A Selective Privacy-Preserving Identity Attributes Protocol for Electronic Coupons

Pau Conejero-Alberola, M. Francisca Hinarejos, Josep-Lluís Ferrer-Gomila

A Selective Privacy-Preserving Identity Attributes Protocol for Electronic Coupons

Pau Conejero-Alberola*, M. Francisca Hinarejos, Josep-Lluís Ferrer-Gomila

University of the Balearic Islands, Ctra. de Valldemossa, km 7,5. 07120, Palma, Spain
Email: {pau.conejero, xisca.hinarejos, jlferrer}@uib.es

Abstract. Electronic coupons (e-coupons) are a very effective marketing tool. In some scenarios, it is necessary to check some customer's personal attributes at the redeeming phase (e.g. age, title, citizenship, etc.). But customers may be reluctant to use e-coupons if their privacy is in danger. Digital certificates and credentials could be suitable for validating customer attributes. However, a bad use of such electronic documents entails a loss of privacy, revealing more identity attributes than necessary. Here, we present the first secure protocol for e-coupons, achieving verification proofs of identity, with selective disclosure of customer's certified attributes. On the other hand, our proposal meets other necessary security requirements, such as forging protection and double-redeem protection.

Keywords: E-coupon, Security, Identity, Privacy

1 Introduction

A coupon is a usually small piece of printed paper that lets you get a service or product for free or at a lower price (Merriam-Webster definition). It is an effective marketing instrument [1], and quite used because merchants and customers are benefited. One one hand, merchants can increase loyalty of their customers or attract new customers. On the other hand, customers can achieve better prices or gifts. In this paper we deal with a type of e-coupon that is addressed to a group of customers that must meet certain identity requirements, for example, being in a certain age group, being resident in a specific country, etc. A real example of this type of promotions is found in a well-known chain of hamburgers [2], whose promotional bases indicate: customers must be over a certain age, resident in X, and have to purchase product Y.

Paper based coupons, redeemed face-to-face, allow the merchant to easily verify compliance with established requirements, using paper documents (such as an identity card). Such verification does not usually result in a high loss of privacy, because verification is instantaneous without the merchant registering private information of the customers in their systems. But in the case of e-coupons, exchanges are made electronically, and therefore "verification documents" must also be in electronic form to be redeemed face-to-machine.

Digital certificates and credentials are suitable to provide authenticated information about customers. These documents are signed by a trusted third party, which assumes the responsibility of validating the data contained in them. But those documents contain more information than may be necessary in certain e-coupon scenarios, with the consequent risk of loss of privacy. As a result, many customers are reluctant to provide personal data, beyond the strictly necessary for the purpose they want to perform. Our goal is to maximize the privacy of customers, that is, only that information strictly necessary to validate compliance with the requirements of the e-coupon must be disclosed.

Contribution. We present the first e-coupon scheme with a selective and verified personal data disclosure mechanism, which provides a better degree of privacy, allowing the issuance of e-coupons for eligible customers. On the other hand, customer's compliance of requirements is verified during the redeem.

This paper is organized as follows. Section 2 reviews the related work. Section 3 defines the proposed scheme, the security requirements and the cryptographic background. Section 4 specifies in detail all the phases of the proposed protocol. Section 5 includes a brief security analysis, and finally, Section 6 lists the conclusions of this paper.

2 Related work

In this section, we will review those most significant proposals that have shown concern about the authentication and privacy of customers.

As a first contribution of e-coupons, Kumar et al. [3] show that targeted e-coupons are intended for a group of customers who meet certain requirements, and they indicate that customers must be identified.

In contrast, Jakobsson et al. [4] state that an e-coupon system should not expose customer privacy more than other advertising system. They present a proposal where no attribute of the customer is verified.

Chang et al. [5] present a scheme with a registration phase, where the customer provides personal information to the issuer. Therefore, e-coupons are identified, but merchants do not receive this information from the customer.

Aigner et al. [6] explain two e-coupon schemes. One of them has an authentication process for the customer in front of the issuer and merchant.

Chang et al. [7] explain a scheme where the customer and merchant must be registered at a trusted third party, indicating a mutual authentication.

Chang et al. [8] provide two e-coupon schemes, one for specific registered customers (with better discounts) and the other for non-specific customers.

