Who Pays Whom? Anonymous EMV-Compliant Contactless Payments

Charles Olivier-Anclin
Universite de Clermont Auvergne, LIMOS
INSA CVL, LIFO, Université d'Orléans, Inria
be ys Pay

Ioana Boureanu, Liqun Chen, C. J. P. Newton
Surrey Centre for Cyber Security, University of Surrey

Tom Chothia, Anna Clee, Andreas Kokkinis
University of Birmingham

Pascal Lafourcade
Universite de Clermont Auvergne, LIMOS

Abstract

EMV is the de-facto worldwide payment system used by Mastercard, Visa, American Express, and such. *In-shop* EMV contactless payments are not anonymous or private: the payers' long-term identification data leaks to Merchants or even to observers. *Anti-Money Laundering (AML)*, *Know Your Customer (KYC)* and *Strong Customer Authentication (SCA)* are payment regulations protecting us from illegal activities, but –in so doing– contribute chiefly to this lack of privacy in EMV payments. Threading the tightrope of AML, KYC and SCA regulations, we provide two *privacy-enhancing*, *EMV-compatible*, *law-abiding* and practicable contactless-payments protocols: `PrivBank` and `PrivProxy`.

We do not use privacy-enhancing technology, like homomorphic encryption, that would break backwards-compatibility with current EMV, but rather we do privacy by engineering design, *adhering to the existing EMV infrastructure, as is*. So, `PrivBank` and `PrivProxy` *provably* achieve strong notions of payers and merchant privacy, anonymity and unlinkability as seen in e-cash or shopping vouchers, whilst being *implementable in EMV as it stands*.

1 Introduction

No Privacy in Contactless EMV by Default. EMVCo [5] is the largest consortium of payment providers, including Visa, Mastercard, American Express and UnionPay. *In-shop* EMV transactions form 94% of the whole payments market [47], with 12.8 billion bankcards in circulation. Despite its modern features, EMV payments do not provision privacy guarantees, for *in-shop*, contactless EMV payments, be it by "plastic"/physical card or mobile device: in these type of transactions, payees and payers identifiable data can be tracked. For starters, payment providers (e.g., Visa, Mastercard) link together mobile and plastic-card payment data, in their standard form, for instance, in the context of loyalty schemes and statistics [19]. Also, card-issuing banks as well as the EMV payment networks always know where we shop, with which merchants, at which location and time, and could be using this to profile us. Worse, despite the fact that all modern, EMV-compliant mobile-apps generate and use a new "account number" with every payment one makes, merchants can link our mobile purchases together, even if we make some with GooglePay and others with ApplePay or SamsungPay. And, there is no complexity to this: just observing the in-shop transaction between the payment device (*i.e.,* card or phone) and the merchant's payment-terminal/PoS (Point of Sale) suffices.

Existent Privacy-aware EMV: Online Only & Potential for Usability Improvements. Certain banks (such as Revolut [17]) have taken steps towards better EMV-payment security and give some (weak) form of pseudonymity by generating cards for one-time use. However, these solutions are only for online shopping and lack a certain degree of end-to-end usability: a one-time card is generated on an app, then the payer has to go online separately and pay with it. Meanwhile, anonymous payments have been considered since the introduction of electronic payments in the 1980 in the form of e-cash [36, 60], and large-scale projects such as GNU TALER [34, 42] aim to revive that; their drawback (with respect to our goals) is that they are not EMV-compliant, and in fact they would need to replace in-shop EMV transactions with a new *online* payment system. Deploying that at a large scale would be costly.

Adding Practicable Privacy to EMV, In-Shop Contactless Payments. There is no in-shop/by-PoS contactless payment solution providing privacy preserving properties, *e.g.,* pseudonymity (payments traceable to longterm identifier) and unlinkability (preventing the linking of two payments) for payers and for merchants, whilst being EMV-compliant, or as practicable as the user-friendly, in-shop EMV contactless payments. Achieving this is not trivial: due to anti-money laundering and fraud-protection regulations, legal requirements inherently do not align with privacy-preservation in EMV contactless in-shop payments, making it hard to attain. Pseudonymity in EMV is hindered by the need for EMV payments to be auditable by the decision-making entities. There are several *Anti-Money Laundering* (AML) regulations [51], including *Know Your Customer (KYC)* and further there are *Strong Customer Authentication (SCA)* [49] regulations to protect customers from fraud. There are also further fraud-protection mechanisms which are not mandatory but lack thereof increases the risk of financial losses for banks and financial bodies, as they have to reimburse customers for unauthorised use of their cards. So, there is one added complexity to proposing a scheme for privacy-preserving EMV contactless, in-shop payments: making that scheme abide by the laws and regulations governing EMV.

