IEICE TRANS. INF. & SYST., VOL.E95–D, NO.1 JANUARY 2012

PAPER Special Section on Trust, Security and Privacy in Computing and Communication System

A Secure E-Ticketing Scheme for Mobile Devices with Near Field Communication (NFC) That Includes Exculpability and Reusability

Arnau VIVES-GUASCH[†a)], ***Student Member***, **Maria-Magdalena PAYERAS-CAPELLÀ**[††b)], **Macià MUT-PUIGSERVER**[††c)], **Jordi CASTELLÀ-ROCA**[†d)], ***and*** **Josep-Lluís FERRER-GOMILA**[††e)], ***Nonmembers***

SUMMARY An electronic ticket is a contract, in digital format, between the user and the service provider, and reduces both economic costs and time in many services such as air travel industries or public transport. However, the security of the electronic ticket has to be strongly guaranteed, as well as the privacy of their users. We present an electronic ticketing system that considers these security requirements and includes the exculpability as a security requirement for these systems, i.e. users and the service provider can not falsely accuse each other of misbehavior. The system ensures that either both parties receive their desired data from other or neither does (fair exchange). Another interesting property is reusability. Thanks to reusability the tickets can be used a predefined number of times with the same security as single tickets. Furthermore, this scheme takes special care of the computational requirements on the users side by using light-weight cryptography. We show that the scheme is usable in practice by means of its implementation using mobile phones with Near Field Communication (NFC) capabilities.

key words: *security, privacy, electronic commerce, e-commerce, electronic ticketing, e-ticketing*

1. Introduction

Information technologies (IT) are becoming usual in our society as they progressively replace the use of paper in many of our common operations. An example of paper ticket could be the air flight boarding pass. Vodafone and Spanair* conducted an e-ticketing test in Spain (May 2007). The passengers received the electronic boarding pass in their mobile phone, and they were able to go directly to the security control area, and later board. The International Air Transport Association (IATA) started in 2004 a program to introduce the use of electronic tickets. IATA estimates that the no-use of paper tickets will reduce the costs by US$ 3000M**, boosting disintermediation by using electronic tickets. The electronic tickets have not been used only as a boarding pass. They can also be used in multiple transport services. The AMSBUS*** booking system from the Czech Republic allows the purchase of SMS tickets. The passenger receives the ticket in her mobile phone; then, the user shows the message to the ticket inspector when needed. Leeds United**** supporters can book one of their sport events and later receive an SMS with the booking confirmation together with some added information such as the assigned seat.

These examples show the progressive introduction of electronic tickets on different kinds of services and the increasing use of mobile phones in all of them as the most suitable e-ticket storage device. In addition to that, the real application of these electronic ticketing systems depends on their security, due to the ease of copy of electronic data. Electronic tickets have to keep the same security that is offered in paper tickets.

The main focus area of the present paper is the development of a secure e-ticketing scheme for mobile devices. Our protocol presents a fair-trading mechanism during the ticket verification in such a way the user pays in exchange of the right to use the agreed service. As in our previous work [1], the protocol is designed to meet the set of security requirements for such schemes described in Sect. 2.1. We would like to highlight the exculpability property, which is a new property that we have first introduced in the e-ticketing schemes (i.e. the service provider can not falsely accuse the user of ticket overspending, and the user is able to demonstrate that she has already validated the ticket before using it). In addition to that, now the protocol has been enhanced with the property of reusability (i.e the tickets can be used a predefined number of times with the same security as single tickets). The final contribution of our paper is the implementation of the proposed e-ticketing protocol using light-weighted mechanisms by means of low computational complexity cryptography and low communicational overhead.

1.1 Organization

An analysis of the previous works related to our proposal is presented in Sect. 2 including also the most common secu-

Manuscript received March 15, 2011.
Manuscript revised June 30, 2011.

[†]The authors are with the Dpt. d'Enginyeria Informàtica i Matemàtiques, UNESCO Chair in Data Privacy, Universitat Rovira i Virgili, Av. Països Catalans 26, E-43007 Tarragona, Spain.
[††]The authors are with the Dpt. de Ciències Matemàtiques i Informàtica, Universitat de les Illes Balears, Ctra. de Valldemossa, km 7,5. 07122 Palma de Mallorca, Spain.
a) E-mail: arnau.vives@urv.cat
b) E-mail: mpayeras@uib.es
c) E-mail: macia.mut@uib.es
d) E-mail: jordi.castella@urv.cat
e) E-mail: jlferrer@uib.es
DOI: 10.1587/transinf.E95.D.78

*http://www.spanair.com/web/es-es/Sobre-Spanair/Noticias-y-eventos/Spanair-y-Vodafone-Espana-presentan-la-tarjeta-de-embarque-movil/
**IATA: http://www.iata.org/pressroom/pr/2008-31-05-01.htm
***AMSBUS: http://www.svt.cz/en/amsbus/
****Leeds United: http://www.leedsunited.com/

rity requirements. Later, in Sect. 3, our scheme is accurately defined. The security and privacy analysis of the scheme is detailed in Sect. 4. Implementation details and performance results are given in Sect. 5. Finally, the conclusions and future work are described in Sect. 6.

2. Previous Work

First of all, in Sect. 2.1, we start presenting the security requirements that have to be achieved in these systems, as well as we introduce a new security requirement that is not achieved in the previous works and which could be taken into account in the future: *exculpability* together with a property that we have recently achieved: *reusability*. Section 2.2 lists other relevant properties of e-ticketing systems. Once the requirements are described, the e-ticketing proposals have been classified in Sect. 2.3.

2.1 Security Requirements

E-ticketing systems have to consider and guarantee the following security requirements:

- Authenticity: A user has to be able to verify if an e-ticket has been issued by an authorized issuer.
- Integrity: A user has to be able to verify if her e-ticket has been altered as regards the one issued by the correspondent authorized issuer.
- Non-repudiation: Once a valid e-ticket has been issued, the issuer cannot deny that she has issued that ticket with its contents. Observe that, in fact, this property comprehends the two previous properties: if the issuer can not deny having issued an e-ticket it means that integrity and authenticity properties have been achieved.
- Unforgeability: Only authorized issuers can issue valid e-tickets.
- Non-Overspending: E-tickets can only be used as agreed between the issuer and the user. Non-reusable e-tickets can not be reused after they have been spent (by the same or other users). Reusable e-tickets can be used exactly the number of times agreed in the moment of issue. Finally, some e-tickets can not be used after their valid period of time. Mechanisms to control overspending can affect the following property: anonymity. Overspending can be prevented or detected. If overspending is detected in the verification phase overspending will not be allowed. If it is detected afterwards, some way to identify the overspender or possible overspenders will be necessary. Some authors [2] call duplication to this property.
- Anonymity: Not all the paper-tickets present the same requirements related to anonymity, so we have to distinguish some possible scenarios for e-tickets. There are three anonymity degrees: non-anonymous (the service requires user identification and authentication), anonymous, and revocable anonymous (the service is anonymous, but it can be revoked if the user misbehaves).
  - Non-anonymous e-tickets: Some e-tickets will have to be non-anonymous; it means that user identity has to be embedded in the e-ticket in some way, in order that the service provider could verify that the user is authorized to spend the e-ticket. This is the case of plane e-tickets. In the boarding phase the auxiliary staff of the Air Company have to be able to verify that the flyer identity is the same as the identity contained in the e-ticket.
  - Anonymous e-ticket: Some paper tickets allow users to remain anonymous in front of the issuer and/or the verifier. Therefore e-tickets will have to maintain the property. Perhaps in the issue phase the user is identified (it depends on the kind of payment used), but that payment has not to be linked to the issued e-ticket. In any case, it has to be able to spend the e-ticket without any kind of identification. Even colluded issuers and service providers should not be able to break anonymity of honest consumers.
  - Revocable anonymity: If overspending is detected after the verification process, it means that the same e-ticket could be used more times than desired (by issuer and/or service provider). For non-anonymous e-ticket this is not a problem: we know overspender user and so actions can be undertaken. For anonymous e-tickets, anonymity has to be revocable in order to identify overspenders. Obviously, fair users would have to remain anonymous or, at least, they would have to be able to prove they are fair users. Some e-ticketing schemes allow also the revocation of the anonymity of the user if she misbehaves using the service.
- Expiry date: A ticket could be only valid during a time interval.
- Online/Offline: Ticket verification can require a persistent connection with a trusted centralized system or Trusted Third Party (online); otherwise, that connection is never necessary (offline).
- Fairness: During the execution of an e-ticketing protocol the parties execute several exchanges of elements. One of these exchanges is produced during the issue phase. Many times an e-ticket is exchanged for a payment or some other element. We can think of some exceptions: donations (between users), free e-ticket (for some events), etc. But in the general case user will have to pay for an e-ticket. During the verification phase another exchange is produced between the user and the provider: the ticket and the service (for that reason they can exchange exculpability proofs). Therefore a protocol for that exchange will have to be designed, and some properties achieved. We are in front of a kind of fair exchange of values (an e-ticket for a payment, a serveice for an e-ticket), and so, some of the following properties will be necessary: fairness, abuse-freeness,

timeliness, verifiability of the TTP, etc. It is out of the scope of this paper to explain these properties that can be found, for instance, in [3].

- Transferability: Some paper tickets can be transferred to other people (spectacle tickets, bus tickets, etc.). Obviously it is not the case of identified e-tickets (plane e-tickets, etc.). People receiving an e-ticket in a transfer (not directly from an authorized issuer) has to be able to verify that this e-ticket is valid (it will be easy if non-repudiation, integrity and authenticity are met) and not spent by the transferring entity. When we are in front of gifts or donations between confident people (a friend, familiar, etc.) no special measures have to be taken, it's a personal matter if afterwards an overspending occurs. But perhaps e-tickets can be resold, or e-tickets (spectacle entrances) can be a present from a third company (in exchange of buying some product from this company). The receiving entity has to be sure that the e-ticket is valid and not spent. But its possible that the user will try to overspend the e-ticket, and the transferring entity has to be able to prove she has not reused the e-ticket. This problem should be specially handled when anonymity is revocable. Transferability will make necessary the fairness property.

2.1.1 Exculpability

None of the analyzed proposals deals with exculpability; that is, the service provider can not falsely accuse the user of ticket overspending, and the user is able to demonstrate that she has already validated the ticket before using it. The exculpability is an important property in our proposal, as the e-ticketing scheme should ensure that either both parties (users and provider) receive their desired data (e-ticket and the validated e-ticket) from other or neither does (fair exchange). The parties agree to reveal its data only if the other part also agrees. If any party deviates from the scheme then it can be identified as the culprit by the Trusted Third Party (TTP). Our scheme defined in Sect. 3 takes exculpability as a security requirement for an e-ticketing system, as the first step to include this security requirement in future works.

2.1.2 Reusability

A ticket could be used once (non-reusable) or many times (reusable). In both cases, ticket overspending has to be prevented. Tickets can be used more than once as is the case of some urban transport, where a transport pass can be used for several travels (and a counter is decreased in every travel) or it can be used over a period of time. Even, sometimes, the same ticket can be used in different places (for instance, bus and underground in the same city). E-tickets have to incorporate security measures that allow using the ticket in the valid period of time or for the number of uses agreed (or a combination of both, time and uses). Some authors name divisibility to this property (probably influenced by the similarities between e-ticket and e-money).

2.2 Other Requirements

There are some other requirements that they are not so directly related to security, but they can be so important than those explained previously.

