_S, (r_s, e_s, \sigma_S))$.

$$\mathsf{Ticket\text{-}Verifier\text{-}Reg}(\mathcal{V}(x_v, Y_V, ID_V, PP) \leftrightarrow \mathcal{CA}(MSK, PP))$$

Ticket-Verifer: $\mathcal{V}$		Central Authority: $\mathcal{CA}$
Selects $x_v \xleftarrow{R} \mathbb{Z}_p$ and computes $Y_V = \xi^{x_v}$. The secret-public key pair is (x_v, Y_V).		
	$\xrightarrow{ID_V, Y_V}$	Selects $e_v, r_v \xleftarrow{R} \mathbb{Z}_p$ and computes
Verifies: $e(\sigma_V, Y_A \mathfrak{g}^{e_v}) \stackrel{?}{=} e(gh^{r_v} Y_V, \mathfrak{g})$.	$\xleftarrow{\sigma_V, r_v, e_v}$	$\sigma_V = (gh^{r_v} Y_V)^{\frac{1}{x_a + e_v}}$.
Keep the credential as (e_v, r_v, σ_V).		Stores $(ID_V, Y_V, (r_v, e_v, \sigma_V))$.

$$\mathsf{User\text{-}Reg}(\mathcal{U}(x_u, Y_U, ID_U, PP) \leftrightarrow \mathcal{CA}(MSK, PP))$$

User: $\mathcal{U}$		Central Authority: $\mathcal{CA}$
Selects $x_u \xleftarrow{R} \mathbb{Z}_p$, and computes $Y_U = \xi^{x_u}$. This secret-public key pair is (x_u, Y_U).		
	$\xrightarrow{ID_U, Y_U}$	Select $e_u, r_u \xleftarrow{R} \mathbb{Z}_p$ and computes
Verifies: $e(\sigma_U, Y_A \mathfrak{g}^{e_u}) \stackrel{?}{=} e(gh^{r_u} Y_U, \mathfrak{g})$.	$\xleftarrow{\sigma_U, e_u, r_u}$	$\sigma_U = (gh^{r_u} Y_U)^{\frac{1}{x_a + e_u}}$.
Keep the credential as $Cred_U = (e_u, r_u, \sigma_U)$.		Stores $(ID_U, Y_U, (r_u, e_u \sigma_U))$.

$$\mathsf{Polic\text{-}Reg}(\mathcal{P}(x_p, Y_P, ID_P, PP) \leftrightarrow \mathcal{CA}(MSK, PP))$$

Polic: $\mathcal{P}$		Central Authority: $\mathcal{CA}$
Selects $x_p \xleftarrow{R} \mathbb{Z}_p$, and computes $Y_P = \xi^{x_p}$. The secret-public key pair is (x_p, Y_P).		
	$\xrightarrow{ID_P, Y_P}$	Select $e_p, r_p \xleftarrow{R} \mathbb{Z}_p$ and computes
Verifies: $e(\sigma_P, Y_A \mathfrak{g}^{e_p}) \stackrel{?}{=} e(gh^{r_p} Y_P, \mathfrak{g})$.	$\xleftarrow{\sigma_P, e_p, r_p}$	$\sigma_P = (gh^{r_p} Y_P)^{\frac{1}{x_a + e_p}}$.
Keep the credential as $Cred_P = (e_p, r_p, \sigma_P)$.		Stores $(ID_P, Y_P, (r_p, e_p, \sigma_P))$.

Fig. 3. Registration Algorithm

$\mathcal{P}$ generates his secret-public key pair (x_p, Y_P) where $x_p \xleftarrow{R} \mathbb{Z}_p$ and $Y_P = \xi^{x_p}$, and sends his identity ID_P and public key Y_P to $\mathcal{CA}$. $\mathcal{CA}$ selects where $r_p, e_p \xleftarrow{R} \mathbb{Z}_p$ and computes $\sigma_P = (gh^{r_p} Y_P)^{\frac{1}{x_a + e_p}}$. $\mathcal{CA}$ sends the credential (r_p, e_p, σ_P) to $\mathcal{P}$, and stores the registration information of $\mathcal{P}$ as $(ID_P, Y_P, (r_p, e_p, \sigma_P))$. (r_p, e_p, σ_P) is a BBS+ signature on $\mathcal{P}$'s public key Y_P.

Ticket Issuing. To prevent verifiers from concluding the number of verifiers included in a ticket, a dummy verifier $\mathcal{V}_{du}$ with identity ID_{du} is used. $\mathcal{U}$ chooses his journey $J_U = \{ID_V\}$ consisting of the identities of verifiers and a secret value $z_u \xleftarrow{R} \mathbb{Z}_p$. For each $ID_V \in J_U$, $\mathcal{U}$ computes a

$\mathsf{Ticket-Issuing}\ (\mathcal{U}(x_u, Cred_U, PP) \leftrightarrow \mathcal{S}(x_s, Crd_S, PP))$

Let ID_{du} be the identity of an dummy verifier $\mathcal{V}_{du}$ which can be used to preventing verifiers from concluding the exact number of stations included in a ticket. Suppose that J_U is $\mathcal{U}$'s journey consisting of the identities of verifiers where $\mathcal{U}$ need pass

User: $\mathcal{U}$

Computes $B_U = gh^{r_u}Y_U$.

Select $v_1, v_2, z_u \xleftarrow{R} \mathbb{Z}_p$ and computes
$v_3 = \frac{1}{v_1}$, $v = r_u - v_2v_3$, $\bar{\sigma}_U = \sigma_U^{v_1}$,
$\tilde{\sigma}_U = \bar{\sigma}_U^{-e_u} B_U^{v_1} (= \bar{\sigma}_U^{x_a})$, $\bar{B}_U = B_U^{v_1} h^{-v_2}$,
$(z_v = H_1(z_u||ID_V), P_V = Y_U Y_P^{z_v},$
$Q_V = \xi^{z_v})_{ID_V \in J_U}$.
Computes the proof $\prod_U^1$:
$\mathrm{PoK}\{(x_u, r_u, e_u, \sigma_U, v_1, v_2, v_3, v,$
$(z_v)_{ID_V \in J_U}) : \frac{\tilde{\sigma}_U}{\bar{B}_U} = \bar{\sigma}_U^{-e_u} h^{v_2}$
$\wedge\ g^{-1} = \bar{B}_U^{-v_3} \xi^{x_u} h^v\ \wedge$
$(P_V = \xi^{x_u} Y_P^{z_v} \wedge Q_V = \xi^{z_v})_{ID_V \in J_U}\}$

$$\xrightarrow[((P_V, Q_V)_{ID_V \in J_U})]{\bar{\sigma}_U, \tilde{\sigma}_U, \bar{B}_U, J_U, \prod_U^1}$$

Ticket Seller: $\mathcal{S}$

Verifies $\prod_U^1$ and $e(\bar{\sigma}_U, Y) \stackrel{?}{=} e(\tilde{\sigma}_U, \mathfrak{g})$;

1. If $|J_U| = 2\lambda$, let $\Omega_U = J_U$;
2. If $|J_U| = 2\lambda - 1$, let $\Omega_U = J_U \cup \{ID_{du}\}$.

Selects $t_u, d_v, w_v, e_v \xleftarrow{R} \mathbb{Z}_p$ and computes
$C_U = \xi^{t_u}$, $D_V = H_2(C_U||ID_V)$, $E_V = \xi^{d_v}$,
$F_V = Y_V^{d_v}$, $K_V = Y_V Y_P^{d_v}$,
$s_v = H_1(P_V||Q_V||E_V||F_V||K_V||Text)$
and $\sigma_v = (gh^{w_v}\tilde{h}^{s_v})^{\frac{1}{x_s+e_v}}$.
For ID_{du}, selects $d', w', e' \xleftarrow{R} \mathbb{Z}_p$, D_{du}, P_{du},
$Q_{du}, F_{du} \xleftarrow{R} \mathbb{G}_1$, and computes $E_{du} = \xi^{d'}$,
$K_{du} = Y_P^{d'} H_3(ID_{du})$, $s' = H_1(P_{du}||Q_{du}||E_{du}$
$||F_{du}||K_{du}||Text)$ and $\sigma_{du} = (gh^{w'}\tilde{h}^{s'})^{\frac{1}{x_s+e'}}$.
Select $w_p, e_p \xleftarrow{R} \mathbb{Z}_p$, and computes
$s_p = H_1(s_1||s_2||\cdots||s_{|\Omega_U|})$ and
$\sigma_P = (gh^{w_p}\tilde{h}^{s_p})^{\frac{1}{x+e_p}}$.
The ticket is
$T_U = ((D_V, P_V, Q_V, E_V, F_V, K_V, s_v,$
$w_v, e_v, \sigma_V)_{ID_V \in \Omega_U}, (s_p, w_p, e_p, \sigma_P))$.

$$\xleftarrow{C_U, T_U, Text}$$

For $ID_V \in \Omega_U$, verifies
$D_V \stackrel{?}{=} H_2(C_U||Y_V)$,
$s_v \stackrel{?}{=} H_1(P_V||Q_V||E_V||F_V||K_V||Text)$,
$e(\sigma_V, \tilde{Y}_S \mathfrak{g}^{e_v}) \stackrel{?}{=} e(gh^{w_v}\tilde{h}^{s_v}, \mathfrak{g})$,
$s_p \stackrel{?}{=} H_1(s_1||s_2||\cdots||s_{|\Omega_U|})$
and $e(\sigma_P, \tilde{Y}_S \mathfrak{g}^{e_p}) = e(gh^{w_p}\tilde{h}^{s_p}, \mathfrak{g})$.
Keeps the ticket as
$T_U = ((D_V, P_V, Q_V, E_V, F_V, K_V, s_v,$
$w_v, e_v, \sigma_V)_{ID_V \in \Omega_U}, (s_p, w_p, e_p, \sigma_P))$
and (z_u, C_U) secret;
otherwise, aborts.

