Privacy-preserving webshopping with attributes

Brinda Hampiholi*, Gergely Alpár*†
*Radboud University, Nijmegen, The Netherlands
{brinda, g.alpar}@cs.ru.nl
†Open University, Heerlen, The Netherlands

***Abstract*—Online shopping is one of the most important applications on the Internet and it is one that has been steadily growing over the last decade. With increasing numbers of online shopping transactions there are also raising concerns over privacy and protection of the customer data collected by the webshops. This is why, we need privacy-preserving technologies for online shopping, in the interest of both users and businesses. To design shopping transactions where privacy is one of the main design considerations, we propose to use attributes. Attributes are pieces of data about an entity that are authenticated by some party. Attribute-based technologies go back more than a decade and they have mainly been used for access control, identity management and encryption. In this paper, however, we demonstrate that they can naturally be employed for various transactions in electronic commerce. In particular, we propose a cryptographic webshopping scheme based on attribute-based credentials. It preserves the functional and security properties required in practice for webshopping, while providing much more privacy for the purchasers. Privacy in this context is defined in terms of data minimization and unlinkability: Purchasers reveal exactly as much information as required in each transaction while leaving no traces that can be linked to their other transactions. In our scheme, a webshop does not learn a purchaser's identity, her financial information (e.g. credit card number or account number) or shipping details (e.g. house address). A bank that processes the payment does not learn the relationship between webshops and purchasers.**

***Index Terms*—webshopping, purchaser privacy, attribute-based credentials, data minimization, unlinkability**

I. Introduction

There has been a profound change in the way we shop in the last 20 years. In a traditional, brick-and-mortar shopping scenario, one walked into a store, collected items, went to the counter and paid by cash. Increasingly we obtain things online. Typically, webshopping includes registering personal details and authentication credentials (e.g. username and password), placing products in the 'shopping cart', logging in using the authentication method, paying with the involvement of a financial service (e.g. bank, PayPal), and finally, initiating product delivery. While in the traditional brick-and-mortar scenario, purchasers can remain anonymous and no personally identifiable information is stored about them, in the online scenario personal information is often registered by several companies: the webshop, the bank and the delivery company. Gradually, our shopping activities have become highly traceable and identifiable. Moreover, the related information is stored basically forever, with no transparency regarding where it is stored, how it is used and with which parties it is shared.

978-1-5386-1027-5/17/$31.00

This huge amount of information places great technical and legal responsibility on the afore-mentioned companies with regard to the protection of personal data. Over the time, the companies have also become victims of hacking[1]. Under certain data protection regulations e.g. European General Data Protection Regulation (GDPR)[2], the companies risk paying high penalties in the case of data violations on their part or failure to report data breaches. And it is not only security problems that arise. Because purchasers give away a lot of personal information, including permanent data (name, address, credit card number, etc.), dynamic data (e.g. purchased items), metadata (e.g. the bank knows the location and time of purchase) and derived data (combined information, behavioral patterns, etc.), data collection and processing may result in different kinds of privacy threats (exclusion, aggregation, secondary use, etc.). For the companies that process customers' personal data, strict compliance with data protection regulations becomes inevitable. Furthermore, some regulations such as the GDPR makes privacy by design and by default mandatory for such companies. This is why both companies and customers have a common interest in countering these security and privacy issues.

There are privacy-friendly approaches to webshopping. In contrast to the fully identifying webshopping paradigm, anonymous online marketplaces such as Silk Road, Agora maintain anonymity for both sellers and purchasers. However, they often become platforms for black markets [1]. In this paper we consider a significantly new approach, called *attribute-based webshopping*. This technology focusses on achieving the purchaser's privacy, while not hiding sellers and products from the public eye. As a result, we strike a balance between the overly-exposing and the overly-hiding paradigms in webshopping.

In our approach, a purchaser reveals the minimum information required to complete a shopping transaction to the various participants: a webshop, a bank and a delivery company. Instead of a delivery company that delivers a package to a purchaser's house, we use an anonymous locker facility from where a purchaser can pickup her delivered package. The main idea is that the participants in a shopping transaction together should learn as little as possible during each interaction with the purchaser. Whenever it is plausible, purchasers are not identified, and no linkable information (not even a pseudonym)

[1]E.g. https://securityintelligence.com/the-top-5-retail-breaches/

[2]More information about GDPR can be found on http://www.eugdpr.org.

is revealed about them. This can be achieved by using attribute-based credentials, which can cryptographically guarantee that the minimum data is revealed in each transaction. The data minimization principle states that "a data controller should limit the collection of personal information to what is directly relevant and necessary to accomplish a specified purpose."[3]. Therefore, in the proposed webshopping scheme, the participants provide tokens with minimal amount of data to each other, and these can be used in other steps with unlinkable zero-knowledge proofs and minimal data disclosure.

With security and data minimization in mind, we have designed our attribute-based shopping paradigm to comprise the following steps. Also see Figure 1.

1) Shopping Cart: The purchaser collects products in her cart and upon closing the cart, she receives the total sum to be paid and a cart identifier from the webshop.
2) Payment: The purchaser selects then a payment method – possibly independently of the webshop – and initiates contact with the financial service, which we call a bank for simplicity. The actual payment for the shopping cart is made at the bank, and as a result, the bank issues a credential to the purchaser.
3) Payment commit and packaging: The purchaser proves to the webshop that she has reserved the money at the bank to pay for the shopping cart, using the bank-issued credential. Then the webshop collects the products according to the cart, packages them and forwards them to a locker facility.
4) Package pickup and payment confirm: The purchaser picks up her package after proving to the locker that she is the rightful owner of the delivered cart items. Then she sends a payment-confirmation-token to the webshop (via the locker).
5) Payment Recovery: The webshop recovers the purchaser's payment from the bank by presenting the purchaser's payment-confirmation-token. Here we note that the bank cannot link the purchaser's payment and the webshop's payment-recovery steps.

In comparison with the widely used, identifying webshopping process, our approach has many benefits for the participants.

- Purchasers do not need to register and authenticate to webshops. Their interaction with companies are more privacy-friendly and they have more control over disclosure and dissemination of their personal data. Specifically, a webshop does not discover a purchaser's identity (e.g. name, date of birth, email address), her financial information (e.g. a credit card number or an account number) and shipping details (e.g. a house address) corresponding to her shopping cart. A bank that processes the payment does not discover the link between the webshops and the purchaser.
- Webshops do not collect and process personal data in our scheme, and consequently, they do not have to worry about data-protection regulations such as GDPR.
- Banks (or financial institutions) can benefit from the new scheme by being able to offer 'customer-privacy-while-shopping' as a new service in the form of anonymous payment credentials.

[3]European Data Protection Authority's Glossary: https://edps.europa.eu/data-protection/data-protection/glossary/d_en.

II. Preliminaries

In this section, we discuss the zero-knowledge proofs, attribute-based credentials and cryptographic primitives used in the issuance and verification of attribute-based credentials.

A. Zero-knowledge proofs

A zero-knowledge proof is a way in which a prover can convince a verifier that she possesses a secret value without giving away any useful information to the verifier. More precisely, the term zero-knowledge refers to the fact that whatever information a verifier learns from such a prover's proof could have been generated by the verifier himself without the assistance of the prover. For example, Schnorr's zero-knowledge proof [3] generated by a prover proves her knowledge of a discrete logarithm and it can be described as $\mathcal{PK}\{x : h = g^x\}$ where x is the discrete logarithm of the number h, group $\mathbb{G}$, its order q, generator g and the number h are known both to the prover and the verifier. Schnorr's honest-verifier interactive zero-knowledge protocol runs as follows.

- The prover commits to a random value $w \in_R \mathbb{Z}_q$ and sends the commitment $a = g^w \mod q$ to the verifier,
- the verifier sends a random challenge $c \in_R \mathbb{Z}_q$ to the prover and,
- the prover responds with $r = w + cx \mod q$ by using the random value w, challenge c and her secret key x.

