Privacy in e-Shopping Transactions: Exploring and Addressing the Trade-Offs

Jesus Diaz[1(✉)], Seung Geol Choi[2], David Arroyo[3], Angelos D. Keromytis[4], Francisco B. Rodriguez[3], and Moti Yung[5]

[1] Blue Indico - BEEVA, Madrid, Spain
jesus.diaz@beeva.com, jesus.diaz.vico@gmail.com
[2] United States Naval Academy, Annapolis, USA
choi@usna.edu
[3] Universidad Autónoma de Madrid, Madrid, Spain
{david.arroyo,f.rodriguez}@uam.es
[4] Georgia Institute of Technology, Atlanta, USA
angelos@gatech.edu
[5] Columbia University, New York, USA
moti@cs.columbia.edu

Abstract. The huge growth of e-shopping has brought convenience to customers, increased revenue to merchants and financial entities and evolved to possess a rich set of functionalities and requirements (e.g., regulatory ones). However, enhancing customer privacy remains to be a challenging problem; while it is easy to create a simple system with privacy, this typically causes *loss of functions*.

In this work, we look into current e-shopping infrastructures and aim at enhancing customer privacy while retaining important features and requiring the system to *maintain the topology and transaction flow of established e-shopping systems* that are currently operational. Thus, we apply what we call the "utility, privacy, and then utility again" paradigm: we start from the state of the art of e-shopping (utility); then we add privacy enhancing mechanisms, reducing its functionality in order to tighten privacy to the fullest (privacy); and finally, we incorporate tools which add back lost features, carefully relaxing privacy this time (utility again).

We also implemented and tested our design, verifying its reasonable added costs.

1 Introduction

Privacy vs. Utility: The Case of Group Signatures. The evolution of privacy primitives in various specific domains often centers around the notion of balancing privacy needs and utility requirements. Consider the notion of "digital signature" [22,39] whose initial realization as a public key infrastructure [37] mandated that a key owner be certified with its identity and its public verification key: a certification authority (CA) signs a record (called certificate) identifying the user and its signature public verification key.

I. Dinur et al. (Eds.): CSCML 2018, LNCS 10879, pp. 206–226, 2018.
https://doi.org/10.1007/978-3-319-94147-9_17

Later on, it was suggested that CA's sign anonymous certificates which only identify the keys (for example, a bulk of keys from a group of users is sent to the CA via a mix-net and the CA signs and publish the certificates on a bulletin board: only the owner of a key can sign anonymously with its certified key. Alternatively the CA blindly signs certificates). This brings digital signing to the domain of anonymous yet certified action (i.e., the action/ message is known to originate from the group that was certified).

However, it was noted quite early that under the mask of anonymity users can abuse their power and sign undesired messages, where no one can find the abuser. Therefore, primitives like group signature [14] or traceable signature [29] were designed, assuring that the anonymity property of a signed message usually stays, but there are *authorities which can unmask abusers, or unmask certain message signatures* in order to keep balance between anonymity of well behaving signers while protecting the community against unacceptable message signing practices.

Privacy by Design for Systems in Production? While privacy by design principles mandate that privacy enhancing mechanisms be taken into account already at the design stage of any system, for well established processes and infrastructures this is not possible. Moreover, trying to re-engineer an existing system from scratch, now including privacy tools by design, must nevertheless be constrained at every step by maintaining the same main processes and information flows. Otherwise, there exists a too high risk of rejection due to the unacceptable chain-effect changes its adoption would imply.

Utility, Privacy, and then Utility Again. The above development on group signatures shows that even in one of the simplest case of anonymity vs. basic message authenticity, there is already certain advantage in providing partial anonymity to perform in a desirable environment which balances various needs. Additionally, the described case of privacy by design for already deployed systems calls out for variants of this methodology. Extrapolating from the above staged methodology that gave us the primitives of group signature and traceable signature, we follow a methodology that can be viewed as "utility, privacy, and then utility again": First translating a primitive to an idealized anonymous primitive, but then identifying lost utility which complete anonymity prevents: and, in turn, relaxing privacy for additional utility.

Application to e-Shopping. We put forward our approach for this methodology through to the involved case of the real world (compound) process of e-shopping, where we find numerous trade-offs which we unveil and discuss (based on utility needed in various steps of the system). We begin by modelling the e-shopping ecosystem, identifying its entities, main processes and added-value mechanisms; then, we implement a fully anonymous system that keeping the entities and main processes, at the cost of losing the added-value parts; finally, we recover them by giving end-users the option to act fully anonymously or pseudonymously. Importantly, our methodology allows us to maintain the main

processes of current e-shopping systems, making it easier to come up with a proposal compatible with the existing complex e-commerce ecosystem.

Note that we have not aimed solely at a theoretical exercise. We demonstrate feasibility of our approach by an exemplifying implementation which demonstrates that we keep a large portion of the utility of the original systems (without anonymity) for a reasonable added performance cost (with anonymity). The achieved practicality of a privacy-respectful system in a real-world context is of relevance, specially considering the latest regulations towards privacy, such as the European GDPR (General Data Protection Regulation[1]) and PSD2 (Payment Services Directive[2].)

1.1 Related Work

The most prolific related area are anonymous payments, e-cash [13] being its main representative, which has seen a huge boost since Bitcoin [34]. While Bitcoin itself does not provide robust privacy, more advanced proposals address this [5,12,24,32][3]. Still, they address only the payment process, and are typically not concerned with additional functionality, except [24], which adds support for regulatory concerns. Some traditional e-cash proposals also incorporate utility to some extent, mainly through tracing (after the payment has been done) [11,18,35] or some kind of spending limitation [35,41]. Privacy respectful payment systems out of the e-cash domain also exist, such as [28], built on mix networks to prevent linking customers and merchants, and [43], which uses discounts based on the (always pseudonymous) users' history. Private purchase systems have been constructed preventing merchants from learning what digital goods customers buy [38], but are not suitable for physical goods; [42] works by interleaving proxies that remove identifiable information about customers. Some works focus specifically on privacy respectful user profiling [17,36,44], mostly for affinity programs, although some approaches are also applicable to fraud prevention [17]. Anonymous delivery systems of physical goods have also been proposed [3,42], covering a crucial phase that has received much less attention. Finally, solutions related to the completion phase (feedback, complaints, etc.) have been basically ignored, although this phase have been shown to allow de-anonymization attacks [33]. Underlying most of these proposals are, often, cryptographic primitives such as oblivious transfer [2] or anonymous credentials [9,15], which are of natural interest in this domain as core building blocks.

The above proposals focus on the two steps of the methodology above (i.e., the "*utility, privacy*" stages), with a few limited exceptions [17,24,35,41], thus restricting the extended utility recovered by our last stage of "*utility again.*" Moreover, none covers all the e-shopping core processes, reducing the privacy of the composed overall system to that of the weakest link [20]. Some proposals

[1] https://www.eugdpr.org/.

[2] https://ec.europa.eu/info/law/payment-services-psd-2-directive-eu-2015-2366_en.

[3] As well as many proposals in non-academic forums. See, for instance, https://z.cash/ (a modified implementation of Zerocash) and https://cryptonote.org/.

introduce extensive changes into the infrastructure and processes [28] or require modifications that conflict with regulations or practical concerns, like requiring the outsourcing of information that would probably be proprietary in many scenarios [17,44]. Therefore, at present, the utility-privacy trade-off is leaning towards utility in the industry and towards full privacy in the literature.