Fig. 4. Ticket Issuing Algorithm

pseudonym $Ps_V = (P_V, Q_V)$ where $P_V = Y_U Y_P^{z_v}$, $Q_V = \xi^{z_v}$ and $z_v = H_1(z_u||ID_V||Time)$. Actually, (P_V, Q_V) is an EIGama encryption of Y_U under Y_P, hence $\mathcal{P}$ can detect the real user from his/her pseudonym Ps_V. To prove he/she is a registered user, $\mathcal{U}$ selects $v_1, v_2 \xleftarrow{R} \mathbb{Z}_p$, and randomises his credential (e_u, r_u, σ_U) to be $(\bar{\sigma}_U, \tilde{\sigma}_U, \bar{B}_U)$, where $\bar{\sigma}_U = \sigma_U^{v_1}$, $\tilde{\sigma}_U = \bar{\sigma}_U^{-e_u} B_U^{v_1} (= \bar{\sigma}_U^{x_a})$, $\bar{B}_U = B_U^{v_1} h^{-v_2}$ and $B_U = g h^{r_u} Y_U$. $\mathcal{U}$ proves the knowledge of his/her credential σ_U and pseudonyms $Ps_V = (P_V, Q_V)$ to $\mathcal{S}$ by sending a proof $\prod_U^1$: PoK$\{(x_u, r_u, e_u, \sigma_U, v_1, v_2, v_3, v, (z_v)_{ID_V \in J_U}) : \frac{\tilde{\sigma}_U}{\bar{B}_U} = \bar{\sigma}_U^{-e_u} h^{v_2} \wedge g^{-1} = \bar{B}_U^{-v_3} g^{x_u} h^v \wedge (P_V = \xi^{x_u} Y_P^{z_v} \wedge Q_V = \xi^{z_v})_{ID_V \in J_U}\}$ where $v_3 = \frac{1}{v_1}$ and $v = r_u - v_2 v_3$.

After verifying the proof $\prod_U^1$, $\mathcal{S}$ checks the number of verifiers included in J_U. If $|J_U| = n = 2\lambda$ is an even number, $\mathcal{S}$ sets $\Omega_U = J_U$; If $|J_U| = n = 2\lambda - 1$ is an odd number, $\mathcal{S}$ sets $\Omega_U = J_U \cup \{ID_{du}\}$. To generate a ticket for $\mathcal{U}$, $\mathcal{S}$ selects $t_u \xleftarrow{R} \mathbb{Z}_p$ and computes $C_U = \xi^{t_u}$. For each $ID_V \in J_U$, $\mathcal{S}$ selects $d_v, w_v, e_v \xleftarrow{R} \mathbb{Z}_p$, and computes $D_V = H_2(C_U||ID_V)$, $E_V = \xi^{d_v}$, $F_V = Y_V^{d_v}$, $K_V = Y_V Y_P^{d_v}$, $s_v = H_1(P_V||Q_V||E_V||F_V||K_V||Text)$ and $\sigma_V = (g h^{w_v} \tilde{h}^{s_v})^{\frac{1}{x_s + e_v}}$. The authentication tag is $Tag_V = (E_V, F_V, K_V, s_v, w_v, e_v, \sigma_V)$ and D_V is the index of Tag_V. Actually, (w_v, e_v, σ_V) is a BBS+ signature s_v, (E_V, Y_V, F_V) is a Diffie-Hellman tuple and (E_V, K_V) is an EIGama encryption of Y_V under Y_P. Similarly, for ID_{du}, $\mathcal{S}$ selects $d', w', e' \xleftarrow{R} \mathbb{Z}_p$ and $D_{du}, P_{du}, Q_{du}, F_{du} \xleftarrow{R} \mathbb{G}_1$, and computes $E_{du} = \xi^{d'}$, $K_{du} = Y_P^{d'} H_3(ID_{du})$, $s' = H_1(P_{du}||Q_{du}||E_{du}||F_{du}||K_{du}||Time)$ and $\sigma_{du} = (g h^{w'} \tilde{h}^{s'})^{\frac{1}{x_s + e'}}$. Finally, to prevent $\mathcal{U}$ from combining different tickets, $\mathcal{S}$ selects $w_u, e_u \xleftarrow{R} \mathbb{Z}_p$, and computes $s_u = H_1(s_1||s_2||\cdots||s_{2\lambda})$ and $\sigma_U = (g h^{w_u} \tilde{h}^{s_u})^{\frac{1}{x + e_u}}$. (w_u, e_u, σ_U) is a BBS+ signature on s_u. The authentication tag for $\mathcal{P}$ is $Tag_P = ((E_1, K_1, s_1), \cdots, (E_{2\lambda}, K_{2\lambda}, s_{2\lambda}), (w_u, e_u, s_u, \sigma_U))$. The ticket is $T_U = ((D_V, P_V, Q_V, E_V, F_V, K_V, s_v, w_v, e_v, \sigma_V)_{ID_V \in J_U}, (s_u, w_u, e_u, \sigma_U))$. $\mathcal{S}$ sends (C_U, T_U) to $\mathcal{U}$.

Finally, $\mathcal{U}$ verifies the correctness of T_U, stores it and keeps (z_u, C_U) secret.

An instantiation of the proof $\prod_U^1$ is as follows. $\mathcal{U}$ select $v_1, v_2, z_u, r'_u, x'_u, e'_u, v'_2, v'_3, v', z'_1, z'_2, \cdots, z'_n \xleftarrow{R} \mathbb{Z}_p$ and computes $v_3 = \frac{1}{v_1}$, $v = r_u - v_2 v_3$, $\bar{\sigma}_U = \sigma_U^{v_1}$, $\tilde{\sigma}_U = \bar{\sigma}_U^{-e_u} B_U^{v_1} (= \bar{\sigma}_U^{x})$, $\bar{B}_U = B_U^{v_1} h^{-v_2}$, $W_1 = \bar{\sigma}_U^{-e'_u} h^{v'_2}$, $W_2 = \bar{B}_U^{-v'_3} g^{x'_u} h^{v'}$, $\left(z_v = H_1(z_u||ID_V||Time), P_V = Y_U Y_P^{z_v}, P'_v = \xi^{x_u} Y_P^{z'_v}, Q_V = \xi^{z_v}, Q'_V = \xi^{z'_v}\right)_{ID_V \in J_U}$. $\mathcal{U}$ computes $c = H_1(\bar{\sigma}_U||\tilde{\sigma}_U||\bar{B}_U||W_1||W_2||P_1||P'_1||Q_1||Q'_1|| P_2||P'_2||Q_2||Q'_2||\cdots||P_n||P'_n||Q_n||Q'_n)$, $\hat{e}_u = e'_u - c e_u$, $\hat{v}_2 = v'_2 - c v_2$, $\hat{v}_3 = v'_3 - c v_3$, $\hat{v} = v' - c v$, $\hat{x}_u = x'_u - c x_u$ and $(\hat{z}_v = z'_v - c z_v)_{ID_V \in J_U}$. $\mathcal{U}$ sends $(\bar{\sigma}_U, \tilde{\sigma}_U, \bar{B}_U, W_1, W_2, (P_V, P'_V, Q_V, Q'_V)_{ID_V \in J_U})$ and $(c, \hat{e}_u, \hat{v}_2, \hat{v}_3, \hat{v}, \hat{x}_u, \hat{z}_1, \hat{z}_2, \cdots, \hat{z}_n)$ to $\mathcal{S}$.

After receiving $(\bar{\sigma}_U, \tilde{\sigma}_U, \bar{B}_U, W_1, W_2, (P_V, P'_V, Q_V, Q'_V)_{ID_V \in J_U})$ and $(c, \hat{e}_u, \hat{v}_2, \hat{v}_3, \hat{v}, \hat{x}_u, \hat{z}_1, \hat{z}_2, \cdots, \hat{z}_n)$, $\mathcal{S}$ checks

$$c \stackrel{?}{=} H_1(\bar{\sigma}_U||\tilde{\sigma}_U||\bar{B}_U||W_1||W_2||P_1||P'_1||Q_1||Q'_1||P_2||P'_2||Q_2||Q'_2||\cdots||P_n||P'_n||Q_n||Q'_n),$$
$$W_1 \stackrel{?}{=} \bar{\sigma}_U^{-\hat{e}_u} h^{\hat{v}_2} (\frac{\tilde{\sigma}_U}{\bar{B}_U})^c, \ W_2 \stackrel{?}{=} \bar{B}_U^{-\hat{v}_3} \xi^{\hat{x}_u} h^{\hat{v}} g^{-c}, \ (P'_V \stackrel{?}{=} \xi^{\hat{x}_u} Y_P^{\hat{z}_v} P_V^c, \ Q'_V \stackrel{?}{=} \xi^{\hat{z}_v} Q_V^c)_{ID_V \in J_U}.$$

Ticket Validation. When verifying the ticket of $\mathcal{U}$, $\mathcal{V}$ with $ID_V \in J_U$ sends its identity ID_V to $\mathcal{U}$. $\mathcal{U}$ first computes the index $D_V = H_2(C_U||ID_V)$, and the searches D_V in T_U. If $D_V \in T_U$, $\mathcal{U}$ goes to the next step; otherwise, $\mathcal{U}$ aborts. To convince $\mathcal{V}$ that he/she is the owner of T_U, $\mathcal{U}$ generates a proof $\prod_U^2$: PoK$\{(x_u, z_v) : P_V = \xi^{x_u} Y_P^{z_v} \ \wedge \ Q_V = \xi^{z_v}\}$ to prove the knowledge of (x_u, z_v) included in the pseudonym Ps_V. $\mathcal{U}$ sends $Tag_V = (E_V, F_V, K_V, s_v, w_v, e_v, \sigma_V)$ and $\prod_U^2$ to $\mathcal{V}$.