- Portability: E-tickets, as paper tickets, have to be portable by users. So, it has not to be necessary a laptop or a personal computer to handle e-tickets. Mobile phones, smart cards, etc. will have to be able to store and process e-tickets.
- Reduced size: Typically, e-tickets will be stored in mobile devices (a mobile terminal as a mobile phone, a smart card, etc.), and sometimes these devices will have a limited memory. Therefore, e-tickets have to be reduced in size as possible.
- Flexibility: We can think of a lot of different tickets (plane tickets, bus tickets, concert tickets, museum tickets, etc.). We can design a specific e-ticket for each application, or we can adapt a general e-ticket for each application. Obviously the later solution is preferred in order to economize the solution, and it will allow a better security analysis.
- Ease of use: We are thinking e-ticketing as a solution for general public (using paper tickets nowadays, and not necessary especially confident in electronic means). E-tickets have to be as easy to use as paper tickets, and without new problems for users.
- Efficiency: We can think efficiency from two points of view. First, mobile terminals can be limited in terms of computational power, and so protocol operations and especially cryptographic operations have to be reduced only to necessary ones. Second, communication capacity can also be limited, and so the protocol has to be designed with this constrain in mind. Whatever, the delay due to verification of validity of e-ticket has to be reasonable to be a valid solution for ticketing by electronic means.
- Payment system: The design of an e-ticketing system has to bear in mind that sometimes it will be necessary a payment system to obtain the e-ticket. So, e-ticketing system has to allow different payment systems to be used in order to pay for the e-ticket (if necessary).
- Globally-spendable: Costumer should be able to spend their e-tickets at any appropriate service provider.
- Offline verification: In some scenarios it will not be possible to contact with external databases or Trusted Third Parties, to verify if e-ticket is valid or not. Perhaps it will not be the general case, but a solution for this case has to be thought. This property is much related to the security mechanisms adopted. If an online solution is preferred, efficiency has to be especially remembered.
- Availability: This property can be seen as a security property, but it is quite difficult to address this problem only as a security one. We are thinking in denial of ser-

vice attacks (difficult to handle), disaster events (more difficult to handle) or temporal malfunction of infrastructure (for instance, a power failure). This can mean that e-tickets can not be verified, and sometimes the event can not be delayed (a concert, a plane, etc.). A procedure to handle these situations has to be designed.

2.3 Classification of Proposals

In this section, the proposals have been differentiated by the devices used in the systems: the *smart-card-based*, and the *non-smart-card-based* ones, with a deeper analysis performed in the non-smart-card-based proposals, taking into account their anonymity compliance.

2.3.1 Smart-Card-Based

Smart-card-based proposals [2], [4]–[9] establish a communication channel with the verification system. Thus, the most sensitive operations are sent to the smart-card through this channel. The smart-card verifies each operation, so that users can not perform any non-allowed action. Security of smart-card-based systems rely on the smart-card security. If the smart-card security is compromised, the security of the entire system is also compromised. One recent example is the case of the Mifare cards, where in [10] the authors succeeded to compromise them. As a result, all the entire public transport system that used exclusively Mifare cards was compromised.

2.3.2 Non-smart-Card-Based

Non-smart-card-based systems [11]–[17] allow to perform applications with higher computation requirements while taking advantage of their high storage capacity and also their wireless short-range communication resources; this is the case of the mobile phones, smart phones or PDA's. However, as these devices are not considered tamper-proof devices, e-ticketing systems require then high-level cryptographic protection in order to assure that users follow the e-ticketing protocol correctly. We classify the non-smart-card-based systems depending on *non-anonymous*, *anonymous* and *revocable anonymous* compliance.

(1) Non-Anonymous proposals

Some proposals are oriented to services where anonymity can not be provided to the user, or simply, these systems are not conceived to achieve anonymity at all. Among the proposals that do not consider anonymity, digital signatures are commonly used [12], [13].

In [13], either the user or the e-ticket should be identified in order to prevent problems such as malicious attacks. There is a real relationship between anonymity and transferability for user and e-ticket identification and reusability is also considered for other ticket information, such as its destination. Online mode is used in this scheme for security reasons, as they say offline systems show weaknesses to malicious attacks.

(2) Anonymous proposals

Haneberg et al. [14] present an electronic onboard ticketing scheme, by using a PDA connected to the system through Bluetooth and using Java for all applications. PDAs are chosen for their short-range wireless communications and the display. Anonymity is achieved in this proposal as no personal data is needed, and anonymity then only depends on the payment method used.

In [15], Quercia and Hailes' e-ticketing system proposal is based on Chaum's e-cash blind signatures, providing anonymity to the user, but the communication cost could be high, and possibly slows down the system. Apart from anonymity, non-repudiation, offline verification as well as portability are achieved in this proposed system.

The great majority of the described proposals that comply with anonymity are based on Chaum's blind signatures [18].

(3) Revocable-Anonymous proposals

In the proposal of Patel and Crowcroft [11] the security requirements are defined, where revocable anonymity is achieved, as well as offline mode, although central authority intervention is needed in order to prevent overspending.

Depending on the services, anonymity, transferability or reusability would be required in the Fujimura et al. proposal [19]. Pseudonyms are proposed if anonymity is required, and overspending is controlled by a central database (online mode).

In [16], Heydt-Benjamin et al. made a proposal using latest advances in e-cash to improve privacy in electronic ticketing systems for public transit. It uses pseudonyms in order to achieve anonymity.

Chen et al. [17] propose the use of mobile devices (mobile phones, smart phones or PDAs) in e-ticketing systems, by taking advantage of their wireless communications. They focus on the compliance of several security requirements, as (revocable) anonymity, non-repudiation, as well as efficient verification. The ticket process is defined in 3 phases in the paper: request, issue and verification. Anonymity is achieved by the use of pseudonyms.

The majority of the studied proposals use pseudonyms in order to achieve revocable anonymity. If pseudonyms are used, real identity information is not put into the ticket, only its pseudonym. But if the issuer could link every pseudonym to its real identity, then anonymity could be compromised. For that reason, only revocable anonymity for the user could be achieved. In this scenario, user traceability could be easily performed if user does not change its pseudonym regularly because the same pseudonym would be used for different tickets. Accumulated data could allow all the involved participants to make user profiles if there were no pseudonym controls.

According to reusability there is a diversity of considerations; Patel and Crowcroft [11] believe that tickets can be reusable; Haneberg et al. [14], Heydt-Benjamin et

al. [16] and Chen et al. [17] do not consider ticket reusability; finally, Quercia and Hailes [15] consider that reusability mainly depends on the service.

There is a remarkable equality between the proposals that use online verification [11], [16] and the ones which use offline verification [14], [15], [17]. Otherwise, ticket transferability is unanimously not considered in the proposals [11], [14]–[17].

3. E-Ticketing Scheme

The e-ticketing scheme has been designed for mobile devices, reducing the computation requirements in the user side, and providing the basic security requirements (authenticity, non-repudiation and integrity) together with expiry date, revocable anonymity, exculpability and reusability. The ticket verification is performed offline when there is only one provider for each service, although the scheme could be extended with multiple providers that offer the same service. In that extended case, the scheme would prevent overspending through online verification, requiring then the interconnection of the service providers which offer the same service. In any case, the connection with the issuer is never necessary. In Table 1, we define the details of our proposal, as well as some notation that is used in the scheme.

3.1 Actors and Phases

The scheme has the following actors: the user $\mathcal{U}$, the ticket issuer $\mathcal{I}$, the service provider $\mathcal{P}$, and the TTP $\mathcal{T}$. The phases of our system consist of the traditional phases (ticket purchase and verification), and we add another phase in order to register and obtain temporal pseudonyms without linkage to user's identity (if user behaves correctly), in order to achieve anonymity. We can see in Fig. 1 the diagram of the entire protocol, with all its actors and phases. Then, the resulting phases are: *Pseudonym Renewal*, where the user obtains a new temporal pseudonym to be used in the system; *Ticket Purchase*, that consists on the payment of the service and reception of the ticket; and *Ticket Verification*, where the user shows the ticket to the service provider in order to be checked and validated. Other phases considered in the system are claims. These claims should only be executed in case of controversial situations during the *Ticket Verification* phase: *Claim* m_2 *Not Received* (when $\mathcal{U}$ sends the first step of the verification m_1 but does not receive m_2 by $\mathcal{P}$, or the information is not correct); *Claim* m_3 *Not Received* (when $\mathcal{P}$ sends the second step of the verification m_2 but does not receive m_3 by $\mathcal{U}$, or the information is not correct); and *Claim* m_4 *Not Received* (when $\mathcal{U}$ sends the third step of the verification m_3 but does not receive m_4 by $\mathcal{P}$, or the information is not correct). These situations will be explained in the following sections.

Users have a digital credential ($\mathsf{Cert}_{\mathcal{U}}$) for authentication to the TTP only, as the system is anonymous, and all further movements in the system are tracked only with the assigned temporal pseudonym ($\mathsf{Pseu}_{\mathcal{U}}$).

Table 1 Details of the proposal: security requirements, actors, ticket information and receipt information.

Security Requirements			
Authenticity		Non-repudiation	
Integrity		Revocable anonymity	
Non-overspending		Offline verification	
Expiry date		Exculpability	
Reusability			
Actors			
User	$\mathcal{U}$	Service Provider	$\mathcal{P}$
Ticket Issuer	$\mathcal{I}$	Trusted Third Party	$\mathcal{T}$
Ticket Information (T)			
Serial number	Sn	Issuer	Is
Service	Sv	Terms and conditions	Tc
User pseudonym	$\mathsf{Pseu}_{\mathcal{U}}$	Attributes	At
Type of ticket	Ty	Verification data	$\delta_{\mathcal{T},\mathcal{P}}$
Validity time	Tv	Date of issue	Ti
Exculpability ($\mathcal{U}$)	$\mathsf{h}_{(r_{\mathcal{U}},n)}$	Exculpability ($\mathcal{P}$)	$\mathsf{h}_{(r_{\mathcal{I}},n)}$
Digital signature of $\mathcal{I}$	$\mathsf{Sign}_{\mathcal{I}}(\mathsf{T})$		
Receipt Information (R)			
Exculpability ($\mathcal{P}$)	$\mathsf{A}_{\mathcal{P}}$	Timestamp	τ_i
Ticket serial number	T.Sn	Digital signature of $\mathcal{P}$	$\mathsf{Sign}_{\mathcal{P}}(\mathsf{R})$

3.1.1 Pseudonym Renewal

The user $\mathcal{U}$ contacts the pseudonym manager $\mathcal{T}$ in order to renew the assigned pseudonym. The certificate $\mathsf{Cert}_{\mathcal{U}}$ identifies $\mathcal{U}$ through a secure connection established between the two parties. The system has the public parameters (α, p, q), where α is a generator of the group G with order p, being p and q large primes achieving $p = 2q + 1$. $\mathcal{U}$ generates a random value $\mathsf{x}_{\mathcal{U}} \overset{R}{\leftarrow} \mathbb{Z}_q$ and computes $\mathsf{y}_{\mathcal{U}} = \alpha^{\mathsf{x}_{\mathcal{U}}} \pmod{p}$ in order to receive a valid signed pseudonym $\mathsf{Pseu}_{\mathcal{U}}$ from $\mathcal{T}$. $\mathcal{U}$ and $\mathcal{T}$ have their own pair of keys used for signature and encryption of the transmitted data between them.

authenticateUser User $\mathcal{U}$ follows the next steps:

1. generates $\mathsf{x}_{\mathcal{U}} \overset{R}{\leftarrow} \mathbb{Z}_q$, and computes $\mathsf{y}_{\mathcal{U}} = \alpha^{\mathsf{x}_{\mathcal{U}}} \pmod{p}$;

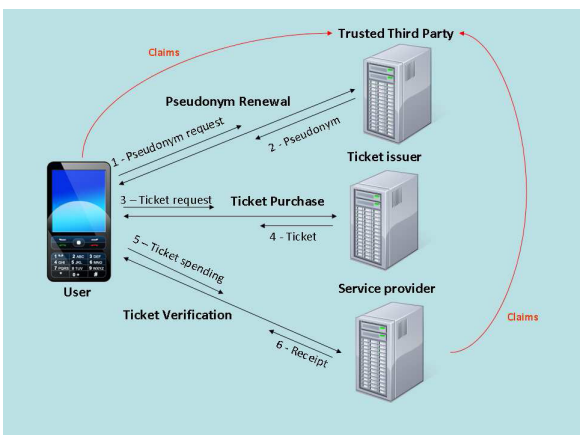

Fig. 1 Diagram of the entire protocol.

Fig. 2 Hash chain.