1.2 Organization

After some preliminaries in Sect. 2, we sketch in Sect. 3 how we apply privacy to the traditional system. We analyze this system to show its shortcomings and recover utility in Sect. 4. We conclude in Sect. 5. For lack of space, we omit formal security definitions and proofs and a detailed analysis on the experiments performed with our prototype. We refer to the full version of this paper for the details [21].

2 Preliminaries

Notation. For an algorithm A, let $A(x_1, \ldots, x_n; r)$ denote the output of A on inputs $x_1, \ldots x_n$ and random coins r; in addition, $y \leftarrow A(x_1, \ldots, x_n)$ means choosing r uniformly at random and setting $y \leftarrow A(x_1, \ldots x_n; r)$. For a set S, let $x \leftarrow S$ denote choosing x uniformly at random from S. We let $\langle O_A, O_B \rangle \leftarrow P(I_C)[A(I_A), B(I_B)]$ denote a two-party process P between parties A and B, where O_A (resp. O_B) is the output to party A (resp. B), I_C is the common input, and I_A (resp. I_B) is A's (resp. B's) private input; when party B does not have output, we sometimes write $O_A \leftarrow P(I_C)[A(I_A), B(I_B)]$. When a single party algorithm P uses a public key pk, we may write $O \leftarrow P_{pk}(I)$ (although we omit it if it is clear from the context). For readability, we assume that if any internal step fails, the overall process fails and stops.

Basic Cryptographic Primitives. We assume readers are familiar with public-key encryption [22,39], digital signature and commitment schemes [8], and zero-knowledge proofs of knowledge (ZK-PoKs) [26]. Let $(\texttt{EGen}, \texttt{Enc}, \texttt{Dec})$ denote a public-key encryption scheme, and $(\texttt{SGen}, \texttt{Sign}, \texttt{SVer})$ denote a digital signature scheme. For readability, we assume that it is possible to extract the signed message from the corresponding signature. We let $\mathsf{com}_m \leftarrow \mathsf{Com}(m; r_m)$ denote a commitment to a message m, where the sender uses uniform random coins r_m; the sender can open the commitment by sending (m, r_m) to the receiver. We use $\pi \leftarrow \texttt{ProveZK}_L(x; w)$ and $\texttt{VerifyZK}_L(x, \pi)$ to refer to creating non-interactive proof π showing that the statement x is in language L (which we sometimes omit if obvious from the context) with the witness w, and to verifying the statement x based on the proof π.

Group Signatures. Group signatures [10,14,29–31] provide anonymity. A public key is set up with respect to a group consisting of multiple members. Any member of the group can create a signature ϱ revealing no more information about the signer than the fact that a member of the group created ϱ. Group

signatures also provide accountability: the group manager (GM) can open signatures and identify the actual signer.

- $(pk_G, sk_G) \leftarrow \texttt{GS.Setup}(1^k)$ sets up a key pair; GM holds sk_G.
- $\langle mk_i, \ell' \rangle \leftarrow \texttt{GS.Join}(pk_G)[M(s_i), GM(\ell, sk_G)]$ allows member M with secret s_i to join group G, generating the private member key mk_i and updating the Group Membership List ℓ to ℓ'.
- $\varrho \leftarrow \texttt{GS.Sign}_{mk_i}(msg)$ issues a group signature ϱ.
- $\texttt{GS.Ver}_{pk_G}(\varrho, msg)$ verifies whether ϱ is a valid group signature.
- $i \leftarrow \texttt{GS.Open}_{pk_G}(sk_G, \varrho)$ returns the identity i having issued the signature ϱ.
- $\pi \leftarrow \texttt{GS.Claim}_{mk_i}(\varrho)$ creates a claim π of the ownership of ϱ.
- $\texttt{GS.ClaimVer}_{pk_G}(\pi, \varrho)$ verifies if π is a valid claim over ϱ.

Traceable Signatures. Traceable signatures [29] are essentially group signatures with additional support of tracing (when we use the previous group signature operations, but with a traceable signature scheme, we use the prefix `TS` instead of `GS`).

- $t_i \leftarrow \texttt{TS.Reveal}_{sk_G}(i)$. The GM outputs the tracing trapdoor of identity i.
- $b \leftarrow \texttt{TS.Trace}(t_i, \varrho)$. Given the tracing trapdoor t_i, this algorithm checks if ϱ is issued by the identity i and outputs a boolean value b reflecting the check.

Partially Blind Signatures. A blind signature scheme [13] allows a user U to have a signer S blindly sign the user's message m. Partially blind signatures [1], besides the blinded message m, also allow including a common public message in the signature.

- $(pk_S, sk_S) \leftarrow \texttt{PBS.KeyGen}(1^k)$ sets up a key pair.
- $(\tilde{m}, \pi) \leftarrow \texttt{PBS.Blind}_{pk_S}(m, r)$. Run by a user U, it blinds the message m using a secret value r. It produces the blinded message $\tilde{m}$ and a correctness proof π of $\tilde{m}$.
- $\tilde{\varrho} \leftarrow \texttt{PBS.Sign}_{sk_S}(cm, \tilde{m}, \pi)$. Signer S verifies proof π and issues a partially blind signature $\tilde{\varrho}$ on $(cm, \tilde{m})$, where cm is the common message.
- $\varrho \leftarrow \texttt{PBS.Unblind}_{pk_S}(\tilde{\varrho}, \tilde{m}, r)$. Run by the user U, who verifies $\tilde{\varrho}$ and then uses the secret value r to produce a final partially blind signature ϱ.
- $\texttt{PBS.Ver}_{pk_S}(\varrho, cm, m)$ checks if ϱ is valid.

3 System with a High Level of Privacy and Less Functionalities

Following the approach of "utility, privacy, and then utility again", we first overview the existing e-shopping system (*utility*) and then add privacy enhancing mechanisms, relaxing its functionality in order to achieve a high level of privacy (*privacy*). In the next section, we add other important features, carefully relaxing privacy (*utility again*).

The General e-Shopping Process. Assuming users have already registered in the system, we may consider four phases: purchase, checkout, delivery and completion (see Fig. 1). The involved parties are customers (C), merchants (M), the payment system (PS), financial entities processing and executing transactions (that we bundle in our abstraction as FN) and delivery companies (DC). PS basically connects merchants and FN, providing advanced services. First, in the *purchase phase*, C picks the products he wants to buy from M and any coupons he may be eligible for (task in which PS may be involved). In the *checkout phase*, the payment and delivery information specified by C are routed to PS, probably through M, and processed and executed by FN. During checkout, M, PS and FN may apply fraud prevention mechanisms and update C's purchase history. Subsequently, in the *delivery phase*, and for physical goods, DC delivers them to C. Finally, in the *completion phase*, C verifies that everything is correct, maybe initiating a complaint and/or leaving feedback.

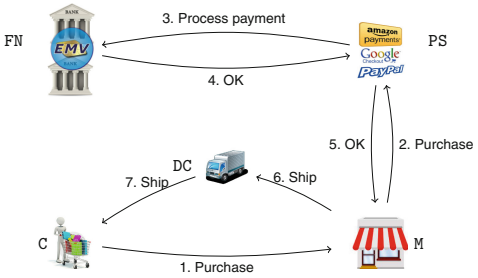

Fig. 1. The overall process of a traditional e-shopping.

