Anonymous reputation based reservations in e-commerce (AMNESIC)

Helena Rifà-Pous
Internet Interdisciplinary Institute
Universitat Oberta de Catalunya
Rambla del Poblenou 156, 08018-Barcelona, Spain
hrifa@uoc.edu

ABSTRACT

Online reservation systems have grown over the last recent years to facilitate the purchase of goods and services. Generally, reservation systems require that customers provide some personal data to make a reservation effective. With this data, service providers can check the consumer history and decide if the user is trustable enough to get the reserve. Although the reputation of a user is a good metric to implement the access control of the system, providing personal and sensitive data to the system presents high privacy risks, since the interests of a user are totally known and tracked by an external entity. In this paper we design an anonymous reservation protocol that uses reputations to profile the users and control their access to the offered services, but at the same time it preserves their privacy not only from the seller but the service provider.

Keywords

anonymous, privacy, reservation, reputation, security

1. INTRODUCTION

Privacy is becoming increasingly important with the dawn of the Internet. The advent of new applications and services that require users to divulge many personal details about themselves, is a privacy thread that must be carefully studied. The practice of surveillance is increasingly being used by the private sector with the aim of data profiling the entities and individuals, and getting information (e.g. strengths, weaknesses) of the people they have to deal with. Citizens may be concerned that they are constantly being watched, and that their personal details and situation can get in the wrong hands and be exploited by unwanted or even criminal people.

One of the Internet activities that is more vulnerable to privacy attacks is the e-commerce. Vendors are interested in profiling the preferences of consumers because they can infer the maximum price each user is willing to pay for a good or services and so, set the prices accordingly and obtain the maximum benefit possible. Of course the profiling can also be interesting from the point of view of the consumers when they get personalized offers and recommendations, but it should be in the consumers' control to decide when they want to be profiled or when not.

Among e-commerce architectures, we focus our work in business to consumer (B2C) platforms for small and medium enterprises (SMEs). In SMEs, e-commerce is promoted as a mean to improve competitiveness. However, a full implementation of an e-commerce site requires an important financial investment that most SMEs can not afford, and poor maintained sites are prone to provide a bad impression to the customer and so be counter-productive. Thus, the winning e-commerce platform among SMEs is structured in two sides with intermediaries in the middle. On one side of the business are vendors that want to reach consumers. On the other side are potential consumers who may or may not be interested in purchasing the offered items and receiving advertising messages. In between there are intermediaries that operate portals to facilitate the connection of vendors and consumers.

Intermediary web portals or business brokers, are key players in the achievement of a purchase agreement. Customers use the platform to compare the offers of different vendors and finally select the goods or services they want to purchase. The broker can get a lot of information from this browsing, analysis, selection, and purchase. Note that a user can be profiled even if she does not transmit personal information with the portal. Only with her attributes (pseudonym, mail, IP address, ..) the vendor can be able to recognize her.

One of the critical factors for the success of an e-commerce web portal is its capacity to induce confidence among consumers. In [18], Kim et al. study the roles of trust and risk in an e-commerce decision and which are the elements to build these roles. Their results show that both privacy and security policies have very strong effects on customers' trust in a website, and that the willingness to make a transaction is inversely proportional to the involved perceived risk. Moreover, the presence of a third-party seal does not increase the consumers' trust, although it can help by reducing the perceived risk of a purchase.

B2C portals can integrate reservation and purchasing facilities. On the one hand, portals that offer buying and paying

for a good are more complex to deploy since there is a financial risk and both consumers and vendors must trust the system. Well designed protocols such as Secure Electronic Transaction (SET) [23] have been proposed, but they have not become widely used due the companies uncertainties to invest in a complex technology that can be difficult to understand and trust from customers, and the implementation cost and overhead associated with it. On the other hand, portals that only provide booking facilities are easier to deploy, and so more extended. However, they also suffer security risks that often have not been appropriately treated. These risks mainly include non-repudiation actions for the vendor and the customer, and privacy issues.

Thus, the propose of this paper is the development of an easy to deploy e-commerce B2C platform suitable for SME, that minimizes the perceived risks of the customers, and that provides effective trust mechanisms. Since the online payment systems are the elements of e-commerce platforms that users are more reluctant to, we design a reservation system that do not requires monetary payments. We present a novel platform, the AMNESIC -AnonyMous reputatioN basEd reServatIons in e-Commerce- that provides access to advertisements, getting information about a product, reading experts and customer's reviews, and reserving a service, preserving the privacy of the consumers. AMNESIC does not require to introduce new agents in actual e-commerce platforms based on a broker, and proposes a protocol that is simple as well as secure.

The remainder of this article is organized as follows. Section 2 reviews literatures related to privacy protecting e-commerce schemes. Section 3 presents the architecture of our framework, whereas the details of the processes involved in the scheme are illustrated in Section 4. Section 5 evaluates the proposed reservation scheme through three viewpoints: costumer, broker, and vendor. Finally, conclusions and further work are given in Section 6.

2. RELATED WORK

Anonymity and reputation seems to be obviously contradictory requirements since reputation systems need some sort of accountability to discern whether users behave in an appropriate way or not. Several papers (eg. [5, 15, 2]) identify the problems associated with technologies offering anonymity and accountability, and agree that there must be a balance between these two properties and that a proper anonymous scheme should not reveal the identity of honest users but should track the identity of dishonest ones.

Different approaches to deal with anonymity and accountability have been proposed. The direct anonymous attestation scheme (DAA) [4] permits the authentication of a user through a special hardware module, called Trusted Platform Module (TPM)[19], that provides a secure environment capable of storing secret information, generating cryptographic keys, and implementing cryptographic functions such as encryption and digital signatures. At the same time, it preserves the privacy of the user that owns the module. Solutions based on DAA can be used for e-commerce applications, but they have the drawback of requiring that clients hold a particular hardware module.