2. computes the signature $\mathsf{Sign}_{\mathcal{U}}(\mathsf{y}_{\mathcal{U}}) = \mathsf{sk}_{\mathcal{U}}(\mathsf{h}_{\mathsf{y}_{\mathcal{U}}})^{\dagger}$ where $\mathsf{h}_{\mathsf{y}_{\mathcal{U}}} = hash(\mathsf{y}_{\mathcal{U}})^{\dagger\dagger}$;
3. encrypts the information to be sent $(\mathsf{y}_{\mathcal{U}}, \mathsf{Sign}_{\mathcal{U}}(\mathsf{y}_{\mathcal{U}}), \mathsf{Cert}_{\mathcal{U}})$ with the $\mathcal{T}$'s public key as a digital envelope: $\mathsf{pk}_{\mathcal{T}}(\mathsf{y}_{\mathcal{U}}, \mathsf{Sign}_{\mathcal{U}}(\mathsf{y}_{\mathcal{U}}), \mathsf{Cert}_{\mathcal{U}})^{\dagger\dagger\dagger}$;
4. sends $\mathsf{pk}_{\mathcal{T}}(\mathsf{y}_{\mathcal{U}}, \mathsf{Sign}_{\mathcal{U}}(\mathsf{y}_{\mathcal{U}}), \mathsf{Cert}_{\mathcal{U}})$ to $\mathcal{T}$;

generatePseudonym Pseudonym Manager $\mathcal{T}$ executes:

1. decrypts $\mathsf{sk}_{\mathcal{T}}(\mathsf{pk}_{\mathcal{T}}(\mathsf{y}_{\mathcal{U}}, \mathsf{Sign}_{\mathcal{U}}(\mathsf{y}_{\mathcal{U}}), \mathsf{Cert}_{\mathcal{U}})) \rightarrow (\mathsf{y}_{\mathcal{U}}, \mathsf{Sign}_{\mathcal{U}}(\mathsf{y}_{\mathcal{U}}), \mathsf{Cert}_{\mathcal{U}})$;
2. verifies $\mathsf{y}_{\mathcal{U}}$: $\mathsf{pk}_{\mathcal{U}}(\mathsf{sk}_{\mathcal{U}}(\mathsf{h}_{\mathsf{y}_{\mathcal{U}}})) \rightarrow (\mathsf{h}_{\mathsf{y}_{\mathcal{U}}}) \stackrel{?}{=} hash(\mathsf{y}_{\mathcal{U}})$;
3. if correct, then computes the signature of $\mathsf{Sign}_{\mathcal{T}}(\mathsf{y}_{\mathcal{U}}) = \mathsf{sk}_{\mathcal{T}}(\mathsf{h}_{\mathsf{y}_{\mathcal{U}}})$;
4. encrypts the signature with the $\mathcal{U}$'s public key: $\mathsf{pk}_{\mathcal{U}}(\mathsf{Sign}_{\mathcal{T}}(\mathsf{y}_{\mathcal{U}}))$; and
5. sends $\mathsf{pk}_{\mathcal{U}}(\mathsf{Sign}_{\mathcal{T}}(\mathsf{y}_{\mathcal{U}}))$ to $\mathcal{U}$;

verifyPseudonym $\mathcal{U}$ computes:

1. decrypts $\mathsf{sk}_{\mathcal{U}}(\mathsf{pk}_{\mathcal{U}}(\mathsf{Sign}_{\mathcal{T}}(\mathsf{y}_{\mathcal{U}}))) \rightarrow (\mathsf{Sign}_{\mathcal{T}}(\mathsf{y}_{\mathcal{U}}))$;
2. verifies the TTP signature of $\mathsf{y}_{\mathcal{U}}$: $\mathsf{pk}_{\mathcal{T}}(\mathsf{sk}_{\mathcal{T}}(\mathsf{h}_{\mathsf{y}_{\mathcal{U}}})) \rightarrow (\mathsf{h}_{\mathsf{y}_{\mathcal{U}}}) \stackrel{?}{=} hash(\mathsf{y}_{\mathcal{U}})$.

Note that *hash*() is a public cryptographic one-way summarizing function that achieves collision-resistance. The notation $hash^n$ (*item*) is used to describe that the hash function is applied n times over the item as a chain (i.e. $hash^n$ (*item*) = $hash(\overset{n-2}{\ldots\ldots}(hash(item)))$). Moreover, the $\mathsf{h}_{(item,n)}$ is used as the value of the calculation of $hash^n$ (*item*) as depicted in Fig. 2.

3.1.2 Ticket Purchase

The user establishes a connection with the ticket issuer $\mathcal{I}$ in order to receive the ticket. This connection could be established through an anonymous channel like TOR [20], guaranteeing then user's privacy. There are current contributions[††††] that have implemented TOR for mobile devices with Android. $\mathcal{I}$ has a key pair and its public key certificate ($\mathsf{Cert}_{\mathcal{I}}$). Users do not use their personal keys (it would cause loss of anonymity); they use the temporal pseudonyms and authenticate through the Schnorr's Zero-Knowledge Proof (ZKP) [21]. The payment method is considered as out of scope in this proposal as we focus on the privacy given to user when joining/exiting the system, and using the service. $\mathcal{I}$ generates the ticket with the information and its digital signature, together with the secret value $\mathsf{r}_{\mathcal{I}}$ and the secret shared key (they are decryptable only by $\mathcal{P}$ and $\mathcal{T}$) in order to let the provider show the secret value $\mathsf{r}_{\mathcal{I}}$ later, in the verification phase. The ticket issuer $\mathcal{I}$ and the user $\mathcal{U}$ follow this protocol:

getService $\mathcal{U}$ executes:

1. selects and pays for the desired service Sv;
2. generates a random value $\mathsf{r}_{\mathcal{U}} \stackrel{R}{\leftarrow} \mathbb{Z}_q$, and computes $\mathsf{h}_{(\mathsf{r}_{\mathcal{U}},n)} = hash^n(\mathsf{r}_{\mathcal{U}})$, where n is the predefined maximum number of times that the e-ticket can be spent;
3. computes $\mathsf{H}_{\mathcal{U}} = \alpha^{r_{\mathcal{U}}} \pmod p$;
4. generates two more random values $a_1, a_2 \stackrel{R}{\leftarrow} \mathbb{Z}_q$ to be used in the Schnorr proof;
5. computes $A_1 = \alpha^{a_1} \pmod p$;
6. computes $A_2 = \alpha^{a_2} \pmod p$;
7. sends $(\mathsf{Pseu}_{\mathcal{U}}, \mathsf{H}_{\mathcal{U}}, A_1, A_2, \mathsf{h}_{(\mathsf{r}_{\mathcal{U}},n)}, \mathsf{Sv})$ to the ticket issuer $\mathcal{I}$;

getChallenge $\mathcal{I}$ follows the next steps:

1. generates and sends a challenge $\mathsf{c} \stackrel{R}{\leftarrow} \mathbb{Z}_q$ for $\mathcal{U}$;
2. asynchronously, for optimization, pre-computes $\mathsf{y}_{\mathcal{U}}{}^{\mathsf{c}} \pmod p$;
3. asynchronously, for optimization, pre-computes $\mathsf{H}_{\mathcal{U}}{}^{\mathsf{c}} \pmod p$;

solveChallenge $\mathcal{U}$ computes:

1. computes $\mathsf{w}_1 = a_1 + \mathsf{c} \cdot \mathsf{x}_{\mathcal{U}} \pmod q$;
2. computes $\mathsf{w}_2 = a_2 + \mathsf{c} \cdot \mathsf{r}_{\mathcal{U}} \pmod q$;
3. encrypts $(\mathsf{w}_1, \mathsf{w}_2)$ and sends it to $\mathcal{I}$: $\mathsf{pk}_{\mathcal{I}}((\mathsf{w}_1, \mathsf{w}_2))$;
4. pre-computes the shared session key used in the ticket verification: $\mathsf{K} = hash(\mathsf{w}_2)$;

getTicket $\mathcal{I}$ follows the next steps:

1. decrypts $\mathsf{sk}_{\mathcal{I}}(\mathsf{pk}_{\mathcal{I}}(\mathsf{w}_1, \mathsf{w}_2)) \rightarrow (\mathsf{w}_1, \mathsf{w}_2)$;
2. computes $\alpha^{\mathsf{w}_1} \pmod p$;
3. computes $\alpha^{\mathsf{w}_2} \pmod p$;
4. verifies $\alpha^{\mathsf{w}_1} \stackrel{?}{=} A_1 \cdot \mathsf{y}_{\mathcal{U}}{}^{\mathsf{c}} \pmod p$;
5. verifies $\alpha^{\mathsf{w}_2} \stackrel{?}{=} A_2 \cdot \mathsf{H}_{\mathcal{U}}{}^{\mathsf{c}} \pmod p$;
6. computes the shared session key: $\mathsf{K} = hash(\mathsf{w}_2)$;
7. obtains a unique serial number Sn, and a random value $\mathsf{r}_{\mathcal{I}} \stackrel{R}{\leftarrow} \mathbb{Z}_p$;
8. computes $\mathsf{h}_{(\mathsf{r}_{\mathcal{I}},n)} = hash^n(\mathsf{r}_{\mathcal{I}})$;
9. composes $\kappa = (\mathsf{K}, \mathsf{r}_{\mathcal{I}})$ and signs it $\kappa^* = (\kappa, \mathsf{Sign}_{\mathcal{I}}(\kappa))$;
10. encrypts κ^* with a digital envelope which is decryptable by the TTP $\mathcal{T}$ and the provider $\mathcal{P}$ for possible future controversial situations during the ticket verification: $\delta_{\mathcal{T},\mathcal{P}} = \mathsf{pk}_{\mathcal{T},\mathcal{P}}(\kappa^*)$. This is a mechanism that prevents $\mathcal{I}$ from forging $\mathsf{r}_{\mathcal{I}}$, because $\mathcal{T}$ can check that information and, demonstrate that $\mathcal{I}$ is the culprit;
11. fills out the ticket information T (Sn, Sv, $\mathsf{Pseu}_{\mathcal{U}}$, Tv, Ti, $\mathsf{h}_{(\mathsf{r}_{\mathcal{I}},n)}$, $\mathsf{h}_{(\mathsf{r}_{\mathcal{U}},n)}$, $\delta_{\mathcal{T},\mathcal{P}}$, etc.);
12. digitally signs the ticket T, and obtains the signed ticket, $\mathsf{Sign}_{\mathcal{I}}(\mathsf{T}) = \mathsf{sk}_{\mathcal{I}}(hash(\mathsf{T}))$, and $\mathsf{T}^* = (\mathsf{T}, \mathsf{Sign}_{\mathcal{I}}(\mathsf{T}))$;
13. sends T^* to the user $\mathcal{U}$;

receiveTicket $\mathcal{U}$ executes:

1. verifies the digital signature $\mathsf{Sign}_{\mathcal{I}}(\mathsf{T})$ of the ticket T using the issuer's certificate;
2. verifies that ticket T data and the performed request match;
3. verifies the ticket validity (T.Ti, T.Tv);
4. verifies $\mathsf{T.Pseu}_{\mathcal{U}}$;
5. stores $(\mathsf{T}^*, \mathsf{r}_{\mathcal{U}}, j = 0)$ in the device. We set up j to 0 because represents the times that the e-ticket has been used. The e-ticket will be totally consumed when $j = n$.

[†] Note that $\mathsf{sk}_{\mathcal{E}}(content)$ means the decryption of *content* or the generation of a signature with its content *content* by using the private key of the entity $\mathcal{E}$

[††] Note that *hash*() is a public cryptographic one-way summarizing function that achieves collision-resistance. The notation $hash(item)^n$ is used to describe that the hash function is applied n times over the *item* (i.e. $hash(item)^n = hash(\overset{n-2}{\ldots\ldots}(hash(item)))$)

[†††] Note that $\mathsf{pk}_{\mathcal{E}}(content)$ means the encryption of *content* or the verification of a signature *content* by using the public key of the entity $\mathcal{E}$

[††††] http://sourceforge.net/apps/trac/silvertunnel/wiki/TorJavaOverview

3.1.3 Ticket Verification

When the user wants to use the service, she must verify the ticket in advance. For simplicity, we present the ticket verification with only one provider, so the service provider $\mathcal{P}$ never needs permanent communication with the ticket issuer. Nonetheless, the protocol can be extended for multiple providers. In that case, all the service providers should be connected to a central repository of spent tickets in order to control ticket overspending. In these situations, when a ticket starts its verification process, the database has to lock its item (keyed with the unique serial number of the ticket) in order to allow concurrent accesses to the database for different tickets; in this case, if another user tried to verify the same ticket in another provider concurrently, it could not be possible.