$\mathsf{Ticket-Validation}\ (\mathcal{U}(x_u, T_U, PP) \leftrightarrow \mathcal{V}(Y_V, PP))$

User: $\mathcal{U}$		Ticket verifier: $\mathcal{V}$ $(ID_V \in J_U)$
	$\xleftarrow{ID_V}$	
Computes $D_V = H_2(C_U \vert\vert Y_V)$ and searches $(D_V, P_V, Q_V, E_V, F_V, K_V, s_v, w_v, e_v, \sigma_V)$; Computes and $z_v = H_1(z_u \vert\vert ID_V)$, and the proof: $\prod_U^2$: PoK$\{(x_u, z_v) : P_V = \xi^{x_u} Y_P^{z_v} \wedge$ $Q_V = \xi^{z_v}\}$.	$\xrightarrow[K_V, s_v, w_v, e_v, \sigma_V]{\prod_U^2, P_V, Q_V, E_V, F_V,}$	
		Checks: (1) The correctness of $\prod_U^2$; (2) $s_v \stackrel{?}{=} H_1(P_V \vert\vert Q_V \vert\vert E_V \vert\vert F_V \vert\vert K_V \vert\vert Text)$; (3) $K_V \stackrel{?}{=} E_V^{x_v}$; (4) $e(\sigma_V, Y_S \mathfrak{g}^{e_v}) \stackrel{?}{=} e(gh^{w_v} \tilde{h}^{s_v}, \mathfrak{g})$. If (1), (2), (3) and (4) hold, the ticket is valid; otherwise, it is invalid.

Fig. 5. Ticket Validation Algorithm

$\mathcal{V}$ checks: (1) The correctness of $\prod_U^2$; (2) $s_v \stackrel{?}{=} H_1(P_j||Q_V||E_V||F_V||K_V||Text)$; (3) $K_V \stackrel{?}{=} E_V^{x_v}$; (4) $e(\sigma_V, Y_S \mathfrak{g}^{e_v}) \stackrel{?}{=} e(gh^{w_v} \tilde{h}^{s_v}, \mathfrak{g})$. If (1), (2), (3) and (4) hold, the ticket is valid; otherwise, it is invalid.

An instantiation of the proof $\prod_U^2$ is as follows. $\mathcal{U}$ selects $x'_u, z'_v \xleftarrow{R} \mathbb{Z}_p$, and computes $P'_V = \xi^{x'_u} Y_P^{z'_v}$, $Q'_V = \xi^{z'_v}$, $c_v = H_1(P_V||P'_V||Q_V||Q'_V)$, $\hat{x}_u = x'_u - c_v x_u$ and $\hat{z}_v = z'_v - c_v z_v$. $\mathcal{U}$ sends (P_V, P'_V, Q_V, Q'_V) and $(c_v, \hat{x}_v, \hat{z}_v)$ to $\mathcal{V}$.

After receiving (P_V, P'_V, Q_V, Q'_V) and $(c_v, \hat{x}_v, \hat{z}_v)$, $\mathcal{V}$ verifiers

$$c_v \stackrel{?}{=} H_1(P_V||P'_V||Q_V||Q'_V),\ P'_V \stackrel{?}{=} \xi^{\hat{x}_u} Y_P^{\hat{z}_v} P_V^{c_v} \text{ and } Q'_V \stackrel{?}{=} \xi^{\hat{z}_v} Q_V^{c_v}.$$

$\mathsf{Ticket\ Trace}$. To trace the journey of $\mathcal{U}$, $\mathcal{P}$ requires $\mathcal{U}$ to submits his ticket T_U. Since each pseudonym (P_V, Q_V) is an encryption of Y_U under Y_P and (E_V, K_V) is an EIGama encryption of Y_V under Y_P, $\mathcal{P}$ can decrypt the encryptions and obtain $Y_U = \frac{P_V}{Q_V^{x_p}}$ and $Y_V = \frac{K_V}{E_V}^{x_p}$ for $ID_V \in J_U$. Furthermore, $\mathcal{P}$ checks: (1) $s_v \stackrel{?}{=} H_1(P_V||Q_V||E_V||K_j||Text)$; (2) $e(\sigma_V, Y_S \mathfrak{g}^{w_v}) \stackrel{?}{=} e(gh^{w_v} \tilde{h}^{s_v}, \mathfrak{g})$; (3) $s_u \stackrel{?}{=} H_1(s_1||s_2|| \cdots ||s_{|\Omega_U|})$; (4) $e(\sigma_U, \tilde{Y}_S \mathfrak{g}^{w_u}) \stackrel{?}{=} e(gh^{w_u} \tilde{h}^{s_u}, \mathfrak{g})$. If (1) and (2) hold, $\mathcal{P}$ adds ID_V into J_U. If (1), (2) (3), (4) and (5) hold, $\mathcal{U}$'s journey information can be determined as: $J_U = \{ID_V\}$ consisting of the identities of verifiers; otherwise, the trace is failed.

Correctness. Our smart ticketing for journey privacy scheme is correct as the following equations hold.

$$e(\sigma_S, Y_A \mathfrak{g}^{e_s}) = e((gh^{r_s} Y_S)^{\frac{1}{x_a + e_s}}, \mathfrak{g}^{x_a + e_s}) = e(gh^{r_s} Y_S, \mathfrak{g}),$$
$$e(\sigma_V, Y_A \mathfrak{g}^{e_v}) = e((gh^{r_v} Y_V)^{\frac{1}{x_a + e_v}}, \mathfrak{g}^{x_a + e_v}) = e(gh^{r_v} Y_V, \mathfrak{g}),$$

$\mathsf{Ticket\text{-}Trace}(\mathcal{U}(T_U) \leftrightarrow \mathcal{P}(x_p, PP))$

User: $\mathcal{U}$		Police: $\mathcal{P}$

	$\xrightarrow{T_U}$	Computes: $Y_U = \frac{P_V}{Q_V^{x_p}}$ and $Y_V \stackrel{?}{=} \frac{K_V}{E_V^{x_p}}$.
		Checks:
		(1) $s_v \stackrel{?}{=} H_1(P_V\|\|Q_V\|\|E_V\|\|K_V\|\|Text)$;
		(2) $e(\sigma_V, Y_S\mathfrak{g}^{w_v}) \stackrel{?}{=} e(gh^{w_v}\tilde{h}^{s_v}, \mathfrak{g})$;
		If (1) and (2) holds, $ID_V \in J_U$; otherwise, $ID_V \notin J_U$.
		(3) $s_p \stackrel{?}{=} H_1(s_1\|\|s_2\|\|\cdots\|\|s_{\|\Omega_U\|})$;
		(4) $e(\sigma_P, \tilde{Y}_S\mathfrak{g}^{w_p}) \stackrel{?}{=} e(gh^{w_p}\tilde{h}^{s_p}, \mathfrak{g})$.
		If (1), (2), (3) and (4) hold,$\mathcal{P}$ can determine that the user $\mathcal{U}$ with public key Y_U has a journey: $J_U = \{ID_V\}$ consisting of the identities of verifiers; otherwise, the trace is failed.

Fig. 6. Ticket Trace Algorithm

$$e(\sigma_U, Y_A\mathfrak{g}^{e_u}) = e((gh^{r_u}Y_U)^{\frac{1}{x_a+e_u}}, \mathfrak{g}^{x_a+e_u}) = e(gh^{r_u}Y_U, \mathfrak{g}),$$
$$e(\sigma_P, Y_A\mathfrak{g}^{e_p}) = e((gh^{r_p}Y_P)^{\frac{1}{x_a+e_p}}, \mathfrak{g}^{x_a+e_p}) = e(gh^{r_p}Y_P, \mathfrak{g}),$$

$$\begin{aligned}\tilde{\sigma}_U &= \bar{\sigma_U}^{-e_u}B_U^{v_1} = \sigma_U^{-e_u v_1}B_U^{v_1} = B_U^{\frac{-e_u v_1}{x_a+e_u}}B_U^{v_1} = B_U^{\frac{-v_1(e_u+x_a)+v_1x_a}{x_a+e_u}}B_U^{v_1} = B_U^{-v_1}B_U^{\frac{v_1x_a}{x_a+e_u}}B_U^{v_1}\\ &= (B_U^{\frac{1}{x_a+e_u}})^{v_1x_a} = (\sigma_U^{v_1})^{x_a} = \bar{\sigma}_U^{x_a},\end{aligned}$$

$$\frac{\tilde{\sigma}_U}{\bar{B}_U} = \frac{\bar{\sigma}_U^{-e_u}B_U^{v_1}}{B_U^{v_1}h^{-v_2}} = \bar{\sigma}_U^{-e_u}h^{v_2},$$

$$\begin{aligned}\bar{B}_U^{-v_3}\xi^{x_u}h^v &= (B_U^{v_1}h^{-v_2})^{-v_3}\xi^{x_u}h^v = ((gh^{r_u}Y_U)^{v_1}h^{-v_2})^{-v_3}\xi^{x_u}h^v = (gh^{r_u}Y_U)^{-1}h^{v_2v_3}\xi^{x_u}h^v\\ &= g^{-1}h^{-r_u}Y_U^{-1}Y_Uh^{v_2v_3+v} = g^{-1}h^{v_2v_3-r_u+v} = g^{-1},\end{aligned}$$

$$e(\sigma_V, \tilde{Y}_S\mathfrak{g}^{e_v}) = e((gh^{w_v}\tilde{h}^{s_v})^{\frac{1}{x_s+e_v}}, \mathfrak{g}^{x_s+e_v}) = e(gh^{w_v}\tilde{h}^{s_v}, \mathfrak{g}),$$

$$e(\sigma_P, \tilde{Y}_S\mathfrak{g}^{e_p}) = e((gh^{w_p}\tilde{h}^{s_p})^{\frac{1}{x_s+e_p}}, \mathfrak{g}^{x_s+e_p}) = e(gh^{w_p}\tilde{h}^{s_p}, \mathfrak{g}),$$

$$E_V^{x_v} = \xi^{x_v d_v} = Y_V^{d_v} = F_V,$$

$$\frac{P_V}{Q_V^{x_p}} = \frac{Y_UY_P^{z_v}}{\xi^{x_pz_v}} = \frac{Y_UY_P^{z_v}}{Y_P^{z_v}} = Y_U,$$

and

$$\frac{K_V}{E_V^{x_p}} = \frac{Y_VY_P^{d_v}}{\xi^{x_pd_v}} = \frac{Y_VY_P^{d_v}}{Y_P^{d_v}} = Y_V.$$

4 Performance

In this section, we implement our scheme by using the code from pairing-based cryptography (PBC) library [26]. To initialise the bilinear group $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$, an elliptic curve should be selected. Notably, for an elliptic curve, the group size l and the embedded degree d are two import factors. To achieve the same security as the 1024-bit RSA scheme, it is required that $l \times d \geq 1024$. Our smart ticketing for journey privacy scheme is implemented on the Type F curve/BN curve with function $y^2 = x^3 + 3$, where p is a 256-bit prime number, $d = 12$, $\mathbb{G}_1 \neq \mathbb{G}_2$ and there are no efficiently computable isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$. The length of one element in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_\tau$ are 160 bits, 320 bits and 1920 bits, respectively. $SHA-256$ is selected as a hash function.