The proof's verification equation $a \stackrel{?}{=} g^r h^{-c}$ (in $\mathbb{G}$) holds only if the prover knows the secret x and she computed the response correctly.

In practice, zero-knowledge proofs are often implemented using the Fiat–Shamir heuristic [4]. In this case the proof is not interactive, the challenge c is not provided by the verifier but computed as a hash value of the commitment a from the first step, context (group elements such as g, q etc.) and possibly some message msg. This turns a proof of knowledge into a signature scheme; e.g. Schnorr signature [3].

B. Attribute-based credentials

Attribute-based credentials (ABCs) are typically used for user identification and/or authentication [5], [6], [7] and can be considered a privacy-enhancing technology. An *attribute* is a characteristic or a qualification of a person. Attributes can either be identifying (e.g. name, date of birth, social security number) or non-identifying (e.g. a student, age>18, gender). An *attribute-based credential* is a cryptographic container of a few attributes that is signed by an authoritative party. All the attributes are bound to the user's secret key. In this paper, we consider a particular type of ABCs which are used in 'I Reveal My Attributes' (IRMA) technology [8]. The IRMA

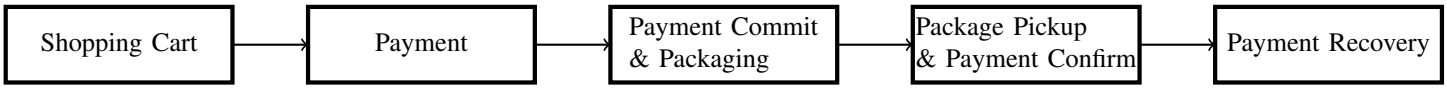

Fig. 1. Web-shopping process

project implements Idemix [9], an ABC scheme developed by IBM Zurich. In contrast to the Idemix cryptographic library, IRMA focusses only on the basic cryptographic features of ABCs such as blind issuance, selective disclosure proofs and composite proofs. This makes the IRMA implementation very simple to use and deploy for attribute-based use case scenarios.

The creation of an attribute-based credential is called *issuance*. This is an interactive cryptographic protocol in which an issuer authority and a receiver take part. As a result, the receiver is provided an ABC, the collection of attributes signed by the issuer. The issued credentials are bound to the user's secret key and are thus untransferable. The issuing procedure can involve special features such as blind issuance. See Section II-C for the description of ABC issuance types.

A user discloses attributes from an ABC to authenticate herself to a verifier. The showing of an ABC provides the *selective disclosure* capability. That is, only a subset of the attributes is revealed from a credential, not necessarily all of them. Moreover, such a showing is a zero-knowledge proof about all non-disclosed attributes. This means that the verifier learns no other information but the revealed attributes and the issuer's identity; conceptually, "these attributes hold for the user and this is asserted by ... issuer". See Section II-D for further explanation.

In addition to the selective disclosure of attributes, an ABC protects the privacy of the user through the following two cryptographic properties:

- Issuer unlinkability: Any information gathered during issuing cannot be used to link a verification of the credential to its issuance.
- Multi-show unlinkability: When a credential is verified multiple times, these verifications cannot be linked.

It provides not only strong privacy properties but also guarantees a high level of security. An ABC gurantees the following security properties:

- Authenticity: A credential originates from an issuer, and this issuer asserts that the attributes hold for the user by digitally signing this credential.
- Integrity: The issuer's digital signature on the credential ensures that the attributes contained in the credential have not been altered since they were issued.
- Non-transferability: A credential is bound to the user's secret key and thus it cannot be transferred to or used by anyone else.

C. Issuance of ABCs

a) Canonical issuance: An issuance protocol takes place between a user's device and an issuer in which a user obtains some personalized attributes as a credential from the issuer. In canonical issuance, both the issuer and the user know the values of all the issued attributes (before and after the issuance) except that the issuer does not learn the value of the user's secret key.

b) Blind issuance: A user can obtain an attribute from an issuer without the issuer learning the attribute value using blind credential issuance protocol, similar to message blinding in a blind signature [10]. The user proves that she actually knows the value of the hidden attribute by presenting a proof of knowledge to the issuer. Actually, all issuance instances are blind: The user always blinds her secret key (technically, one of her attributes in each of her credentials) from the issuer while the issuer issues other attributes following the user's proof of knowledge of this secret key. However, in some practical cases (e.g. e-voting, e-cash) not only the secret key but other attribute values may also be blinded during issuance. We use the blind issuance feature of ABCs in our webshopping protocol.

D. Selective disclosure of attributes from ABCs

While authenticating with ABCs, a user can choose to reveal only a few attributes from credentials. This is cryptographically realized by a selective disclosure (SD) protocol. An SD protocol involves two parts: (1) the user discloses a particular subset of attributes from the available credentials and, (2) she proves – using zero-knowledge techniques – to the verifier the validity of the hidden attributes within the credentials and that the disclosed attributes are indeed part of the credentials.

a) Interactive SD (ISD) proof: In an interactive SD proof, the verifier chooses a random challenge and interactively sends it to the user during the protocol.

b) Non-Interactive SD (NISD) proof: The Fiat–Shamir heuristic converts the interactive SD protocol to a non-interactive version. In NISD protocol, the user hashes the commitment, context (system parameters) and the verifier's nonce to create the unpredictable challenge. The verifier's nonce provides freshness to the resulting proof thereby preventing the replay attacks. The NISD protocol can be used as a signature on a message where the message could be anything i.e. text, a random number, hash of something etc. In this case, the user includes the message to be signed in the hash during the computation of the challenge. The meaning of this signature conceptually is as follows: "The user characterized by the attributes ... signs [anything]" [11]. We only use NISD proofs in the webshopping protocol described in this paper.

c) AND proofs: Multiple statements that can be proven using zero-knowledge proofs, can often be combined into a single AND zero-knowledge proof. For instance, in a proof $\mathcal{PK}\{x, y : h_1 = g_1^x \wedge h_2 = g_2^x g_3^y\}$ where g_1, g_2 are public,

both statements can be verified together while keeping both x and y hidden. Similarly, when attributes from different credentials are to be disclosed to a verifier, then SD proofs can be aggregated into a single SD proof.

E. Notation for ABCs

ABCs provide many flexible cryptographic operations. We introduce some notation for the techniques to make it easier to describe our protocols.

- A credential issuance is an operation carried out interactively by an issuer I and a user. This operation is denoted by

$$C^I \leftarrow Cred^I(a_1, .., a_l)$$

  where $a_1, .., a_l$ are the attributes in that credential. As a result, the user stores the credential C^I, I's signature over the attributes $a_1, .., a_l$ inside C^I.
- A credential with attributes is denoted as

$$C^I(a_1, [a_2], .., a_l)$$

  The notation with regard to attributes within a credential is explained in the Table I:

TABLE I
NOTATION USED FOR HANDLING ATTRIBUTES.

Attribute	Issuance	Selective disclosure
a_1	the value of a_1 is known to both the issuer and the user	the value of a_1 is revealed
$[a_2]$	the value of a_2 is known only to the user and not to the issuer	the value of a_2 is known only to the user, hidden from the verifier

- In this paper, the $Cred^I(..)$ operation always contains the secret key sk of the user (typically, the purchaser in online shopping). In principle, it is $Cred^I([sk], a_1, .., a_l)$, we leave it implicit in our notation.
- A credential verification is a selective disclosure operation carried out by the user.
  - A selective disclosure proof is a non-interactive (NISD) proof and it is denoted as

$$\mathcal{SD}\{C^I(a_1, [a_2])\}(nonce)$$

    where the attributes a_1 is disclosed and a_2 is hidden from the verifier. In transactions, a non-interactive proof is always created using an unpredictable nonce provided by the verifier. This is defined by $nonce$ in the above $\mathcal{SD}$ proof.
  - A selective disclosure proof can be used as the user's signature on a message msg and it is denoted by

$$\mathcal{SD}\{C^I(a_1, [a_2])\}(nonce, msg)$$

III. Attribute-based webshopping

A typical shopping transaction within the attribute-based webshopping scheme consists of four participants: Purchaser, Webshop, Bank and Locker. Figure 1 shows the main stages in an attribute-based shopping transaction.