The user only interacts with the service provider, but in controversial situations, she and/or the service provider could interact directly with the TTP through a resilient connection in order to preserve the security requirements of the protocol. If user misbehaved, her identity could be revoked, enabling to take further actions.

$\mathcal{U}$ sends the ticket T^*, and $\mathcal{P}$ checks it. If passed, $\mathcal{P}$ sends the commitment so that $\mathsf{r}_\mathcal{I}$ will be disclosed if $\mathcal{U}$ behaves correctly. Once the user sends the secret value $\mathsf{r}_\mathcal{U}$ encrypted through a shared key, then she receives the secret $\mathsf{r}_\mathcal{I}$ together with the receipt R^* from $\mathcal{P}$. The service provider $\mathcal{P}$ and the user $\mathcal{U}$ do the following steps:

showTicket $\mathcal{U}$ computes:

1. sends ticket $\mathsf{m}_1 = (\mathsf{T}^*, i)$ to $\mathcal{P}$. As a general case, we suppose that the service costs s of the n times that the e-ticket can be spent. So, the value i is computed as $i = j + s$;

verifyTicket $\mathcal{P}$ executes:

1. verifies the ticket signature, $\mathsf{T.Sv}$, $\mathsf{T.Ti}$, and $\mathsf{T.Tv}$;
2. if the verifications fail, $\mathcal{P}$ omits m_1, and aborts the ticket verification;
3. else $\mathcal{P}$ looks for the ticket T^* in the database using $\mathsf{T.Sn}$ and locking this item; later, it verifies that the ticket has not been spent by retreiving the information related to the ticket $(j, \mathsf{h}_{(\mathsf{r}_\mathcal{U}, n-j)})$ in the provider's database (if no information is found, then j is set to $j = 0$):
   a. if $(i > j)$ then:
      i. computes $\mathsf{A}_{\mathcal{P},i} = PRNG(\mathsf{h}_\mathsf{K}) \oplus \mathsf{h}_{(\mathsf{r}_\mathcal{I}, n-i)}$, where $PRNG(\mathsf{h}_\mathsf{K})$ is a secure pseudorandom number generator and, $\mathsf{h}_\mathsf{K} = hash(\mathsf{K})$ is the seed. Note that K and $\mathsf{r}_\mathcal{I}$ are obtained from $\delta_{\mathcal{T},\mathcal{P}}$, then the provider is able to compute $\mathsf{h}_{(\mathsf{r}_\mathcal{I}, n-i)} = hash^{(n-i)}(\mathsf{r}_\mathcal{I})$;
      ii. encrypts $\mathsf{A}_{\mathcal{P},i}$ with the public key of the TTP $\mathcal{T}$: $\mathsf{pk}_\mathcal{T}(\mathsf{A}_{\mathcal{P},i})$;
      iii. stores $\mathsf{A}_{\mathcal{P},i}$ for future use;
      iv. assigns $\mathsf{V}_\mathsf{succ} = (\mathsf{T.Sn}, \mathsf{flag}_1, \tau_1, \mathsf{pk}_\mathcal{T}(\mathsf{A}_{\mathcal{P},i}), j)$, ($\tau_1$ is the verification timestamp). The flag flag_1 indicates that the ticket is valid and has not been spent yet. The signature is noted: $\mathsf{V}_\mathsf{succ}{}^* = (\mathsf{V}_\mathsf{succ}, \mathsf{sk}_\mathcal{P}(hash(\mathsf{V}_\mathsf{succ})))$;
      v. sends $\mathsf{m}_2 = \mathsf{V}_\mathsf{succ}{}^*$ to $\mathcal{U}$;
   b. if $(i \leq j)$ then:
      i. computes $\mathsf{h}_{(\mathsf{r}_\mathcal{U}, n-i)} = hash^{(j-i)}(\mathsf{h}_{(\mathsf{r}_\mathcal{U}, n-j)})$
      ii. assigns $\mathsf{V}_\mathsf{fail} = (\mathsf{T.Sn}, \mathsf{h}_{(\mathsf{r}_\mathcal{U}, n-i)}, \mathsf{flag}_0, i, \tau_1)$. The flag_0 indicates that the ticket has been spent, i.e. is not valid. The signature is noted: $\mathsf{V}_\mathsf{fail}{}^* = (\mathsf{V}_\mathsf{fail}, \mathsf{sk}_\mathcal{P}(hash(\mathsf{V}_\mathsf{fail})))$;
      iii. sends $\mathsf{m}_2 = \mathsf{V}_\mathsf{fail}{}^*$ to $\mathcal{U}$;

showProof $\mathcal{U}$ executes:

1. verifies $\mathcal{P}$'s signature;
2. if $\mathsf{V}_\mathsf{succ}{}^*$ or either $\mathsf{V}_\mathsf{fail}{}^*$ are not received, the *Claim* m_2 *Not Received* is called;
   a. if $\mathsf{V}_\mathsf{fail}{}^*$ is received, $\mathcal{U}$ aborts the verification process. If the response is not correct, $\mathcal{U}$ can contact with the TTP to reconsider the situation by calling *Claim* m_2 *Not Received*;
   b. if $\mathsf{m}_2 = \mathsf{V}_\mathsf{succ}{}^*$ is received, $\mathcal{U}$ has to verify the signature and data. If verifications are correct she continues the protocol. Otherwise, $\mathcal{U}$ can contact the TTP by calling *Claim* m_2 *Not Received*;
3. calculates $\mathsf{A}_{\mathcal{U},i} = PRNG(\mathsf{K}) \oplus \mathsf{h}_{(\mathsf{r}_\mathcal{U}, n-i)}$, using the shared value K as seed;
4. sends $\mathsf{m}_3 = (\mathsf{T.Sn}, \mathsf{A}_{\mathcal{U},i})$ to $\mathcal{P}$;

verifyProof $\mathcal{P}$ follows the next steps:

1. if $\mathsf{h}_{(\mathsf{r}_\mathcal{U}, n-i)}$ is not received, the *Claim* m_3 *Not Received* is called;
2. obtains $\mathsf{T.Sn}$, and computes $\mathsf{h}_{(\mathsf{r}_\mathcal{U}, n-i)} = \mathsf{A}_{\mathcal{U},i} \oplus PRNG(\mathsf{K})$;
3. verifies $\mathsf{h}_{(\mathsf{r}_\mathcal{U}, n-j)} \stackrel{?}{=} hash^s(\mathsf{h}_{(\mathsf{r}_\mathcal{U}, n-i)})$;
4. if $\mathsf{h}_{(\mathsf{r}_\mathcal{U}, n-i)}$ does not match, the *Claim* m_3 *Not Received* is called;
5. generates τ_2 and verifies it using the ticket expiry date ($\mathsf{T.Ti}$, $\mathsf{T.Tv}$) and the timestamp τ_1;
6. signs $\mathsf{A}_{\mathcal{P},i}$ approving then the verification with timestamp τ_2: $\mathsf{R} = (\mathsf{A}_{\mathcal{P},i}, \mathsf{T.Sn}, \tau_2)$, and $\mathsf{R}^* = (\mathsf{R}, \mathsf{sk}_\mathcal{P}(hash(\mathsf{R})))$;
7. stores, updates its database, and unlocks the Sn from the database: $[\mathsf{R}^*, (\mathsf{h}_{(\mathsf{r}_\mathcal{U}, n-j)} \blacktriangleleft \mathsf{h}_{(\mathsf{r}_\mathcal{U}, n-i)}), (j \blacktriangleleft i)]$[†];
8. sends $\mathsf{m}_4 = \mathsf{R}^*$ to $\mathcal{U}$;

getValidationConfirmation $\mathcal{U}$ follows the next steps:

1. checks the signature of R^*;
2. computes $\mathsf{h}_{(\mathsf{r}_\mathcal{I}, n-i)} = \mathsf{A}_{\mathcal{P},i} \oplus PRNG(\mathsf{h}_\mathsf{K})$;
3. verifies $\mathsf{h}_{(\mathsf{r}_\mathcal{I}, n-j)} \stackrel{?}{=} hash^{(i-j)}(\mathsf{h}_{(\mathsf{r}_\mathcal{I}, n-i)})$;
4. if all verifications are correct, then stores and updates her database $[\mathsf{R}^*, (\mathsf{h}_{(\mathsf{r}_\mathcal{U}, n-j)} \blacktriangleleft \mathsf{h}_{(\mathsf{r}_\mathcal{U}, n-i)}), (j \blacktriangleleft i)]$; or else calls *Claim* m_4 *Not Received* to the TTP.

The *Ticket Verification* protocol is a fair exchange protocol with the existence of an offline TTP [22] between the user and the provider of the service (a valid e-ticket is given in exchange for the permission to use the service). This enables dispute resolution protocols in case of incorrect behaviour of the actors so as to preserve the security of the system. In case of dispute, they can contact the TTP following these protocols:

3.1.4 Claim m_2 Not Received

This protocol can be executed if $\mathcal{U}$ sends m_1 and says that she has not received $\mathsf{m}_2 = \mathsf{V}_\mathsf{succ}{}^*$ from $\mathcal{P}$.

Claim User $\mathcal{U}$ executes:

1. sends the ticket $\mathsf{m}_1 = (\mathsf{T}^*, i)$ to the TTP $\mathcal{T}$;

Response TTP $\mathcal{T}$ follows the next steps:

[†]The notation $a \blacktriangleleft b$ is used to describe a database update operation where the value represented by a is replaced by b.

1. checks the information, signature and timestamp;
2. if the verification is correct, generates $(\mathsf{T.Sn}, \tau_3)$; then
3. signs the information $\mathsf{m}_5 = ((\mathsf{T.Sn}, \tau_3), \mathsf{sk}_{\mathcal{T}}(hash((\mathsf{T.Sn}, \tau_3)))$; and
4. sends m_5 to both $\mathcal{U}$ and $\mathcal{P}$. This entails acceptance of $\mathcal{U}$'s sent information and then $\mathcal{P}$ has the responsibility to unblock and send a correct m_2 to continue with the verification phase at sub-phase *verifyTicket*. After that, if the service could not be finally guaranteed, $\mathcal{U}$ could demonstrate to a third party (by showing m_5) that $\mathcal{U}$ behaved correctly and $\mathcal{P}$ was the responsible of the denial of service;

verifyTicketWithTTP Service provider $\mathcal{P}$ executes:

1. executes *verifyTicket* normally;
2. sends m_2 to both $\mathcal{T}$ and $\mathcal{U}$, and continues the *Ticket Verification* steps at point *showProof*. The TTP has to store m_2 and m_5 because the user can go to an external dispute resolution system (if m_2 is still wrong) to solve the problem. In this case, the TTP will be able to provide these evidences.

3.1.5 Claim m_3 Not Received

This protocol can be executed if $\mathcal{P}$ sends m_2 and says that has not received $\mathsf{m}_3 = \mathsf{A}_{\mathcal{U},i}$ (with a correct $\mathsf{h}_{(\mathsf{r}_{\mathcal{U}},n-i)}$ inside) from $\mathcal{U}$.

Claim Provider $\mathcal{P}$ executes:

1. blocks the ticket $\mathsf{T.Sn}$ till the reception of m_3 from $\mathcal{U}$ or m_5 by $\mathcal{T}$;
2. another $\mathsf{T.Sn}'$ received from the same connection could not be accepted and $\mathsf{m}_2 = \mathsf{V}_{\mathsf{succ}}{}^*$ would be repeatedly sent in order to unblock the ticket identified by $\mathsf{T.Sn}$.

3.1.6 Claim m_4 Not Received

This protocol can be executed if $\mathcal{U}$ sends m_3 and says that has not received $\mathsf{m}_4 = \mathsf{R}^*$ (with the contained $\mathsf{h}_{(\mathsf{r}_{\mathcal{I}},n-i)}$) from $\mathcal{P}$.