Many aspects in this process enter in conflict with privacy (e.g., coupons, fraud prevention and physical delivery), but they are necessary to foster industry acceptance.

3.1 Privacy Goal

We assume that merchants can act maliciously, but PS, FN and DC are semi-honest. Informally, we aim at achieving customer privacy satisfying the following properties:

– *Hide the* identity of a customer *and reveal it only if necessary:* The identity of a customer is sometimes sensitive information, and we want to hide it from other parties as much as possible. In the overall e-shopping process merchants, PS, and DC don't really need the identity of the customer in order for the transaction to go through. However, FN must know the identity to withdraw the actual amount of money from the customer's account and to comply with current regulations.

– *Hide the* payment information *and reveal it only if necessary:* The information about the credit card number (or other auxiliary payment information) that a customer uses during the transaction is quite sensitive and thereby needs to be protected. In the overall e-shopping process, like the case of the customer identity, observe that only FN must know this information to complete the financial transaction.
– *Hide the* product information *and reveal it only if necessary:* The information about which product a customer buys can also be sensitive. However, note that PS and FN don't really need to know what the customer is buying in order for the transaction to go through, but the merchants and DC must handle the actual product.

3.2 Approach for Privacy-Enhancements

In the full version of this paper, we describe in detail the privacy enhanced system. Below, we highlight our approach towards privacy and sketch the system in Fig. 2.

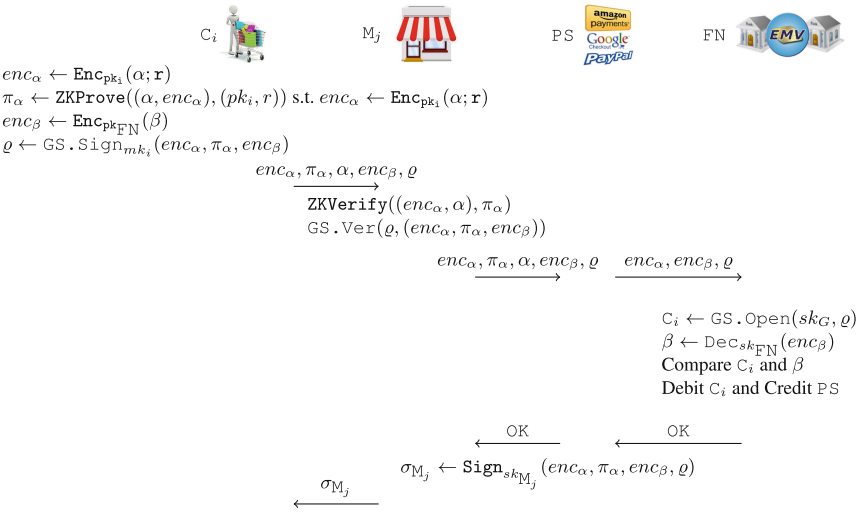

Fig. 2. The overall process of the system. Here, α and β are the product and purchase information respectively. α has been obtained previously by $\mathtt{C}_i$, browsing $\mathtt{M}_j$'s web anonymously.

Controlling the Information of Customer Identity. We use the following privacy-enhancing mechanisms to control the information of customer identity.

– *Sender anonymous channel from customers*: Customers use sender-anonymous channels such as Tor [23] for their communications.

– *Customer group signatures on transaction data:* The transaction data on the customer side is authenticated by the customer's group signature. In our context, FN takes the role of the group manager. Thus, if a merchant M verifies the group signature included by a customer in a transaction, M is confident that the customer has an account with FN. Moreover, due to the group signatures, the customer's identity is hidden from other parties based on. However, since FN takes the role of the group manager, it can identify the customer by opening the signature if required, but it is otherwise not requested to take any active role with respect to managing the group or processing group signatures. Note that the group manager must be a trusted entity concerning the group management tasks, although this trust can be reduced with threshold techniques like those in [6].

Controlling the Payment Information. Customers encrypt their payment information with FN's public key. Thus, only FN can check if the identity in the payment information matches the one extracted from the customer's group signature.

Controlling the Product Information. The customer encrypts the information about the product he wants to purchase using a key-private public key encryption scheme (e.g., ElGamal encryption) [4]; he generates a key pair and uses the public key to encrypt the product information. The key pair can be used repeatedly since the scheme is key-private[4], and the public encryption key is never sent to other parties. The main purpose of doing this is for logging. Once FN logs the transactions, the customer can check the product information in each transaction by simply decrypting the related ciphertext.

Obviously, the encryption doesn't reveal any product information to other parties. Yet, merchants must obtain this data to proceed. To handle it, customers send the product information both in plaintext and ciphertext, and then prove consistency using a ZK proof. When this step is cleared, only the ciphertext part is transferred to other entities.

Note that this system satisfies all our privacy goals. However, it reduces utility, as is not compatible with many features required by the industry (or by regulation), specifically, marketing and fraud prevention tools, or extensions like customer support, subscriptions or taxation [20].

4 Privacy-Enhanced System with Richer Functionality

Next, we add important functionalities, in particular marketing and antifraud mechanisms, to the system described in Sect. 3, carefully relaxing privacy (*utility again*).

[4] Key-privacy security requires that an eavesdropper in possession of a ciphertext not be able to tell which specific key, out of a set of known public keys, is the one under which the ciphertext was created, meaning the receiver is anonymous from the point of view of the adversary.

Adding Marketing Tools: Utility vs Privacy. We would like the payment system PS (or merchants) to use marketing tools (e.g., coupons) so as to incentivize customers to purchase more products and thereby increase their revenue. For clarity of exposition, we will consider adding a feature of coupons and discuss the consequential privacy loss; other marketing features essentially follow the same framework.

When we try to add this feature to the system, PS *must at least have access to the amount of money each customer has spent so far*; otherwise, it's impossible for the coupons to be issued for more loyal customers. Obviously, revealing this information is a privacy loss. However, this trade-off between utility and privacy seems to be unavoidable, if the system is to be practically efficient, ruling out the use of fully-homomorphic encryptions [25] or functional encryptions [7], which are potentially promising but, as of now, prohibitively expensive to address our problem. The main question is as follows:

- Can we reveal *nothing more than* the purchase history of encrypted products?
- Can we provide the customers with an option to *control the leakage of this history*? In other words, can we give the customers an option to *exclude some or all of their purchase activities* from the history?

We address both of the above questions affirmatively. In order to do so, we first allow *each customer to use a pseudonym selectively*. That is, the payment system can aggregate the customer's purchase history of encrypted products only if the customer uses his pseudonym when buying a product. If the customer wants to exclude some purchase activity from this history, he can proceed with the transaction anonymously.

Still, there are a couple of issues to be addressed. First, we would like the system to work in *a single work flow* whether a customer chooses to go *pseudonymously or anonymously*. More importantly, we want a customer to be able to *use coupons even if he buys a product anonymously*. We will show below how we address these issues, when we introduce the notion of a checkout-credential.