Liu et al. [9] provide a proposal, in which customers remain anonymous if they are honest. Their scheme achieves traceability against dishonest customers.

As a conclusion, we can affirm that there is no previous solution that requires the authentication of some attributes of the customer, and that this authentication takes place without revealing other data related to that customer. That is, the selective disclosure of attributes is a problem that has not been addressed so far in the area of e-coupons.

3 Scheme: Scenario and Security

In this section, we describe our proposal. First, we detail the scenario, the entities involved, their role and we outline the e-coupon structure. Then, the security requirements and the cryptographic background are defined.

3.1 Scenario

We define a custom environment, offering an online distribution marketing portal, using daily e-coupons promotions to be redeemed in manufacturer's branches for eligible and registered customers. The proposed scenario allows different grades of privacy using an *Idemix* [10] service, a selective method to disclose customer identity attributes. We consider the following entities: Trusted Third Party (TTP), Issuer, Merchant and Customer.

Trusted third party $\mathcal{T}$ is in charge of issuing the *Idemix* credentials based on a digital certificate, and carrying the system public parameters.

Issuer $\mathcal{I}$ is in charge of issuing and distributing e-coupons. In this scenario, $\mathcal{I}$ has to ensure that the credential has not been used previously for registering, and only one-time e-coupon is delivered per customer $\mathcal{C}$. A registered $\mathcal{C}$ can request the e-coupon policy from the portal, which defines the redemption requirements to compute the identity proof. If the proof is valid then $\mathcal{I}$ issues the e-coupon.

Merchant $\mathcal{M}$ is in charge of verifying if the e-coupon is valid and has not been used before at the same $\mathcal{M}$. As the e-coupon was issued for a specific $\mathcal{C}$, the merchant $\mathcal{M}$ has to ensure if $\mathcal{C}$ has the right to use it.

$\mathcal{C}$ is the actor who makes use of the platform to generate the identity proofs of its credential attributes, and requests and redeems the e-coupons.

Next, the promotional e-coupon is detailed as follows: SN represents the identifier of the selected e-coupon, I_1 includes the offer data of the promotion, I_2 contains all the public parameters to prove the ownership of the e-coupon during the redemption phase, and $T_1 - T_2$ define the time interval for redeeming.

3.2 Security requirements

The protocol has to consider and guarantee the following security requirements:

- Owner authentication: Merchants should be able to verify the customer's e-coupon ownership during the redeem phase.
- Anonymity: The system has to provide full anonymity during the registration phase and redeem phase. The scheme allows a selective personal data disclosure during the issuance phase.
- Unforgeability: All the parts have to be able to verify if an e-coupon has been issued by an authorized issuer, and if the e-coupon has not been manipulated.
- Double-redeem protection: The system must provide protection about intents of reusing an e-coupon at the same merchant in an offline mode.
- Issuer non-repudiation: Once a valid e-coupon has been issued, issuer should not able to deny it.
- Non-transferability: Customers should not be able to share e-coupons.

3.3 Cryptographic background

We briefly review the cryptography techniques that will be used in our proposal.

ZKP: Zero-knowledge proof

The aim of a zero-knowledge proof is to prove the validity of a statement given to a verifier. This kind of proof is suitable to prove the possession of a secret without revealing the content. The Schnorr protocol [11] is an identification scheme based on discrete logarithms, that can be used as a interactive zero knowledge proof. Figure 1 shows in detail the three steps (commitment, challenge and response) involved in the protocol.

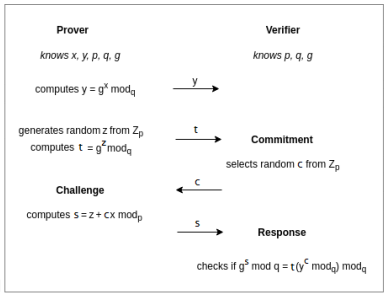

Fig. 1. Schnorr protocol

ABC: Attribute-Based Credential

The aim of an attribute-based credential is to provide an authentication mechanism based on a selective attributes method. In this way, every credential contains attributes that the user can either reveal or keep hidden. *Idemix* is an ABC protocol based on a Camenisch-Lysyanskaya (CL) signature scheme [12], that provides a solution for strong privacy-preserving authentication with a disclosure method of certified attributes. Table 1 defines the *Idemix* parameters involved in the protocol.