Our Research Questions (RQ) & Contributions.

We state below our contributions as answers to research questions (RQ), providing intuitions for these answers.

RQ1: How can one create in-shop, contactless payments that are privacy preserving, and EMV-compliant from the viewpoint of system requirements engineering, and transaction flow of current in-shop, contactless EMV?

When answering this question, the solution would not add heavy privacy-enhancing cryptographic machinery (such as homomorphic encryption, *etc.*) on top of EMV payments, as that would immediately lead to a system that is not compliant with today's EMV back-ends. This would also likely increase the duration of a payment, infringing on the timing constraints that exist in place today (*i.e.,* a contactless payment end-to-end takes only 12 seconds [20]). So, a payment scheme answering RQ1 will likely follow closely existing EMV payments, adding to it not privacy-enhancing cryptography but rather privacy-enhancing parties such as anonymising proxies (also called instant escrows). Indeed, privacy proxies are a well-known privacy-enhancing technology (PET), accepted and catered for even by internet standardisation bodies [54]. In our solutions, we pursue this idea; see Section 6.

RQ2: If EMV-compliant privacy preserving payments rely on privacy-enhancing proxies being added to the infrastructure, how do these proxies fit in with pseudonymity-hindering AML, SCA laws and fraud prevention mechanisms?

At present time, these pseudonymity-hindering AML and SCA laws and fraud prevention mechanisms are primarily observed and implemented by the card-issuing banks. With this in mind, there are two avenues forward for payment-anonymising proxies in EMV: (a) the issuing banks provision these proxies themselves; (b) third-party proxies are used, but then there will be a trade-off of liability and risks between them and the issuing banks.

In this work, we provide two privacy enhancing designs, `PrivBank` (see Section 6.1) and `PrivProxy` (see Section 6.3), which can be plugged directly into the banking system. They modulate liability and risks differently, as per (a) and (b) above. We evaluate and compare their alignment with existing contactless EMV-based in-shop payment systems and prevailing regulations; see Section 6.2, 6.4 and 6.5.

RQ3: What type of privacy protection that complies with EMV and the law would be desirable and possible to achieve?

We formulate this notion of privacy with adequate pseudonymity and unlinkability requirements of payment schemes (as well as a way to reason about them) in a way that is clear and easy to understand by laypersons, since they will need to choose the levels of privacy (formalised as pseudonymity and unlinkability of various entities) and products suited to them; see Section 3 (pseudonymity notions) and Section 7 (analysis for `PrivBank` and `PrivProxy`). Moreover, these privacy properties need to be general enough to fit not just EMV but also other payment systems, enabling direct comparison; see Section 4. That said, our privacy properties can be made generic in some ways (*e.g.,* akin to those applying to large classes of protocols [33]), and are limited in others (*e.g.,* not covering metadata attacks [40], or counter-based attacks [33]). The limitations of our privacy notions and comparisons with other models are discussed in Section 7.5.

2 Related Work

Research into EMV is vast, ranging from applied works such as [52, 62, 65] to formal treatments [25, 28–31, 41, 56, 59].

Formal Analysis of EMV. Most formalisms for EMV analysis are based on the symbolic/Dolev-Yao model [43], very few are computational (*e.g.,* [32]) like the one we give in the extended version of this manuscript [58]. Moreover, most formalisms focus on security, with few addressing privacy. We are the first to give a model based on mathematical relations (see Section 7) to encode privacy in EMV. The closest idea to this, not in EMV, speaks of traceability relations [33]; but these are complex links made between protocol layers. Next, we will cover directly related works, on privacy and traditional payment systems.