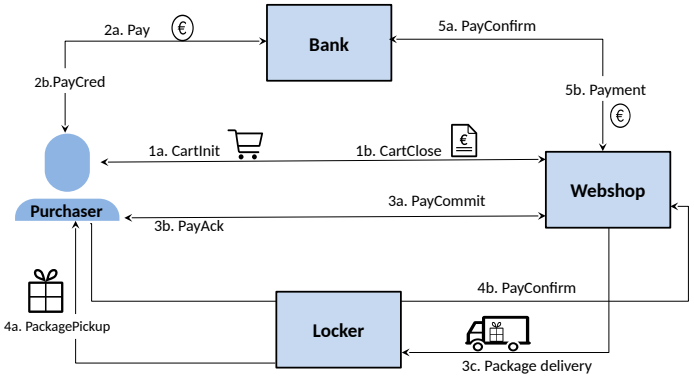

Fig. 2. Overview of the steps carried out in a shopping transaction under our webshopping scheme.

Figure 2 outlines the steps carried out between the participants in a shopping transaction under our webshopping scheme. First, the Purchaser anonymously initializes a cart and fills it up at the Webshop (CartInit). When the cart is ready, the shop issues a credential (CartClose). Second, using some out-of-the-scope payment method (cash, bank transfer, credit card, PayPal, Bitcoin, etc.) and some (hidden) information from the CartClose credential, the Purchaser receives a credential (PayCred) from the Bank that certifies that she has paid. Third, the Purchaser goes back to the Webshop, proves that the cart has been paid for at the Bank and gives a proof (PayCommit). If the proof is correct, then the Webshop issues an acknowledgement credential (PayAck) to the Purchaser that enables her to pick up the package later at the Locker. Fourth, the Purchaser proves to the Locker that she is the cart owner and the Webshop has accepted her PayCommit, using PayAck credential and picks up her package from the Locker. Then she confirms her payment to the Webshop with a PayConfirm token. Fifth, the Webshop presents the PayConfirm token to the Bank and claims its payment for the Purchaser's shopping cart. (Note that the Webshop has not learnt the Purchaser's identity and as the PayConfirm token does not identify the Purchaser to the Bank, the Bank does not learn the relation between the Purchaser and the Webshop.)

In our scheme, we follow the data-minimization principle while deciding which attributes are necessary to complete each stage of a shopping transaction. We make our own policy that defines which data is collected from which party during each step of the transaction. For instance, we make the webshop to issue a cart identifier attribute during the shopping phase that allows the webshop to keep track of the purchaser's items and the statuses of the corresponding payment and delivery phases. Here we emphasize that attributes allow the scheme implementors to define and execute their own policies

regarding the nature of the attributes, how they are issued (canonically or blindly) and disclosed (all, few or none) from the credentials. Thus, ABCs technology provides flexibility for defining contextual policies over them to any extent but we strive to achieve the minimal set of attributes to be issued and disclosed in our webshopping scheme. Furthermore, we stress that irrespective of the revealed or unrevealed attributes, ABCs provide a high degree of reliability about the binding of a credential to its holder because of the secret key (that is never revealed) and the cryptographic robustness (see Section II-B for details). That's why they are especially suitable to achieve security and privacy simultaneously. In the rest of this section we give the technical details of the scheme.

A. Assumptions

Following are the assumptions that we make in our attribute-based webshopping scheme with regard to the participants and the communications that take place between them over the course of a shopping transaction.

- All communications between a purchaser and other participants happen over an encrypted anonymous channel.
- All communications are purchaser-controlled i.e., each communication in a webshopping transaction is initiated by the purchaser.
- All ABC issuance and selective disclosure instances are implemented on the server and the client sides properly, i.e., security and privacy properties of ABCs are ensured.
- A suitable public-key infrastructure is assumed to be in place for the companies involved in the scheme. In particular, the bank and the webshop have public-private key-pairs (required for the credential issuance) for the ABC system. For instance, the webshop verifies and accepts the credentials issued by the bank.
- The browser cookies during a shopping transaction are session-bound, and once the session is over, the purchaser's browser deletes all the cookies set by the webshop.
- The bank is semi-trusted (a.k.a. honest but curious), i.e., it follows the protocol but it wishes to learn as much as possible about the purchaser's shopping (where, when, what).
- The webshop is semi-trusted; in particular, it honestly follows the protocol but may be curious to find out the purchaser's identity.
- The webshop may control the anonymous locker facility.
- The prices of the items at the webshop are fixed i.e., the webshop will not charge the purchasers differently for the same product.
- An external adversary does not have full access to the internal states and databases of more than one participant at any point in time.

B. Table of notation

Table II briefly summarizes the notation for the webshopping participants and the most important objects.

Symbol	Interpretation
$\mathcal{P}, \mathcal{W}, \mathcal{L}, \mathcal{B}$	Purchaser, Webshop, Locker, Bank respectively
$\mathcal{W}_{id}, \mathcal{L}_{id}$	Identifiers for $\mathcal{W}$, $\mathcal{L}$ respectively
ID_c	Cart identifier
σ	Total price of all items in the cart
$sk_{\mathcal{P}}$	Secret key of $\mathcal{P}$ associated with its ABCs
$pk_{\mathcal{W}}, sk_{\mathcal{W}}$	Public, private signing keys of $\mathcal{W}$
$pk_{\mathcal{B}}, sk_{\mathcal{B}}$	Public, private signing keys of $\mathcal{B}$
$C_1^{\mathcal{W}}, C_2^{\mathcal{W}}$	Cart, PayAck credentials issued by $\mathcal{W}$ to $\mathcal{P}$
$C^{\mathcal{B}}$	Payment-credential issued by $\mathcal{B}$ to $\mathcal{P}$
$C_3^{\mathcal{W}}, C_4^{\mathcal{W}}, C_5^{\mathcal{W}}$	Return, Voucher, Refund credentials issued by $\mathcal{W}$ to $\mathcal{P}$
$\mathcal{SD}$	Selective disclosure proof

TABLE II
NOTATION USED IN OUR WEBSHOPPING PROTOCOL.

C. Shopping

In this and the following subsections we describe all five steps of the proposed scheme depicted in the high-level Figure 2 and in the technically more detailed Figure 3. Each rectangular box in Figure 3 represents a secure session between any two participants within a shopping transaction.

The shopping phase consists of the shopping cart initialization, addition of items to the cart and finally closing of the cart. The communication between the purchaser $\mathcal{P}$ and the webshop $\mathcal{W}$ that takes place in this phase is described in the following steps and in **1. Cart** box in Figure 3.

- `CartInit`: $\mathcal{W}$ assigns a cart identifier ID_c to $\mathcal{P}$. ID_c can be viewed as a session-specific pseudonym for $\mathcal{P}$.
- $\mathcal{P}$ browses through the items on $\mathcal{W}$'s website, makes her choice and adds items to the cart ID_c. Let the total price of all the added items in the cart be σ.
- `CartClose`: After all the items have been added to the cart, $\mathcal{P}$ closes the cart and then $\mathcal{W}$ issues a cart-credential to $\mathcal{P}$:

$$C_1^{\mathcal{W}} \leftarrow Cred^{\mathcal{W}}(ID_c, \sigma)$$

D. Payment

Payment is the next stage of the scheme, in which $\mathcal{P}$ essentially reserves money at the bank $\mathcal{B}$ (within a separate secure session) to pay for the shopping cart at $\mathcal{W}$ and gets a credential from the bank in return. This phase is separately carried out by $\mathcal{P}$, independent of $\mathcal{W}$, to prevent leaking $\mathcal{P}$'s financial information (e.g. bank account number) to $\mathcal{W}$. The steps carried out by $\mathcal{P}$ and $\mathcal{B}$ in this phase are given below. Also see **2. Payment** box in Figure 3.