Table 1. *Idemix* protocol parameters

$\mathcal{S}$	CL-signature scheme, Credential structure, Context
x	Master secret key
m_{H_1}	Master secret key attribute
m_{H_i}	Credential/Proof hidden attributes
m_{K_i}	Credential/Proof known attributes
$common$	Public parameters
T	Commitment prove values (Aggregation of t-values)
T'	Commitment verify values (Aggregation of t-values)
c	Challenge prove
c'	Challenge verify
s_i	Response values (s-values)
$\mathcal{P}$	Proof = (c,s_i,common)
n_i	Random value

4 Proposal

This section is divided into two main subsections. The first subsection defines the prerequisites for the system set-up, and the second subsection explains in detail each step of the proposed protocol.

4.1 System Set-up

In this scheme, the entities $\mathcal{T}$ and $\mathcal{I}$ previously generated an asymmetric key pair. As a first step, $\mathcal{C}$ generates a master secret key x as m_{H_1} and enrolls a digital residence certificate with two fields: $PersonalID$ as m_{H_2} and $Zipcode$ as m_{H_3} in $\mathcal{T}$. Then, $\mathcal{T}$ generates a serial number attribute to identify the credential as m_{K_1}. Once the structure of the credential is defined, both parties agree to run the CL-signature scheme (Algorithm 1) to create the signature over the attributes specified in the credential structure. As a first step, $\mathcal{C}$ calls the function `CL.commit` to build the commitment for each of the hidden $\{m_{H_i}\}$ attributes using the system parameters retrieved from $\mathcal{T}$ $((n, S, Z, \{R_i\}_{i \in M}))$. As a result, C obtains U as an aggregation of commitments. Then, $\mathcal{C}$ sends U to I, and I has to call the function `CL.sign` to prepare the pre-signature with the aggregation of U and the known $\{m_{K_i}\}$ attributes. Finally, I sends the pre-signature to C, and C calls the function `CL.build` to compute the signature. Figure 2 shows the credential [(m_{H_1}, m_{H_2}, m_{H_3}, m_{K_1}),(CL Signature)].

Algorithm 1 Camenisch-Lysyanskaya signature

function CL.COMMIT($\{m_i\}_{i \in M_H}, (n, S, Z, \{R_i\}_{i \in M})$)
  $v' \leftarrow$ RANDOM( )
  $U \leftarrow S^{v'} \mod n$
  for each $i \in M_H$ **do**
    $U \leftarrow U \cdot R_i^{m_i} \mod n$
  return (U, v')
function CL.SIGN($U, \{m_i\}_{i \in M \setminus M_H}, (n, S, Z, \{R_i\}_{i \in M}), (p', q')$)
  $v'' \leftarrow$ RANDOM( )
  $U \leftarrow U \cdot S^{v''} \mod n$
  for each $i \in M \setminus M_H$ **do**
    $U \leftarrow U \cdot R_i^{m_i} \mod n$
  $Q \leftarrow Z \cdot U^{-1} \mod n$
  $e \leftarrow$ RANDOMPRIME( )
  $d \leftarrow e^{-1} \mod (p' \cdot q')$
  $A \leftarrow Q^d \mod n$
  return (A, e, v'')
function CL.BUILD($v', (A, e, v'')$)
  $v \leftarrow v' + v''$
  return (A, e, v)

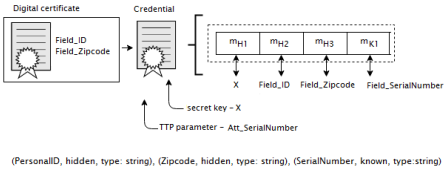

Fig. 2. *Idemix* credential structure

4.2 Phases

The phases of our system are: *Customer registration*, where the system checks the eligibility of the user and gets registered in the portal; *Issue*, that consists

of selecting an eligible promotion and disclose the required identity attributes for issuing an e-coupon; *Redeem*, where the customer redeems an e-coupon at a merchant. Table 2 defines the notation used in the description of the phases.