4.1 Benchmark Time

The time consumed by different operations on the bilinear group $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ is calculated on a MacBook Pro laptop with Intel Core i7 CPU (3.1 GHz) and 16 GB RAM. The time is obtained by calculating the average of running each operation 10 times with random inputs by using the test code from the PBC library [26]. Table 2 describes the running times consumed by different operations on the bilinear group from Type F curve. Pa, PPa, E_1, PE_1, E_2, PE_2, E_τ and PE_τ stand for the time of executing a pairing operation, executing one pairing operation with preprocessing, executing one exponent on $\mathbb{G}_1$, executing one exponent on $\mathbb{G}_1$ with preprocessing, executing one exponent on $\mathbb{G}_2$, executing one exponent on $\mathbb{G}_2$ with preprocessing, executing one exponent on $\mathbb{G}_\tau$ and executing one exponent on $\mathbb{G}_\tau$ with preprocessing, respectively.

Table 2. Benchmark Time (ms)

Curve	Pairing		$\mathbb{G}_1$		$\mathbb{G}_2$		$\mathbb{G}_\tau$		SHA-256
	Pa	PPa	E_1	PE_1	E_2	PE_2	E_τ	PE_τ	
Type F Curve	48.111	47.682	0.739	0.090	1.789	0.226	11.656	1.880	0.007

4.2 Evaluation

The computation cost and communication required by the algorithms in our smart ticketing for journey privacy scheme are described by Fig. 7 and Fig. 8, respectively.

The computation cost of the ticket issuing algorithm is linear with the number of verifiers included in a journey J_U. When $|J_U| = 10$, it takes about 1.2 second to generate a ticket. It takes a verifier $\mathcal{V}$ about 0.11 second to validate a ticket.

The secret-public key sizes of $\mathcal{S}$, $\mathcal{V}$, $\mathcal{U}$ and $\mathcal{P}$ are 92 bytes, 52 bytes, 52 bytes and 52 bytes, respectively, while the size of credentials is the same (84 bytes). The size of a ticket for a journey J_U is linear with the number of verifiers included in J_U. When there are 10 verifiers in J_U, the ticket size is up to 2.56 KB. Notably, the size of each authentication tag Tag_V is only 260 bytes.

5 Security Analysis

Theorem 3. *Our smart-ticketing scheme with journey privacy in Fig. 2, Fig. 3, Fig. 4, Fig. 5 and Fig. 6 is* $(q', \epsilon'(\ell), T')$ *seller secure if the* $(\epsilon(\ell), T)$ *JOC version q-strong Diffie-Hellman*

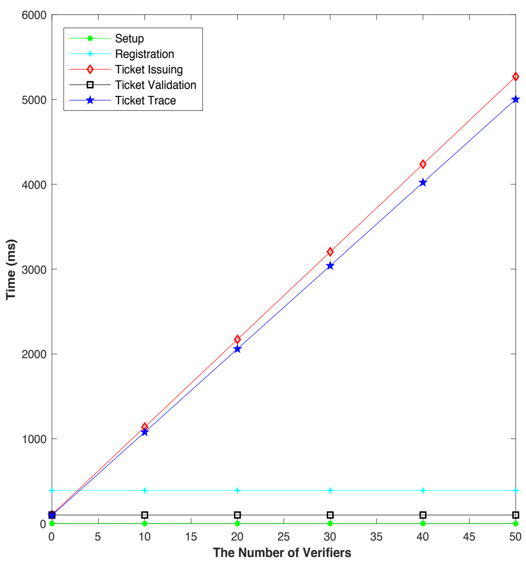

Fig. 7. The Computation Cost of Our JASTS

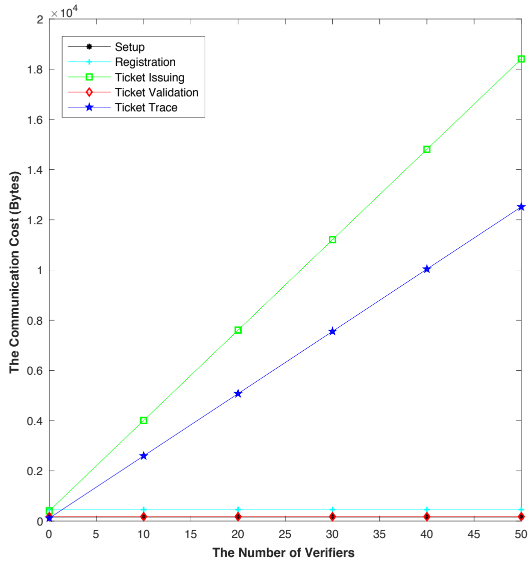

Fig. 8. The Communication Cost of Our JASTS

(JOC-q-SDH) assumption holds on the bilinear group $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau)$ *and* H_1, H_2, H_3 *are secure cryptographic hash functions, where* q' *is the total number of verifiers selected by* $\mathcal{A}$ *to query tickets,* $q' \leq q$, $\epsilon(\ell) = (\frac{p-q}{p} + \frac{1}{p} + \frac{p-1}{p^3})\epsilon'(\ell)$ *and* $T' = \mathcal{O}(T)$.

Proof. If there exists an adversary $\mathcal{A}$ which can break the seller security of our smart ticketing for journey privacy scheme with the advantage $\epsilon'(\ell)$, we can constructs an algorithm $\mathcal{B}$ which can use $\mathcal{A}$ as a subroutine to break the JOC-q-SDH assumption as follows. Given a $(q+3)$-tuple $(g, g^x, \cdots, g^{x^q}, \mathfrak{g}, \mathfrak{g}^x)$, $\mathcal{B}$ will output $(c, g^{\frac{1}{x+c}})$ where $c \in \mathbb{Z}_p - \{-x\}$.

Setup. $\mathcal{B}$ selects $e_1, e_2, \cdots, e_{q-1} \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and sets $f(x) = \prod_{i=1}^{q-1}(x + e_i) = \sum_{i=0}^{q-1} \alpha_i x^i$, $f_i(x) = \frac{f(x)}{x+e_i} = \sum_{j=0}^{q-2} \beta_{i_j} x^j$, $\tilde{g} = \prod_{i=0}^{q-1}(g^{x^i})^{\alpha_i} = g^{f(x)}$, $\hat{g} = \prod_{i=0}^{q-1}(g^{x^{i+1}})^{\alpha_i} = \tilde{g}^x$. $\mathcal{B}$ selects $e, a, k \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computs $h = ((\hat{g}\tilde{g}^e)^k \tilde{g}^{-1})^{\frac{1}{a}} = \tilde{g}^{\frac{(x+e)k-1}{a}}$. $\mathcal{B}$ selects $x_a, \gamma, \vartheta \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and computes $Y_A = \mathfrak{g}^{x_a}$, $\xi = \tilde{g}^\gamma$ and $\tilde{h} = h^\vartheta$. $\mathcal{B}$ selects four three functions $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p$, $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_p$ and $H_3 : \{0,1\}^* \rightarrow \mathbb{G}_1$. $\mathcal{B}$ sends $(e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau, \tilde{g}, h, \xi, \tilde{h}, \mathfrak{g}, Y_A, H_1, H_2, H_3)$ to $\mathcal{A}$.

Registration Query. $\mathcal{A}$ can make the following queries.

1. Ticket Seller Registration Query. $\mathcal{B}$ sets $\tilde{Y}_S = \mathfrak{g}^x$, and computes $Y_S = (\hat{g})^\gamma$. $\mathcal{B}$ selects $e_s, r_s \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and computes $\sigma_S = (\tilde{g}h^{r_s}Y_S)^{\frac{1}{x_a+e_s}}$. $\mathcal{B}$ sends $(Y_S, \tilde{Y}_S, \sigma_S)$ to $\mathcal{A}$.
2. Ticket Verifier Registration Query. Let $Corrupt_V$ be the set consisting of the identities of verifiers who are corrupted by $\mathcal{A}$. $\mathcal{A}$ submits an identity ID_V. $\mathcal{B}$ selects $x_v, e_v, r_v \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and computes $Y_V = \xi^{x_v}$ and $\sigma_V = (\tilde{g}h^{r_v}Y_S)^{\frac{1}{x+e_v}}$. If $ID_V \in Corrupt_V$, $\mathcal{B}$ sends $(x_v, Y_V, e_v, r_v, \sigma_V)$. If $ID_V \notin Corrupt_V$, $\mathcal{B}$ sends $(Y_V, e_v, r_v, \sigma_V)$ to $\mathcal{A}$. $\mathcal{A}$ can adaptively make this registration queries multiple times.