- $\mathcal{P}$ requests $\mathcal{B}$ to debit an amount σ from her account in exchange for a payment-credential. The actual payment method is out of scope in this study; that is, the purchaser can pay in any way that $\mathcal{B}$ accepts, including cash, bank transfer, debit card/credit card payment, PayPal or Bitcoin. Then $\mathcal{P}$ blinds the values of $\mathcal{W}$'s identity $\mathcal{W}_{id}$ and $\mathcal{W}$-assigned cart identifier ID_c and sends them to $\mathcal{B}$.
- `PayCred`: $\mathcal{B}$ processes $\mathcal{P}$'s request, debits σ from $\mathcal{P}$'s account and issues a payment-credential to $\mathcal{P}$:

$$C^{\mathcal{B}} \leftarrow Cred^{\mathcal{B}}([\mathcal{W}_{id}], [ID_c], \sigma)$$

where only attribute value σ is visible to $\mathcal{B}$. Blinding the attributes $\mathcal{W}_{id}$ and ID_c from $\mathcal{B}$ during the issuance of `PayCred` aims to prevent $\mathcal{B}$ from linking the issuance of a payment credential from its usage. A payment credential is bound to a purchaser's secret key, her shopping cart and the webshop and thus the credential is non-transferable and specific to a shopping transaction. Further, it is similar to a 'dinner cheque' or 'present cheque' that one can spend but can never get back the money.

E. Payment Commit and Packaging

Obviously, as a next step $\mathcal{P}$ should send a payment token involving the payment credential to $\mathcal{W}$ which would enable $\mathcal{W}$ to get the cart amount reimbursed by $\mathcal{B}$. However, we consider a threat scenario in which $\mathcal{W}$ claims and gets its payment from $\mathcal{B}$ immediately after receiving the payment token from $\mathcal{P}$ and aborts the protocol without sending the cart items to $\mathcal{P}$. To counter this threat, we include a payment commit phase in which $\mathcal{P}$ just commits to pay $\mathcal{W}$ for her shopping cart (not finalize the payment yet) as follows. Also see **3. Payment Commit** box in Figure 3.

- `PayCommit`: $\mathcal{P}$ proves that she is the owner of the cart ID_c and she has got the payment-credential from the bank to pay $\mathcal{W}$ for the cart, by creating a selective disclosure proof:

$$\mathcal{SD}\left\{C_1^{\mathcal{W}}(ID_c, \sigma) \wedge C^{\mathcal{B}}(\mathcal{W}_{id}, ID_c, \sigma)\right\}\left(n_1, msg_1\right)$$

  where $\mathcal{P}$ discloses all the attributes from both $C_1^{\mathcal{W}}$ and $C^{\mathcal{B}}$ credentials. In addition to the attributes, the input to this proof includes $\mathcal{W}$'s nonce n_1 and $msg_1 =$ "$PayCommit$". Basically this $\mathcal{SD}$ proof is $\mathcal{P}$'s signature on n_1 that adds freshness to the proof and "$PayCommit$" message that adds context to it.

The packaging of $\mathcal{P}$'s cart items follows the payment commit step. After the `PayCommit` proof is verified successfully, $\mathcal{P}$ and $\mathcal{W}$ continue their communication as follows.

- $\mathcal{P}$ sends her choice of locker location $\mathcal{L}_{id}$ to $\mathcal{W}$.
- `PayAck`: To indicate that it acknowledges the purchaser's payment-commitment, $\mathcal{W}$ issues a second credential to $\mathcal{P}$:

$$C_2^{\mathcal{W}} \leftarrow Cred^{\mathcal{W}}(\mathcal{L}_{id}, ID_c, D_{date})$$

  where $\mathcal{L}_{id}$ attribute denotes the identifier of the locker to which $\mathcal{P}$'s package will be delivered, ID_c attribute is the cart identifier that binds the shopping cart and the package and D_{date} is the delivery date on or after which $\mathcal{P}$ can pickup her package from the locker.
- Then $\mathcal{W}$ transports $\mathcal{P}$'s package to the locker facility $\mathcal{L}_{id}$.

F. Package pickup and Payment Confirmation

To pickup the package from the assigned locker, $\mathcal{P}$ has to first prove to the locker $\mathcal{L}$ that she is the cart owner and $\mathcal{W}$ has acknowledged her payment commit. After picking up her package from the locker, $\mathcal{P}$ sends the cart's payment confirmation to $\mathcal{W}$ through the locker. The above actions are described in more technical detail below and also in **4. Package Pickup** box in Figure 3.

- `ProvePayAck`: $\mathcal{P}$ proves that she holds a valid `PayAck` credential issued by the webshop $\mathcal{W}$ by creating the following proof:

$$\mathcal{SD}\left\{C_2^{\mathcal{W}}(\mathcal{L}_{id}, ID_c, D_{date})\right\}\left(n_2, msg_2\right)$$

  where $\mathcal{P}$ discloses all the attributes from $C_2^{\mathcal{W}}$ to $\mathcal{L}$. `ProvePayAck` proof is considered as $\mathcal{P}$'s signature on $\mathcal{L}$'s nonce n_2 and $msg_2 =$ "$ProvePayAck$".
- If this proof is correct, then $\mathcal{L}$ opens the locker for $\mathcal{P}$.
- $\mathcal{P}$ picks up her package and creates a `PayConfirm` proof:

$$\mathcal{SD}\left\{C^{\mathcal{B}}(\mathcal{W}_{id}, ID_c, \sigma)\right\}\left(n_3, msg_3\right)$$

  in which she discloses all the attributes from the `PayCred` $C^{\mathcal{B}}$ and sends it to $\mathcal{W}$ via the locker $\mathcal{L}$. `PayConfirm` proof is essentially $\mathcal{P}$'s signature on $\mathcal{L}$'s nonce n_3 and $msg_3 =$ "$PayConfirm$". The disclosed attribute σ in this proof indicates the amount that the bank $\mathcal{B}$ has to pay $\mathcal{W}$ in the payment recovery phase.

In practice, `ProvePayAck` and `PayConfirm` proofs can be automatically sent from $\mathcal{P}$ to $\mathcal{L}$ by introducing a predetermined delay, say, five minutes between them. This delay allows $\mathcal{P}$ to ensure that her package has indeed been delivered at the locker and if not, she can block the `PayConfirm` proof from reaching $\mathcal{W}$. This type of transferring both the proofs to $\mathcal{L}$ could be the default action that would be carried out during the pickup phase, unless $\mathcal{P}$ explicitly blocks the second proof in the case of some dissatisfaction e.g. no package is found inside the locker. If $\mathcal{P}$ is dissatisfied with the product at a later time, she can return the product in exchange for a replacement, a voucher or cash according to the procedure described in Section III-H, within a return period stipulated by the webshop, say, two weeks from the delivery date.

Further, it is upto the implementors of the scheme to trust either $\mathcal{P}$'s authenticating device or the locker to buffer the `PayConfirm` proof during the delay period before passing it on to the webshop $\mathcal{W}$. We emphasize that the order of the messages in the pickup phase can be altered based on the trust assumptions made by the implementors because it only affects the fairness in the exchange of money and the goods between the purchaser and the webshop, but not the security or the cryptographic robustness of our webshopping protocol.

G. Payment recovery

$\mathcal{W}$ approaches $\mathcal{B}$ in a separate session with $\mathcal{P}$-provided `PayConfirm` proof to redeem the payment for the shopping cart. $\mathcal{B}$ maintains a double-spend database that logs all the previous transaction identifiers corresponding to the payment recovery claims. The interaction between $\mathcal{W}$ and $\mathcal{B}$ is detailed in the following steps and in **5. Payment Recovery** box in Figure 3.