Table 2. Protocol parameters

x	Master secret key
y	Schnorr public key
z	Schnorr value
t	Schnorr commitment
c	Schnorr challenge
r	Schnorr response
$Policy$	e-coupon promotion policy
$Coupon$	$SN \mid I_1 \mid I_2 \mid T_1 \mid T_2$
$Sign(Coupon)$	$\mathcal{I}$'s signature on the e-coupon
$pk_{\mathcal{I}}$	Issuer public key
$pk_{\mathcal{T}}$	Trusted Third Party public key

Phase 1 *Customer registration*

As a first step, $\mathcal{C}$ contacts with $\mathcal{I}$ to prove her eligibility to be registered in the portal, following the protocol flow described in Figure 3. Then, $\mathcal{I}$ sends a random value n_i to $\mathcal{C}$, required to build a proof (Algorithm 2). First, $\mathcal{C}$ has to define the proof specification, which contains the disclosed and undisclosed attributes. That is, m_{K_1} as a known attribute, while m_{H_1}, m_{H_2}, m_{H_3} remain hidden. During the registration phase there is not a disclosure of hidden attributes. C calls the function `ProveCL.randomise` to generate a randomised signature, and the function `ProveCL.tvalues` to compute the commitment T over the aggregation of the undisclosed attributes, and the randomised signature (t-values). Next, C calls the `ProveCL.challenge` function to compute the challenge c, and computes the responses $\{s_i\}$ for every t-value. As a result, the following certified and signed $\mathcal{P}$ proof is generated: $\mathcal{P}$=(c,($\hat{e}, \hat{v}, \{\hat{m}_1\}, \{\hat{m}_2\}, \{\hat{m}_3\}$), $common$). The computed proof $\mathcal{P}$ will be used to demostrate the eligibility of $\mathcal{C}$ as a physical person who has certified attributes.

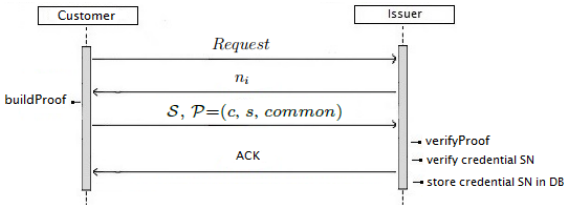

Fig. 3. Redeem protocol flow

$\mathcal{C}$ sends the generated proof $\mathcal{P}$ to $\mathcal{I}$, and $\mathcal{I}$ has to verify the proof $\mathcal{P}$ (Algorithm 3). Then, $\mathcal{I}$ calls the function `VerifyCL.tvalues` to validate the received $\mathcal{P}$, computing the aggregation of the randomised signature, the disclosed $\{m_{K_i}\}$ attributes and the undisclosed $\{m_{H_i}\}$ attributes. Then, I calls the function `VerifyCL.challenge` to compute the challenge c' to be compared against the received proof $\mathcal{P}$. If the proof $\mathcal{P}$ is accepted, $\mathcal{I}$ has to check the non-reusability of the credential, checking the credential serial number in an internal database. If the credential has not been used, $\mathcal{I}$ sends an ACK to $\mathcal{C}$. At this point, $\mathcal{C}$ is eligible to get registered in the portal. Moreover, $\mathcal{I}$ has to store the serial number of the credential in an internal database to prevent multiple registration using the same credential.

Algorithm 2 Build Proof

Protocol : $BuildProof$ [IN: $\{m_1$,$\mathcal{S}$,n_i,OUT: $\{\mathcal{P}\}$]
$ProveCL.randomise(\mathcal{S}) \rightarrow (e, v')$,common
$ProveCL.tvalues(\mathcal{S}$,common$) \rightarrow (\tilde{e}, \tilde{v}, \{\tilde{m}_i\}_{i \in m_H}), T$
$ProveCL.challenge$(context,$common$,T,$n_i) \rightarrow c$
$ProveCL.svalues(\mathcal{S}, (e, \tilde{e}, \tilde{v}, \{\tilde{m}_i\}_{i \in m_H}$,c$) \rightarrow s\text{-}values$
return $\mathcal{P} = (c$,$s\text{-}values$,$common)$