Claim User $\mathcal{U}$ follows the next steps:

1. sends to the TTP $\mathcal{T}$: $(\mathsf{m}_1, \mathsf{m}_2, \mathsf{m}_3) = (\mathsf{T}^*, \mathsf{V}_{\mathsf{succ}}{}^*, (\mathsf{T.Sn}, \mathsf{A}_{\mathcal{U},i}))$;

Response TTP $\mathcal{T}$ executes:

1. checks the entire information; if verification fails, aborts the claim;
2. computes $\mathsf{A}_{\mathcal{P},i} = PRNG(\mathsf{h}_{\mathsf{K}}) \oplus \mathsf{h}_{(\mathsf{r}_{\mathcal{I}},n-i)}$ using K and $\mathsf{r}_{\mathcal{I}}$. Note that K and $\mathsf{r}_{\mathcal{I}}$ can be obtained by decrypting $\delta_{\mathcal{T},\mathcal{P}}$ and then $\mathcal{P}$ can computes $\mathsf{h}_{(\mathsf{r}_{\mathcal{I}},n-i)} = hash^{(n-i)}(\mathsf{r}_{\mathcal{I}})$;
3. checks that $\mathsf{A}_{\mathcal{P},i}$ (from the previous step) $\stackrel{?}{=} \mathsf{m}_2.\mathsf{V}_{\mathsf{succ}}.\mathsf{A}_{\mathcal{P},i}$;
4. verifies that $\mathsf{h}_{(\mathsf{r}_{\mathcal{I}},n-i)}$ (computed at step 2) matches with $\mathsf{T.h}_{(\mathsf{r}_{\mathcal{I}},n)}$;
5. checks that $\mathsf{m}_1.i > \mathsf{m}_2.\mathsf{V}_{\mathsf{succ}}.j$
6. computes $\mathsf{h}_{(\mathsf{r}_{\mathcal{U}},n-i)} = \mathsf{A}_{\mathcal{U},i} \oplus PRNG(\mathsf{K})$ and checks that $hash^i(\mathsf{h}_{(\mathsf{r}_{\mathcal{U}},n-i)}) \stackrel{?}{=} \mathsf{T.h}_{(\mathsf{r}_{\mathcal{U}},n)}$
7. if everything is successful, then generates $(\mathsf{T.Sn}, \mathsf{A}_{\mathcal{P},i}, \mathsf{A}_{\mathcal{U},i}, \tau_4)$; otherwise, it publishes which entity misbehaved in accordance with the above verifications;
8. signs the information $\mathsf{m}_6 = ((\mathsf{T.Sn}, \mathsf{A}_{\mathcal{P},i}, \mathsf{A}_{\mathcal{U},i}, \tau_4), \mathsf{sk}_{\mathcal{T}}(hash((\mathsf{T.Sn}, \mathsf{A}_{\mathcal{P},i}, \mathsf{A}_{\mathcal{U},i}, \tau_4)))$;
9. sends m_6 to $\mathcal{U}$.

Message m_6 can be used as an evidence in case of a user demand for the right to use the service in an external dispute resolution system.

3.2 Multiple Providers

The described proposal considers that only one provider is able to give a certain service, enabling then offline verification. Nevertheless, this scenario could be extended to the existence of multiple providers that give a certain service and accept the same ticket in different places, but guaranteeing the control of ticket overspending through online verification between all the providers. In this extended scenario, $\mathsf{sk}_{\mathcal{P}}$ would be shared between the providers, enabling the access to K and $\mathsf{r}_{\mathcal{I}}$. There should be also special care to the distribution and control of used tickets (controlled by the existence of $\mathsf{r}_{\mathcal{U}}$ in the database for that ticket). There should be a central database where all the providers could store all the used tickets, and then the verification would be online by imperative. In this scenario, the central server would only control the database, as the providers could be able to verify signatures and make all the cryptographic operations in order to perform all the critical real-time operations. This central database can be in the cloud; nonetheless, this can cause a delay that should be studied in detail in future work. Another option is to have all the databases actively connected one each other, and achieve Atomic Broadcast as in [23] in order to perform atomic operations to the databases (i.e. avoiding concurrent verifications using the same ticket in different providers). Expired tickets could be removed from the database for storage efficiency, and, moreover, only ticket serial number would be necessary to be stored in the database instead of storing all the ticket information.

4. Security and Privacy of the System

Proposition 1: The proposed e-ticketing system preserves authenticity, non-repudiation, integrity and the expiry date of the e-ticket.

Claim 1: It is computationally unfeasible to make a new fraudulent e-ticket.
Security Argument. A valid e-ticket has the form $\mathsf{T}^* = (\mathsf{T}, \mathsf{Sign}_{\mathcal{I}}(\mathsf{T}))$. Then, the first step that the provider $\mathcal{P}$ does when an e-ticket is received is the verification of the signature. The *Ticket Verification* protocol will continue only if this verification ends correctly; otherwise, $\mathcal{P}$ refuses $\mathcal{U}$'s request. Thus, making a new fraudulent valid e-ticket would be equivalent to break the signature scheme and that would be computationally unfeasible as we have supposed that the issuer $\mathcal{I}$ uses a secure signature scheme.

Claim 2: The issuer can not deny the emission of a valid e-ticket.
Security Argument. A valid e-ticket has $\mathcal{I}$'s signature and the signature scheme used is secure. Consequently, the identity of the issuer is associated to the ticket; the signature is a non-repudiation evidence of origin.

Claim 3: The content of the e-ticket can not be modified.

Security Argument. Suppose that someone modifies the content of the ticket, then a new $\mathcal{I}$'s signature has to be generated over the modified content; otherwise, the e-ticket will not be valid. Again, if it is computationally unfeasible to forge the $\mathcal{I}$'s signature, it is unfeasible to modify the content of the e-ticket.

Claim 4: The e-ticket will not be longer valid after the ticket validity time $\mathsf{T.Tv}$.
Security Argument. The provider $\mathcal{P}$ receives the e-ticket from the user at the *Ticket Verification* protocol before allowing access to the service. First of all $\mathcal{P}$ checks the correctness of the e-ticket (obviously that includes the verification of $\mathsf{T.Tv}$). If the verification is not correct $\mathcal{P}$ stops the protocol and the user will have no access to the service. Also, according the *Claim 3*, the user can not tamper $\mathsf{T.Tv}$.

Result 1: According to the definitions given at Sec. 3 and the Claims 1, 2, 3 and 4 we can assure that the protocol achieves the properties specified at Proposition 1.

Proposition 2: The e-ticketing system described in Sect. 3 is anonymous. The service offered is revocable anonymous.

Claim 5: An e-ticket is anonymous.
Security Argument. A valid e-ticket has the following information $\mathsf{T} = (\mathsf{Sn}, \mathsf{Sv}, \mathsf{Pseu}_{\mathcal{U}}, \mathsf{Tv}, \mathsf{Ti}, \mathsf{h}_{(\mathsf{r}_{\mathcal{I}},n)}, \mathsf{h}_{(\mathsf{r}_{\mathcal{U}},n)}, \delta_{\mathcal{T},\mathcal{P}}, \ldots)$. The information related to the user's identity is solely $\mathsf{Pseu}_{\mathcal{U}} = (\mathsf{y}_{\mathcal{U}}, \mathsf{sk}_{\mathcal{T}}(hash(\mathsf{y}_{\mathcal{U}})))$, where $\mathsf{y}_{\mathcal{U}} = \alpha^{\mathsf{x}_{\mathcal{U}}} \pmod{p}$. The user's identity is $\mathsf{x}_{\mathcal{U}}$, thus an enemy has to solve the problem of computing the discrete logarithm to know the identity of the user. Currently no efficient algorithms are known to compute that.

Claim 6: The purchase of an e-ticket is anonymous.
Security Argument. As the protocol in Sect. 3.1.2 specifies, the channel between $\mathcal{U}$ and $\mathcal{I}$ of the ticket is anonymous. The protocol uses a Schnorr's ZKP to provide the user identity to the $\mathcal{I}$, so that the issuer can be sure that the connected user who wants to buy the ticket is the right holder of the pseudonym $\mathsf{Pseu}_{\mathcal{U}}$ without disclosing her real identity. Thus, the user does not need to reveal her identity to buy an e-ticket.

Claim 7: A fake user cannot buy an e-ticket impersonating other user.
Security Argument. In order to buy a ticket, the user has to perform a Schnorr's ZKP to prove knowledge of the identity to the issuer without revealing it. The user has to compute w_1 such as $\alpha^{\mathsf{w}_1} \stackrel{?}{=} A_1 \cdot \mathsf{y}_{\mathcal{U}}{}^{\mathsf{c}} \pmod{p}$. As far as any user preserves the privacy of her identity $\mathsf{x}_{\mathcal{U}}$ (which links to $\mathsf{Cert}_{\mathcal{U}}$ through the cooperation of $\mathcal{T}$), anyone else will not be able to compute such w_1. In this case, user can only be accused through $\mathsf{x}_{\mathcal{U}}$ of ticket overspending (supposing that the user keeps $\mathsf{x}_{\mathcal{U}}$ secretly), because she solely has the information to perform the *Ticket Verification* protocol. Thus, the e-ticketing system also preserves the exculpability property.

Result 2: According to the definitions given at Sect. 3 and the Claims 5, 6 and 7 we can assert the Proposition 2. The e-ticketing system is anonymous and this anonymity could be revocable in case of a user's fraudulent action. The pseudonym manager $\mathcal{T}$ knows the correspondence between $\mathsf{x}_{\mathcal{U}}$ and $\mathsf{y}_{\mathcal{U}}$ (see Algorithm: 'Pseudonym Renewal'). Therefore, $\mathcal{T}$ could reveal the association between $\mathsf{x}_{\mathcal{U}}$ and $\mathsf{y}_{\mathcal{U}}$ due to law enforcement (e.g. a judge could request the user's identity to $\mathcal{T}$).

Proposition 3: The protocol satisfies the property of exculpability and a malicious service provider cannot reduce the times that a reusable ticket can be used.

Claim 8: The user $\mathcal{U}$ is able to prove that she has already validated the ticket.
Security Argument. If a user $\mathcal{U}$ executes successfully the *Ticket Verification* protocol, $\mathcal{U}$ will obtain the exculpability proof $\mathsf{r}_{\mathcal{I}}$. She can use this proof to demonstrate that the ticket has been validated. If the *Ticket Verification* protocol is stopped and $\mathcal{U}$ does not obtain the exculpability proof after the revelation of $\mathsf{r}_{\mathcal{U}}$, she can execute *Claim* m_4 *Not Received*. This way $\mathcal{U}$ would obtain an alternative exculpability proof from the TTP.

Claim 9: The service provider can not falsely accuse the user of ticket overspending.
Security Argument. When the service provider $\mathcal{P}$ receives the message m_1 in step 1 of the *Ticket Verification* protocol (*showTicket*), $\mathcal{P}$ looks for the ticket that matches with the received serial number in its database. If the ticket had been already spent, the service provider will find the overspending proof $\mathsf{r}_{\mathcal{U}}$ together with the ticket. The service provider has to show this element to accuse the user of overspending. If the user had not validated the ticket before, then the service provider does not have the element ($\mathcal{U}$ will send it in step 3: *showProof*), as the inversion of the *hash* function is believed to be computationally infeasible, and also there can not exist collisions in this *hash* function, so $\mathcal{P}$ can not falsely accuse the user of overspending. If the service provider, even not being able to prove the overspending, decides to deny the service to the user, the user can contact the TTP in order to solve the situation through *Claim* m_2 *Not Received*.