EMV Payments & Privacy. The card-to-PoS channel is insecure as per current EMV specifications, and [35, 53] work on next-generation EMV, where this channel will be secure. In this setting, considering a corruption model and linkability attack stronger than one against Unlnk, [53] find next–generation EMV payments to be linkable. The authors of [35] build on [53] and extend their model to dishonest terminals, achieving unlinkability and pseudonymity for smart card-based payments. Both proposals are non-EMV compliant.

If we go back to standard-EMV systems and and their insecure card-to-PoS channels, then we should mention [29]; they showed that long-term *Primary Account Nmbers* (PANs, identifier written on the payment card) can be used to track people. This gave rise to mobile-devices having "tokenised PANs", which would go towards payers' anonymity, were it not for the introduction of an element called "PAR (Payment Account Reference)" to mobile EMV payments. Such payments were studied, in particular w.r.t. tokenization, in [25, 38], but from security perspectives, not privacy ones.

We fill in these gaps in comparisons and analyses of EMV-payments' privacy. Concretely, in Section 4 and in the extended version [58], we compare *our solutions* with the most common and contemporary payment schemes — ranging from cash and PayPal to mobile payments via platforms like Curve and shop vouchers — evaluating them based on the privacy they offer and the regulations they are subject to.

3 Payments' Privacy Notions

We propose privacy notions from the perspective of different entities in the payment systems: a *payer* who pays for goods/services, a *merchant* selling them, an *issuer* who gives the payer a means of payments (bank account, cards, banknotes, *etc.*), and a *proxy* who mediate the purchase from the payer to the merchant.

3.1 Entities Identification in EMV

Law-Enforced Payer Identifications. In EMV payments, the payer has to be identified at the on-boarding phase with their banks. To obtain a bank card, customers must provide a piece of identification such as a passport and proof of address to the card-issuer. These measures fall under the *Know Your Customer (KYC)* regulations. Secondly, in EMV payments, the payers have to be identified during transactions: knowledge (*e.g.,* PIN, Personal Identification Number), possession (*e.g.,* of the card) or physiological characteristics (biometrics) *etc.* must be checked as they pay; this is known as *Strong Customer Authentication (SCA)* and it is governed by Payments Security Directive (PSD2) regulations [50]. Thirdly, payers must be identified (even beyond SCA) for all transfers or payments amounting to certain values over a certain period (*e.g.,* EUR/GBP1000 per month in the EU/UK). Together, it is all driven by fraud protection (e.g., in the case of SCA) and/or AML which includes the Money Laundering and Terrorist Financing Regulations (AML) regulations [51] [24] [23]. The Financial Conduct Authority (FCA)'s Handbook FCG 3.2.5 [27] requires regulated firms such as issuers to perform (real-time) *monitoring* of transactions and submit "Suspicious Activity Reports" (SAR) (containing the payer identity with the payment details) to the FCA, if concerns arise. Such reports lead to the full identification of the payers and their actions, rendering our privacy notions inapplicable under SAR.

Merchant Identification. In EMV payment systems, the issuers usually get the following merchant information with each of their customers' transactions: *Merchant Category Code (MCC)*, *merchant's name*, *Merchant Risk Index (MRI)*, *Merchant Location (ML)*.

Issuer Identification. The identity of the issuer is generally disclosed to the entities participating in a payment, and is of no interest to third parties. If there is a proxy in the system, this proxy may learn who someone banks with, as is the case in many existing settings, such as those offered by Curve, Monzo, and similar services.

3.2 Our Payment-Privacy Notions

Now, we are in a position to put forward a set of desirable privacy notions, general enough that they can be meaningful in the context of most payment systems. These properties can be from the perspective of the different relevant entities involved: merchants, issuers, proxies and observers.

Payer Pseudonymity ($\mathsf{P}_{\mathtt{Ps}}$). An instance of $\mathsf{P}_{\mathtt{Ps}}$ holds if a given entity cannot recover the payer's identity *ID* or a long-term pseudonym based on actions its sees.