Adding Antifraud Mechanisms: Utility vs Privacy. Merchants need to be protected against fraudulent or risky transactions, e.g. transactions that are likely to end up in non-payments, or that are probably the result of stolen credit cards and similar cases. This is typically done by having the PS send a risk estimation value to merchants, who can also apply their own *filters* based on the specifics of the transaction (number of items, price, etc.). At this point, we have an utility-privacy trade-off. In particular, if the risk estimation is too specific and identifying, it will hinder the system from supporting anonymous transactions. We believe that this trade-off is inherent, and in this paper, we treat the specificity of risk estimation to be given as an appropriately-chosen system parameter, depending on the volume of the overall transactions and only

mildly degrading the quality of anonymity in anonymous transactions. The main question we ask is:

Can we relax anonymity of transactions but only to reveal *the risk estimation*?

As with the marketing tools, we use the checkout-credential for implementing this.

4.1 Our Approach

Checkout Credentials. We want to allow customers to perform unlinkable (anonymous) purchases, and we also need to provide merchants with the fraud estimation of a transaction based on each customer's previous transactions. This goal is achieved in a privacy-respectul manner through the checkout-credential retrieval process.

The checkout-credential retrieval process is carried out before the actual checkout, and it is executed between PS and the customer. The resulting checkout-credential is the means used by PS to aggregate the available information related to each pseudonym and provide the marketing and antifraud information for merchants without violating each customer's privacy. Figure 3 shows the augmented information flow of the *purchase* and *checkout* phases in our system. Delivery and completion are not depicted in Fig. 3 since, as we show in the following description, they are quite straightforward and do not suffer further modifications (with respect to the system in Sect. 3) besides integrating them with the new *purchase* and *checkout* processes. Specifically, note that while we have partitioned the main processes in multiple sub-processes, the overall flow is still the same. That is, *purchase* → *checkout* → *delivery* → *completion*. Finally, note also that the parties involved in each process are maintained compared to current systems.

Basically, a checkout-credential is a partially blind signature, requested by a customer and issued by PS, where the common message includes *aggregated*

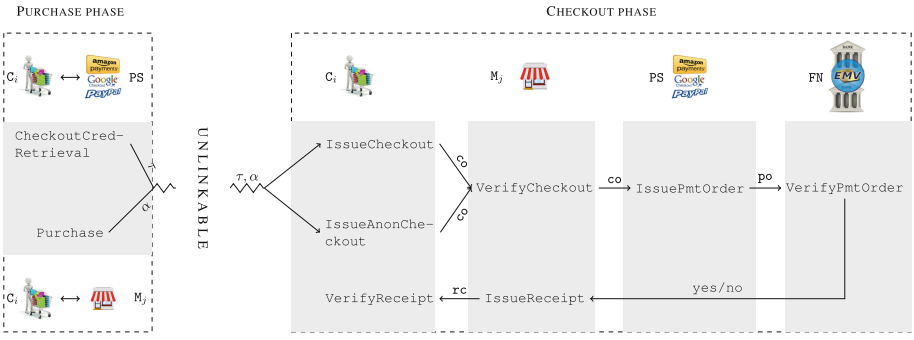

Fig. 3. System process flow. Here, τ is the checkout-credential and α is the product information.

data related to fraud and marketing and the blinded message is a *commitment to the customer key*. During checkout, a customer proves to merchants in ZK that he knows the committed key embedded in the checkout credential. *Since it was blindly signed,* PS *and merchants cannot establish a link* beyond what the aggregated common information allows.

At this point, when the customer decides to perform a pseudonymous checkout (in this case, the pseudonym is also shown during checkout), PS will be able to link the current checkout to the previous ones and update the customer's history (updating his eligibility to promotions and risk estimation). If he chooses an anonymous checkout, PS will not be able to link this transaction with others.

Protection Against Fraudulent Anonymous Transactions. There is an additional issue. An attacker may execute a large volume of pseudonymous transactions honestly, making its pseudonym have a low risk-estimate value, and then perform a fraudulent anonymous transaction. Note in this case, the checkout-credential will contain low risk estimate and the transaction will likely go through, but problematically, *because of unlinkability of this fraudulent transaction,* PS *cannot reflect this fraud into the pseudonym's transaction history.* Moreover, taking advantage of this, the attacker can repeatedly perform fraudulent anonymous transactions with low risk estimate. However, in this variant of our system, we use traceable signatures. Thus, if an anonymous transaction proves to be fraudulent a posteriori, FN can open the signature and give PS the tracing trapdoor associated with the token (i.e., the traceable signature). Given this trapdoor, PS can update the risk estimation even for anonymous checkouts.

Note that customers are offered a trade-off. When customers always checkout anonymously, they have no previous record and receive worse promotions and fraud estimates. When they always checkout pseudonymously, they get better offers and probably better fraud estimates, in exchange of low privacy. But there are also intermediate options. In all cases, they can take advantage of any coupons they are eligible for and receive fraud estimates based on previous pseudonymous purchases.

However, we emphasize that our system is natively compatible with many antifraud techniques in the industry without needing to resort to tracing and which are also applicable with anonymous checkouts and do not reduce privacy (see [21]).

4.2 System Description

In this section, we describe our system. The processes composing each phase are defined next. The flow for purchase and checkout is depicted in Fig. 3.

Setup. FN, PS, and every merchant $\mathtt{M}_j$ and customer $\mathtt{C}_i$ run their corresponding setup processes in order to get their keys, according to the processes in Fig. 4. In particular, FN runs FNSetup to generate traceable signature and encryption keys. PS runs PSSetup to generate a key pair for partially blind signatures. $\mathtt{M}_j$ runs MSetup to generate signing keys. $\mathtt{C}_i$ and FN interact in order to generate key pairs for $\mathtt{C}_i$, running CSetup. $\mathtt{C}_i$ contacts FN, creates an account and joins a group

G, obtaining a membership key mk_i using a secret s_i. In this case, $\mathtt{C}_i$ also sets up a pseudonym P_i, known to FN. The pseudonym P_i is a traceable signature on a random message created using his membership key mk_i; we let $P_i.r$ denote the random message and $P_i.\varrho$ the traceable signature on $P_i.r$. During the process, FN updates its membership database ℓ into ℓ'.

$\mathtt{FNSetup}(1^k)$:
$(pk_G, sk_G) \leftarrow \mathtt{TS.Setup}(1^k)$
$(pk_{\mathtt{FN}}, sk_{\mathtt{FN}}) \leftarrow \mathtt{EGen}(1^k)$
$\mathrm{PK}_{\mathtt{FN}} \leftarrow (pk_{\mathtt{FN}}, pk_G)$
$\mathrm{SK}_{\mathtt{FN}} \leftarrow (sk_{\mathtt{FN}}, sk_G)$

$\mathtt{PSSetup}(1^k)$:
$(pk_{\mathtt{PS}}, sk_{\mathtt{PS}}) \leftarrow \mathtt{SGen}(1^k)$
$(pk_{\mathtt{PBS}}, sk_{\mathtt{PBS}}) \leftarrow \mathtt{PBS.KeyGen}(1^k)$
$\mathrm{PK}_{\mathtt{PS}} \leftarrow (pk_{\mathtt{PS}}, pk_{\mathtt{PBS}})$
$\mathrm{SK}_{\mathtt{PS}} \leftarrow (sk_{\mathtt{PS}}, sk_{\mathtt{PBS}})$

$\mathtt{MSetup}(1^k)$:
$(pk_{\mathtt{M}_j}, sk_{\mathtt{M}_j}) \leftarrow \mathtt{SGen}(1^k)$
$\mathrm{PK}_{\mathtt{M}_j} \leftarrow pk_{\mathtt{M}_j}; \mathrm{SK}_{M_j} \leftarrow sk_{\mathtt{M}_j}$

$\mathtt{CSetup}(pk_G)[\mathtt{C}_i(s_i), \mathtt{FN}(sk_G, \ell)]$:
$\langle mk_i, \ell' \rangle \leftarrow \mathtt{TS.Join}(pk_G)[\mathtt{C}_i(s_i), \mathtt{FN}(\ell, sk_G)]$
$(pk_i, sk_i) \leftarrow \mathtt{EGen}(1^k)$
$\mathtt{C}_i$ chooses $r \leftarrow \{0,1\}^*$
$\mathtt{C}_i$ computes $\varrho \leftarrow \mathtt{TS.Sign}_{mk_i}(r; r_{P_i})$
$\mathtt{C}_i$ sends $P_i = (r, \varrho)$ to FN
$\mathrm{SK}_{\mathtt{C}_i} \leftarrow (P_i, mk_i, r_{P_i}, pk_i, sk_i)$

Fig. 4. Full system setup processes.