3. User Registration Query. Let $Corrupt_U$ be the set consisting of the identities of users who are corrupted by $\mathcal{A}$. $\mathcal{A}$ submits an identity ID_U. $\mathcal{B}$ selects $x_u, e_u, r_u \xleftarrow{R} \mathbb{Z}_p$, and computes $\sigma_U = (\tilde{g}h^{r_u}Y_U)^{\frac{1}{x+e_u}}$. If $ID_V \in Corrupt_V$, $\mathcal{B}$ sends $(x_u, Y_U, e_u, r_u, \sigma_U)$ to $\mathcal{A}$. If $ID_U \notin Corrupt_U$, $\mathcal{B}$ sends $(Y_U, e_u, r_u, \sigma_U)$ to $\mathcal{A}$. $\mathcal{A}$ can adaptively make this registration queries multiple times.
4. Police Registration Query. $\mathcal{B}$ selects $x_p, e_p, r_p \xleftarrow{R} \mathbb{Z}_p$ and computes $Y_P = \xi^{x_p}$ and $\sigma_P = (\tilde{g}h^{r_p}Y_S)^{\frac{1}{x_a+e_p}}$. $\mathcal{B}$ sends $(Y_P, r_p, e_p, \sigma_P)$ to $\mathcal{A}$.

Ticket Issuing Query. $\mathcal{A}$ can adaptively submit a journey J_U, a set of pseudonyms $Ps_U = \{(P_V, Q_V)_{ID_V \in J_U}\}$ and a proof $\prod_U^1$: PoK$\{(x_u, r_u, e_u, \sigma_U, v_1, v_2, v_3, (z_v)_{ID_V \in J_U}) : \frac{\tilde{\sigma}_U}{\bar{B}_U} = \bar{\sigma}_U^{-e_u}h^{v_2} \wedge g^{-1} = \bar{B}_U^{v_3} g^{x_u} h^{r_u - v_2 v_3} \wedge (P_V = \xi^{x_u}Y_P^{z_v} \wedge Q_V = \xi^{z_v})_{ID_V \in J_U}\}$. $\mathcal{B}$ verifies $\prod_U^1$ and $e(\bar{\sigma}_U, Y) \stackrel{?}{=} e(\tilde{\sigma}_U, \mathfrak{g})$. If the verification is unsuccessful, $\mathcal{B}$ aborts; otherwise, $\mathcal{B}$ works as follows: 1. If $|J_U| = 2\lambda$, let $\Omega_U = J_U$; 2. If $|J_U| = 2\lambda - 1$, let $\Omega_U = J_U \cup \{ID_{du}\}$.

For $ID_V \in J_U$, let $f_v(x) = \frac{f(x)}{x+e_v} = \sum_{k=1}^{q-2} \beta_{v_k} x^k$. $\mathcal{B}$ selects $t_u, d_v, w_v \xleftarrow{R} \mathbb{Z}_p$ and computes $C_U = g^{t_u}$, $D_V = H_2(C_U||ID_V)$, $E_V = \xi^{d_v}$, $F_V = Y_V^{d_v}$, $K_V = Y_V Y_P^{d_v}$, $s_v = H_1(P_V||Q_V||E_V||F_V||K_V||Text)$ and

$$\sigma_V = \prod_{k=0}^{q-2} (g^{x^k})^{\beta_{v_k}(1+\frac{(ek-1)s_v}{a})} \prod_{k=0}^{q-2} (g^{x^{k+1}})^{\frac{\beta_{v_k} k s_v}{a}}.$$

We claim that (w_v, e_v, σ_v) is a valid signature on s_v. We have

$$\begin{aligned}
\sigma_V &= \prod_{k=0}^{q-2} (g^{x^k})^{\beta_{v_k}(1+\frac{(ek-1)(w_v+\vartheta s_v)}{a})} \prod_{k=0}^{q-2} (g^{x^{k+1}})^{\frac{\beta_{v_k} k(w_v+\vartheta s_v)}{a}} \\
&= \prod_{k=0}^{q-2} (g^{\beta_{v_k} x^k})^{(1+\frac{(ek-1)(w_v+\vartheta s_v)}{a})} \prod_{k=0}^{q-2} (g^{\beta_{v_k} x^k})^{\frac{xk(w_v+\vartheta s_v)}{a}} \\
&= (g^{\sum_{k=0}^{q-2} \beta_{v_k} x^k})^{(1+\frac{(ek-1)(w_v+\vartheta s_v)}{a})} (g^{x\sum_{k=0}^{q-2} \beta_{v_k} x^k})^{\frac{k(w_v+\vartheta s_v)}{a}} \\
&= (g^{f_v(x)})^{(1+\frac{(ek-1)(w_v+\vartheta s_v)}{a})} (g^{xf_v(x)})^{\frac{k(w_v+\vartheta s_v)}{a}} \\
&= (g^{f(x)})^{(1+\frac{(ek-1)(w_v+\vartheta s_v)}{a})\frac{1}{x+e_v}} (g^{xf(x)})^{\frac{k(w_v+\vartheta s_v)}{a(x+e_v)}} \\
&= \tilde{g}^{(1+\frac{(ek-1)(w_v+\vartheta s_v)}{a})\frac{1}{x+e_v}} \tilde{g}^{\frac{xk(w_v+\vartheta s_v)}{a(x+e_v)}} \\
&= (\tilde{g}^{(1+\frac{(ek-1)(w_v+\vartheta s_v)}{a})} \tilde{g}^{\frac{xk(w_v+\vartheta s_v)}{a}})^{\frac{1}{x+e_v}} \\
&= (\tilde{g}\tilde{g}^{\frac{ekw_v}{a}} \tilde{g}^{\frac{-w_v}{a}} \tilde{g}^{\frac{ek\vartheta s_v}{a}} \tilde{g}^{\frac{-\vartheta s_v}{a}} \tilde{g}^{\frac{xkw_v}{a}} \tilde{g}^{\frac{xk\vartheta s_v}{a}})^{\frac{1}{x+e_v}} \\
&= \left(\tilde{g}\tilde{g}^{(\frac{(k(e+x)-1)w_v}{a}} \tilde{g}^{(\frac{(k(e+x)-1)\vartheta s_v}{a}}\right)^{\frac{1}{x+e_v}} \\
&= \left((\tilde{g}(\tilde{g}^{\frac{k(e+x)-1}{a}})^{w_v}((\tilde{g}^{\frac{(k(e+x)-1}{a}})^{\vartheta})^{s_v}\right)^{\frac{1}{x+e_v}} \\
&= (\tilde{g}h^{w_v}\tilde{h}^{s_v})^{\frac{1}{x+e_v}}
\end{aligned} \tag{1}$$

For ID_{du}, let $f_d(x) = \frac{f(x)}{x+e_d} = \sum_{k=0}^{q-2} \beta_{d_k} x^k$, where $e_d \in \{e_1, e_2, \cdots, e_{q-1}\}$. $\mathcal{B}$ selects $d', w' \xleftarrow{R} \mathbb{Z}_p$ and $D_{du}, P_{du}, Q_{du}, F_{du} \xleftarrow{R} \mathbb{G}_1$, computes $E_{du} = \xi^{d'}$ $K_{du} = Y_P^{d'} H_3(ID_{du})$, $s' = H_1(P_{du}||Q_{du}||E_{du}||F_{du}||K_{du}||Text)$ and

$$\sigma_{du} = \prod_{k=0}^{q-2} (g^{x^k})^{\beta_{d_k}(1+\frac{(ek-1)(w'+\vartheta s')}{a})} \prod_{k=0}^{q-2} (g^{x^{k+1}})^{\frac{\beta_{d_k} k(w'+\vartheta s')}{a}}.$$

According to Equation (1), (w', e_d, σ_{du}) is a BBS+ signature on s'.

Let $f_p(x) = \frac{f(x)}{x+e_p} = \sum_{k=0}^{q-2} \beta_{p_k} x^k$, where $e_p \in \{e_1, e_2, \cdots, e_{q-1}\}$. $\mathcal{B}$ selects $w_p \xleftarrow{R} \mathbb{Z}_p$ and computes $s_p = H_1(s_1||s_2||\cdots||s_{|\Omega_U|})$ and

$$\sigma_P = \prod_{k=0}^{q-2} (g^{x^k})^{\beta_{p_k}(1+\frac{(ek-1)(w_p+\vartheta s_p)}{a})} \prod_{k=0}^{q-2} (g^{x^{k+1}})^{\frac{\beta_{p_k} k(w_p+\vartheta s_p)}{a}}.$$

According to Equation (1), (w_p, e_p, σ_P) is a BBS+ signature on s_p.

If the q-th signature is required, $\mathcal{B}$ computes $w_p = a - \vartheta s_u$ and $\sigma_P = \tilde{g}^k$. We claim that (w_p, e, σ_P) is valid signature on s_p. We have

$$\sigma_P = \tilde{g}^k = (\tilde{g}\tilde{g}^{\frac{a(k(x+e)-1)}{a}})^{\frac{1}{x+e}} = (\tilde{g}\tilde{g}^{\frac{(w_p+\vartheta s_p)(k(x+e)-1)}{a}})^{\frac{1}{x+e}} = (\tilde{g}\tilde{g}^{\frac{w_p(k(x+e)-1)}{a}}\tilde{g}^{\frac{\vartheta s_p(k(x+e)-1)}{a}})^{\frac{1}{x+e}}$$
$$= \left(\tilde{g}(\tilde{g}^{\frac{k(x+e)-1}{a}})^{w_p}((\tilde{g}^{\frac{k(x+e)-1}{a}})^{\vartheta})^{s_p}\right)^{\frac{1}{x+e}} = (\tilde{g}h^{w_p}\tilde{h}^{s_p})^{\frac{1}{x+e}}.$$

The ticket is $T_U = ((D_V, P_V, Q_V, E_V, F_V, K_V, s_v, w_v, e_v, \sigma_V)_{ID_V \in J_U}, (s_p, w_p, e_p, \sigma_P))$. $\mathcal{B}$ sends $(C_U,\ T_U, Text)$ to $\mathcal{A}$. Let QT be the set consisting of the tickets queried by $\mathcal{A}$ and is initially empty. $\mathcal{B}$ adds (T_U, C_U) into QT.

Output. $\mathcal{A}$ outputs a ticket $T_{U^*} = ((D_{V^*}, P_{V^*}, Q_{V^*}, E_{V^*}, F_{V^*}, K_{V^*}, s_{v^*}, w_{v^*}, e_{v^*}, \sigma_{V^*})_{ID_{V^*} \in J_{U^*}}$, $(s_p, w_p, e_p, \sigma_P))$. Let $(s^*, w^*, e^*, \sigma^*) \in \left((s_{v^*}, w_{v^*}, e_{v^*}, \sigma_{V^*})_{ID_{V^*} \in J_{U^*}}, (s_p, w_p, e_p, \sigma_P)\right)$ be a forged signature/authentication tag.