Algorithm 2.1 Prove CL

function ProveCL.randomise($[(A, e, v), (n, S, Z, \{R_i\}_{i \in M})]$)
    $r \leftarrow$ Random( )
    $A' \leftarrow A \cdot S^r \mod n$
    $Common \leftarrow A'$
    $v' \leftarrow v - e \cdot r$
    return $(e, v'), A'$
function ProveCL.tvalues($[\{m_i\}_{i \in m_H}, (n, S, Z, \{R_i\}_{i \in M})], (A')$)
    $\tilde{e} \leftarrow$ Random( )
    $\tilde{v} \leftarrow$ Random( )
    $\tilde{Z} \leftarrow A'^{\tilde{e}} \cdot S^{\tilde{v}} \mod n$
    for each $i \in m_H$ **do**
        $\tilde{m}_i \leftarrow$ Random( )
        $\tilde{Z} \leftarrow \tilde{Z} \cdot R_i^{\tilde{m}_i} \mod n$
    return $(\tilde{e}, \tilde{v}, \{\tilde{m}_i\}_{i \in m_H}), \tilde{Z}$
function ProveCL.challenge($context, common$,T, n_i)
    $c := H(context, common$,T, $n_i)$
    return $(\hat{e}, \hat{v}, \{\hat{m}_i\}_{i \in m_H})$
function ProveCL.svalues($\{m_i\}_{i \in m_H}, (e, \tilde{e}, \tilde{v}, \{\tilde{m}_i\}_{i \in m_H}), c$)
    $\hat{e} \leftarrow \tilde{e} + c \cdot e$
    $\hat{v} \leftarrow \tilde{v} + c \cdot v'$
    for each $i \in m_H$ **do**
        $\hat{m}_i \leftarrow \tilde{m}_i + c \cdot m_i$
    return $(\hat{e}, \hat{v}, \{\hat{m}_i\}_{i \in m_H})$

Algorithm 3 Verify Proof

Protocol : $VerifyProof$ [IN: $\{\mathcal{S}$, (c,s-values,common), $n_i\}$, OUT: { accept or reject $\mathcal{P}\}$]
$VerifyCL.tvalues(\mathcal{S},\mathcal{P}) \rightarrow T$'
$VerifyCL.challenge$(context,$common$,T',$n_i) \rightarrow c$'
return If $c \equiv c$' accept $\mathcal{P}$ or reject otherwise

Algorithm 3.1 Verify CL

function VerifyCL.tvalues($[\{m_i\}_{i \in m_K}), (n, S, Z, \{R_i\}_{i \in M}), [c, (\hat{e}, \hat{v}, \{\hat{m}_i\}_{i \in m_H}), A']$)
    $\hat{Z} \leftarrow Z^{-c} \cdot A'^{\hat{e}} \cdot S^{\hat{v}} \mod n$
    for each $i \in m_K$ **do**
        $\hat{Z} \leftarrow \hat{Z} \cdot R_i^{c \cdot m_i} \mod n$
    for each $i \in m_H$ **do**
        $\hat{Z} \leftarrow \hat{Z} \cdot R_i^{\hat{m}_i} \mod n$
    return $\hat{Z}$
function VerifyCL.challenge($context, common$,T$'$, n_i)
    $c' := H(context, common$,T, $n_i)$
    return c'

Phase 2 *Issue protocol*

When $\mathcal{C}$ wants to request a promotion from the portal, has to follow the protocol flow described in Figure 4. As a first step, she has to request the specified e-coupon using its SN. Next, $\mathcal{I}$ replays with the policy of the selected e-coupon and a fresh random number n_i to $\mathcal{C}$, required to build a proof (Algorithm 2). First, $\mathcal{C}$ has to use the proof specification of the received promotion policy, which defines the disclosed and undisclosed attributes to be proven. In our schema all the promotions have to be redeem by customers who live in a specified area. So, the zip code attribute m_{H_3} has to be disclosed as m_{K_2} to check the residence requirement. That is, m_{K_1}, m_{K_2} are known attributes, while m_{H_1}, m_{H_2} remain hidden. $\mathcal{C}$ calls the function `ProveCL.randomise` to generate a randomised signature, and the function `ProveCL.tvalues` to compute the commitment T over the aggregation of the undisclosed attributes and the randomised signature (t-values). Next, C calls the `ProveCL.challenge` function to compute the challenge c, and computes the responses $\{s_i\}$ (s-values) for every t-value. As a result, the following certified and signed proof $\mathcal{P}$ is generated: $\mathcal{P}$=(c,($\hat{e}, \hat{v}, \{\hat{m}_1\}, \{\hat{m}_2\}$), *common*). The computed proof $\mathcal{P}$ will be used to demostrate the residence area of $\mathcal{C}$, in particular the zip code.