- $\mathcal{W}$ authenticates to $\mathcal{B}$ in some form (e.g. logging in with its bank-credentials) independent of this scheme and then presents $\mathcal{P}$'s `PayConfirm` proof to $\mathcal{B}$.
- $\mathcal{B}$ checks if
  - $\mathcal{W}$ claimed identity during authentication and the payee's identity attribute $\mathcal{W}_{id}$ are the same;
  - transaction identifier ID_c (disclosed from `PayConfirm`)) is not present in its double-spend database, and
  - `PayConfirm` proof verifies correctly i.e., it is a valid signature involving $C^{\mathcal{B}}$ on the message "$PayConfirm$" (and nonce n_3).

  If all the three checks are successful, $\mathcal{B}$ gives or transfers the money worth σ amount to $\mathcal{W}$.
- $\mathcal{B}$ stores the ID_c attribute in its double-spend database so that it can verify if the same proof is presented to it for the second time.

Note that $\mathcal{B}$ cannot link an ID_c to the payment phase. Now we elaborate on the payment and recovery phases from the bank's perspective. It is kept implicit in our scheme that the number of outstanding payments at the bank is always equal to the number of payment recovery claims by the potential payees (i.e. webshops). The bank $\mathcal{B}$ maintains a pool (multiset) of outstanding payments following the issuance of payment credentials to its customers. During a payment recovery, $\mathcal{B}$ sees the disclosed attributes – the payee's identifier $\mathcal{W}_{id}$ and the transaction's identifier ID_c – belonging to a payment claim i.e., a `PayConfirm` proof for the first time. As the bank does not know these identifiers earlier to the submission of the payee's claim, it cannot link the payment and payment recovery stages which is equivalent to saying that $\mathcal{B}$ cannot link the issuance and showing of the payment-credential (ABCs' issuer unlinkability feature – see Section II-B for explanation). Let us consider an example in which the bank's pool has ten outstanding payments of 50 euros each. When a payee claims for the recovery of 50 euros and provides a valid `PayConfirm` proof, the bank reimburses the payee with any one of the ten outstanding payments from its pool. The bank does not know which of its customer's 50-euro payment is going to that particular payee.

H. Return-Replacement-Refund scenario

In this subsection, we discuss a scenario in which a purchaser wishes to return the delivered product back to the webshop and requests the webshop for a product replacement or refund in the form of a gift voucher or cash. The return scenario is not included in Figure 3 as we do not primarily focus on it in our webshopping scheme. The communication between th $\mathcal{P}$ and $\mathcal{W}$ in such a situation is described in the following steps.

1) $\mathcal{P}$ visits the returns section of $\mathcal{W}$'s website, submits a return request with the following proof using the `PayAck` credential:

$$\mathcal{SD}\Big\{C_2^{\mathcal{W}}(\mathcal{L}_{id}, ID_c, D_{date})\Big\}\big(n_4, msg_4\big)$$

This NISD proof is $\mathcal{P}$'s signature on $\mathcal{W}$'s nonce n_4 and $msg_4 =$ "$ReturnRequest$". It proves to $\mathcal{W}$ that $\mathcal{P}$ had committed to pay for a shopping cart with identifier ID_c and had received the package at locker location $\mathcal{L}_{id}$. $\mathcal{W}$ can also track the `PayConfirm` token corresponding to ID_c.

2) If the proof verifies successfully and if $\mathcal{W}$ approves $\mathcal{P}$'s request for returning the delivered product, then it issues a return-credential to $\mathcal{P}$:

$$C_3^{\mathcal{W}} \leftarrow Cred^{\mathcal{W}}(\mathcal{L}_{id}, ID_c, R_{date}, opt)$$

where $\mathcal{L}_{id}$ is the locker's identifier (same locker as in first package pickup step unless $\mathcal{P}$ explictly mentions a different locker in her return request), ID_c is the original shopping cart identifier, R_{date} is the date of return and opt is the option attribute whose value could be either 'replacement', 'exchange for a voucher' or 'cash refund'. Then $\mathcal{W}$ notifies $\mathcal{P}$ to go to the locker facility and deposit the package to be returned.

3) $\mathcal{P}$ shows this credential to the locker system $\mathcal{L}$ and deposits the package in the locker.
4) $\mathcal{W}$ retrieves the package, checks the returned product's condition or its defect (in case a defective item is returned) and then does either of the following based on the value of opt attribute:
   a) replaces the returned product at the locker. $\mathcal{P}$ can pick it up at the locker after proving the attributes from $C_3^{\mathcal{W}}$ credential with an $\mathcal{SD}$ proof:

$$\mathcal{SD}\Big\{C_3^{\mathcal{W}}(\mathcal{L}_{id}, ID_c, R_{date}, opt = \text{“}replacement\text{”})\Big\}\big(n_5, msg_5\big)$$

   where n_5 is a nonce provided by the locker and $msg_5 =$ "$ReplacedProduct$".
   b) issues a voucher-credential:

$$C_4^{\mathcal{W}} \leftarrow Cred^{\mathcal{W}}(\sigma, V_{id}, V_{val}, opt = \text{“}voucher\text{”})$$

   where σ is the voucher's worth, V_{id} is the voucher identifier and V_{val} is the voucher validity. V_{id} attribute is randomly chosen by $\mathcal{P}$ and blindly issued to $\mathcal{P}$ by $\mathcal{W}$. It is included to prevent the double spending of the voucher by $\mathcal{P}$ at $\mathcal{W}$. $\mathcal{P}$ can use this voucher-credential to purchase some other item worth σ amount at $\mathcal{W}$ within the voucher's validity period by carrying out `VoucherCommit` and `VoucherConfirm` steps similar to the `PayCommit` and `PayConfirm`. The only difference is that, in the payment-by-voucher scenario, the bank $\mathcal{B}$ is not involved; $\mathcal{W}$ checks if the presented voucher identifier is present in its voucher-double-spend database and if not, $\mathcal{W}$ accepts the payment in the form of a voucher. $\mathcal{W}$ cannot link a voucher's issuance and its use by a specific purchaser.
   c) issues a cash-refund-credential:

$$C_5^{\mathcal{W}} \leftarrow Cred^{\mathcal{W}}(\sigma, ID_c, opt = \text{“}refund\text{”}, \mathcal{W}_{id}, \mathcal{W}_{ac})$$