Also, $\mathsf{P}_{\mathtt{Ps}}$ comes in various flavours w.r.t. what is the identifiable "object". For instance, in EMV-like payment systems:

$\mathsf{P}_{\mathtt{Ps}}{}^{ID}$**:** imposes that the merchant does not learn the payer's long-term identity *ID*;

$\mathsf{P}_{\mathtt{Ps}}{}^{C_{ID}}$**:** imposes that the merchant does not learn the payer's long-term card-number or bank account.

However, in EMV payments, even those made via mobile phones, the merchant always learns a long-term pseudonym of the payer. While there is generally no direct relationship between $\mathsf{P}_{\mathtt{Ps}}{}^{ID}$ and $\mathsf{P}_{\mathtt{Ps}}{}^{C_{ID}}$, as shown below, our constructions achieve both simultaneously.

$\mathsf{P}_{\mathtt{Ps}}{}^{C_{ID}} \not\Rightarrow \mathsf{P}_{\mathtt{Ps}}{}^{ID}$**.** For instance, paying for goods with the long-term card currently reveals the PAN/PAR of the card C_{ID} to the (PoS of the) merchant, but the identity of the payer *ID* is not disclosed in the payment.

$\mathsf{P}_{\mathtt{Ps}}{}^{ID} \not\Rightarrow \mathsf{P}_{\mathtt{Ps}}{}^{C_{ID}}$**.** A payer utilising a payment-proxy system (*e.g.,* Revolut, see Appendix B) may undergo a KYC procedure with the proxy thus leaking their *ID*, but said proxy may never know a long-term card C_{ID} of the payer as they may always top up their proxy account from one-time cards.

Payments' Unlinkability (Unlnk). An instance of Unlnk holds if a relevant entity in payment systems will stay unable to link payments made by the same payer.

So, we ask a merchant not distinguish if two payments are made by the same payer, as with cash transactions, say.

Merchant/Seller Pseudonymity ($\mathsf{M}_{\mathtt{Ps}}$). An instance of $\mathsf{M}_{\mathtt{Ps}}$ holds if a payment-system entity (*e.g.,* an issuer) cannot infer the identity of the merchant involved in a payment.

Discussions. $\mathsf{P}_{\mathtt{Ps}}$ and Unlnk are considered in front of the merchant or an observer, since the issuer may always know the payer's identity due to KYC. Similarly, $\mathsf{M}_{\mathtt{Ps}}$ sits most naturally in front of the issuer.

Additional relations between these properties apply: payer pseudonymity is required for unlinkability, *i.e.,* $\mathsf{Unlnk} \Rightarrow \mathsf{P}_{\mathtt{Ps}}$. Indeed, an entity deducing the identity of the payer or their information persistent across transactions can link said transactions. However, the reverse does not hold, *i.e.,* $\mathsf{P}_{\mathtt{Ps}} \not\Rightarrow \mathsf{Unlnk}$. For $\mathsf{P}_{\mathtt{Ps}}{}^{C_{ID}}$ specifically, an example of this is provided by EMV payment tokenisation [46]: there, each payment yields an ephemeral identifier for the card/payer called *tokenised Primary Account Number (PAN)*: *i.e.,* $\mathsf{P}_{\mathtt{Ps}}{}^{C_{ID}}$ holds. But payments by the same payer are linkable together (by the merchant and any observer) via an EMV element called *Payment*