Claim 10: The provider $\mathcal{P}$ cannot falsely accuse the user of spending a coupon $\mathsf{h}_{(\mathsf{r}_{\mathcal{U}},n-i)}$, which has not already been used.
Security Argument. The provider $\mathcal{P}$ cannot deduce any $\mathsf{h}_{(\mathsf{r}_{\mathcal{U}},n-k)} \forall k < j$. When the user $\mathcal{U}$ spends the ith coupon of her ticket, she sends $\mathsf{h}_{(\mathsf{r}_{\mathcal{U}},n-i)}$ to the provider $\mathcal{P}$ (see step *shoowProof* of the *Ticket Verification* protocol). Then the provider stores this value in order to avoid overspending. According to the hash functions properties and, as it shows in Figs. 3 and 4, from $\mathsf{h}_{(\mathsf{r}_{\mathcal{U}},n-i)}$ it is only possible to deduct an $\mathsf{h}_{(\mathsf{r}_{\mathcal{U}},n-k)} \forall k > j$. Because it is not possible to go in the direction of $\mathsf{r}_{\mathcal{U}}$. So, the provider is not able to deduce any non-spent value of the hash function chain.

Result 3: According to the definitions given at Sect. 3 and the Claims 8, 9 and 10 we can assure that the protocol

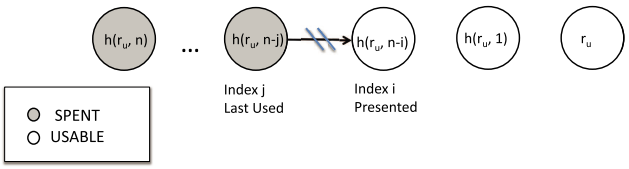

Fig. 3 Correct use.

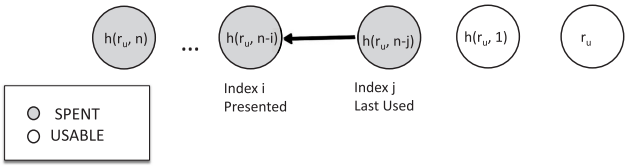

Fig. 4 Reutilization.

achieves the property specified at the Proposition 3. The ticket verification process is a fair exchange: any part can obtain the exculpability proof of the other part without revealing its own proof.

Proposition 4: The tickets issued by the protocol described in Sect. 3.1 can be preset to be reusable tickets, both for a limited number of verifications or a limited period of time.

Claim 11: The protocol allows the creation of N-usable tickets maintaining the security properties of the non reusable tickets, including exculpability.

Security Argument. During the execution of the *Ticket verification* protocol, $\mathcal{U}$ uses the last element of the chain of proofs $\mathsf{h}_{(r_{\mathcal{U}},n)}$ and receives in exchange an element containing the last element of the chain of issuer proofs $\mathsf{h}_{(r_{\mathcal{I}},n)}$. Due to the properties of hash functions, $\mathcal{U}$ cannot generate $\mathsf{h}_{(r_{\mathcal{I}},n-i)}$ and $\mathcal{P}$ cannot generate $\mathsf{h}_{(r_{\mathcal{U}},n-i)}$. The successive verifications will use the remaining elements of the chain in reverse order.

Claim 12: The protocol allows the creation of period-usable tickets maintaining the security properties of the non reusable tickets, including exculpability.

Security Argument. In this case the concept of overspending is not applicable. The user will obtain a verification proof each time he executes the *Ticket Verification* protocol obtaining an exculpability proof provided the time of the verification attempt is less than the limit of the validity period.

Result 4: According to the Claims 11 and 12 we can assert the Proposition 4. The protocol is flexible enough to be used with all kinds of services, with independence of its reusability requirements.

Proposition 5: The protocol avoids overspending with minimum requirements of persistent connections with a centralized system.

Claim 13: The protocol avoids overspending.
Security Argument. If a user tries to overspend a ticket, she will send to $\mathcal{P}$ a spent hash value (a $\mathsf{h}_{(r_{\mathcal{U}},n-i)}$ with $i \leq j$). The provider will verify that $\mathsf{h}_{(r_{\mathcal{U}},n-j)} \stackrel{?}{=} hash^s(\mathsf{h}_{(r_{\mathcal{U}},n-i)})$. This verification will always fail, because the hash chain goes in the opposite direction (see Figs. 3 and 4). Then the provider will block the ticket identified by $\mathsf{T.Sn}$ till the reception of a non-spent hash value.

Claim 14: If the ticket can only be validated by one provider then the verification is totally offline.
Security Argument. The provider $\mathcal{P}$ maintains a database with the serial numbers of the e-tickets that have been already validated (together with their exculpability proofs) until their expiry date. With the contents of this database the provider has enough information to decide if $\mathcal{P}$ accepts and validates a new ticket because $\mathcal{P}$ can check both the issuer's signature and the fact that the e-ticket has not been spent before. So the provider does not need to contact to any party during the verification of an e-ticket.

Claim 15: If the ticket can be validated with several providers then the providers must be connected and share a database of spent tickets.
Security Argument. The set of providers maintain a shared database with the serial numbers of the e-tickets that have been already validated (together with their exculpability proofs) until their expiry date. The contents of this database are used by the providers to decide if they accept and validate a new ticket. So the provider does not need connection to the issuer during the verification of an e-ticket, but the set of providers must have a shared database instead.

Result 5: According to the definitions given at Sect. 3 and the Claims 13, 14 and 15 we can assure that the protocol achieves Proposition 5. The issuer is offline during the verification phase and the providers contact among them only in some kind of services. In all cases, the protocol prevents ticket overspending.

5. Implementation Details and Experimental Results

There are several important factors to consider when we design an e-ticketing system that should be usable in practice. The response time is one of them. Thus, we have implemented our protocol and we present some results regarding its time performance.

In Sect. 5.1, we describe the developed components, the development environment and the hardware that we have used. Next, the testing methodology is described in Sect. 5.2, i.e. the system can be configured using different key lengths. We can assume that we would obtain more security with larger keys but the computational cost will be higher. We want to study how the key length influence the computational cost. Next, we present the obtained results differentiating the costs in the user side (Sect. 5.3) and the server side (Sect. 5.4).

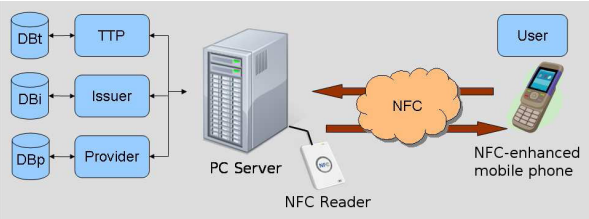

Fig. 5 Architecture of the testing environment.

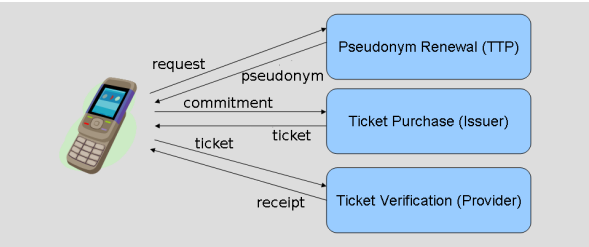

Fig. 6 Protocols of the client component.

5.1 E-ticketing System Development and Configuration Details

As introduced in Sect. 3, our system comprises three main phases: pseudonym renewal, ticket purchase and ticket verification; and there are four actors: the user, the service provider, the ticket issuer and the pseudonym manager. So that, the system implementation requires four components: one for each entity in the system. Nonetheless, we have grouped in one server the service provider, the ticket issuer and the pseudonym manager for practical reasons, see Fig. 5. The server takes the role of different servers in a PC. The server component has been developed with the Java programming language (Java 2 Standard Edition), allowing portability in a great number of platforms.

The user interacts with the other actors (the service provider, the ticket issuer and the pseudonym manager) by means of a mobile phone, so that the user component (client) should be executed in a mobile phone. The Fig. 6 shows the protocols implemented in the client component. Given that a great number of mobile phones can execute Java applications, we have developed the client in Java 2 Micro Edition (J2ME). The mobile phone used has been a Nokia 6212 Classic with an embedded API for NFC communication.

The communication between server and client is performed via Near Field Communication (NFCIP-1, ISO18092). Nowadays, there are few mobile phones with NFC technology. Nonetheless, in a near future the main smart-phones platforms will include the NFC technology. Nokia has announced that for this year (2011) every Nokia smartphone will have NFC[†]. Android Gingerbread 2.3 will support near field communications to read RFID tags as well as communicate with other phones, payment systems and other possible applications[††], and Apple is testing an iPhone with NFC chips[†††]. The server uses an Arygon NFC Reader (ADRA-USB) in order to connect with the mobile phone. The equipment of the entire scheme is detailed in Table 2.

Table 2 Equipment specification details.

Computer (server)	CPU	AMD Athlon 64 X2 Dual 5000+ (2.59 GHz)
	RAM	2 GB
	OS	Windows XP
	Java version	Java 6
	NFC reader	Arygon ADRA-USB
Mobile phone (client)	Model	Nokia 6212 NFC classic
	Java version	J2ME (Series 40 SDK 1.0 with JSR 257 extension)

It should be taken into consideration that the mobile phone acts as the initiator of the transactions, and the server is the target, i.e. the server is waiting for $\mathcal{U}$'s requests. Finally, we have used the BouncyCastle crypto library[††††] for all the cryptographic operations in both J2SE and J2ME.

5.2 Testing Methodology

The e-ticketing system can be configured with the key length parameter l. This parameter l refers to the key size (in bits) of the RSA cryptosystem used in the protocol, as well as the number of bits of the generated prime numbers for the generation of the $\mathbb{Z}_q$ and $\mathbb{Z}_p$. The larger the parameter is, the harder is the cryptosystem to be broken, i.e. we have a system more secure. On the other hand, the time consumption is also increased, and has to be evaluated.

We have run the protocols with different key sizes of 512, 1024 and 2048 bits, respectively. The results are studied in Sects. 5.3 and 5.4 evaluating the costs in the user side and the server side, respectively. Regarding the length of the keys l, at the present time a size of $l = 1024$ bits is considered computationally safe [24]. According to that, we have tested our scheme with a smaller length ($l = 512$ bits) and a larger one ($l = 2048$ bits). In this way, we can examine how the key length influences the system performance.

We have executed several test for every key length and protocol, so that the times shown in the following sections are the average of these times.

5.3 Cost Results in the Client (User) Side

We have studied the global times of the protocol in 5.3.1 as well as the partial times of each protocol (pseudonym renewal, ticket purchase and ticket verification, see Sects. 5.3.2, 5.3.3 and 5.3.4 respectively), in order to identify the most costly parts of each protocol in the client (user) side.

[†]http://www.nearfieldcommunicationsworld.com/2010/06/17/33966/all-new-nokia-smartphones-to-come-with-nfc-from-2011/

[††]http://thenextweb.com/google/2010/11/15/googles-schmidt-android-gingerbread-to-have-near-field-communication-support/

[†††]http://blogs.computerworld.com/16749/new_apple_hire_shows_iphone_iwallet_future

[††††]http://www.bouncycastle.org/

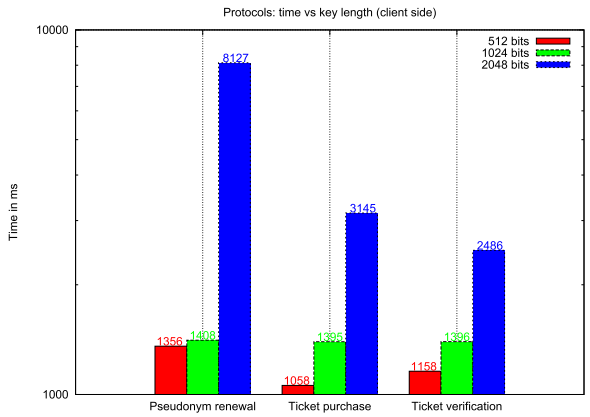

Fig. 7 Computational cost of every protocol using several key lengths in the client side.

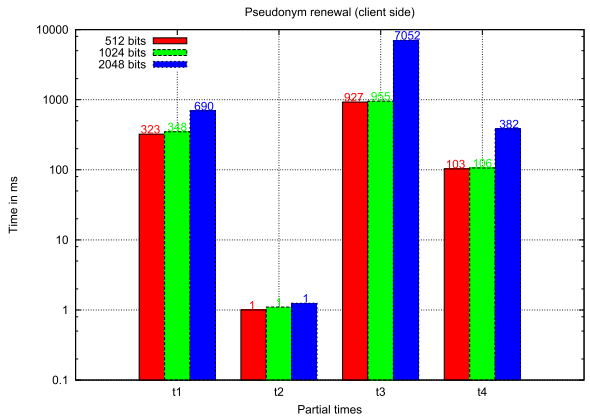

Fig. 8 Partial times of the pseudonym renewal. (see Table 3 for t_i details)

Table 3 Details of the pseudonym renewal partial times.