Checkout-Credential Retrieval and Purchase. The purchase phase includes the Purchase and CheckoutCredRetrieval processes. The purpose of this phase is for $\mathtt{C}_i$ to obtain a description of the products to buy from $\mathtt{M}_j$ and a credential authorizing him to proceed to checkout, including information necessary to apply marketing and antifraud tools.

During CheckoutCredRetrieval, $\mathtt{C}_i$ interacts pseudonymously with PS. The protocol starts by having the customer $\mathtt{C}_i$ send his pseudonym P_i. Then, PS retrieves the information of how loyal P_i is (i.e., rk), whether (and how) P_i is eligible for promotion (i.e., pr), and the deadline of the checkout-credential to be issued (i.e., dl), sending back $(\mathtt{rk}, \mathtt{pr}, \mathtt{dl})$ to $\mathtt{C}_i$. $\mathtt{C}_i$ chooses a subset $\mathtt{pr}'$ from the eligible promotions pr. Finally, $\mathtt{C}_i$ will have PS create a partially blind signature such that its common message is $(\mathtt{rk}, \mathtt{pr}', \mathtt{dl})$ and its blinded message is a commitment , to his membership key mk_i. We stress that the private member key mk_i of the customer $\mathtt{C}_i$ links the pseudonym (i.e., $P_i.\varrho \leftarrow \mathtt{TS.Sign}_{mk_i}(P_i.r)$) and the blinded message (i.e., $\mathsf{com} \leftarrow \mathsf{Com}(mk_i; r_{com})$). The customer is supposed to create a ZK-PoK ϕ showing this link. Upon successful execution, the checkout-credential is set to τ. We use $\tau.\mathtt{rk}$, $\tau, \mathtt{pr}$, $\tau.\mathtt{dl}$, $\tau.,$, $\tau.\varrho$ to denote the risk factor, promotion, deadline, commitment to the member key, and the resulting blind signature respectively. Refer to Fig. 5 for pictorial description. A checkout-credential issued with the process in Fig. 5 would be verified during checkout using the VerifyCheckoutCred process, defined as follows:

$\mathtt{VerifyCheckoutCred}_{\mathrm{PK}_{\mathtt{PS}}}(\tau)$: **return** $\mathtt{PBS.Ver}_{pk_{\mathtt{PBS}}}(\tau.\varrho, (\tau.\mathtt{pr}, \tau.\mathtt{rk}, \tau.\mathtt{dl}), \tau.\mathsf{com})$

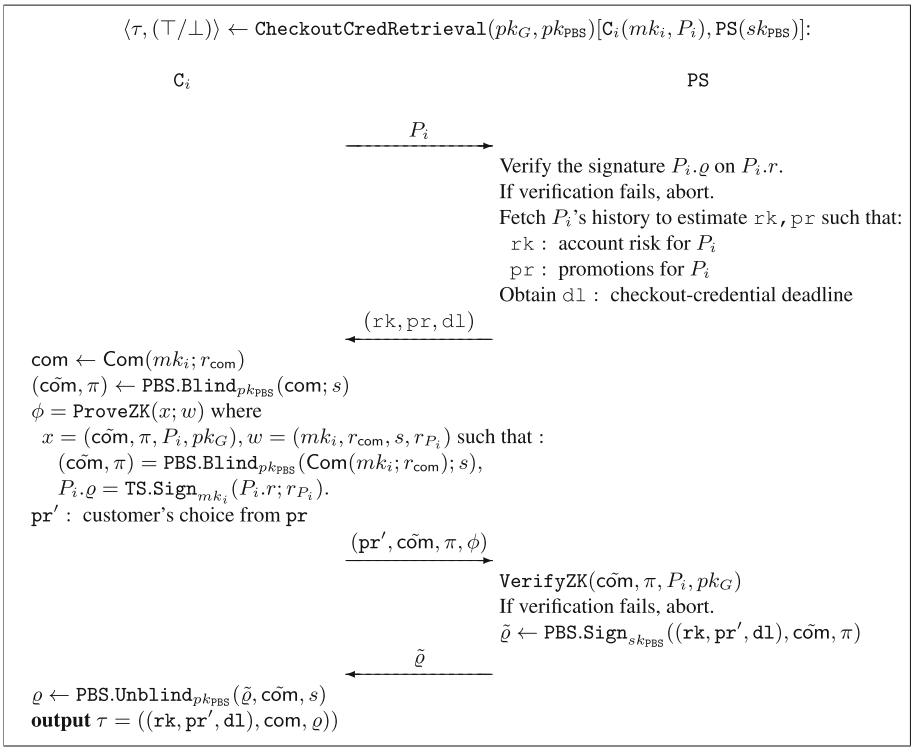

Fig. 5. The `CheckoutCredRetrieval` process.

Concurrently, $\mathtt{C}_i$ obtains through the `Purchase` process a product description of the items he wants to buy. Note that this can be done just by having $\mathtt{C}_i$ browse $\mathtt{M}_j$'s website using sender anonymous channels:

$$\alpha \leftarrow \mathtt{Purchase}[\mathtt{C}_i, \mathtt{M}_j] : \textbf{return} \text{ product description from } \mathtt{M}_j\text{'s website}$$

Finally, with both the product description α and the checkout-credential τ, $\mathtt{C}_i$ can initiate the checkout phase.

Checkout. After receiving the checkout-credential τ and having obtained a product description, $\mathtt{C}_i$ decides whether to perform an anonymous (`IssueAnonCheckout`) or pseudonymous (`IssueCheckout`) checkout process. Let α be the product information with the product name, merchant, etc.; also, let \$ be the price of the product and let β be the customer's payment information containing a random number uniquely identifying each transaction. The checkout process is formed as follows (refer to Fig. 6 for a detailed description of the algorithms). Note that the information flow is equivalent to that in Fig. 2, but here we include additional cryptographic tokens.