Payment method	SCA		KYC		Pseudonymity		
	Issuer	Proxy	Issuer	Proxy	Issuer	Merchant	Proxy
1. Cash	no	×	×	×	M_{Ps}	$\mathsf{P}_{\mathsf{Ps}}, \mathsf{Unlnk}$	×
2. Cheque	yes	×	yes	×	$\neg\mathsf{M}_{\mathsf{Ps}}$	$\neg\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}$	×
3. E-cash	yes	×	yes	×	M_{Ps}	$\mathsf{P}_{\mathsf{Ps}}, \mathsf{Unlnk}$	×
4. Physical cards	yes	×	yes	×	$\neg\mathsf{M}_{\mathsf{Ps}}$	$\neg\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}$	×
5. Google, Apple Pay, *etc.*	yes	×	yes	×	$\neg\mathsf{M}_{\mathsf{Ps}}$	$\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}$	×
6a. Top-up cards	yes	(yes)	yes	(yes)	$\neg\mathsf{M}_{\mathsf{Ps}}$	$\mathsf{P}_{\mathsf{Ps}}/\neg\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}$	$(\neg\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}, \neg\mathsf{M}_{\mathsf{Ps}})$
6b. Pre-Paid/gifts cards	no	(no)	no	(no)	$\mathsf{M}_{\mathsf{Ps}}/\neg\mathsf{M}_{\mathsf{Ps}}$	$\mathsf{P}_{\mathsf{Ps}}, \mathsf{Unlnk}/\neg\mathsf{Unlnk}$	$(\neg\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}, \neg\mathsf{M}_{\mathsf{Ps}})$
7. Virtual cards	yes	(yes)	yes	(yes)	$\mathsf{M}_{\mathsf{Ps}}/\neg\mathsf{M}_{\mathsf{Ps}}$	$\mathsf{P}_{\mathsf{Ps}}, \mathsf{Unlnk}$	$(\neg\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}, \neg\mathsf{M}_{\mathsf{Ps}})$
8a. PayPal	yes	yes/no	yes	yes/no	M_{Ps}	$\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}$	$\neg\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}, \neg\mathsf{M}_{\mathsf{Ps}}$
8b. Curve	yes	yes	yes	yes	$\neg\mathsf{M}_{\mathsf{Ps}}$	$\neg\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}$	$\neg\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}, \neg\mathsf{M}_{\mathsf{Ps}}$
9. Online Marketplaces	yes	no	yes	yes/no	M_{Ps}	$\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}$	$\neg\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}, \neg\mathsf{M}_{\mathsf{Ps}}$
10. `PrivBank`	yes	no	yes	no	M_{Ps}	$\mathsf{P}_{\mathsf{Ps}}, \mathsf{Unlnk}$	$\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}, \neg\mathsf{M}_{\mathsf{Ps}}$
11. `PrivProxy`	yes	yes	no	yes	M_{Ps}	$\mathsf{P}_{\mathsf{Ps}}, \mathsf{Unlnk}$	$\neg\mathsf{P}_{\mathsf{Ps}}, \neg\mathsf{Unlnk}, \neg\mathsf{M}_{\mathsf{Ps}}$

Table 1: SCA, KYC and pseudonymity properties of payment methods from the point of view of the Issuer, Merchant and the Proxy when it exists. $\neg\mathsf{P}_{\mathsf{Ps}}$ and $\neg\mathsf{Unlnk}$ holds in all systems for the Issuer and $\neg\mathsf{M}_{\mathsf{Ps}}$ holds in all systems for the Merchant. Detailed explanations are provided in the Appendix B. *(Notation: × stands for "not applicable",* P_{Ps} *for* $\mathsf{P}_{Ps}{}^{ID}$ *and* $\mathsf{P}_{Ps}{}^{C_{ID}}$*, brackets are used when the proxy may not necessarily exist in all systems and / when deployment can lead to different properties.)*

Account Reference (PAR): *i.e.,* Unlnk does not hold.

3.3 Threat Model

For the privacy-preserving properties of `PrivBank` and `PrivProxy`, we assume that parties, payers, issuers, merchants, proxies, can be corrupted, at any point in the execution, and they can be made to behave arbitrarily. Yet, due to AML auditability requirement, the issuer and proxy involved in a given executions cannot be simultaneously corrupted, otherwise, in compliance with the law, trivially breaking any pseudonymity property. Thus, excluding all scenarios where identity disclosure is deemed necessary, such as through *suspicious activity reports* mechanisms or other mandatory auditing requirements. Our corruption and adversaries vary with the properties. In general, any party can be corrupted, except for specific ones designated for each property.

For P_{Ps} and Unlnk:(1) the payer against whom the property is considered and at least one other banking with the same issuer; (2) an adversary can eavesdrop payments made by the payers to merchants' PoS[1];

For M_{Ps}:(1) the merchant against whom M_{Ps} is considered and at least one other; (2) an adversary can eavesdrop all payments submitted by the proxies to the issuers[2].