Some works propose building an infrastructure that can issue anonymous accountable tickets [11, 13]. Although effective, these solutions are complex since several parties and transactions are required in order to get the elements to carry out an e-commerce operation. For example, in [13] users need to hold at least two real identity certificates issued from separate Certification Authorities in order to operate in the system. Moreover, the generation of an anonymous ticket to make an e-commerce transaction, involves a trusted party that is the responsible to generate the ticket, the above-mentioned CAs, and a trusted Legal Authority that will trace the anonymous ticket to its corresponding real-identity certificate if the user misbehaves.

Other proposals treat the problem of doing an e-commerce transaction with an anonymous e-cash protocol. The aim is that a user is able to buy goods on-line and that nor the vendor knows who is the consumer, nor the bank knows the items that a user has bought. The solutions usually involve a third party that is involved in the purchase process in order to help the consumer to get the required anonymity [25], or, in some more efficient schemes [7, 20], it helps the system to trace users operations when a bank wants to identify a dishonest consumer. Canard and Gouget propose in [8] a construction that does not require a trusted third party. However the complexity and computations of the protocol are notable, and quite superior than the ones of schemes that deal with a trusted third party.

The closest related works to our proposal are the designs of [17, 26, 3], which define general frameworks to ensure accountability in reputation systems while maintaining anonymity. For example, Buttyan et al. [6] present an architecture based on the existence of a customer care agency that is fully trusted by both client and service provider, and design a ticket-based system that allows anonymous accesses to the services. However, like most of the payment system proposals [24], this solution suffers the problems of including a trusted third party in the system.

The proposals that do not require a trusted third party are based on the collaboration of different users/entities of the system. For example, the Accountable Self-Organizing Communities (A2SOCs) scheme proposed in [17] generates on time identities for the users of the e-commerce system, and the association between the real identity of the users and their virtual one is hidden using public cryptography. To disclose this association a master key is required, and this key is distributed among the members of the network using threshold cryptography. Therefore, the collaboration of several members of the community is required to trace a user.

The A2SOC proposal is extended in [26], where the authors design a layered architecture to reduce the possible exposure of users' private information while enforcing accountability.

Besides, [3] proposes a system in which users requests the authorization access to an e-commerce platform using a one time identifier supported by some attribute credentials from external referees. External referees may request to know the identity of the user to issue a positive credential, but do not know which is the purpose and the destination of this.

3. ARCHITECTURE OVERVIEW

The research case of our designed B2C platform, examines a scenario where a number of consumers intend to make a reservation of a good or service through an e-commerce portal in the Internet. Users get the reserved item by a face encounter with the vendor, and in that moment they make the payment of it. The system can be used for getting tickets for some performances (theatre, sports, concerts, ..), making reservations in hotels and restaurants, getting appointments in a medical centre, or reserving a good in a shop.

The scheme involves three participants: the Broker (B), the Vendor (V), and the Customer (C). Besides, a Certification Authority (CA) is assumed to issue identity certificates for the involved parties in order to participate in the scheme. Figure 1 depicts a general overview of the architecture of the system.

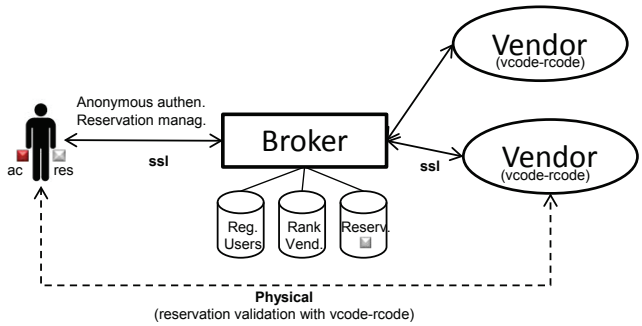

Figure 1: System Architecture

The communication between the customer and the broker is over an SSL server authentication session. The broker and the vendor use a mutual authentication SSL channel to communicate with. And finally, the data interchange between the customer and the vendor must be granted that it can be provided without technological means, using a physical channel. This limitation assures that the system can be deployed easily and without a great inversion to any running company, and that it can reach all the customers that already use the Internet to find and compare goods and services. Regardless of this condition, the communication between the customer and the vendor can also be automatized using a wireless personal area network, like Bluetooth.

For illustrative purposes, we will take the particular case of making a reservation in a restaurant to explain the system architecture and the processes involved in it. From the point of view of the reservation portal, it can no longer track the users; However, since users have to return to the web site after the reservation is executed in a restaurant, the portal can take profit of this to obtain more feedback from its services and that of the restaurants. Thus, the information of the portal can be improved, being more accurate,

The role of the system participants' is the following:

1. Certification Authorities: the system is supported by one or more external and trustworthy PKIs.
2. Broker: web portal that advertises and classifies the restaurants. It maintains a ranking of the restaurants based on customers' opinions. It also holds reviews.
3. Vendors: restaurants.
4. Customers: people that have registered in the web portal and are potential customers of the restaurants.

The system let users reserve a restaurant from the web portal without paying any fee. However, users have an associated reputation and restaurants may state that people whose reputation is below a certain threshold cannot reserve a table in their premises. If the reservation request of a user is accepted, she gets her reputation decreased in one point and in exchange, she receives a reservation code (named $rcode$).