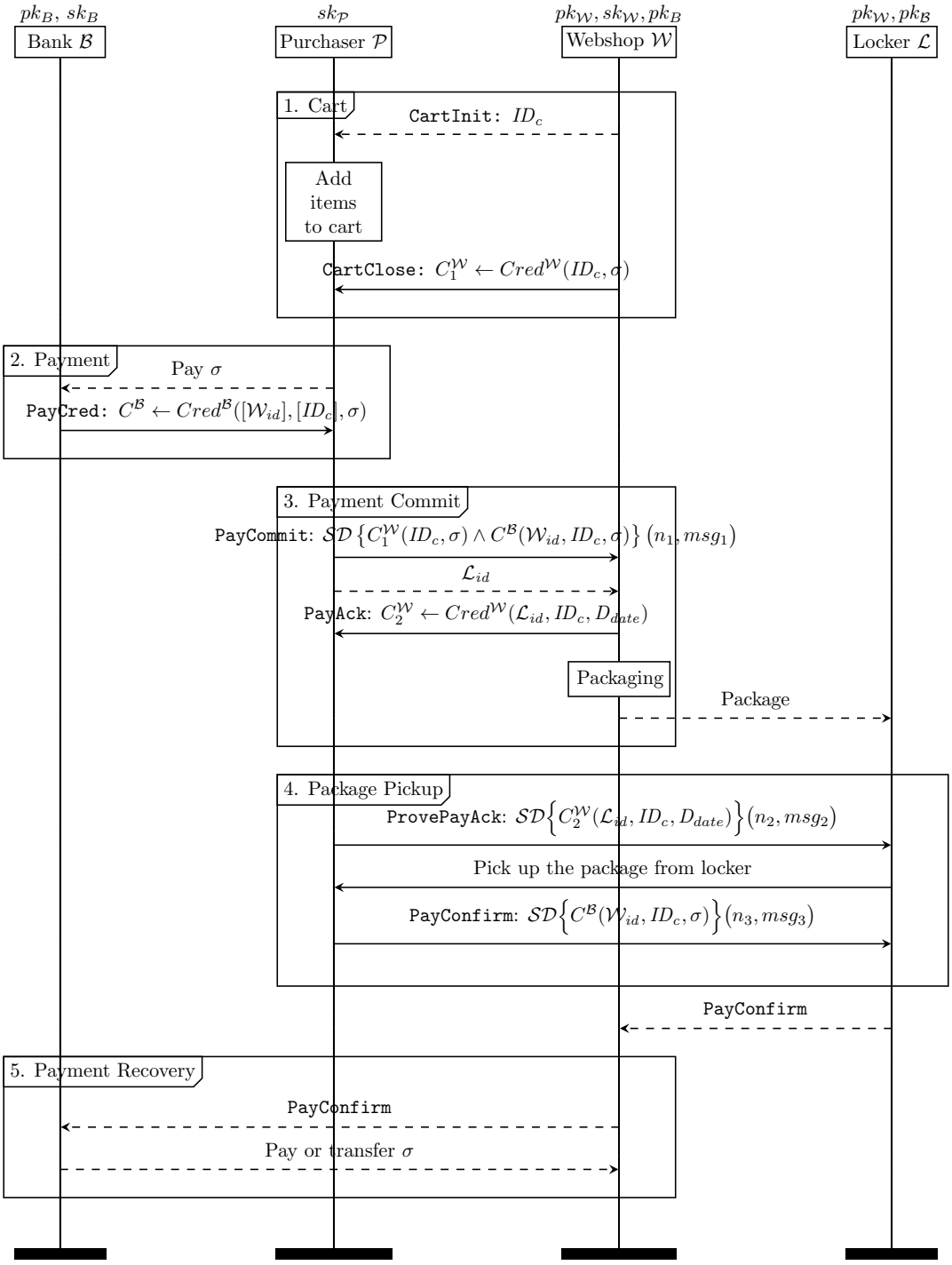

Fig. 3. Attribute-based online shopping process. (Each rectangular box represents a secure session between any two participants within a shopping transaction; ABC transactions are denoted by $\longrightarrow$, other transactions by $- - \rightarrow$)

using which $\mathcal{P}$ can prove to the bank $\mathcal{B}$ that she has rightfully received the cash refund from the webshop $\mathcal{W}$. The bank transfers σ amount from the $\mathcal{W}$'s account $\mathcal{W}_{ac}$ to the purchaser's account.

Options a) and b) maintain the anonymity of the purchaser towards the webshop and the bank but option c) reveals to the bank that the purchaser had previously bought something worth σ amount at the webshop. If the webshop colludes with the bank, then it can deanonymize the purchaser in the case of cash refund.

IV. Protocol Analysis

Under the assumptions stated in Section III-A, we analyze our attributes-based webshopping scheme and list the possible attack scenarios in this section.

A. Privacy attacks

Due to the use of secure, encrypted and anonymous channel, an external adversary cannot gather much information about the purchaser $\mathcal{P}$'s shopping session nor can it see any data exchanged between $\mathcal{P}$ and the webshop $\mathcal{W}$. Two instances of our protocol observed by an external adversary are indistinguishable from each other. The potential attacks against $\mathcal{P}$'s privacy and the ways in which they are countered in our scheme are described below.

1) The webshop $\mathcal{W}$ colludes with the bank $\mathcal{B}$ to link the shopping and the payment made by $\mathcal{P}$.
   Solution: Our protocol does not allow this as $\mathcal{B}$ blindly issues the attributes in the payment-credential. $\mathcal{B}$ cannot link a `PayConfirm` token to a specific $\mathcal{P}$'s payment-credential based on the value of ID_c because $\mathcal{B}$ had not known the value of ID_c during the `PayCred` issuance.
2) Based on the transcripts of two shopping transactions, $\mathcal{W}$ attempts to trace a purchaser or find out if the same purchaser was involved in both the transactions.
   Solution: Linking of two transactions to a particular $\mathcal{P}$ by $\mathcal{W}$ is not possible due to the multi-show unlinkability feature of ABCs (See Section II-B for details.).
3) $\mathcal{B}$ could link $\mathcal{P}$'s identity with the cart price σ, the time of payment-credential `PayCred` request by $\mathcal{P}$ and the time when $\mathcal{W}$ approaches $\mathcal{B}$ with a `PayConfirm` token to recover this payment.
   Solution: This attack is difficult to carry out when the anonymity set for the total cart amount is considerably large. One way to increase the anonymity set for the amount is by rounding up the cart prices to their next multiple of 10. In this case many purchasers would receive payment credentials of similar amount and it will become hard for $\mathcal{B}$ to correlate a `PayConfirm` token with a specific $\mathcal{P}$ purely based on σ. In the client side implementation, purchasers could be given two choices: whether to accept the total cart price as it is or to anonymize cart price by rounding it up.

B. Security attacks

1) $\mathcal{P}$ tries to create new shopping or payment-acknowledgement credentials and claims for products from $\mathcal{W}$ with fake credentials.
   Solution: Relying on the unforgeability of the ABCs, this attack is not possible because $\mathcal{P}$ does not have $\mathcal{W}$'s secret key.
2) $\mathcal{P}$ steals someone's credentials and tries to prove as her own.
   Solution: This attack is not possible because all attribute-based credentials are associated to a specific user's secret key and without that secret key, $\mathcal{P}$ cannot create a proof for that credential and thus cannot succeed in authenticating with somebody else's credentials. The way the secret key is bound to the user is out-of-scope here as it is part of the ABC implementation.
3) $\mathcal{P}$ tries to create a new `PayCred` without $\mathcal{B}$'s involvement.
   Solution: This attack is not possible because `PayCred` credential is signed by $\mathcal{B}$ using its own private key and $\mathcal{P}$ does not have the $\mathcal{B}$'s private key to create such a credential.
4) $\mathcal{P}$ does not commit to pay the correct amount for her shopping cart.
   Solution: In our protocol, $\mathcal{W}$ does not proceed to the packaging stage without receiving a valid payment commitment i.e. `PayCommit` proof from $\mathcal{P}$.
5) $\mathcal{P}$ uses an old `Paycred` to pay for the current shopping session – double-spending scenario.
   Solution: This attack is not possible as a typical `PayCred` includes shopping cart identifier ID_c. $\mathcal{W}$ can detect double-spending if it sees same ID_c in consecutive `PayCommit` proofs (involving `PayCred` credentials) presented by $\mathcal{P}$. Thus, a `PayCred` can be used only to pay for a particular shopping transaction (i.e. a payment-credential is specific to a cart, its contents and the webshop).
6) $\mathcal{P}$ denies receiving the package.
   Solution: $\mathcal{P}$ provides a proof to $\mathcal{L}$ that she holds the correct cart identifier ID_c from the $\mathcal{W}$-issued `PayAck` credential before picking up her package and then sends a payment-confirmation – `PayConfirm` proof. These proofs can be considered as $\mathcal{P}$'s signature at the time of package delivery which ensures that $\mathcal{P}$ cannot deny receiving the package later.
7) $\mathcal{W}$ tries to modify the amount in $\mathcal{P}$'s `PayConfirm` proof.
   Solution: This attack is not possible because the proof will not be valid if any change is made to it and $\mathcal{W}$ will need $\mathcal{B}$'s and the $\mathcal{P}$'s private keys for re-signing the `PayCred` after changing. Here we rely on the ABC's integrity guarantee.
8) $\mathcal{W}$ tries to withdraw twice by repeatedly presenting the same `PayConfirm` proof from $\mathcal{P}$ at $\mathcal{B}$ during payment recovery.

Solution: This attack is not possible because the $\mathcal{B}$ stores the `PayConfirm` proof (Each `PayConfirm` contains the cart identifier which is unique for a shopping session) and if it is presented for the second time, then $\mathcal{B}$ checks it against its records and rejects it.