Step 1: Client issues a checkout object. A customer $\mathtt{C}_i$ enters the checkout phase by creating a checkout object `co`, executing `Issue(Anon)Checkout` using the

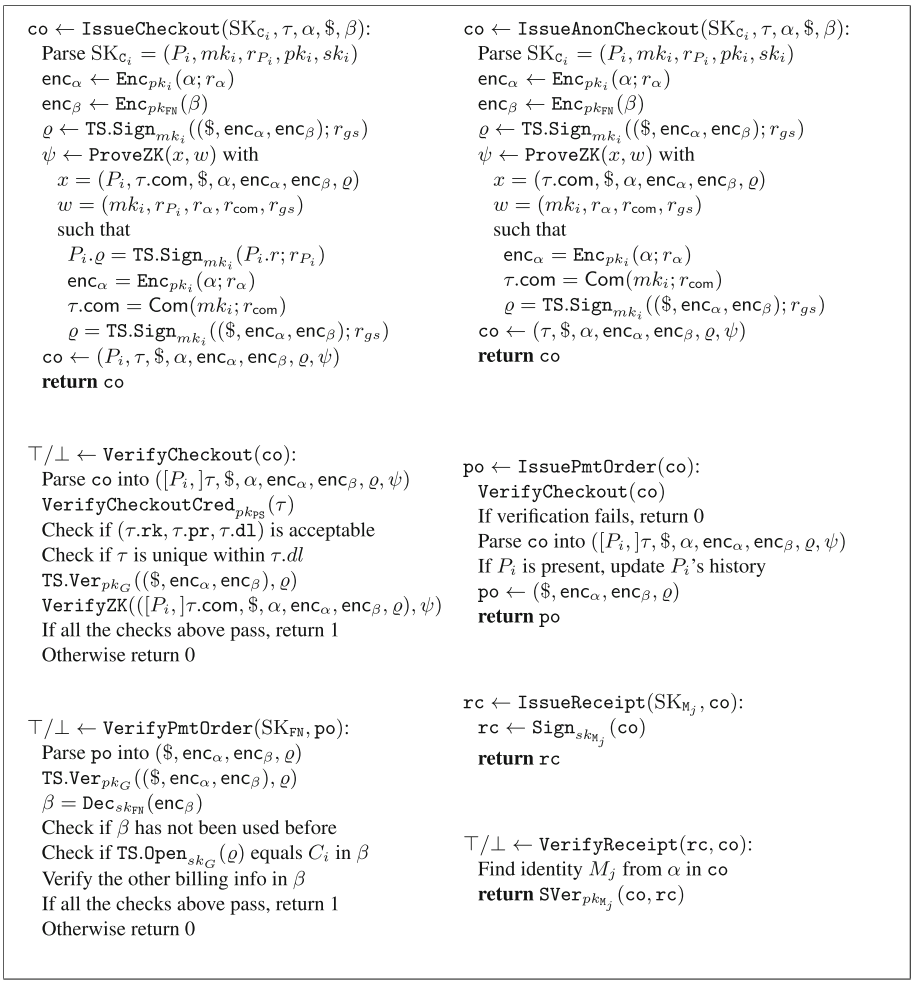

Fig. 6. Checkout algorithms.

checkout-credential τ obtained during checkout-credential retrieval. In either procedure, $\mathtt{C}_i$ generates a traceable signature ϱ on $(\$, \mathsf{enc}_\alpha, \mathsf{enc}_\beta)$, where enc_α is an encryption of the product information α, and enc_β is an encryption of the payment information β, and \$ is the price of the product. Then, $\mathtt{C}_i$ generates a ZK proof ψ showing that the checkout-credential and the traceable signature (and the pseudonym for `IssueCheckout`) use the same mk_i. In summary, we have $\mathtt{co} = ([P_i,]\tau, \$, \alpha, \mathsf{enc}_\alpha, \mathsf{enc}_\beta, \varrho, \psi)$.

Step 2: Merchant processes checkout co. When $\mathtt{M}_j$ receives the checkout object co (which includes the product information α in the clear, as well as encrypted), verifies it with `VerifyCheckout`. If verification succeeds, $\mathtt{M}_j$ passes co to PS. Note that τ needs to be checked for uniqueness to prevent replay attacks. However,

a used credential τ only needs to be stored up to $\tau.dl$. It is also possible for $\mathtt{M}_j$ to include additional antifraud information, like an Address Verification Service value[5] (see [21]).

Step 3: PS *issues a payment order po.* On receiving co from $\mathtt{M}_j$, PS verifies co, runs IssuePmtOrder and issues a payment order *po* with the minimum information required by FN for processing the payment that is, $po = (\$, \mathsf{enc}_\alpha, \mathsf{enc}_\beta, \varrho)$.

Step 4–5: Payment confirmations. Given the payment order *po*, FN verifies it by running VerifyPmtOrder. If the verification succeeds, FN processes the order and notifies PS of the completion; PS in turn sends the confirmation back to $\mathtt{M}_j$.

Step 6: $\mathtt{M}_j$ *issues a receipt.* $\mathtt{M}_j$ receives the confirmation from PS and runs IssueReceipt, issuing rc, a signature on co. Finally, $\mathtt{C}_i$ verifies rc with VerifyReceipt.

Delivery. Once $\mathtt{C}_i$ receives rc, he can use it to prove in ZK that he actually payed for some transaction co, and initiate additional processes, like having DC deliver the goods through APOD [3]. This proof is obtained with the processes in Fig. 7. In the showing process, if $\mathtt{C}_i$ received a receipt rc, he shows rc along with the corresponding checkout object co; then, using his membership key mk_i, he claims ownership of a traceable signature contained in co. Even if he did not receive a receipt, he can prove ownership of ϱ to FN (using ShowReceiptZK too). Since FN is semi-honest, $\mathtt{C}_i$ may ask FN to cancel the associated payment (or *force* PS and $\mathtt{M}_j$ to reissue the receipt).

$\pi \leftarrow \mathtt{ShowReceiptZK}(\mathrm{SK}_{\mathtt{C}_i}, \mathtt{rc}, \mathtt{co})$:
Parse $\mathtt{co} = ([P_i,]\tau, \$, \alpha, \mathsf{enc}_\alpha, \mathsf{enc}_\beta, \varrho, \psi)$
$\pi \leftarrow \mathtt{TS.Claim}_{mk_i}(\varrho)$
return π

$\top/\bot \leftarrow \mathtt{VerifyReceiptZK}(\mathtt{rc}, \mathtt{co}, \pi)$:
Parse $\mathtt{co} = ([P_i,]\tau, \$, \alpha, \mathsf{enc}_\alpha, \mathsf{enc}_\beta, \varrho, \psi)$
VerifyReceipt(rc, co)
$\mathtt{TS.ClaimVer}_{pk_G}(\pi, \varrho)$
If all the checks pass, return 1
Otherwise return 0

Fig. 7. Full system processes for claiming rc in Zero-Knowledge.

In order to interconnect with APOD, $\mathtt{C}_i$ proves $\mathtt{M}_j$ being the owner of rc (through ShowReceiptZK). Then, $\mathtt{M}_j$ issues the credential cred required by APOD as in [3]. Note however that the incorporation of APOD incurs in additional costs and the need for further cryptographic tokens for merchants (who could delegate this task to PS). A less anonymous delivery method, but probably good enough for many contexts, could be using Post Office boxes (or equivalent delivery methods) [20].

Completion. When $\mathtt{C}_i$ receives the goods, the completion phase may take place. In this phase, $\mathtt{C}_i$ may leave feedback or initiate a claim, for which he needs to prove having purchased the associated items. For this purpose, $\mathtt{C}_i$ can again make use of the ShowReceiptZK and VerifyReceiptZK processes, defined in Fig. 7.

[5] https://en.wikipedia.org/wiki/Address_Verification_System.