When the user executes the reservation, she goes to the solicited restaurant and presents the reservation code $rcode$. The restaurant checks if it has a reservation identified with this number, and if it does, it offers a table to the consumer and gives her a validation code (named $vcode$).

Afterwards, the customer can enter the received $vcode$ in the web portal and recover the reputation point she had used to pay for the reservation. Moreover, if she gives feedback of the experience in the restaurant, she gains another extra point that increases her reputation. Contrary, if the customer does not introduce the $vcode$ in the application, it is assumed that she did not fulfil her obligations with the restaurant and so, she can not reclaim her spent reputation point.

Therefore, the system works such that users do not have to pay money to make a reservation, but they have an associated reputation (which is measured with points) that expresses if they finally meet their reserves or not. This reputation is the metric used to evaluate whether to allow users to make new reservations when they demand it to.

Besides, the restaurants also have a reputation from the evaluations that former customers have reported of their service. This reputation is public for all system users' so that they can use it to decide if they want to reserve a table in a particular restaurant or not.

The security properties of the system guarantee the right execution of the protocols involved in the reservation and the privacy protection of the customers that participate in it. In particular, the aimed properties of the system are the following.

1. From the customer's point of view:
   - No one (neither the broker nor the vendor) know the identity or the user profile of the customer who is making a reservation.
   - No one but the customer herself is able to make a reservation using her own reputation.
   - Only the customer specified in the reservation (or someone in whom she trusts and delegates) can receive the good or service she has reserved.
   - Only the proper customer can recover the funds she has deposited to make a reservation.

- The customer can be certain of being correctly credited for the reservations and deals with the vendors.

2. From the broker's point of view:
   - The broker can detect a fraudulent use of a customer's deposit.
3. From the vendor's point of view:
   - The vendor cannot deny that a customer has completed the deal agreed in a reservation.

4. PROTOCOL

The main characteristic of the AMNESIC system is that provides anonymity. To this end, we use partially blind signatures. Blind signatures were first introduced by Chaum [10] and satisfy the unforgeability (only the signer is able to generate valid signatures) and unlinkability (no one can derive the association between the message and the blind signature, except the signature requester) properties. Partially blind signatures, introduced later by Abe and Fujisaki [1], allow that moreover a signer can include some piece of information to the signed data, under an agreement with the signature requester.

However, Coron et al. [12] found both Chaum and Abe-Fujisaki algorithms presented the problem of being vulnerable to chosen-plaintext attacks. Fan et al. [16] proposed a randomization technique to withstand the attack on blind signatures. Using randomization the message to be signed is determined by both of the signature requester and the signer and so, the attackers cannot obtain the signature of an arbitrarily chosen message. Cao et al. [9] proposed a randomized RSA-based partially blind signature which meets the unforgeability and unlinkability properties. Our reservation scheme is based on the Cao et al. algorithm.

The functionalities provided by the scheme are classified in six distinct actions: (1) the registry in which a customer first enters in the web Portal, (2) the reservation phase in which a customer books a service in the web Portal for a specific vendor, (3) the cancellation, that is produced when a customer wants to vacate a reservation, (4) the service phase, in which the customer gets the service she had reserved and pays for it, (5) the deposit phase where the reputation of the customer and vendor is updated, and (6) the renewal phase in which the customer updates her pseudonymous before it expires.

Details of the six actions of the scheme are shown in the following. First, we introduce in table 1 the notations we use.

It is assumed that the potential customers of the scheme, the vendors, and the broker, possess an RSA digital certificate issued by a trusted CA. The public key of the broker satisfies $2^{d+1} < pbk_b$, with d the length of the hash function h used in the reservation protocol.

4.1 Registry

At the start of the protocol a user U registers herself in the restaurants' web portal (managed by a broker B) sending her identity certificate $cert_u$.

uid	The identity of a customer U
bid	The identity of a broker B
vid	The identity of a vendor V
psd	The temporal pseudonymous of customer U - $psd(t)$ is the authentication pseudonymous in time t-
pvk_k, pbk_k	The private and public key of entity K
n_k	Working modulus of the pair key of entity K
$cert_k$	A digital certificate of the entity K signed by a TTP
rep	The reputation account of a customer U
exp	The expiration date of temporal pseudonymous and its associated reputation
$h(\cdot)$	one-way hash function
d	the length of the one-way hash function output
w	The wallet of a customer U
db	The data base of a broker B
$rcode$	A reservation code of a vendor
$vcode$	A validation code of a vendor
db_U	Web portal database with information of registered users and their initial reputation
db_R	Web portal database with information about the unexpired reservations requested by its customers, and the pseudonymous used for it.
log	Web portal database with a list of already used but unexpired pseudonymous
db_V	Vendor database with information about the pending reserves
$Z^*_{n_k}$	Set of all positive integers less than and relative prime to n_k
$\|$	concatenation function

Table 1: Notation

On receipt of the first message, the broker B verifies the certificate $cert_u$ and if it is correct, it sends to U an integer $y = x^{pbk_b} \ (mod\ n_b)$, with $x \in Z^*_{n_b}$ a random number, and pbk_b its public key. The integer y plays two roles in this context: (1) the signer's commitment to the randomizing factor x, and (2) a challenge for checking that the customer owns the declared private key.