Partial time	Description
t_1	Sending of the pseudonym request
t_2	Reception of the pseudonym response
t_3	Decryption of the signed pseudonym
t_4	Pseudonym's signature verification

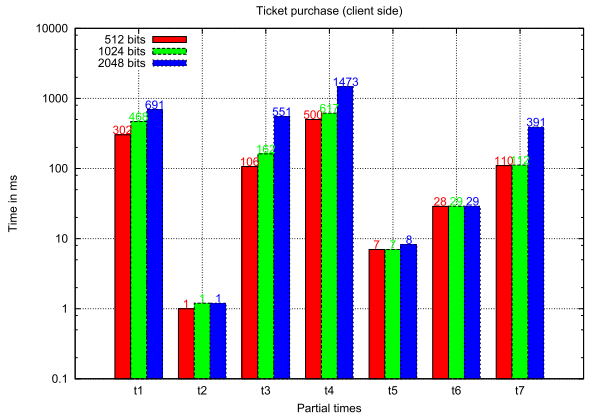

Fig. 9 Partial times of the ticket purchase. (see Table 4 for t_i details)

Table 4 Details of the ticket purchase partial times.

Partial time	Description
t_1	Sending of the commitment
t_2	Reception of the challenge
t_3	Computation of the Schnorr's ZKP response
t_4	Sending of the Schnorr's ZKP response
t_5	Computation of the shared symmetric key
t_6	Reception of the ticket
t_7	Verification of the ticket data

5.3.1 Global Time Cost Results

Figure 7 shows the average time (in ms) required to complete each transaction (*Pseudonym Renewal*, *Ticket Purchase* and *Ticket Verification* phases) taking into account the interaction with the other entities. These results are given depending on the used key length l (in bits) with its values 512, 1024 and 2048, respectively. We focus specially on the *Ticket Verification* phase, where the delay time has to be strongly reduced if we assume a mass-transit scenario. This delay varies from 1.1 to 2.5 seconds depending on the key length l parameter, what makes the proposal definitely practical. In general, all the transactions are considered practical in 1024 bits (cost lower than 2s), and they become increased in 2048 bits. We detail the times of each phase by considering the costs of all their subphases in Sects. 5.3.2, 5.3.3 and 5.3.4 in order to show the most costly operations in terms of delay times.

5.3.2 Detailed Time Cost of the Pseudonym Renewal

Figure 8 shows the partial time intervals of the *Pseudonym Renewal* phase. As expected, the decryption of the signed pseudonym (t_3) is the most costly operation in this phase, and increases obviously depending on the key length l parameter. This operation is performed by the user in the mobile phone. There are not great remarks in the other operations, as precomputation of the non-interactive values helps to reduce the global transaction times.

We have obtained similar results for 512 and 1024, where the variation between them is few milliseconds; when we use the 2048-bit key, the computational times are higher as we expected. This is due to the actual computational power, i.e. the times required to compute modular exponentiations with 512 and 1024 bits are quite low and they are practically the same. In this case (512 and 1024), the communication costs can have more influence in the final time than the computational cost.

5.3.3 Detailed Time Cost of the Ticket Purchase

Figure 9 shows the partial time intervals of the *Ticket Purchase* phase. The main costs remain on the computation and transmission of the Schnorr's ZKP (t_3 and t_4), as well as the communication cost of the first commitment (t_1), specially all of them with 2048 bits. The verification of the ticket signature (t_7) varies from 100 to 400 ms. Once again, some values have been precomputed to reduce the time of the pro-

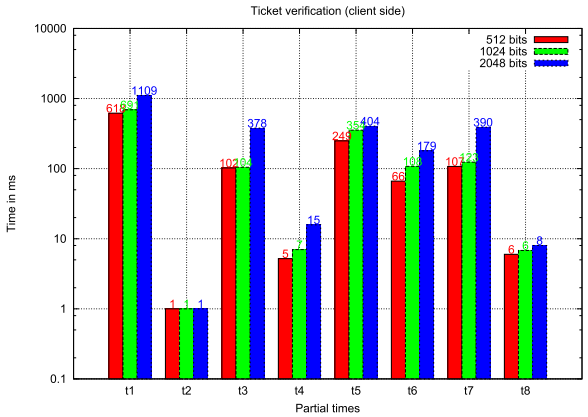

Fig. 10 Partial times of the ticket verification. (see Table 5 for t_i details)

Table 5 Details of the ticket verification partial times.

Partial time	Description
t_1	Sending of the ticket
t_2	Reception of the response
t_3	Verification of the response
t_4	Computation of the symmetric encryption of $\mathsf{r}_{\mathcal{U}}$
t_5	Sending of the symmetric encryption of $\mathsf{r}_{\mathcal{U}}$
t_6	Reception of the receipt
t_7	Verification of the receipt data
t_8	Computation and verification of the symmetric decryption of $\mathsf{r}_{\mathcal{I}}$

tocol execution.

5.3.4 Detailed Time Cost of the Ticket Verification

Figure 10 shows the partial time intervals of the *Ticket Verification* phase. The most remarkable costs remain on the connection and sending of the ticket (t_1), depending on the amount of data with its key length (parameters and signature), followed by the signature verification of the response (t_3), the sending of the symmetric encryption of the parameter $\mathsf{r}_{\mathcal{U}}$ (t_5), and finally the verification of the receipt (t_7). Other times such as the reception of the response (t_2), the computation of the symmetric encryption of $\mathsf{r}_{\mathcal{U}}$ (t_4), the reception of the receipt (t_6) and the computation of the symmetric decryption of $\mathsf{r}_{\mathcal{I}}$ (t_8) have become not costly operations.

Independently from the key length l parameter, there is a variation in the communication times depending on the steps of the protocol, as the server and the client have to synchronize their protocol steps in order to exchange their information.

5.4 Cost Results in the Server Side

We have studied the global times of the protocol in 5.4.1 as well as the partial times of each protocol (*Pseudonym Renewal*, *Ticket Purchase* and *Ticket Verification* phases, see Sects. 5.4.2, 5.4.3 and 5.4.4 respectively), in order to identify the most costly parts of each protocol in the server side.

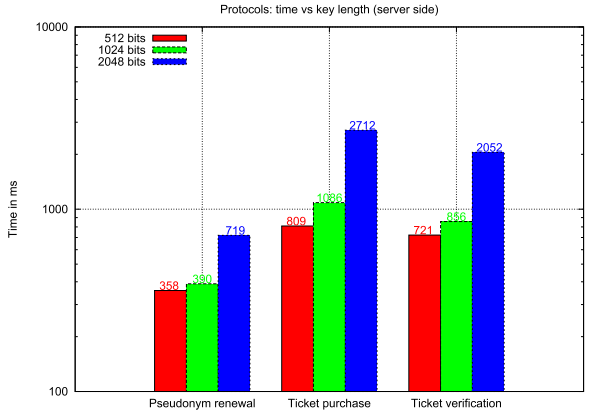

Fig. 11 Computational cost of every protocol using several key lengths in the server side.

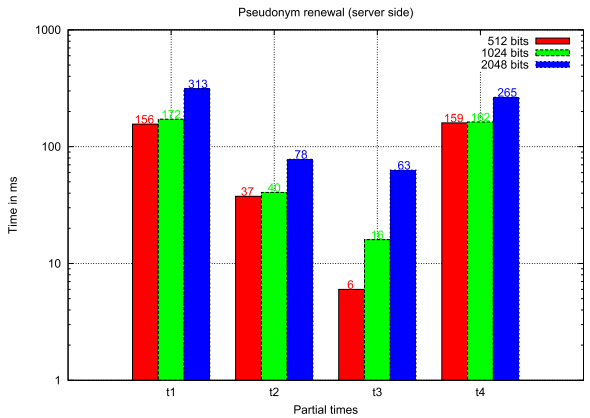

Fig. 12 Partial times of the pseudonym renewal. (see Table 6 for t_i details)

5.4.1 Global Time Cost Results

Figure 11 shows the average time (in ms) required to complete each transaction (*Pseudonym Renewal*, *Ticket Purchase* and *Ticket Verification* phases) taking into account the interaction with the user by each entity (Trusted Third Party, Issuer and Service Provider). These results are given depending on the used key length l (in bits) with its values 512, 1024 and 2048, respectively. We focus specially on the *Ticket Verification* phase, where the delay time has to be strongly reduced if we assume a mass-transit scenario. This delay varies from 0.7 to 2 seconds depending on the key length l parameter, what makes the proposal definitely practical, specially until 1024 bits. We detail the times of each phase by considering the costs of all their subphases in Sects. 5.4.2, 5.4.3 and 5.4.4 in order to show the most costly operations in terms of delay times.

5.4.2 Detailed Time Cost of the Pseudonym Renewal

Figure 12 shows the partial time intervals of the *Pseudonym Renewal* phase. In this part, the communication with the

Table 6 Details of the pseudonym renewal partial times.

Partial time	Description
t_1	Reception of the pseudonym request
t_2	Pseudonym extraction
t_3	Verification & signature of the pseudonym
t_4	Sending the signed pseudonym

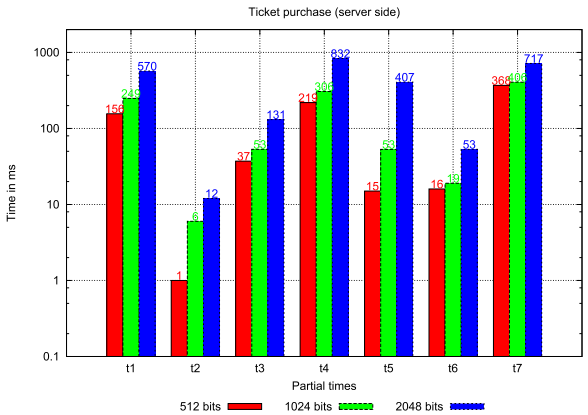

Fig. 13 Partial times of the ticket purchase. (see Table 7 for t_i details)

Table 7 Details of the ticket purchase partial times.

Partial time	Description
t_1	Reception of the commitment
t_2	Signature verification
t_3	Computation & sending of the challenge
t_4	Reception of the ZKP response
t_5	Verification of the ZKP response
t_6	Generation of the ticket
t_7	Sending of the ticket

client (reception of the request and sending of the signed pseudonym) is the major part of the protocol.

5.4.3 Detailed Time Cost of the Ticket Purchase

Figure 13 shows the partial time intervals of the *Ticket Purchase* phase. The main costs are, another time, for communication (and synchronization) with the client (t_1, t_4, t_7), and only the verification of the Zero-Knowledge Proof is the most costly computation part (mainly for 2048 bits).

5.4.4 Detailed Time Cost of the Ticket Verification

Figure 14 shows the partial time intervals of the *Ticket Verification* phase. The main costs are related to communication also (t_1 for sending of the ticket, t_5 for sending the symmetric encryption of $r_{\mathcal{U}}$), but there are also some computation costs to be taken into account (t_3 verification of the response, and t_7 verification of the receipt data), specially at 2048 bits.

5.5 Database Size and Other System Requirements

We show in Table 9 the size of every register in the database. This size depends on the key length parameter l: 512, 1024 or 2048 bits. In the table we detail also which are the attributes which are stored into the database, and also which are their partial sizes.

We analyse the capacity requirements of this protocol for a real mass-transport system. The Tokyo Subway is the metro system which has most annual passenger rides, where in 2009 registered 3160 M rides, what makes an average of 8.7 M daily rides[†]. If we take our 1024-bit results (1533 B per register), it would require a new daily capacity of 12.43 GiB. According to what we suggested for the maintenance of the database, the tickets have a validity time, and after that period they could be removed. If we set, for example, a 60-day validity period, it would require a capacity of 750 GiB, which we think it could be practical in this kind of mass-transport system.