4.3 Security

We assume that customers and merchants can act maliciously. PS is assumed to be semi-honest during checkout-credential retrieval, but malicious otherwise. FN is semi-honest.

Here, for lack of space, we informally describe the security properties of our system. We give formal security definitions and proofs in the full version [21].

Privacy. The system possesses the following privacy properties.

- *Customer anonymity.* If a customer executes the checkout process anonymously, no coalition of merchants, PS, and other customers should be able to determine the identity or pseudonym of the customer from the checkout process beyond what the common message in the checkout credential reveals.
- *Transaction privacy against merchants and* PS. No coalition of merchants, PS and other customers should be able to determine the payment information associated to the checkout process.
- *Transaction privacy against* FN. The financial network FN should not be able to determine the detail of a customer's transaction beyond what is necessary, i.e., the customer identity and the amount of payment; in particular, M_j's identity and the product information should be hidden from FN.
- *Unlinkable checkout-credential retrieval and checkout.* If a customer runs an anonymous checkout, no coalition of merchants, PS, and other customers should be able to link the customer or his pseudonym to the corresponding checkout-credential retrieval procedure beyond what the common message in the credential reveals.

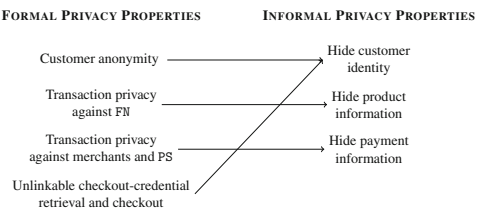

Fig. 8. Mapping between informal properties in Sect. 3.1 and formal properties in this section.

Note that this properties map to the properties in Sect. 3.1, with some additional conditions (see Fig. 8 for a pictorial representation). It is also worth noting that there are indirect connections between them. For instance, *Transaction privacy against* FN and *Transaction privacy against merchants and* PS undoubtedly improves resistance against differential privacy attacks aimed at deanonymizing customers (hence, affecting the *Customer anonymity*). However, as stated in the conclusion, a detailed analysis of these aspects is out of the scope of this work and is left for future work.

Robustness. The system also ensures the following robustness properties.

- *Checkout-credential unforgeability.* A customer should not be able to forge a valid checkout-credential with a risk factor, promotions or deadline set by his own choice.
- *Checkout unforgeability.* When $\texttt{C}_i$ receives a checkout-credential from PS, it cannot be used by $\texttt{C}_j$ $(i \neq j)$ to create a valid co, even if they collude.
- *Fraudulent transaction traceability.* When $\texttt{C}_i$ performs a fraudulent transaction, FN and PS can trace the pseudonym used by $\texttt{C}_i$ even if the transaction is anonymous.
- *Receipt unforgeability.* No coalition of customers, merchants (other than the target $\texttt{M}_j$), and PS should be able to forge a valid receipt that looks originating from M_j.
- *Receipt claimability.* For any valid receipt issued to an uncorrupted customer, no other customer should succeed in claiming ownership of the receipt.

4.4 Outline of the Methodology and Experiments Summary

We achieve a privacy-enhanced e-shopping system by applying the *utility, privacy and utility again* methodology as follows:

- *(Utility, privacy)* Following [20], we first identify the core components of the existing e-shopping system as follows:
  - The participating parties: users, merchants, payment systems, financial network, and delivery companies.
  - The basic e-shopping processes: purchase, checkout, delivery, completion.
  - Added-value tools: marketing and fraud prevention.

  When applying the privacy-enhancing mechanisms, we minimize the modification of these core functionalities. In particular, we change neither the participating parties nor the actual transaction flow. However, we add full anonymity at the cost of marketing and fraud prevention tools.
- *(Utility again)* In this stage, we add the following important real-world features:
  - Marketing tools such as targeted coupons.
  - Fraud preventions measures, allowing to include unpayment risk estimations.

  When providing these important utility features, we carefully relax privacy. In particular, each customer is associated with a pseudonym, and fraud prevention and marketing tools are applied by aggregating certain pieces of transaction history based on the pseudonym. Yet, we allow customers to act anonymously in each transaction, ensuring *privacy is not reduced beyond what this aggregation implies.*

Finally, we have implemented a prototype of our system. Here, for lack of space, we do not include a full report on our results, which will be made available in the full version [21]. As as a summary, we point out that in an unoptimized

version of our prototype, we achieve between 1–3 full-cycle purchases per second. For comparison, other similar systems (e.g., Magento) report between 0.17 and 0.7 purchases per second[6]. It is important to note that we have simplified some parts of the process, such as payments (simulated through a database modification). This, however, is likely to be a relatively negligible operation within the overall process: e.g. VISA processed 141 billion transactions in 2016[7], which makes roughly 4500 transactions per second. Concerning the sizes of the groups of customers in the group signature schemes, we note that this is a highly configurable aspect. For instance, groups can be set based on geographies, based on sign up time, or other heuristics. As for the impact on performance of the sizes of the groups, we refer to [19], which we used to implement our prototype and offers some statistics about the group sizes and throughput of the main operations.

5 Conclusion

We have put forth our proposal for reaching a balance between privacy and utility in e-shopping. This is a complex scenario, where the diverse set of functionalities required by the industry makes it hard to provide them in a privacy respectful manner [20]. Moreover, the restriction of maintaining a similar system topology, limits the application of traditional privacy by design principles. With respect to the related work, our proposal integrates all core components of e-shopping (purchase, checkout, delivery and completion) and the advanced functionality in industry systems (marketing and fraud prevention). To the best of our knowledge this is an unsolved problem [20,40].

Note that our system provides a basic infrastructure for building privacy respectful systems requiring user profiling. Specifically, users pseudonymously obtain customized credentials based on their history, and then anonymously prove possession of those credentials unlinkably to the pseudonymous phase. We have also implemented a prototype of our system, showing its practicability and low added costs. We refer to the full paper for further details on experiments, formal security proofs and possible extensions [21].

Nevertheless, further work is necessary. We include aggregated antifraud and promotions information that is publicly accessible from the checkout-credential. Hence, an open problem is reducing the impact of this leak for reidentification.

Finally, we used a "*utility, privacy, and then utility again*" methodology for designing our system. This strategy is can be applied to transition from policy to engineering in privacy protection in already deployed systems [16]. In other words, our work contributes to build up the Business, Legal, and Technical framework [27] demanded to reconcile economic interests, citizens' rights, and users' needs in today's scenario.

[6] https://magento.com/sites/default/files/White%20Paper%20-%20Magento%202.0%20Performance%20and%20Scalability%2003.31.16.pdf.

[7] https://usa.visa.com/dam/VCOM/global/about-visa/documents/visa-facts-figures-jan-2017.pdf.

Acknowledgements. The work of Jesus Diaz was done in part while visiting the Network Security Lab at Columbia University. The work of Seung Geol Choi was supported in part by the Office of Naval Research under Grant Number N0001415WX01232. The work of David Arroyo was supported by projects S2013/ICE-3095-CM (CIBERDINE) and MINECO DPI2015-65833-P of the Spanish Government. The work of Francisco B. Rodriguez was supported by projects MINECO TIN2014-54580-R and TIN2017-84452-R of the Spanish Government. The work of Moti Yung was done in part while visiting the Simons Institute for Theory of Computing, UC Berkeley.