User U demonstrates knowledge of her private key pvk_u by computing a signature over the received challenge y, which she then sends to B together with a message np ($next\text{-}pseudonymous$) . To build np, the customer first generates a random sequence of bytes, psd, that constitutes the pseudonymous she will use to authenticate herself to the broker the next time she gets a session. Then, she generates two random numbers: (1) a randomizing factor $v \in Z^*_{n_b}$, (2) a blinding factor $r \in Z^*_{n_b}$. She computes the message np doing

$$np = h(psd|(v^{pbk_b}y))r^{pbk_b}v\ mod(n_b)$$

On the receipt of the third message, B checks whether the signature of the challenge is correct using the certificate $cert_u$ obtained in the first step of the protocol. If the signature is verified, B establishes an initial reputation rep for the user, sets the expiration date exp until which the new pseudonymous of the user will be valid, and it computes the public information of the signature, $i = rep|exp$. Then, it formats the common information as:

$$\tau(i) = 2^d + h(i)$$

So, $\tau(i) \in [2^d, 2^{d+1}]$, and it can be assured that $\tau(i')$ is not divisible by $\tau(i)$ when $i' \neq i$. Finally, B computes the blinded signature doing:

$$sp = ((np.x)^{pvk_b.\tau(i)})^{-1} \; mod(n_b)$$

The signature sp (*signed-pseudonymous*) is a mixed-blind signature because the broker has signed a message $np.x$ which contains information from both the user and the broker. Indeed, part of the sp message is hidden with a random number that only the customer knows, and the signing private key includes common information between the customer and the broker: the reputation and expiration attributes of the customer.

Figure 2 depicts the steps of the registry process.

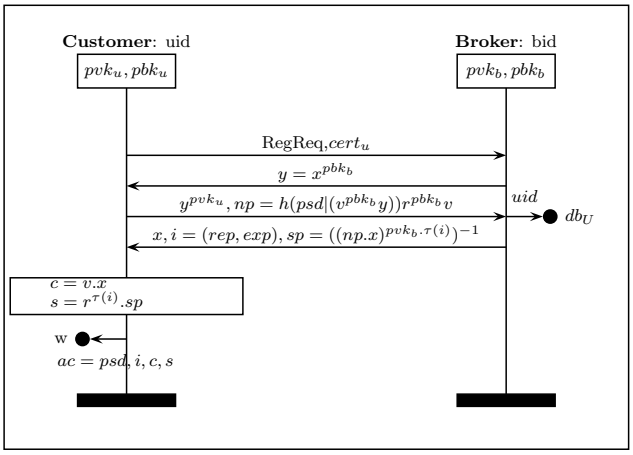

Figure 2: Customer Registry

The broker sends the signature sp along with its randomization factor x and the reputation and expiration parameters (rep, exp) to the customer. Since the customer knows her randomization factor v and the brokers' x, she can compute the joint randomization factor $c = v.x \; mod(n_b)$. Then she removes the blinding factor r from the received sp message to get the broker's signature s over her new pseudonymous psd: $s = r^{\tau(i)} sp \; (mod \, n_b)$.

To ensure that the reputation and expiration values that the broker has included in the signature are right, and verify that the signature is correctly bound to her account, the customer checks if $s^{pbk_b}.(H(psd||c^{pbk_b}).c)^{\tau} \equiv 1 \; (mod \, n_b)$.

The customer stores in her wallet all the account information (ac) for later use in the reservation phase. The stored data is: (1) the pseudonymous (psd), (2) the reputation of the customer (rep), (3) the expiration date of all this account data (exp), (4) the signature joint randomization factor (c), and (5) the broker's signature of the above fields (s). Note that the broker does not know neither the pseudonymous of the user nor their joint randomization factor, so the next time she initiates a session, the broker could neither track nor profile the habits of the customer.

4.2 Reservation

When a customer wants to reserve a table in a restaurant, she sends a reservation request ($ResReq$) message to the broker. In the message, the user includes the vendor identifier (vid) of the restaurant she wants to reserve, as well as the attributes of the reservation ($vatt$), such as the day, time, and number of people.

Once the broker receives the $ResReq$ message, it checks the availability of the vid restaurant to make a reservation with $vatt$ characteristics and sends back a response message $ResRes$ to the user with this information. Moreover, if the response is positive, the broker invites the customer to proceed with the anonymous reservation by attaching in the message a randomizing factor $y = x^{pbk_b} \; (mod \, n_b)$.

After receiving $ResRes$, the customer sends a message identifying herself with the account information (ac) she stores in her wallet. She also computes a new pseudonymous psd' to be used the next time she operates with the broker. User U blinds the new pseudonymous psd' using two new random factors v' and r'. The resultant blind account information np is attached in the message $ResRes$ message.

On the receipt of $ResRes$, the broker first verifies the customer has access to the system by:

1. Verifying the signature of the account information ac is correct $(s^{pbk_b}.(H(psd||c^{pbk_b}).c)^{\tau} \equiv 1 \; (mod \, n_b))$
2. The expiration date of the account data has not passed
3. The account data is not in the list of already used logins log

If the account information is valid, the broker stores the account information in the list of already used logins log. The log list prevents that customers use a particular ac data to enter the system several times. The list stores used accounts till they expire. If a user tries to get a session with an expired account, the broker requests her to register in the system again and initializes her reputation rep with a penalization (p.e. 10 points less the normal initialization rate, which can be set at 0 points). Each time the user has to register again, the penalization in her initial reputation is bigger. Brokers can give the users the opportunity to recover some reputation points up to the standard initial level (0 points) by paying a fee.

When a user makes a reservation request, the broker contacts the solicited restaurant and asks for a reservation with $vatt$ characteristics and associated to a user with a rep reputation level. If the restaurant agrees, it generates a random $vcode$ of 12-bytes and computes $rcodeT = h(vcode)$ and $rcode = rcodeT \; (mod \; 2^{64})$. The restaurant sets a reservation entry in its database db_V with the tuple (rcode, rcodeT, vcode), and sends to the broker the values $rcode$ and $rcodeT$. The broker also creates a new entry in its reservation database db_R with the data (psd, rcode, rcodeT). Then, the broker subtracts a unit from the user's reputation (provided in the account information ac in the reservation request message) and signs a message sp that includes the customer's next reputation and expiration parameters

(rep', exp'), as well as a hidden link to her next pseudonymous psd'.