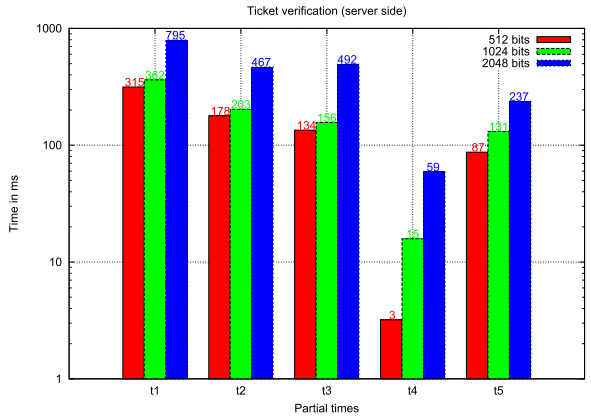

Fig. 14 Partial times of the ticket verification. (see Table 8 for t_i details)

Table 8 Details of the ticket verification partial times.

Partial time	Description
t_1	Reception of the ticket
t_2	Generation of the response
t_3	Reception of the symmetric encryption of $r_{\mathcal{U}}$
t_4	Generation of the receipt
t_5	Sending of the receipt

Table 9 Details of the database register sizes (in bytes) depending on the key length parameter l.

Parameter	Key size		
	512 bits	1024 bits	2048 bits
T^*	870B	995 B	1765 B
R^*	218 B	281 B	538 B
$r_{\mathcal{I}}$	64 B	128 B	256 B
$r_{\mathcal{U}}$ or $h_{(r_{\mathcal{U}},use)}$	64 B	128 B	256 B
use	1 B	1 B	1 B
TOTAL	1217 B	1533 B	2816 B

6. Conclusions

We have proposed an e-ticketing scheme with revocable anonymity and exculpability as security requirements, and

[†]http://geography.about.com/od/urbaneconomicgeography/a/Busiest-Subways.htm

also reusability, in order to allow multi-verifiable tickets. Moreover, we have used personal mobile devices in the system, lightening then the computational requirements on the users' side. We assume, for simplicity, that only one provider is able to give a certain service; then, the ticket verification phase is totally offline from the ticket issuer.

An analysis of the security and privacy of the proposal has been also performed, breaking down all the security requirements and evaluating the achievements.

The proposed system has been implemented and we give the details and the experimental results in order to verify that has a reasonable response time, i.e. it is usable in practice. We have analyzed the global time results for all the phases, and also the partial time results for each phase of the protocol. The results obtained, specifically the verification time (1.4 seconds using a 1024-bit key length in the user side, less than 1 second in the server side), allow to use the system in practice for mass-transit systems.

Finally, as a future work, we want to extend this e-ticketing NFC system (for one provider giving one determined service) by analysing its performance in a scenario with multiple service providers giving the same service, evaluating delay differences between a central server (cloud, remote desktop) and an online scenario connecting all the databases of the service providers. This last scenario would use Atomic Broadcast in order to make atomic operations to the databases as in [23]. Moreover, our intention is to continue improving the delay response times and also to include new security requirements such as transferability.

Disclaimer and Acknowledgements

Partial support by the Spanish MICINN (projects TSI2007-65406-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004, TSI2007-62986, "RIPUP" TIN2009-11689 and AUDIT TRANSPARENCY IPT-430000-2010-0031), the Spanish Ministry of Industry, Commerce and Tourism (projects eVerification TSI-020100-2009-720 and SECloud TSI-020302-2010-153), and the Government of Catalonia (grant 2009 SGR1135) is acknowledged. The authors are solely responsible for the views expressed in this paper, which do not necessarily reflect the position of UNESCO nor commit that organization.

References

[1] A. Vives-Guasch, M. Payeras-Capella, M. Mut-Puigserver, and J. Castellà-Roca, "E-ticketing scheme for mobile devices with exculpability," Data Privacy Management (DPM), Fifth International Workshop, Lect. Notes Comput. Sci., vol.6514, pp.79–92, 2010. ISSN 0302-9743.

[2] W. Siu and Z. Guo, "Application of electronic ticket to online trading with smart card technology," Proceedings of the 6th INFORMS Conference on Information Systems and Technology (CIST-2001), pp.222–239, Miami Beach, Florida (US), 2001.

[3] J.L. Ferrer-Gomilla, J.A. Onieva, M. Payeras, and J. Lopez, "Certified electronic mail: Properties revisited," Computers and Security, vol.29, no.2, pp.167–179, ISSN 0167-4048, 2010.

[4] J. Elliot, "The one-card trick multi-application smart card e-commerce prototypes," Computing & Control Engineering Journal, vol.10, no.3, pp.121–128, 1999.

[5] D. Haneberg, "Electronic ticketing: risks in e-commerce applications," Digital excellence, pp.55–66, 2008. ISBN 3540726209.

[6] K. Kuramitsu, T. Murakami, H. Matsuda, and K. Sakamura, "Ttp: Secure acid transfer protocol for electronic ticket between personal tamper-proof devices," 24th Annual International Computer Software and Applications Conference (COMPSAC2000), vol.24, pp.87–92, Taipei, Taiwan, Oct. 2000.

[7] K. Kuramitsu and K. Sakamura, "Electronic tickets on contactless smartcard database," Proceedings of the 13th International Conference on Database and Expert Systems Applications, pp.392–402, LNCS 2453, 2002.

[8] S. Matsuo and W. Ogata, "Electronic ticket scheme for its," IEICE Trans. Fundamentals, vol.E86-A, no.1, pp.142–150, Jan. 2003.

[9] I. Siu and Z. Guo, "The secure communication protocol for electronic ticket management system," 8th Asia-Pacific Software Engineering Conference (APSEC2001), University of Macau, 2001.

[10] G. Koning Gans, J.H. Hoepman, and F.D. Garcia, "A practical attack on the mifare classic," CARDIS '08: Proc. 8th IFIP WG 8.8/11.2 international conference on Smart Card Research and Advanced Applications, pp.267–282, Berlin, Heidelberg, Springer-Verlag, 2008.

[11] B. Patel and J. Crowcroft, "Ticket based service access for the mobile user," Proc. 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'97), pp.223–233, Budapest, Hungary, 1997.

[12] K. Fujimura, H. Kuno, M. Terada, K. Matsuyama, Y. Mizuno, and J. Sekine, "Digital-ticket-controlled digital ticket circulation," 8th USENIX Security Symposium, pp.229–240, 1999.

[13] F. Bao, "A scheme of digital ticket for personal trusted device," 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC04), vol.4, pp.3065–3069, 2004.

[14] D. Haneberg, K. Stenzel, and W. Reif, "Electronic-onboard-ticketing: Software challenges of an state-of-the-art m-commerce application," Workshop Mobile Commerce, ed. K. Pousttchi and K. Turowski, Lecture Notes in Informatics (LNI), vol.42, pp.103–113, Gesellschaft für Informatik (GI), 2004.

[15] D. Quercia and S. Hailes, "Motet: Mobile transactions using electronic tickets," 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks, Proceedings, Athens, Greece, vol.24, pp.374–383, Sept. 2005.

[16] T.S. Heydt-Benjamin, H.J. Chae, B. Defend, and K. Fu, "Privacy for public transportation," 6th Workshop on Privacy Enhancing Technologies (PET 2006), pp.1–19, LNCS 4258, 2006.

[17] Y.Y. Chen, C.L. Chen, and J.K. Jan, "A mobile ticket system based on personal trusted device," Wireless Personal Communications: An International Journal, vol.40, no.4, pp.569–578, 2007.

[18] D. Chaum, "Blind signatures for untraceable payments," Advances in Cryptology - CRYPTO'82, pp.199–203, 1983.

[19] K. Fujimura and Y. Nakajima, "General-purpose digital ticket framework," 3rd USENIX Workshop on Electronic Commerce, pp.177–186, 1998.

[20] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," Proceedings of the 13th USENIX Security Symposium, 2004.

[21] C.P. Schnorr, "Efficient signature generation by smart cards," J. Cryptology, vol.4, no.3, pp.161–174, 1991.

[22] S. Kremer, O. Markowitch, and J. Zhou, "An intensive survey of fair non-repudiation protocols," Computer Communications, vol.25, pp.1606–1621, 2002.

[23] F. Pedone, "A two-phase highly-available protocol for online validation of e-tickets," Hewlett-Packard Labs Technical Reports, 2000. HPL-2000-116 20000929.

[24] N.I. of Standards and Technology, "Special publication 800-57 part 1," 2007. Recommendation for Key Management.

Arnau Vives-Guasch (Valls, 1983) is a Ph.D. student at the CRISES Research Group (UNESCO Chair in Data Privacy) of the Universitat Rovira i Virgili. He achieved his title of Engineer in Computer Science from Universitat Rovira i Virgili in 2006, and his Master degree in Computer Engineering and Security in 2008 from the same University. His research motivation focuses on the fields of applied cryptography, privacy, secure e-commerce protocols and mobile applications. He has published and participated in several international and national conferences, and he is author of a patent.

Maria-Magdalena Payeras-Capellà (Mallorca, 1973) got her title as Telecommunication Engineer (1998) from the Polytechnic University of Catalonia (UPC) and her Ph.D. in Computer Science (2005) from the University of the Balearic Islands (UIB). She has held several posts in the University of the Balearic Islands since 1998 in the department of mathematics and computer science. Her research is focused primarily in the security in communications networks, protocols for electronic commerce and technical-legal aspects of electronic commerce. Within these lines of research she has had an active pace of publications in international journals and in international and national conferences. Author of several articles published in international journals included in Journal Citation Reports (JCR). She has been a member of the scientific and organizing committee in several international congresses. She has participated in European-funded and Spanish-funded research projects. She has participated in the creation of an awarded spin-off. The company KEYRON dedicated to telemedicine, was formed in 2006. She won the award to the best doctoral thesis in 2005 from the COIT, Engineers Spanish Association.

Macià Mut-Puigserver (Mallorca, 1966) achieved his graduate in Computer Science in 1990 at the Autonomous University of Barcelona. He received his Ph.D. in 2006 at the University of Balearic Islands (UIB). In 1996 he joined to the Department of Mathematic and Computer Science at the UIB as Associate Professor. He began teaching as a Vocational Training Professor specialized in Computer Science in 1990. He combines his job at the University with the job at the College where he has been the Computer Science Department Head for 10 years and the Mobility Coordinator of Erasmus Programme. His current research interests are: design of secure protocols, secure e-commerce, mobile applications and applied cryptography. He is author of articles on these topics in international journals and in international and national conferences. He has been a member of the scientific and organizing committee in several international congresses. He has also participated in several transfer projects.

Jordi Castellà-Roca (Menàrguens, 1975) is tenured assistant professor at Rovira i Virgili University, he is currently a member of the UNESCO Chair in Data Privacy. He got his title of Engineer in Computer Systems from University of Lleida in 1998, the title of Engineer in Computer Science from Rovira i Virgili University in 2000 and Ph.D. in Computer Science from the Autonomous University of Barcelona in 2005. His research focuses on the fields of cryptography (cryptographic protocols) and privacy. He has published over 35 works in international journals, book chapters, international and national congresses. Now he is part of the advisory board of an international magazine, and he has been a member of the scientific and organizing committee in several international congresses. He has participated in 23 Spanish-funded and Catalan-funded research projects. He has been the main researcher in two research projects funded by Rovira i Virgili University, one funded by the Ministry of Science and innovation and two funded by the Ministry of Industry, Tourism and Trade. He has also participated in several transfer projects, and he is the author of six patents, five of them international and in operation. He is a founding partner of three technology companies that have been awarded.

Josep-Lluís Ferrer-Gomila (Mallorca, 1967) got his BSc degree as Telecommunication Engineer (1991) from the Polytechnic University of Catalonia (UPC) and Ph.D. degree in Computer Science (1998) from the University of the Balearic Islands (UIB). He is associate professor in the University of the Balearic Islands since 1995 in the department of mathematics and computer science. His main topic of research is security in electronic commerce. He is author of publications in international journals and in international and national conferences, some of them published in international journals included in Journal Citation Reports (JCR). He has been a member of the scientific and organizing committee in several international conferences. He has participated in European-funded and Spanish-funded research projects, main researcher in four. He has participated in the creation of an awarded spin-off dedicated to telemedicine (2006).