References

1. Abe, M., Fujisaki, E.: How to date blind signatures. In: Kim, K., Matsumoto, T. (eds.) ASIACRYPT 1996. LNCS, vol. 1163, pp. 244–251. Springer, Heidelberg (1996). https://doi.org/10.1007/BFb0034851
2. Aiello, B., Ishai, Y., Reingold, O.: Priced oblivious transfer: how to sell digital goods. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 119–135. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44987-6_8
3. Androulaki, E., Bellovin, S.M.: APOD: anonymous physical object delivery. In: Privacy Enhancing Technologies, pp. 202–215 (2009)
4. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 566–582. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_33
5. Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: decentralized anonymous payments from bitcoin. In: 2014 IEEE Symposium on Security and Privacy, pp. 459–474 (2014)
6. Benjumea, V., Choi, S.G., Lopez, J., Yung, M.: Fair traceable multi-group signatures. In: Tsudik, G. (ed.) FC 2008. LNCS, vol. 5143, pp. 231–246. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85230-8_21
7. Boneh, D., Sahai, A., Waters, B.: Functional encryption: definitions and challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_16
8. Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. J. Comput. Syst. Sci. **37**(2), 156–189 (1988)
9. Camenisch, J., Dubovitskaya, M., Neven, G.: Oblivious transfer with access control. In: ACM CCS, CCS 2009, pp. 131–140. ACM (2009)
10. Camenisch, J., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 61–76. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_5
11. Camenisch, J., Piveteau, J.-M., Stadler, M.: An efficient fair payment system. In: ACM Conference on Computer and Communications Security, pp. 88–94 (1996)
12. Campanelli, M., Gennaro, R., Goldfeder, S., Nizzardo, L.: Zero-knowledge contingent payments revisited: attacks and payments for services. In: Proceedings of the 2017 ACM SIGSAC CCS, pp. 229–243 (2017)
13. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology, pp. 199–203. Springer, Boston (1983). https://doi.org/10.1007/978-1-4757-0602-4_18
14. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-46416-6_22

15. Coull, S.E., Green, M., Hohenberger, S.: Access controls for oblivious and anonymous systems. ACM Trans. Inf. Syst. Secur. **14**, 10:1–10:28 (2011). http://doi.acm.org/10.1145/1952982.1952992
16. Danezis, G., Domingo-Ferrer, J., Hansen, M., Hoepman, J.-H., Le Metayer, D., Tirtea, R., Schiffner, S.: Privacy and data protection by design-from policy to engineering. Technical report, ENISA (2014)
17. Danezis, G., Kohlweiss, M., Livshits, B., Rial, A.: Private client-side profiling with random forests and hidden Markov models. In: Fischer-Hübner, S., Wright, M. (eds.) PETS 2012. LNCS, vol. 7384, pp. 18–37. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31680-7_2
18. Davida, G., Frankel, Y., Tsiounis, Y., Yung, M.: Anonymity control in E-cash systems. In: Hirschfeld, R. (ed.) FC 1997. LNCS, vol. 1318, pp. 1–16. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63594-7_63
19. Diaz, J., Arroyo, D., de Borja Rodríguez, F.: libgroupsig: an extensible C library for group signatures. IACR Cryptology ePrint Archive, 2015:1146 (2015)
20. Diaz, J., Choi, S.G., Arroyo, D., Keromytis, A.D., Rodriguez, F.B., Yung, M.: Privacy threats in e-Shopping (position paper). In: Garcia-Alfaro, J., Navarro-Arribas, G., Aldini, A., Martinelli, F., Suri, N. (eds.) DPM/QASA -2015. LNCS, vol. 9481, pp. 217–225. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29883-2_14
21. Diaz, J., Choi, S.G., Arroyo, D., Keromytis, A.D., Rodriguez, F.B., Yung, M.: A methodology for retrofitting privacy and its application to e-Shopping transactions (2018, to appear)
22. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Inf. Theor. **22**(6), 644–654 (1976)
23. Dingledine, R., Mathewson, N., Syverson, P.: Tor: the second-generation onion router. In: USENIX Security Symposium, SSYM 2004, Berkeley, CA, USA, pp. 21–21. (2004)
24. Garman, C., Green, M., Miers, I.: Accountable privacy for decentralized anonymous payments. IACR Cryptology ePrint Archive, 2016:61 (2016)
25. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) 41st ACM STOC, pp. 169–178. ACM Press, May/June 2009
26. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. **18**(1), 186–208 (1989)
27. Greenwood, D., Stopczynski, A., Sweatt, B., Hardjono, T., Pentland, A.: The new deal on data: a framework for institutional controls. In: Privacy, Big Data, and the Public Good: Frameworks for Engagement, p. 192 (2014)
28. Jacobson, M., M'Raïhi, D.: Mix-based electronic payments. In: Tavares, S., Meijer, H. (eds.) SAC 1998. LNCS, vol. 1556, pp. 157–173. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48892-8_13
29. Kiayias, A., Tsiounis, Y., Yung, M.: Traceable signatures. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 571–589. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_34
30. Libert, B., Peters, T., Yung, M.: Group signatures with almost-for-free revocation. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 571–589. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_34
31. Libert, B., Yung, M.: Fully forward-secure group signatures. In: Naccache, D. (ed.) Cryptography and Security: From Theory to Applications. LNCS, vol. 6805, pp. 156–184. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28368-0_13

32. Miers, I., Garman, C., Green, M., Rubin, A.D.: Zerocoin: anonymous distributed e-cash from bitcoin. In: 2013 IEEE Symposium on Security and Privacy (2013)
33. Minkus, T., Ross, K.W.: I know what you're buying: privacy breaches on eBay. In: De Cristofaro, E., Murdoch, S.J. (eds.) PETS 2014. LNCS, vol. 8555, pp. 164–183. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08506-7_9
34. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2009). http://www.bitcoin.org/bitcoin.pdf
35. Nakanishi, T., Haruna, N., Sugiyama, Y.: Unlinkable electronic coupon protocol with anonymity control. ISW 1999. LNCS, vol. 1729, pp. 37–46. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-47790-X_4
36. Partridge, K., Pathak, M.A., Uzun, E., Wang, C.: PiCoDa: privacy-preserving smart coupon delivery architecture (2012)
37. ITU-T Recommendation. X.509. Information technology - open systems interconnection - the directory: authentication framework, June 1997
38. Rial, A., Kohlweiss, M., Preneel, B.: Universally composable adaptive priced oblivious transfer. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 231–247. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03298-1_15
39. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
40. Ruiz-Martinez, A.: Towards a web payment framework: State-of-the-art and challenges. Electron. Commer. Res. Appl. **14**, 345–350 (2015)
41. Sander, T., Ta-Shma, A.: Flow control: a new approach for anonymity control in electronic cash systems. In: Franklin, M. (ed.) FC 1999. LNCS, vol. 1648, pp. 46–61. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48390-X_4
42. Stolfo, S., Yemini, Y., Shaykin, L.: Electronic purchase of goods over a communications network including physical delivery while securing private and personal information of the purchasing party. US Patent App. 11/476,304, 2 November 2006
43. Tan, C., Zhou, J.: An electronic payment scheme allowing special rates for anonymous regular customers. In: DEXA Workshops, pp. 428–434 (2002)
44. Toubiana, V., Narayanan, A., Boneh, D., Nissenbaum, H., Barocas, S.: Adnostic: privacy preserving targeted advertising. In: NDSS (2010)