Finally, the broker returns to the customer a reservation response message with information about its randomization factor x, the user's reputation and expiration parameters (rep', exp'), a blind signature of user's account information (sp), and the short reservation code of the service she has solicited $(rcode)$. $rcode$ is sent to the user in order to present a reservation identifier in front of the final vendor. To this end, $rcode$ is short (only 6-bytes), and is sent to the user in base64 encoding (it occupies 8 characters) so that it is easy to manage.

After receiving the message, the customer verifies the signature and stores in her wallet the information of her account and her reservation.

Figure 3 summarises the steps required to make a reservation.

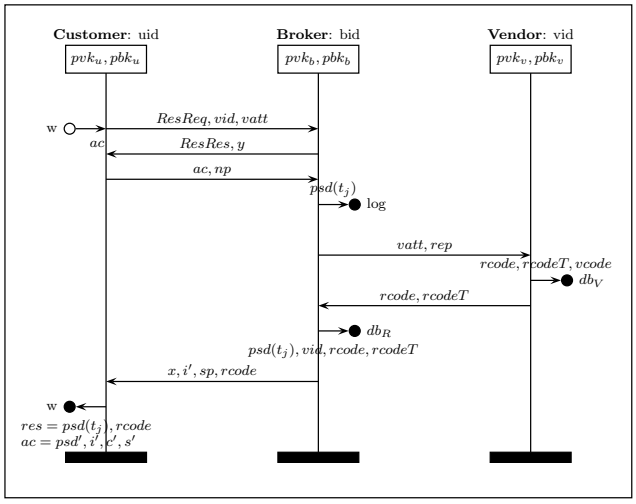

Figure 3: Reservation

4.3 Cancellation

In case the customer wants to cancel a reservation she sends a cancellation request $CanReq$ to the broker. The $CanReq$ contains information of the reservation to cancel (res).

On the receipt of a $CanReq$ message, the broker verifies if the vendor's policies allow the cancellation. B responds to the user whether she can proceed with the cancellation and it includes in the replying $CanRes$ message a randomizing factor $y = x^{pbk_b} \ (mod\, n_b)$.

After receiving $CanRes$, the customer identifies herself with her account information (ac) and the blind account information np to be used in the next session.

When the broker receives $CanRes$, it verifies the customer has access to the system in the same way as in the reservation phase (see subsection 4.2), and stores the account information in the list of already used logins log. Then, the broker checks the reservation information (res) is correct and, if the cancellation policies allow it, it deletes the reservation and sends the cancellation to the vendor. The broker also increases the reputation of the customer in one unit, and signs a message sp to the customer that includes the customer's next reputation and expiration parameters (rep, exp), as well as a hidden link to her next pseudonymous np.

At last, the broker returns to the customer a cancellation response message with information about her reputation and expiration parameters (rep, exp), and a blind signature of her account information (sp). The broker also informs the vendor of the reserve cancellation by sending it a message.

After receiving the message, the customer verifies the signature and stores in her wallet the information of her account.

Figure 4 summarises the steps required to cancel a reservation.

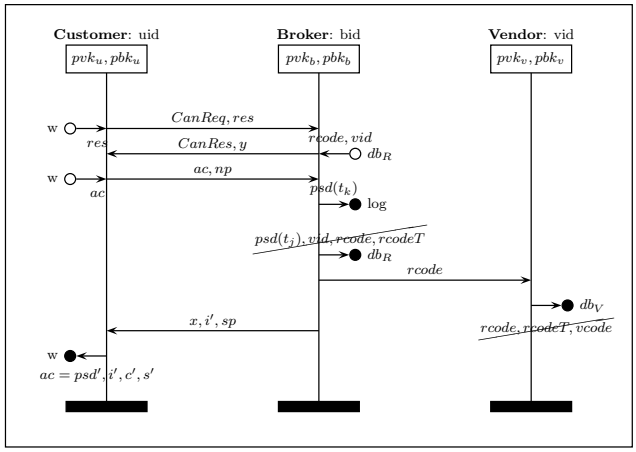

Figure 4: Cancel Reservation

4.4 Service

When the customer wants to get the service she has reserved, she goes the vendor and shows the $rcode$ value (which is part of the res record she had stored in her wallet in the reservation phase, part 4.2).

The vendor verifies that there is actually a reservation with code $rcode$ and if it is, it responds to the user with the validation code $vcode$. Users will only be able to demonstrate to the broker that they have fulfilled a reservation if they provide the $vcode$ linked to a reservation $rcode$, since $vcode$ satisfies that $rcode = h(vcode) \ (mod\, 2^{64})$. The vendor gives the $vcode$ to the user either automatically (i.e. through a Bluetooth push connection) or manually printed in a ticket. $vcode$ is represented in base64 so that it can be printed without problems. Its printed length is 16 characters, which is short enough to be managed by hand. Once the user has received the $vcode$ of the vendor, she can verify that it is correctly performing the hash operation over it. Note that if the user holds a smart phone, all the operations of the protocol can be easily and automatically executed by the mobile itself which has the computational resources to do it [21, 22].

The reason for *rcode* to be so short is twofold: in one hand it is easy to manage by the users that may have to handle with this code manually. On the other hand it prevents brute force attacks by the customers that try to get the associated *vcode* of a reservation to earn good reputation in the broker system without having achieved their responsibilities with the final vendor. Each *rcode* can be derived from an average of 2^{48} vcodes, and for a customer it is impossible to know which one of them is the real one. So the probability of lunching a brute force attack and succeed, is lower than $10^{(-14)}$.