We consider the following three cases.

- Case-I. $e^* \notin \{e_1, e_2, \cdots, e_{q-1}, e\}$. Let $f_1^*(x) = \frac{f(x)}{x+e^*} = \sum_{i=0}^{q-2} c_i x^i$, $f_2^*(x) = \frac{f(x)(e+x)}{x+e^*} = \sum_{i=0}^{q-1} \tilde{c}_i x^i$ and $f(x) = (x+e^*)c(x) + \theta_0$ where $c(x) = \sum_{i=0}^{q-2} c_i x^i$. Therefore,

$$\sigma^* = (\tilde{g}h^{w^*}\tilde{h}^{s^*})^{\frac{1}{x+e^*}} = \tilde{g}^{\frac{1}{x+e^*}}(h^{w^*}\tilde{h}^{s^*})^{\frac{1}{x+e^*}}.$$

We have

$$\begin{aligned}
\tilde{g}^{\frac{1}{x+e^*}} &= \sigma^* \cdot (h^{w^*}\tilde{h}^{s^*})^{\frac{-1}{x+e^*}} \\
&= \sigma^* \cdot (\tilde{g}^{\frac{w^*((e+x)-1)}{a}}\tilde{g}^{\frac{\vartheta s^*((e+x)-1)}{a}})^{\frac{-1}{x+e^*}} \\
&= \sigma^* \cdot \tilde{g}^{\frac{-(w^*+\vartheta s^*)(x+e)}{a(x+e^*)}} \cdot \tilde{g}^{\frac{w^*+\vartheta s^*}{a(x+e^*)}} \\
&= \sigma^* \cdot g^{\frac{-f(x)(w^*+\vartheta s^*)(x+e)}{a(x+e^*)}} \cdot g^{\frac{f(x)(w^*+\vartheta s^*)}{a(x+e^*)}} \\
&= \sigma^* \cdot g^{\frac{-(w^*+\vartheta s^*)f_2^*(x)}{a}} \cdot g^{\frac{(w^*+\vartheta s^*)f_1^*(x)}{a}} \\
&= \sigma^* \cdot \prod_{k=0}^{q-1} (g^{x^k})^{\frac{-\tilde{c}_k(w^*+\vartheta s^*)}{a}} \cdot \prod_{k=0}^{q-2} (g^{x^k})^{\frac{c_i(w^*+\vartheta s^*)}{a}}.
\end{aligned}$$

Let $\Gamma = \sigma^* \cdot \prod_{k=0}^{q-1} (g^{x^k})^{\frac{-\tilde{c}_k(w^*+\vartheta s^*)}{a}} \cdot (g^{x^i})^{\frac{c_k(w^*+\vartheta s^*)}{a}}$. We have

$$\Gamma = \tilde{g}^{\frac{1}{x+e^*}} = g^{\frac{f(x)}{x+e^*}} = g^{\frac{c(x)(x+e^*)+\theta}{x+e^*}} = g^{c(x)} g^{\frac{\theta_0}{x+e^*}}.$$

Hence,

$$g^{\frac{1}{x+e^*}} = (\Gamma \cdot g^{-c(x)})^{\frac{1}{\theta}} = \left(\sigma^* \cdot \prod_{k=0}^{q-1} (g^{x^i})^{\frac{-\tilde{c}_k(w^*+\vartheta s^*)}{a}} \cdot \prod_{k=0}^{q-2} (g^{x^i})^{\frac{c_k(w^*+\vartheta s^*)}{a}} \cdot \prod_{k=0}^{q-2} (g^{x^k})^{-c_k}\right)^{\frac{1}{\theta}}.$$

- Case-II. $e^* \in \{e_1, e_2, \cdots, e_{q-1}, e\}$. We have $e^* = e$ with the probability $\frac{1}{q}$. Since $e \notin \{e_1, e_2, \cdots, e_{q-1}\}$, $\mathcal{B}$ can output $g^{\frac{1}{x+e}}$ using the same technique above.
- Case-III. $e^* = e_v$, $\sigma^* = \sigma_V$, but $s^* \neq s_v$. Since $\sigma^* = (\tilde{g}h^{w^*}\tilde{h}^{s^*})^{\frac{1}{x+e^*}}$ and $\sigma_v = (\tilde{g}h^{w_V}\tilde{h}^{s_v})^{\frac{1}{x+e_v}}$. We have $h^{w^*}\tilde{h}^{s^*} = h^{w_v}\tilde{h}^{s_v}$, $\tilde{h} = h^{\frac{w^*-w_v}{s_v-s^*}}$ and $log_h\tilde{h} = \frac{w^*-w_v}{s_v-s^*}$. $\mathcal{B}$ can use $\mathcal{A}$ to break the discret logarithm assumption. Therefore $\mathcal{B}$ can use $\mathcal{A}$ to break the JOC-q-SDH assumption since JOC-q-SDH assumption is included in discrete logarithm assumption.

Therefore, the advantage with which $\mathcal{B}$ can break the q-SDH assumption is

$$\begin{aligned} Adv_{\mathcal{B}}^{q-SDH} &= \Pr[\mathsf{Case-I}] + \Pr[\mathsf{Case-II}] + \Pr[\mathsf{Case-III}] \\ &\geq \frac{p-q}{p}\epsilon'(\ell) + \frac{q}{p} \times \frac{1}{q}\epsilon'(\ell) + \frac{1}{p} \times \frac{1}{p} \times \frac{p-1}{p}\epsilon'(\ell) \\ &= (\frac{p-q}{p} + \frac{1}{p} + \frac{p-1}{p^3})\epsilon'(\ell). \end{aligned}$$

Theorem 4. *Our smart ticketing for journey privacy scheme in Fig. 2, Fig. 3, Fig. 4, Fig. 5 and Fig. 6 is $(\epsilon'(\ell), T')$ user secure if the $(\epsilon(\ell), T)$ decisional Diffie-Hellman (DDH) assumption holds on the bilinear group $(e, p, \mathbb{G}, \mathbb{G}_\tau)$, and H_1, H_2, H_3 are cryptographic hash functions, where $\epsilon(\ell) = \frac{\epsilon'(\ell)}{2}$ and $T' = \mathcal{O}(T)$.*

Proof. If there exists an adversary $\mathcal{A}$ can $(\epsilon'(\ell), T)$ break the user security in our smart ticketing for journey privacy scheme, we can construct an algorithm $\mathcal{B}$ which can use $\mathcal{A}$ as a subroutine to break the decisional Diffie-Hellman (DDH) assumption as follows. Given $(\xi, \xi^\alpha, \xi^\beta)$, $\mathcal{C}$ flips an unbiased coin with $\{0,1\}$, and obtains a bit $b \in \{0,1\}$. If $b = 0$, $\mathcal{C}$ sends $T = \xi^{\alpha\beta}$ to $\mathcal{B}$; If $b = 1$, $\mathcal{C}$ sends $T = R$ to $\mathcal{B}$, where $R \xleftarrow{R} \mathbb{G}_2$. $\mathcal{B}$ will output his guss b' on b.

Initialisation. $\mathcal{A}$ submits two verifiers identities $ID_{V_0^*}$ and $ID_{V_1^*}$. $\mathcal{B}$ flip unbiased coin with $\{0,1\}$ and obtains a bit $\mu \in \{0,1\}$. $\mathcal{B}$ sets $Y_{V_\mu^*} = \xi^\alpha$ and $Y_{V_{1-\mu}^*} = \xi^\gamma$ where $\gamma \xleftarrow{R} \mathbb{Z}_p$.

Setup. $\mathcal{B}$ selects $x_a \xleftarrow{R} \mathbb{Z}_p$, $g, h, \xi, \tilde{h}, \xleftarrow{R} \mathbb{G}_1$ and $\mathfrak{g} \xleftarrow{R} \mathbb{G}_2$. $\mathcal{B}$ computes $Y_A = \mathfrak{g}^{x_a}$, and selects $H_1 : \{0,1\}^* \to \mathbb{Z}_p$, $H_2 : \{0,1\}^* \to \mathbb{Z}_p$ and $H_3 : \{0,1\}^* \to \mathbb{G}_1$. $\mathcal{B}$ sends the public parameters $PP = (e, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_\tau, g, h, \xi, \tilde{h}, \mathfrak{g}, Y_A, H_1, H_2, H_3)$ to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ can make the following queries.

Registration Query. $\mathcal{A}$ can make the following registration queries.