We also assume two realistic aspects. One, the (application implementing our solution on the) payer's phone will not be corrupted, unless the Payer is also corrupted. Second, other data (values, dates, *etc.*) in funds' transfers are independent of the Payer's long-term identity; it means that the Payer does not encode their identity in the time or value of the payment. Moreover, we assume that spending caps, applicable to all payment methods and potentially enforced in `PrivBank` and `PrivProxy` due to AML legal requirements, are never exceeded by payers when assessing the Unlnk property. Similarly, we assume that the two honest payers requesting a payment always share the same acceptance profile, leading to indistinguishable acceptance of their transactions.

[1] This is pertinent in the case where the channel between the PoS and payer is public/un-encrypted, which is the case today; in this setting, the eavesdropper adversary is weaker than a corrupted merchants/payers. However, there are proposals to make this channel secure in EMV 2nd Gen [45].

[2] This is a very strong adversary in practice, as it would mean that there is breach in the backend of the payment networks.

4 Scrutinising KYC & Privacy in Payments

We now empirically evalutate *traditional payment systems* and some *modern* ones, and discuss where they sit w.r.t. KYC and SCA regulations and how they fare against our privacy requirements P_{Ps}, Unlnk, and M_{Ps}. We exclude crypto currencies [57] and QR-code-based payments [12]. The reason for this is that their infrastructure is totally different from the rest of the long-established payments, especially the EMV-based systems, which we aim to augment here. Thus, the systems of interest in our analysis here are: 1. cash, 2. cheque, 3. e-cash [36], 4. physical/classical bankcards, 5. mobile-phone apps [11, 15], 6. top-up and pre-paid cards [7, 13], 7. virtual/one-time cards [17], 8. payment service providers such as PayPal [6] and Curve [8], 9. online marketplaces [9, 16]. The readers are likely familiar with most of these systems and can judge if they have SCA, KYC, P_{Ps}, Unlnk, M_{Ps}, so we left most of their full descriptions in Appendix B. We give below just the description of Curve, as it is less well-known.

Curve. Curve [8] is a payment proxy providing a payer with a card and a payment application. Curve users must satisfy the KYC rules. The payer registers multiple bankcards issued by one or several banks in the Curve app. When paying with a Curve card, authorisation by the payment network (*e.g.,* Visa, Mastercard) and the bank is carried out on the basis of Curve card data and merchant information, as it is the Curve card interacting with the merchant's PoS. Curve then pays the merchant on behalf of the payer on the spot and one of the payer's Curve-registered bankcards is charged. So, there is an intricate payment authorisation process between Curve, traditional bankcard issuers and the payment network.

Analysis of KYC, $\mathsf{P_{Ps}}$, Unlnk, $\mathsf{M_{Ps}}$ in Payments. Table 1 surveys the way payment systems fare against SCA and KYC (described in Section 3.1), and our notions $\mathsf{P_{Ps}}$, Unlnk, $\mathsf{M_{Ps}}$ (Section 3.2). We got to these results via an empirical analysis, brief justifications for our claims are provided in Appendix B. For instance, e-cash, physical cards, payment apps (Google Pay, Apple Pay, *etc.*), top-up and virtual cards are payment methods for which the SCA/KYC apply to the issuer unlike pre-paid cards which fall under exemption cases. PayPal applies strict limits unless the customer is identified, rules application therefore depends on the usage scenario.

Takeaway Message. Cash (rows 1 in Table 1) offers the best pseudonymity and unlinkability, with no SCA/KYC, but is arguably not the modern comodity. In turn, modern payment methods (*e.g.,* rows 4, 5, 7 in Table 1) need to adhere to SCA and/or KYC, and in doing so lose most privacy. Also, when a payment proxy is used (e.g., rows 6 to 9 in Table 1), merchant pseudonymity is typically lost, in part due to AML requirements. Given the variety of results in Table 1, this section suggests, that meeting in the middle between proxied solutions, cash and modern payments methods (like EMV cards, mobile wallets) in terms of $\mathsf{P_{Ps}}$, $\mathsf{M_{Ps}}$, Unlnk properties, KYC/SCA regulations, while maintaining the EMV infrastructure and complying to AML as well, is likely non-trivial.