4.5 Deposit

When a customer that had a reservation completes the action she had reserved (i.e. going out for a dinner in a restaurant), she contacts the broker to update her reputation (see Figure 5 for a summary of the required steps).

The customer triggers the deposit protocol sending a *DepReq* to the broker. The *DepReq* contains information of the reservation that has been completed (*res*).

On the receipt of a *DepReq* message, the broker verifies that it has the reservation code in its database db_R. B responds to the user whether she can proceed with the deposit and it includes in the replying *DepRes* message a randomizing factor $y = x^{pbk_b} \ (mod\, n_b)$.

After receiving *DepRes*, the customer sends a message identifying herself with her account information (*ac*) and providing blind account information *np* to be used in the next session. The customer also sends the validation code *vcode* she received when she visited the reserved restaurant, and some optional feedback (*rnote*) about the service obtained by the vendor. Only the validated customers of a restaurant can express their opinions of its services and enrich the reviews analysis managed by the web portal.

When the broker receives *DepRes*, it verifies the customer has access to the system and stores the account information in the list of already used logins *log*. Then, it checks if the provided *vcode* is valid doing $rcodeT \equiv h(vcode)$. If the code is valid, it deletes the reservation from its database db_R, increases the reputation of the customer, and signs a message *sp* to the customer that includes the customer's next reputation and expiration parameters (rep, exp), as well as a hidden link to her next pseudonymous *np*.

Brokers spur customers to send review reports of their experience in the reserved service by prizing them with good reputation. Thus, when a costumer makes a deposit after visiting a restaurant, she recovers the reputation point she lost in the reservation process. If she moreover provides some feedback about the vendor, she receives one extra reputation point.

After receiving the *sp* message, the customer verifies the signature and stores in her wallet the information of her account.

4.6 Pseudonymous Renewal

The customer renews her pseudonymous each time she executes an operation with the broker. Yet, the pseudonymous has an expiration date, and if a customer do not use the broker's services during a long time, she could get her pseudonymous expired. Therefore, the AMNESIC system defines a renewal operation which only purpose is to renovate the pseudonymous of that users that do not enter in the system regularly.

The client application of the user must have a timeout that automatically initiates a pseudonymous renewal process when the user has not updated her account information during a long period. The renewal action is initiated by a *RenReq* message. On the receipt of a *RenReq* message, the broker sends back a *RenRes* message with a randomizing factor $y = x^{pbk_b} \ (mod\, n_b)$. After receiving *RenRes*, the customer identifies herself with her account information (*ac*) and the blind account information *np* to be used in the next session. The broker verifies the customer holds a valid pseudonymous and stores her account information in the list of already used logins *log*. Then, B signs a message *sp* to the customer that includes the reputation parameter *rep* she had before, and that carries a hidden link to the user's next pseudonymous *np*. After receiving the message, the customer verifies the signature and stores in her wallet the information of her account.

Figure 6 summarises the steps of a pseudonymous renewal process.

5. DISCUSSION

In this section we evaluate the system by the security and anonymous properties announced in section 3.

1. *The customer identity and profile are unknown*

Users give personal information (name, public key) to the web portal when they register to it, but this happens only once. Then, in the subsequent processes involved in a reservation, the broker controls the access of the users in the system using a signed authorization message that contains:

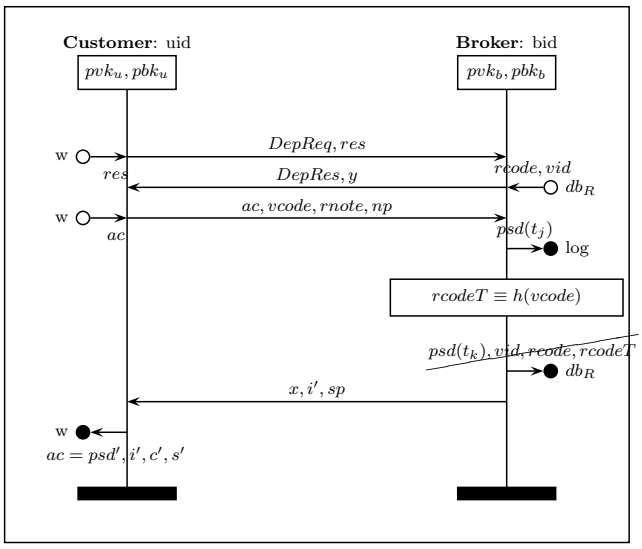

Figure 5: Coupon Validation

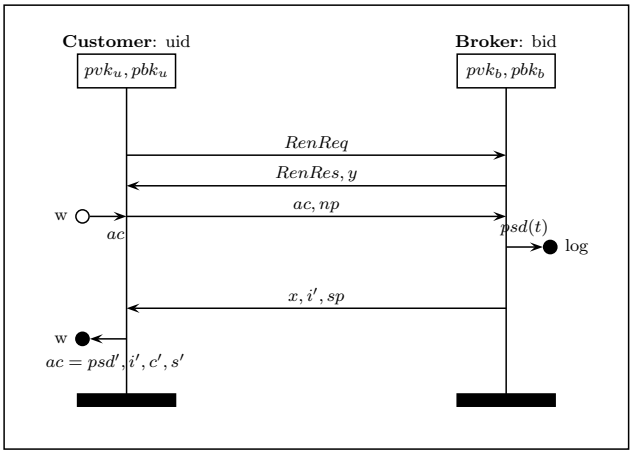

Figure 6: Pseudonymous renewal

(1) a random one-time pseudonymous, (2) the user's reputation, (3) an expiration date. All three parameters can be authenticated because they are generated using the RSA-partially blind signature scheme proposed by Cao et al. [9], which has been proved to satisfy the blindness and unforgeability properties, even in front of chosen plain text attacks. The pseudonymous can not be used to track the users since it is generated using the blind information of the signature and so, the broker have only access to them once. However, the user's reputation and expiration date are part of the common information of the signature. Nevertheless we state that these parameters cannot be used to track customers since they are very generic (we set the expiration dates to have a big granularity, i.e. weeks) and shared between a great proportion of other customers. Besides, we assume the connections of the users to the web portal can not be tracked using low layer techniques (cookies are disabled, the IP changes dynamically, ..).

2. *Only the customer can make a reservation using her own reputation*

The reputation of a customer is stored by the customer herself in her signed authorization message. The authorization ticket is sent to the broker using an SSL protected channel, so nobody can steal its information.

3. *Only the customer specified in the reservation can receive the service she has reserved*

When a user reserves a table in a restaurant, she gets a unique reservation code. The reservation code is known by the vendor, the broker, and the customer herself, but not by other users of the system. The transmission of the reservation code is done through SSL protected channels and so, it can not be recovered by any malicious user.

4. *Only the proper customer can recover the funds she has deposited to make a reservation*

To recover the funds of a reservation the customer has to present to the broker two items:

- the validation code (*vcode*) associated to the reservation. *vcode* is only know by the restaurant, and is given to the user by hand when she goes to the restaurant in fulfilment of her reservation.
- the pseudonymous (*psd*) that was used to get the reservation. The pseudonymous is only known by the user and the broker

The only person that knows both the *vcode* and the *psd* of the reservation is the user when she has completed the service she had reserved.

5. *The customer can be certain of being correctly credited for the reservations and deals with the vendors*

Users can always validate if the web portal is managing their reputation in the expected way. The reservation points of a customer are part of her signed authorization tokens. These tokens are generated by the broker, but can be verified by anyone. Thus, a customer can verify the amount of points in her account each time she gets an authorization message. Irregular reputation updates will be immediately detected by the users.

6. *The broker can detect a fraudulent use of a customer's deposit*

The broker checks that a customer can not use a pseudonymous (and its associated reputation) more than once. When a user identifies herself using a pseudonymous, the broker stores this identifier in a database until the identifier expires. During the lifetime of the identifier, for each access request in the system, the broker will check that the given pseudonymous has not been used before.

7. *The vendor cannot deny that a customer has completed the deal agreed in a reservation*

The only way that a customer can know the *vcode* associated with a reservation, is because the vendor has given this to her.

The function h is collision resistant, which means that it is computationally infeasible to find two inputs $x' \neq x$ which are mapped to the same value $y = h(x) = h(x')$. There are a number of practical cryptographic functions that are assume to be collision resistant, e.g. SHA.256, SHA-384 and SHA-512 [14]. Their collision resistance strengths are half the lengths of their hash value.

The reservation and validation codes (*rcode* and *vcode*) used in AMNESIC are quite short due to the limitation of being able to deal with them without using automatic processes but transcripts. In particular, the reservation codes are the result of hashing a 12-byte *vcode* string. In this case, the

collision resistance offered by the hash function is half the length of the input, resulting in a strength of 48 bits. A security strength of 48 bits is below the general NIST recommendations, since a brute force attack to the hash function would be able to succeed. Thus, to increase the security of the system, AMNESIC proposes that the reservation code that is sent to the customer *rcode* is not the output of the hash function but a truncated version of it. In this way, a malicious customer does not have any computational way to find out *vcode* from *rcode*.

If a customer presents a valid *vcode* in the broker (i.e. $rcodeT = h(vcode)$), it means that she has received this code from the vendor, and the vendor cannot deny that the reservation has been fulfilled.

6. CONCLUSIONS

In this paper we have described the AMNESIC system for anonymous on-line reservations. A key concept in AMNESIC is the provision of user privacy protection both in front of brokers and final vendors. Moreover the system is easy to deploy and do not require great investments from the vendors or the customers. We have specified the protocols of the processes involved in a reservation service and have analysed their security and robustness.

The reservation service has been applied in the context of a web portal that promotes restaurants and permits making reservations. Clients benefit from the anonymous reservation system in the sense that neither the broker nor the restaurant knows who they are (eventually the restaurant will know their faces, but not their identity). The broker cannot gather information about how often a certain user goes to a restaurant, with how many people, or which type of food he enjoys. On the other hand, the system spurs users to give feedback about the restaurants they visited, so the broker benefits of having updated reports of the quality of the restaurants to make a thorough and valuable review of them. Finally, restaurants benefit from the system because clients are more honest in their reservations, otherwise they are penalized.

Our further work focuses on extending the AMNESIC concepts to a multi-device context, i.e., each user can connect to the broker using different terminals (private or public).

Acknowledgments

This work has been supported by the projects JC2010-0081, TSI2007-65406-C03-03 E-AEGIS, and CONSOLIDER CSD 2007-00004 ARES.

References

[1] M. Abe and E. Fujisaki. How to date blind signatures. In K. Kim and T. Matsumoto, editors, *Advances in Cryptology - ASIACRYPT*, volume 1163 of *Lecture Notes in Computer Science*, pages 244–251. Springer Berlin / Heidelberg, 1996. 10.1007/BFb0034851.