1. Ticket Seller Registration Query. $\mathcal{A}$ submits an identity ID_S. $\mathcal{C}$ selects $x_s \xleftarrow{R} \mathbb{Z}_p$, and computs $Y_S = \xi^{x_s}$, and $\tilde{Y}_S = \mathfrak{g}^{x_s}$. $\mathcal{B}$ selects $e_s, r_s \xleftarrow{R} \mathbb{Z}_p$, and computes $\sigma_S = (gh^{r_s}Y_S)^{\frac{1}{x_a+e_s}}$. $\mathcal{B}$ sends $(r_s, e_s, \sigma_S, Y_S, \tilde{Y}_S)$ to $\mathcal{A}$.
2. Ticket Verifier Registration Query. Let $Corrupt_V$ and QR_V be the set consisting of the identities of verifiers corrupted by $\mathcal{A}$ and the set consisting of the information of ticket verifier registration queries, respectively. $\mathcal{A}$ submits an identity $ID_V \notin \{ID_{V_0^*}, ID_{V_1^*}\}$. $\mathcal{B}$ selects $x_v, e_v, r_v \xleftarrow{R} \mathbb{Z}_p$, and computes $Y_V = \xi^{x_v}$ and $\sigma_V = (gh^{r_v}Y_V)^{\frac{1}{x_a+e_v}}$. If $ID_V \in Corrupt_V$, $\mathcal{B}$ sends $(x_v, Y_V, e_v, r_v, \sigma_V)$ to $\mathcal{A}$; If $ID_V \notin Corrupt_V$, $\mathcal{B}$ sends $(Y_V, e_v, r_v, \sigma_V)$ to $\mathcal{A}$. For $ID_V \in \{ID_{V_0^*}, ID_{V_1^*}\}$, $\mathcal{B}$ selects $e_\mu, r_\mu, e_{1-\mu}, r_{1-\mu} \xleftarrow{R} \mathbb{Z}_p$, and computes $\sigma_{V_\mu^*} = (gh^{r_\mu}Y_{V_\mu^*})^{\frac{1}{x_a+e_\mu}}$ and $\sigma_{V_{1-\mu}^*} = (gh^{r_{1-\mu}}Y_{V_{1-\mu}^*})^{\frac{1}{x_a+e_{1-\mu}}}$. $\mathcal{B}$ sends $(Y_{V_\mu^*}, r_\mu, e_\mu, \sigma_{V_\mu^*})$ and $(Y_{V_{1-\mu}^*}, r_{1-\mu}, e_{1-\mu}, \sigma_{V_{1-\mu}^*})$ to $\mathcal{A}$. $\mathcal{B}$ adds $(ID_V, Y_V, e_v, r_v, \sigma_V)$ into QR_V. $\mathcal{A}$ can adaptively make this registration query multiple times.

3. User Registration Query. Let $Corrupt_U$ and RQ_U be the set consisting of the identities of users corrupted by $\mathcal{A}$ and the set consisting of the information of user registration queries, respectively. $\mathcal{A}$ submits an identity ID_U. $\mathcal{B}$ selects $x_u, e_u, r_u \xleftarrow{R} \mathbb{Z}_p$, and computes $Y_U = \xi^{x_u}$ and $\sigma_U = (g_0 h^{r_u} Y_U)^{\frac{1}{x_a+e_u}}$. If $ID_U \in Corrupt_U$, $\mathcal{B}$ sends $(x_u, Y_U, r_u, e_u, \sigma_U)$ to $\mathcal{A}$. If $ID_U \notin Corrupt_U$, $\mathcal{B}$ sends $(Y_U, r_u, e_u, \sigma_U)$ to $\mathcal{A}$. $\mathcal{B}$ adds $(ID_U, Y_U, e_u, r_u \sigma_U)$ to QK_U. $\mathcal{A}$ can adaptively make this registration queries multiple times.
4. Police Registration Query. $\mathcal{A}$ submits an identity ID_P. $\mathcal{P}$ selects $x_p, e_p, r_p \xleftarrow{R} \mathbb{Z}_p$ and computs $Y_P = \xi^{x_p}$ and $\sigma_P = (gh^{r_p} Y_P)^{\frac{1}{x_a+e_p}}$. $\mathcal{B}$ sends $(Y_P, r_p, e_p, \sigma_P)$ to $\mathcal{A}$.

Ticket Issuing Query. $\mathcal{A}$ submits an identity $ID_U \in QR_U$, a journey J_U, a set of pseudonym $Ps_U = \{(P_V, Q_V)_{ID_V \in J_U})\}$, and a proof $\mathrm{PoK}\{(x_u, r_u, e_u, \sigma_U, v_1, v_2, v_3, (z_v)_{ID_V \in J_U}) : \frac{\tilde{\sigma}_U}{\bar{B}_U} = \bar{\sigma}_U^{-e_u} h^{v_2} \wedge g_0^{-1} = \bar{B}_U^{v_3} g^{x_u} h^{r_u - v_2 v_3} \wedge (P_V = \xi^{x_u} Y_P^{z_v} \wedge Q_V = \xi^{z_v})_{ID_V \in J_U}\}$. If the proof is incorrect, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ works as follows. If $|J_U| = 2\lambda$, let $\Omega_U = J_U$; If $|J_U| = 2\lambda - 1$, let $\Omega_U = J_U \cup \{ID_{du}\}$.

For $ID_V \in J_U$, $\mathcal{B}$ selects $t_u, d_v, w_v, e_v \xleftarrow{R} \mathbb{Z}_p$ and computes

$$C_U = \xi^{t_u},\ D_V = H_2(C_U || ID_V),\ E_V = \xi^{d_v},\ F_V = Y_V^{d_v},\ K_V = Y_V Y_P^{d_v},$$
$$s_v = H_1(P_V || Q_V || E_V || F_V || K_V || Text) \text{ and } \sigma_V = (gh^{w_v} \tilde{h}^{s_v})^{\frac{1}{x_s+e_v}}.$$

For ID_{du}, $\mathcal{B}$ selects $d', w', e' \xleftarrow{R} \mathbb{Z}_p$ and $D_{du}, P_{du}, Q_{du}, F_{du} \xleftarrow{R} \mathbb{G}_1$, and computes $E_{du} = \xi^{d'}$, $K_{du} = Y_P^{d'} H_3(ID_{du})$, $s' = H_1(D_{du} || P_{du} || Q_{du} || E_{du} || F_{du} || K_{du} || Text)$ and $\sigma_{du} = (gh^{w'} \tilde{h}^{s'})^{\frac{1}{x_s+e'}}$.

$\mathcal{B}$ select $w_p, e_p \xleftarrow{R} \mathbb{Z}_p$, and computes $s_p = H_1(s_1 || s_2 || \cdots || s_{|\Omega_U|})$ and $\sigma_P = (gh^{w_p} \tilde{h}^{s_p})_{\frac{1}{x+w_p}}$. The ticket is $T_U = ((D_V, P_V, Q_V, E_V, F_V, K_V, s_v, w_v, \sigma_V)_{ID_V \in J_U}, (s_p, w_p, \sigma_P))$. $\mathcal{B}$ returns $(C_U, T_U, Text)$ to $\mathcal{A}$. Let QT be the set consisting of the tickets queried by $\mathcal{A}$ and is initially empty. $\mathcal{B}$ adds $(Ps_U, J_U, T_U, t_u, (d_v)_{ID_V \in J_U})$ into QT.

Ticket Validation Query. $\mathcal{A}$ submits $(P_V, Q_V, E_V, F_V, K_V, s_v, w_v, e_v, \sigma_V)$ and a proof $\prod_U^2 : \mathrm{PoK}\{(x_u, z_v) : P_V = \xi^{x_u} Y_P^{z_v} \ \wedge\ Q_V = \xi^{z_v}\}$. B checks whether $(P_V, Q_V, E_V, F_V, K_V, s_v, w_v, \sigma_V) \in QT$. If not, $\mathcal{B}$ aborts; otherwise, $\mathcal{B}$ computes $Y_V = F_j^{\frac{1}{d_v}}$ and checks $D_V \stackrel{?}{=} H_2(C_U || ID_V)$, $s_v \stackrel{?}{=} H_1(|| P_V || Q_V || E_V \ || F_V || K_V || Text)$ and $e(\sigma_V, Y_S \mathfrak{g}^{e_v}) \stackrel{?}{=} e(gh^{w_v} \tilde{h}^{s_v}, \mathfrak{g})$. If the above equations hold. $\mathcal{B}$ returns ID_V to $\mathcal{A}$; otherwise, $\perp$ is returned to indicate failure. Let QV be the set consisting of ticket validation queries made by $\mathcal{A}$ and initially empty. $\mathcal{B}$ adds $(P_V, Q_V, E_V, F_V, K_V, s_v, w_v, e_v, \sigma_V)$ into QV.

Ticket Trace Query. $\mathcal{A}$ submits a ticket T_U. $\mathcal{B}$ computes $Y_U = \frac{P_V}{Q_V^{x_P}}$ and $Y_V = \frac{K_V}{E_V^{x_P}}$, and checks: (1) $s_v \stackrel{?}{=} H_1(P_{Vj} || Q_V || E_V || K_V || Text)$; (2) $e(\sigma_V, Y_S \mathfrak{g}^{e_v}) \stackrel{?}{=} e(gh^{w_v} \tilde{h}^{s_v}, \mathfrak{g})$; If (1) and (2) hold, $ID_V \in J_U$; otherwise, $ID_V \notin J_U$. (3) $s_p \stackrel{?}{=} H_1(s_1 || s_2 || \cdots || s_{|\Omega_U|})$; (4) $e(\sigma_P, \tilde{Y}_S \mathfrak{g}^{w_p}) \stackrel{?}{=} e(gh^{w_p} \tilde{h}^{s_p}, \mathfrak{g})$. If (1), (2), (3) and (4) hold, $\mathcal{B}$ sends the public key Y_U and the journey information $J_U = \{ID_V\}$ to $\mathcal{A}$. Let QT be a set consisting of the ticket trace queries made by $\mathcal{A}$ and initially empty. $\mathcal{B}$ adds T_U into QT.

Challenge. $\mathcal{B}$ selects $z_\mu^*, t_\mu^*, w_\mu^*, e_\mu^*, w^*, e^* \xleftarrow{R} \mathbb{Z}_p$, and computes $P_\mu^* = Y_U Y_P^{z_\mu^*}$, $Q_\mu^* = \xi^{z_\mu^*}$, $C_\mu^* = \xi^{t_\mu^*}$, $D_\mu^* = H_2(C_U^* || Y_{V_\mu^*})$, $E_\mu^* = \xi^\beta$, $F_\mu^* = T$, $K_\mu^* = (E_\mu^*)^{x_P} Y_{V_\mu^*}$, $s_\mu^* = H_1(P_U^* || Q_U^* || E_\mu^* || F_\mu^* || K_\mu^* || Text)$, $\sigma_\mu^* = (gh^{w_\mu^*} \tilde{h}^{s_\mu^*})^{\frac{1}{x_s+e_\mu^*}}$, $s^* = H_1(s_\mu^*)$ and $\sigma^* = (gh^{w^*} \tilde{h}^{s^*})^{\frac{1}{x_s+e^*}}$. $\mathcal{B}$ sends $((P_\mu^*, Q_\mu^*, E_\mu^*, F_\mu^*, K_\mu^*, s_\mu^*, w_\mu^*, e_\mu^*, \sigma_\mu^*), (s^*, w^*, e^*, \sigma^*))$ to $\mathcal{A}$.