9) $\mathcal{W}$ sends a different price to $\mathcal{P}$ as a part of the cart-credential in the shopping phase.
Solution: This inconsistency would be easily detected by $\mathcal{P}$, when she compares the product price attribute as a part of $\mathcal{W}$-issued cart-credential with the original product price present on the $\mathcal{W}$'s website.
10) $\mathcal{W}$ redeems cash from $\mathcal{B}$ as soon as it receives the `PayCommit` proof from $\mathcal{P}$ and does not send the ordered items to $\mathcal{P}$.
Solution: This attack is not possible as $\mathcal{W}$ cannot use `PayCommit` proof to redeem its money from the bank. It needs `PayConfirm` token to recover the cart amount which it gets only after the $\mathcal{P}$ has picked up he package from the locker.
11) $\mathcal{W}$ omits some zero-knowlege proof components of `PayCommit` to make it look like a `PayConfirm` proof and claims the cart payment at $\mathcal{B}$.
Solution: The above attack is not possible because both proofs are $\mathcal{P}$'s signatures over different nonces and messages. Due to unforgeability property of ABCs, $\mathcal{W}$ cannot modify $\mathcal{P}$'s `PayCommit` proof into another valid signature from $\mathcal{P}$ without $\mathcal{P}$'s secret inputs (secret key and randomness for the commitments).
12) $\mathcal{P}$ picks up the package but does not send payment confirmation for $\mathcal{W}$.
Solution: If the package is delivered correctly, $\mathcal{P}$ does not have any motivation to block the cart payment to $\mathcal{W}$ because of two reasons: first, she cannot use the payment credential obtained from the bank to pay for any other purpose – it is only meant to pay the webshop for a particular cart, and next, she cannot reclaim her money from the bank because the credential is like a 'dinner cheque' that can only be spent and bank only reimburses the payee whose identifier is present in the `PayConfirm` proof. Furthermore, in the last paragraph of Section III-F, we describe a mechanism in which the locker $\mathcal{L}$ receives both `ProvePayAck` and `PayConfirm` simultaneously from $\mathcal{P}$ but it forwards the `PayConfirm` proof to $\mathcal{W}$ only after a predetermined delay. This mechanism handles the instances where a purchaser forgets to send `PayConfirm` proof after picking up her package. However, we do not solve the issue when a purchaser solely wishes to cause damage to the webshop by blocking payment confirmation at the locker during the delay period. Here we note that achieving fairness in e-commerce schemes is an orthogonal problem and the implementors of our scheme may choose to handle the afore-mentioned issue by using some ideas proposed in fairness protocols existing in the literature (e.g. [14]).
13) $\mathcal{B}$ fails to add an outstanding payment entry (maliciously or erroneously) to its pool after a customer's payment which results in possible denial of payment to $\mathcal{W}$ during payment recovery.
Solution: Under the above circumstance, if $\mathcal{W}$ produces a valid `PayConfirm` proof with a fresh ID_c that is not present in $\mathcal{B}$'s double-spend database, then $\mathcal{B}$ following the scheme pays the cart amount to $\mathcal{W}$. Here $\mathcal{B}$ relies on the unforgeability property of ABCs which makes it impossible for a payee (i.e., $\mathcal{W}$) to have come up with a fake payment-credential and a corresponding `PayConfirm` proof with this credential, even it had colluded with the purchaser $\mathcal{P}$.

V. Related work

Smith et al. [15] survey the existing technologies that promote consumer privacy in e-commerce. They split the range of privacy-enhancing technologies that have been proposed in the literature into two main categories: 1. Those that attempt to preserve an individual's privacy by enabling anonymous communication channels for interaction between a customer and an e-business; 2. Those that attempt to minimize the amount of personal information given to an e-business during the interaction. In this paper, we assume that all the protocol communication take place within an anonymous channel and mainly focus on achieving privacy through data minimization.

In their position paper [16], Diaz et al. review the e-shopping process and discuss privacy threats in each of its stages (i.e., purchase, payment, delivery and completion). They argue that it is not enough to protect a single stage but rather that a complete solution that deals with threats and data leaks in every stage and interconnections between the stages is necessary. In this paper, we devise a privacy-preserving shopping scheme using ABCs and corresponding protocols for the overall online shopping process that deals with the following privacy threats mentioned in [16]: leaking of shopped products to the bank, linking of a purchaser and a webshop by the bank or third parties and the webshop or the third parties learning a purchaser's delivery address.

To make the purchase anonymous, many cryptographic e-cash schemes have been proposed in the literature (e.g [17], [18], [19], [20]) which make the cash withdrawal and deposit independent of each other. However, our online shopping scheme relies on traditional money and centralized banks for payment, but it uses blind issuance and selective disclosure properties of ABCs to make the cash withdrawal (payment credential issuance) and deposit (payment recovery by the webshop) stages independent of each other.

Zhang et al. [12] propose a true fair exchange protocol that handles dispute resolution automatically and also incorporates physical delivery by using a delivery cabinet. Although their protocol ensures anonymity of the customer and the merchant, it does not achieve unlinkability of a customer's transactions at the merchant as our protocol, because it heavily relies on public-key encryption and signatures.

Alqahtani proposes an e-commerce protocol in [13] which ensures fair exchange of information and digital goods be-

tween a customer and a merchant with the help of a semi-trusted third party. Their protocol also hides the identity of the customer from the merchant by using digital cash for payment and anonymous channel for information exchange. In contrast, our protocol is not restricted to digital goods and the payment works with regular money. Thus our protocol has a wider scope and can be more easily integrated with the existing infrastructure for online shopping.

VI. Discussion

A. Implementation aspects

ABCs have been implemented on several platforms since they were first designed. Most notably, there have been efficient implementations on smart cards [21], [22], [23] and on Android-based smartphones[4]. According to the most recent and efficient implementations, creating a zero-knowledge proof during issuance and selective disclosure takes 1-3 seconds on the smartcards [24] and 12-17 milliseconds on smartphones [25]. The performance on smartphones demonstrate that ABCs can be efficiently used for real-world transactions on the web and moreover smartphones provide user interfaces and are more user-friendly than smartcards. Thus we have chosen to use a smartphone implementation of ABCs from the open-source IRMA project [8] to implement our entire shopping scheme. We call IRMA's ABC phone app that stores the purchaser's ABCs and creates selective disclosure proofs on her behalf as *ABC app* henceforth.

We now discuss two models for putting our attribute-based webshopping protocols into practice. See Table III.

1) A purchaser shops on her personal computer (PC) via a shopping website and uses her smartphone's ABC app for receiving and showing credentials during the shopping transaction.
2) A purchaser shops on her smartphone and uses the ABC app on the same device; the ABC app is invoked at every credential issuance and showing instance over the course of a shopping transaction. Here, both the apps on the smartphone communicate via inter-app communication.

TABLE III
Implementation models for attribute-based webshopping scheme

Model	Shopping app	ABC app
1	PC(Desktop/Laptop)	Smartphone
2	Smartphone	Smartphone

We have developed a prototype implementation of our webshopping scheme that follows the first model (shown in Table III), that is, the webshop is on the purchaser's PC and the ABC app is on her smartphone. On the client (i.e. purchaser) side, the prototype makes use of the ABC app on the smartphone in addition to a web browser, the Tor network (or any other anonymous network) and TLS on the PC. On the server (i.e. webshop) side, it runs a web server that calls IRMA's credential issuer and verifier modules. This prototype successfully demonstrates that:

- developing an online shopping framework by using existing ABC implementations is easy, and
- privacy-preserving webshopping transactions with ABCs are not only feasible but also efficient. There is no observable delay in comparison with traditional webshopping transactions.

B. Comparison with anonymous marketplaces

In the last decade, we have witnessed the emergence, flourishing and eventual demise of many online anonymous marketplaces (e.g. Silk Road, Agora, Silk road 2.0). Such marketplaces are designed to provide an online rendezvous place for sellers and buyers. Some of their features are listed below.