[2] G. Antoniou, L. Batten, and U. Parampalli. An anonymity revocation technology for anonymous communication. In *Information Systems Development*, pages 329–337. Springer US, 2010. ISBN 978-0-387-84810-5.

[3] R. Au, H. Vasanta, K.-K. R. Choo, and M. Looi. A user-centric anonymous authorisation framework in e-commerce environment. In *International conference on Electronic commerce*, ICEC, pages 138–147, New York, NY, USA, 2004. ACM. ISBN 1-58113-930-6. doi: http://doi.acm.org/10.1145/1052220.1052238.

[4] E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *Proceedings of the ACM conference on Computer and communications security*, pages 132–145, New York, NY, USA, 2004. ACM. ISBN 1-58113-961-6. doi: http://doi.acm.org/10.1145/1030083.1030103.

[5] M. Burmester, Y. Desmedt, R. Wright, and A. Yasinsac. Accountable privacy. In B. Christianson, B. Crispo, J. Malcolm, and M. Roe, editors, *Security Protocols*, volume 3957 of *Lecture Notes in Computer Science*, pages 83–95. Springer Berlin / Heidelberg, 2006.

[6] L. Buttyan and J.-P. Hubaux. Accountable anonymous access to services in mobile communication systems. In *In Symposium on Reliable Distributed Systems*, pages 384–389, 1999.

[7] J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In R. Cramer, editor, *Advances in Cryptology - EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 566–566. Springer Berlin / Heidelberg, 2005.

[8] S. Canard and A. Gouget. Divisible e-cash systems can be truly anonymous. In M. Naor, editor, *Advances in Cryptology - EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 482–497. Springer Berlin / Heidelberg, 2007.

[9] T. Cao, D. Lin, and R. Xue. A randomized rsa-based partially blind signature scheme for electronic cash. *Computers & Security*, 24(1):44–49, 2005. ISSN 0167-4048. doi: DOI: 10.1016/j.cose.2004.05.008.

[10] D. Chaum. Blind signatures for untraceable payments. In *CRYPTO*, pages 199–203, 1982.

[11] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24:84–90, February 1981. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/358549.358563.

[12] J.-S. Coron, D. Naccache, and J. Stern. On the security of rsa padding. In M. Wiener, editor, *Advances in Cryptology - CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 789–789. Springer Berlin / Heidelberg, 1999.

[13] D. Critchlow and N. Zhang. Security enhanced accountable anonymous pki certificates for mobile e-commerce. *Computer Networks*, 45(4):483–503, 2004. ISSN 1389-1286. doi: DOI: 10.1016/j.comnet.2004.02.010.

[14] Q. Dang. Recommendation for applications using approved hash algorithms. *NIST Special Publication*, (800-107), February 2009.

[15] C. Diaz and B. Preneel. Accountable anonymous communication. In M. Petkovic and W. Jonker, editors, *Security, Privacy, and Trust in Modern Data Management*, Data-Centric Systems and Applications, pages 239–253. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-69861-6.

[16] C.-I. Fan, W.-K. Chen, and Y.-S. Yeh. Randomization enhanced chaum's blind signature scheme. *Computer Communications*, 23(17):1677 – 1680, 2000. ISSN 0140-3664. doi: DOI: 10.1016/S0140-3664(00)00254-1.

[17] C. Farkas, G. Ziegler, A. Meretei, and A. Lörincz. Anonymity and accountability in self-organizing electronic communities. In *ACM workshop on Privacy in the Electronic Society*, WPES, pages 81–90, New York, NY, USA, 2002. ACM. ISBN 1-58113-633-1. doi: http://doi.acm.org/10.1145/644527.644536.

[18] D. J. Kim, D. L. Ferrin, and H. R. Rao. A trust-based consumer decision-making model in electronic commerce: The role of trust, perceived risk, and their antecedents. *Decision Support Systems*, 44(2): 544 – 564, 2008. ISSN 0167-9236. doi: DOI: 10.1016/j.dss.2007.07.001.

[19] S. Pearson. *Trusted Computing Platforms: TCPA Technology in Context*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2002. ISBN 0130092207.

[20] C. Popescu. An anonymous mobile payment system based on bilinear pairings. *Informatica*, 20:579–590, December 2009. ISSN 0868-4952.

[21] H. Rifà-Pous and J. Herrera-Joancomartí. Cryptographic Energy Costs Are Assumable in Ad Hoc Networks. *IEICE Trans. Inf.& Syst.*, 92:1194–1196, 2009. doi: 10.1587/transinf.E92.D.1194.

[22] H. Rifà-Pous and J. Herrera-Joancomartí. Computational and energy costs of cryptographic algorithms on handheld devices. *Future Internet*, 3(1):31–48, 2011. ISSN 1999-5903. doi: 10.3390/fi3010031.

[23] Visa and M. Card. *Secure Electronic Transaction Standard - Book 3: Formal Protocol Definitions*. SETCo, 1997.

[24] H. Wang, L. Sun, Y. Zhang, and J. Cao. Anonymous access scheme for electronic-services. In *Australasian conference on Computer science - Volume 26*, ACSC, pages 295–304, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[25] H. Wang, J. Cao, and Y. Zhang. A flexible payment scheme and its role-based access control. *IEEE Trans. on Knowl. and Data Eng.*, 17:425–436, March 2005. ISSN 1041-4347. doi: http://dx.doi.org/10.1109/TKDE.2005.35.

[26] G. Ziegler, C. Farkas, and A. Lorincz. A framework for anonymous but accountable self-organizing communities. *Information and Software Technology*, 48 (8):726 – 744, 2006. ISSN 0950-5849. doi: DOI: 10.1016/j.infsof.2005.08.007.