Phase 2. It is the same as in Phase 1 with the limitation that $(E^*_\mu, F^*_\mu) \notin QV$ and $(E^*_\mu, F^*_\mu) \notin QT$.

Output. $\mathcal{A}$ outputs his guess μ' on μ. If $\mu' = \mu$, $\mathcal{B}$ outputs $b' = 0$; otherwise, $\mathcal{B}$ outputs $b = 1$.

Now, we compute the probability with which $\mathcal{B}$ can break the DDH assumption. If $b = 0$ and $T = \xi^{\alpha\beta}$, $(D^*_\mu, P^*_\mu, Q^*_\mu, E^*_\mu, F^*_\mu, K^*_\mu, w^*_\mu, e^*_\mu, \sigma^*_\mu)$ is a valid authentication tag, so $\mathcal{A}$ can outputs $\mu' = \mu$ with $\Pr[\mu' = \mu | b = 0] \geq \frac{1}{2} + \epsilon'(\ell)$. When $\mu' = \mu$, $\mathcal{B}$ outputs $b' = 0$. Hence, $\Pr[b' = b | b = 0] \geq \frac{1}{2} + \epsilon(\ell)$. If $b = 1$ and $T = R$, $(D^*_\mu, P^*_\mu, Q^*_\mu, E^*_\mu, F^*_\mu, K^*_\mu, w^*_\mu, e^*_\mu, \sigma^*_\mu)$ are random elements in $\mathbb{G}_1$, so $\mathcal{A}$ can output $\mu' \neq \mu$ with $\Pr[\mu' \neq \mu | b = 1] = \frac{1}{2}$. When $\mu' \neq \mu$, $\mathcal{B}$ outputs $b = 1$. Hence, $\Pr[b' = b | b = 1] = \frac{1}{2}$.

Therefore, the advantage with which $\mathcal{B}$ can break the DDH assumption is

$$Adv^{DDH}_{\mathcal{B}} = \left| \frac{1}{2} \times \Pr[b' = b | b = 0] - \frac{1}{2} \times \Pr[b' = b | b = 1] \right| \geq \frac{1}{2}(\frac{1}{2} + \epsilon'(\ell)) - \frac{1}{2} \times \frac{1}{2} = \frac{\epsilon'(\ell)}{2}.$$

6 Conclusion

Privacy-preserving smart ticketing schemes have been proposed to protect customers' personal identity information, but customers' journey privacy has not been focused extensively. However, journey information is sensitive since malicious party can infer users' lifestyles, private businesses, relationships, health condition, *etc.*

To protect customers' personal identity information and enable them to control release their journey information, this paper proposed a smart ticketing for journey privacy scheme. This scheme provides the following features: (1) For a journey, only one ticket is issued to a user, even if he/she needs multiple transits; (2) Users can purchase tickets from the ticket seller anonymously without releasing anything about their personal identity information, namely the ticket seller cannot detect whether two journeys are from two different users or a same user; (3) Ticket verifiers can be convinced that whether a user is authorised to pass the stations and cannot profile the user's journey even if they collude; (4) For public safety, a trusted party named police is authorised to trace a user's journey if required; (5) The journey in a ticket is fixed to prevent a user from using a cheaper ticket to have a long journey when there are multiple hops between the starting station and the destination station.

Acknowledgement

This work is partially supported by the project ****

References

1. Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 271–286. Springer, 2000.
2. Ghada Arfaoui, Jean-Franois Lalande, Nicolas Desmoulins Jacques Traoré, Pascal Berthomé, and Saïd Gharou. Practical set-membership proof for privacy-preserving nfc mobile ticketing. *Proceedings on Privacy Enhancing Technologies*, (2):25–45, 2015.
3. Man Ho Au, Willy Susilo, and Yi Mu. Constant-size dynamic k-taa. In Roberto De Prisco and Moti Yung, editors, *SCN 2006*, volume 4116 of *Lecture Notes in Computer Science*, pages 111–125. Springer, 2006.

4. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer, 2004.
5. Dan Boneh and Xavier Boyen. Short signatures without random oracles and the sdh assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, 2008.
6. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Christian Cachin and Jan L. Camenisch, editors, *CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.
7. Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*. Springer, 2001.
8. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. *Journal of Cryptology*, 17(4):297–319, 2004.
9. Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In Birgit Pfitzmann, editor, *CCS 2004*, pages 68–177. ACM, 2004.
10. Stefan Brands. Untraceable off-line cash in wallets with observers (extended abstract). In Douglas R. Stinson, editor, *CRYPTO 1993*, volume 773 of *Lecture Notes in Computer Science*, pages 302–318. Springer, 1993.
11. Jan Camenisch, Rafik Chaabouni, and abhi shelat. Efficient protocols for set membership and range proofs. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *Lecture Notes in Computer Science*, page 234?252. Springer, 2008.
12. Jan Camenisch, Manu Drijvers, and Anja Lehmann. Anonymous attestation using the strong Diffie Hellman assumption revisited. In Michael Franz and Panos Papadimitratos, editors, *TRUST 2016*, volume 9824 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2016.
13. Jan Camenisch, Aggelos Kiayias, and Moti Yung. On the portability of generalized schnorr proofs. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 425–442. Springer, 2009.
14. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer, 2001.
15. Jan Camenisch and Markus Michels. Proving in zero-knowledge that a number is the product of two safe primes. In Jacques Stern, editor, *EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 107–122. Springer, 1999.
16. Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups (extended abstract). In Burton S. Kaliski Jr, editor, *CRYPTO 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 1997.
17. David Chaum and Torben P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *CRYPTO 1992*, volume 1993 of *Lecture Notes in Computer Science*, pages 89–105. Springer, 1993.
18. Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Information Theory Society*, 22(6):644–654, 1976.
19. EZ-LINK. Card system and technology.
20. Chun-I Fan and Chin-Laung Lei. Multi-recastable ticket schemes for electronic voting. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, E81-A(5):940–949, 1998.
21. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.
22. Essam Ghadafi, Nigel P. Smart, and Bogdan Warinschi. Groth-Sahai proofs revisited. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 177–192. Springer, 2010.
23. Ivan Gudymenko. A privacy-preserving e-ticketing system for public transportation supporting fine-granular billing and local validation. In *SIN 2014*, pages 101–107. ACM, 2014.
24. Florian Kerschbaum, Hoon Wei Lim, and Ivan Gudymenko. Privacy-preserving billing for e-ticketing systems in public transportation. In *WPES 2013*, pages 143–154. ACM, 2013.

25. Nicolai Kuntze and Andreas U. Schmidt. Trusted ticket systems and application. In Hein Venter-Mariki, EloffLes Labuschagne, and Jan Eloff andRossouw von Solms, editors, *SEC 2007*, volume 232 of *IFIP International Federation for Information Processing*, pages 49–60. Springer, 2007.
26. Ben Lynn. The pairing-based cryptography library, 2006.
27. Milica Milutinovic, Koen Decroix, Vincent Naessens, and Bart De Decker. Privacy-preserving public transport ticketing system. In Pierangela Samarati, editor, *DBSec 2015*, volume 9149 of *Lecture Notes in Computer Science*, pages 135–150. Springer, 2015.
28. Macià Mut-Puigserver, M. Magdalena Payeras-Capellà, Josep-Lluís Ferrer-Gomila, Arnau Vives-Guasch, and Jordi Castellà-Roca. A survey of electronic ticketing applied to transport. *Computers & Security*, 31(8):925–939, 2012.
29. Toru Nakanishi, Nobuaki Haruna, and Yuji Sugiyama. Unlinkable electronic coupon protocol with anonymity control. In Masahiro Mambo and Yuliang Zheng, editors, *ISW 1999*, volume 1729 of *Lecture Notes in Computer Science*, pages 37–46. Springer, 1999.
30. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999.
31. Torben Pryds Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *CRYPTO 1991*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 1999.
32. Daniele Quercia and Stephen Hailes. Motet: Mobile transactions using electronic ticket. In *SECURECOMM 2005*, pages 1–10. IEEE, 2005.
33. Andy Rupp, Gesine Hinterwälder, Foteini Baldimtsi, and Christof Paar. P4r: Privacy-preserving pre-payments with refunds for transportation systems. In Ahmad-Reza Sadeghi, editor, *FC 2013*, volume 7859 of *Lecture Notes in Computer Science*, pages 205–212. Springer, 2013.
34. Andy Rupp, Gesine Hinterwälder, Foteini Baldimtsi, and Christof Paar. Cryptographic theory meets practice: Efficient and privacy-preserving payments for public transport. *ACM Transactions on Information and System Security*, 17(3):10:01–10:31, 2015.
35. Claus-Peter Schnor. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.
36. Ronggong Song and Larry Korba. Pay-TV system with strong privacy and non-repudiation protection. *IEEE Transactions on Consumer Electronics*, 49(2):408–413, 2003.
37. Arnau Vives-Guasch, Jordi Castellà-Roca, M. Magdalena Payeras-Capella, and Macià Mut-Puigserver. An electronic and secure automatic fare collection system with revocable anonymity for users. In *MoMM2010*, pages 387–392. ACM, 2010.
38. Arnau Vives-Guasch, Magdalena Payeras-Capellá, Maciá Mut Puigserver, Jordi Castellá-Roca, and Josep Lluŝ Ferrer-Gomila. A secure e-ticketing scheme for mobile devices with near field communication (nfc) that includes exculpability and reusability. *IEICE Transactions on Information and Systems*, 95(D(1)):78–93, 2012.
39. Arnau Vives-GuaschM, Magdalena Payeras-Capellà, Macià Mut-Puigserver, Jordi Castellà-Roca, and Josep-Lluís Ferrer-Gomila. Anonymous and transferable electronic ticketing scheme. volume 8247 of *Lecture Notes in Computer Science*, pages 100–113. Springer, 2013.