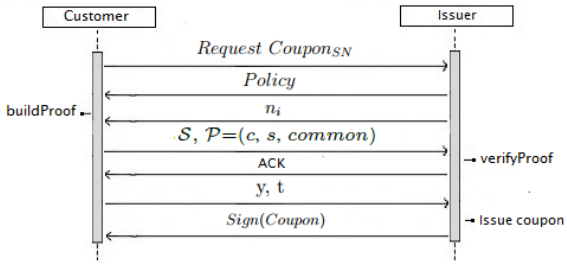

Fig. 4. Issue protocol flow

$\mathcal{C}$ sends the generated proof $\mathcal{P}$ to $\mathcal{I}$, and $\mathcal{I}$ has to verify the proof $\mathcal{P}$ (Algorithm 3). Then, $\mathcal{I}$ calls the function `VerifyCL.tvalues` to validate the received $\mathcal{P}$, computing the aggregation of the randomised signature, the disclosed $\{m_{K_i}\}$ attributes and the undisclosed $\{m_{H_i}\}$ attributes. Then, I calls the function `VerifyCL.challenge` to compute the challenge c' to be compared against the received proof $\mathcal{P}$. If the proof $\mathcal{P}$ is accepted, $\mathcal{I}$ sends an ACK to $\mathcal{C}$. Then, $\mathcal{C}$ contacts with $\mathcal{T}$ to obtain the Schnorr public parameters to compute the Schnorr public key y and the Schnorr commitment t. After that, both values are sent to $\mathcal{I}$. Next, $\mathcal{I}$ includes the Schnorr parameters obtained from C inside I_2, and prepares all the remaining data to be included inside the *Coupon* as the offer information and the time interval to be redeem. As the last step, $\mathcal{I}$ signs the issued *Coupon* and delivers *Sign*(Coupon) to $\mathcal{C}$.

Phase 3 *Redeem protocol*

During the redeem protocol (see Figure 5), $\mathcal{M}$ has to check that *Sign*(Coupon) presented by $\mathcal{C}$, is not a fake copy. Thus, $\mathcal{M}$ has to verify the signature using the $\mathcal{I}$'s $pk_{\mathcal{I}}$. If the verification is successful, $\mathcal{M}$ has to check if $\mathcal{C}$ is the legitimate

owner of $Coupon$. To do that, $\mathcal{M}$ starts the Schnorr identity protocol (see Section 3.3) to verify if $\mathcal{C}$ is able to answer a generated challenge using the information inside the I_2. $\mathcal{M}$ generates a time-variant random challenge c and sends it to $\mathcal{C}$. Then, $\mathcal{C}$ has to compute the response s and send it back to $\mathcal{M}$. As a result, $\mathcal{M}$ verifies the received response to know if $\mathcal{C}$ is eligible to redeem $Coupon$. If the verification is successful, $\mathcal{M}$ checks if the SN has not been used before, and if the time interval between T_1 and T_2 have not expired. If both verifications are successful, $\mathcal{M}$ redeem $Coupon$ and stores the SN in a local non-persistent database. In our scheme, portal offers e-coupons with a small limited lifetime period of use since the moment that $\mathcal{I}$ issues the e-coupons. As a last step, $\mathcal{M}$ sends an ACK to $\mathcal{C}$.

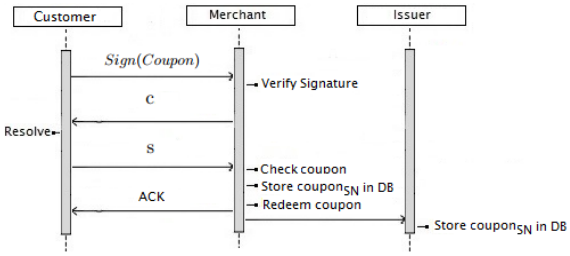

Fig. 5. Redeem protocol flow

5 Security Analysis

In this section we include a brief security analysis of the following requirements:

1. Owner authentication: During the issuance phase, $\mathcal{C}$ generates a public value y linked to the master secret key x. Only the legitime owner of x will be able to resolve the challenge-response step during the redemption phase.
2. Anonymity: During the registration phase, $\mathcal{C}$ sends the disclosed attributes to $\mathcal{I}$. In our scheme only the known attributes, as the serial number of the credential, was disclosed, while the other identity attributes remains hidden.
3. Unforgeability: $\mathcal{I}$ generates a digital signature $Sign$(Coupon) during the issuing phase. Any forged e-coupon that is not generated by $\mathcal{I}$, or it is modified will be detected during the redemption verification as illegal.
4. Double-redeem protection: In our scheme we only consider the double-redeem issue if $\mathcal{C}$ tries to use the e-coupon more than one time in the same $\mathcal{M}$, which is resolved by using a memory database to check the redeemed e-coupons. Also, there is a lifetime period to mitigate the impact if the same $\mathcal{C}$ tries to use the e-coupon in different $\mathcal{M}$. In addition, it is possible to use a global database between all the merchants.
5. Issuer non-repudiation: A credential proof can be verified with the corresponding $pk_{\mathcal{T}}$, so the $\mathcal{T}$ can not deny it. An e-coupon can be verified with the corresponding $pk_{\mathcal{I}}$, so the $\mathcal{I}$ can not deny it.
6. Non-transferability: If $\mathcal{C}$ shares the e-coupon, it has to expose its master secret key x to other customers to pass the challenge-response step during the redeem phase. Thus, the protocol must to discourage customers to reveal a valuable secret which is linked with the credential.

6 Conclusions and further work

Our proposal offers a portal to retrieve e-coupons, depending on the issuer promotion requirements. Customers can generate proofs from a digital certificate to prove identity requirements with a selective privacy mechanism. We offer a trusted registration mechanism providing anonymity. The design of the system has been performed taking into account security and privacy requirements described for e-coupons. As a future work, we want to extend this solution to other scenarios and build a prototype. Moreover, the security and the performance analysis will be presented in a formal way.

Acknowledgments. This work is partially financed by the European Social Fund and the Spanish Government under the projects TIN2014-54945-R and TIN2015-70054-REDC.

References

[1] Coupon Savings Report. 2016. `https://www.nchmarketing.com/nchpressreleases.aspx` (accessed 17 August 2017)
[2] Terms and Conditions - Christmas promotion 31 days. `https://app.mcdonalds.es/landing/legal/legal31DiasLocosNavidad.html` (accessed 17 August 2017)
[3] Kumar, M., Anand, R., Jhingran, A. and Mohan, R. Sales promotions on the Internet. Proceedings of the 3rd USENIX Workshop on Electronic Commerce, pp. 167–176, Boston, 1998.
[4] Jakobsson, M., Mackenzie, P.D. and Stern, J. P. Secure and lightweight advertising on the web. Journal of Computer and Telecommunications Networking, 31(11):1101–1109, 1999.
[5] Chang, C.C., Wu, C.C. and Lin, I.C. A Secure E-coupon System for Mobile Users. International Journal of Computer Science and Network Security, 6(1):273–280, 2006.
[6] Aigner, M., Dominikus, S., and Feldhofer, M. A system of secure virtual coupons using NFC technology. In Proceedings of IEEE International Conference on Pervasive Computing and Communication Workshops, pp. 362– 366, New York, 2007.
[7] Chang, C.C. and Sun, C.Y. A secure and efficient authentication scheme for e-coupon systems. Wireless Personal Communications, 77(4):2981–2996, 2014.
[8] Chang, C.C., Lin, I.C. and Chi, Y.L. Secure electronic coupons. Proceedings - 10th Asia Joint Conference on Information Security, AsiaJCIS 2015, pp. 104–109, 2015.
[9] Liu, W. Mu, Y., Yang, G. and Yu, Y. Efficient E-coupon systems with strong user privacy. Telecommunication Systems, pp. 695–708, 2017.
[10] Camenisch, J. , Modersheim, S. and Sommer, D. A formal model of identity mixer. Formal Methods for Industrial Critical Systems: 15th International Workshop, FMICS 2010, pp. 198–214, 2010.
[11] Schnorr, C.P. Efficient signature generation by smart cards. J. Cryptology, 4(3):161–174, 1991.
[12] Camenisch, J. and Lysyanskaya, A. A Signature Scheme with Efficient Protocols. Security in Communication Networks: 3rd International Conference, pp. 268–289, 2003.