- The marketplaces themselves do not sell.
- They enforce that the buyers and sellers manually log in to view the listings and initiate a shopping transaction.
- They act as risk management platforms by providing payment escrow and dispute resolution mechanisms.
- They provide strong anonymity guarantees to buyers and to sellers.

Emboldened by the anonymity properties of marketplaces such as Silk Road, sellers and buyers often traded narcotics and contraband. These marketplaces were eventually seized by law enforcement agencies, voluntarily shut down or fraudulently closed due to absconding operators [1]. Although, nowadays, new marketplaces[5] have replaced the old ones in response to market and user demand, their future seems very uncertain.

We believe that events such as marketplace shutdowns can be avoided if some control can be exercised on who is selling what. If an online marketplace recruits only the registered (or tax paying) sellers who sell legal goods, then fraud is automatically curbed. This is precisely why we do not focus on seller or product anonymity in our proposed scheme. We protect the privacy of the buyers alone and enable direct communication between identified sellers and unidentified buyers without a trusted third party such as a marketplace operator. As we also use traditional money and banks to handle payment, we identify the seller and the value of the transaction to the bank. This allows the banks to exercise some control at payment recovery, for instance, to detect fraudulent transactions, to ask the webshop to reveal the nature of the goods involved in such transactions, and to take suitable follow-up actions, such as abort the payment to the webshop. Our aim is to provide privacy (anonymity and unlinkability of transactions) for the buyers. That is, in our scheme, buyers can freely buy any product (regular or sensitive) at a legitimate webshop without being watched over by either the webshop or the bank. Blacklisting purchasers by webshops is currently

[4]Read more about IRMA smartphone implementation on https://www.irmacard.org/irmaphone/ and https://privacybydesign.foundation/irma-begin/ (only in Dutch).

[5]Dark web-market list: https://darkwebnews.com/dark-web-market-list/

not possible with our scheme, however, technically, we can use the epoch-based revocation scheme [26] (or other ABC revocation mechanisms) for that purpose. Nevertheless, our focus is the construction of privacy-friendly webshopping; revocation/blacklisting is an orthogonal problem, and out of scope in this research.

VII. Conclusion

Attribute-based credentials (ABCs) make it possible to design applications with security and privacy simultaneously and they also allow great flexibility for defining and enforcing contextual policies in relation to the attributes. Using the two specific ABC protocols – issuing and selective disclosure – a wide variety of web transactions can be described. Unlike earlier work which mostly focused on authorization and encryption with attributes, we have demonstrated a more general approach. In this paper we have described how ABCs can be used in the design of privacy-preserving electronic commerce which offers privacy for purchasers. Our data-minimizing webshopping scheme is also incentivizing for webshops which are data controllers, because data protection regulations such as the upcoming GDPR in Europe will soon make privacy by design and by default mandatory for all the data controllers and impose high penalties for the rule violators. The scheme also creates new business opportunities for the banks, such as facilitating anonymous payments. Furthermore, our scheme can be efficiently implemented with existing components of ABCs on smartphones.

It is expected that ABCs will be applied in various other contexts, such as anonymous donations, discount vouchers, and dissemination of electronic goods (e.g. media streaming, e-books). With this research, we aspire to encourage a privacy-preserving way of thinking about the applications on the web, and the authors hope that it will inspire other researchers and developers as well.

References

[1] Kyle Soska and Nicolas Christin. Measuring the longitudinal evolution of the online anonymous marketplace ecosystem. In *USENIX Security*, volume 15, 2015.

[2] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. 2010.

[3] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174, 1991.

[4] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 186–194. Springer, 1986.

[5] Stefan A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, Cambridge, MA, USA, 2000.

[6] Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Computer and Communications Security (CCS 2002)*, pages 21–30. ACM, November 2002.

[7] Jan Camenisch, Ioannis Krontiris, Anja Lehmann, Gregory Neven, Christian Paquin, Kai Rannenberg, and Harald Zwingelberg. D2.1 Architecture for Attribute-based Credential Technologies. Technical report, ABC4Trust, 2011.

[8] IRMA – I Reveal My Attributes. https://www.irmacard.org, https://privacybydesign.foundation/irma-begin/.

[9] Idemix. https://www.zurich.ibm.com/identity_mixer/.

[10] David Chaum. Blind signatures for untraceable payments. In *Advances in cryptology*, pages 199–203. Springer, 1983.

[11] Brinda Hampiholi, Gergely Alpár, Fabian van den Broek, and Bart Jacobs. Towards practical attribute-based signatures. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 310–328. Springer, 2015.

[12] Qing Zhang, Konstantinos Markantonakis, and Keith Mayes. A practical fair-exchange e-payment protocol for anonymous purchase and physical delivery. In *IEEE International Conference on Computer Systems and Applications, 2006.*, pages 851–858. IEEE, 2006.

[13] Fahad A Alqahtani. A fair exchange & customer anonymity protocol using a trusted third party for electronic commerce transactions & payments. *International Journal of Network Security & Its Applications*, 6(1):59, 2014.

[14] Alfredo Rial. *Privacy-preserving e-commerce protocols*. PhD thesis, Doctoral dissertation, Doctoral Dissertation, KU Leuven University, Belgium. Retrieved from: https://www. cosic. esat. kuleuven. be/publications/thesis-220. pdf, 2013.

[15] Rhys Smith and Jianhua Shao. Privacy and e-commerce: a consumer-centric perspective. *Electronic Commerce Research*, 7(2):89–116, 2007.

[16] Jesus Diaz, Seung Geol Choi, David Arroyo, Angelos D Keromytis, Francisco B Rodriguez, and Moti Yung. Privacy threats in e-shopping (position paper). In *International Workshop on Data Privacy Management*, pages 217–225. Springer, 2015.

[17] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In *Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '88, pages 319–327, London, UK, 1990. Springer-Verlag.

[18] Stefan Brands. Electronic cash on the internet. In *Network and Distributed System Security, 1995., Proceedings of the Symposium on*, pages 64–84. IEEE, 1995.

[19] Gesine Hinterwälder, Felix Riek, and Christof Paar. Efficient E-cash with Attributes on MULTOS Smartcards. In *International Workshop on Radio Frequency Identification: Security and Privacy Issues*, pages 141–155. Springer, 2015.

[20] Georg Fuchsbauer and Markulf Kohlweiss. Anonymous transferable e-cash. In *Public-Key Cryptography–PKC 2015: 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30–April 1, 2015, Proceedings*, volume 9020, page 101. Springer, 2015.

[21] Wojciech Mostowski and Pim Vullers. Efficient U-Prove implementation for anonymous credentials on smart cards. In George Kesidis and Haining Wang, editors, *Security and Privacy in Communication Networks – SecureComm 2011*, volume 96 of *LNICST*, pages 243–260. Springer, 2011.

[22] Pim Vullers and Gergely Alpár. Efficient selective disclosure on smart cards using Idemix. In Simone Fischer-Hübner, Elisabeth de Leeuw, and Chris Mitchell, editors, *Policies and Research in Identity Management (IDMAN)*, pages 53–67. Springer, 2013.

[23] Joerg Abendroth, Vasiliki Liagkou, Apostolis Pyrgelis, Christoforos Raptopoulos, Ahmad Sabouri, Eva Schlehahn, Yannis Stamatiou, and Harald Zwingelberg. D7.1 Application Description for Students. Technical report, ABC4Trust, 2012.

[24] Pim Vullers et al. *Efficient Implementations of Attribute-based Credentials on Smart Cards*. Uitgever niet vastgesteld, 2014.

[25] Sietse Ringers. *Quantization using Jet Space Geometry and Identity Management using Credential Schemes*. Uitgever niet vastgesteld, 2016.

[26] Wouter Lueks, Gergely Alpár, Jaap-Henk Hoepman, and Pim Vullers. Fast revocation of attribute-based credentials for both users and verifiers. *Computers & Security*, 2016.