5 Our Main EMV Ingredients

We recall the notions[3] related strictly to EMV payments which are relevant to us.

5.1 From Card Issuing to Payment Processing

We divide EMV card-based payments in 4 main stages.

1. Card-issuing, KYC & AML. A future Payer opens a bank-account with a *card issuer* (*i.e.,* bank). We discuss the case where they receive a credit/debit card, associated with the account. The card is supplied by one of the current *card providers* (e.g,. Mastercard, Visa, American Express, *etc.*). We will refer to the collection of Issuers, card providers and the proxies linking them loosely as *payment networks*. To obtain such a card, the customers must provide a piece of identification such a passport and proof of address to the card issuer, in line with KYC regulations. KYC falls under the AML regulations [51] (see Section 3.1).

2. Making a Card Payment & Relevant Payment Data: PAN, MCC, ML. A *cardholder* goes to pay with their card to a *Merchant*, using their card readers, known also as *points of sale*: the cardholder inserts the card into the PoS or taps the PoS in the case of a contactless transaction. The basic operations of the protocol between the card and the PoS are defined in the set of standards from EMVCo. A partial example of the data exchanged between the card and the reader, called the *payment transcript*, is available in the extended version of this manuscript [58].

Due to the transmission of the PAN, plastic-card EMV does not have $\mathsf{P_{Ps}}$ from the viewpoint of the Merchant, or an observer between the card and the Merchant's PoS.

At the end of the protocol, certain transaction data is signed by the card and returned to the PoS for its checks. Equally, the card sends a MAC of certain transaction data to the PoS to forward it to the card-issuers for checks therein. They check this data and based on it, they approve/decline the transaction.

Alongside with the card-centric data sent on the back-end from the Merchant's PoS to the Issuer, payment networks and others, the Merchant's PoS also adds all or some of the following merchant-identifying details, relevant to us: *Merchant category code (MCC)*, *Merchant's name (MN)*, *Merchant risk index (MRI)*, *Merchant location (ML)*. Thus, due to their transmission of the MN, plastic-card EMV does not have $\mathsf{M_{Ps}}$ from the viewpoint of the Issuer.

3. Customer Identification During Payments. When the card is presented to the Merchant's PoS for payment, the SCA/PSD2 [49, 50] regulation require two factor authentication of the Payer (*e.g.,* possession of card and associated PIN). The Issuer checks the payment data sent by the Merchant along with this SCA identification-data of the payer. Should the checks fail, the payment is declined by the bank. There are variations to card and PIN verification, especially if the payment is not made by card[4]. If the payment is contactless, derogation from the SCA rules can apply and single-factor authentication is required instead. SCA remains required every few payments or after a set of payments has exceeded set value (*e.g.,* EUR/GBP150 in EU/UK).

4. Payment Authorisation and Clearing. Funds are settled during the final phase, called *clearing*, as follows. (i) The Merchant, via their acquirer, requests payment from the card issuer. The issuer verifies details like transaction location, payer identity, and merchant information. (ii) If the cardholder has sufficient funds, the issuer deducts the amount from their

[3] Capital letters are used to refer formally to entities in the system: e.g., "payer" – a personal paying, in Sections 1-4, vs "Payer" – a formal algorithmic party, from Section 5 onwards.

[4] E.g., if the Payer uses a smartphone SCA verification by the issuing bank is replaced by *Consumer Device Cardholder Verification Method* (CDCVM) executed on the phone. That is, the payers fingerprint or face identification is read by the phone and used as customer authentication. The result of that is later checked by the issuing bank.

account[5]. The final authorization is handled by the issuing bank, possibly in consultation with payment networks like Visa or Mastercard. Once approved, the funds are transferred to the Merchant's/acquiring bank.

As explained in the extended version of this manuscript [58], the information necessary for a payment authorisation varies based on the business model (*e.g.,* from Visa to Mastercard) and not all Merchant information is truly necessary. For example, Curve [8] operates in the following way (and *e.g.,* Visa incentivises it [63]): they over-submit Merchant data especially if their MRI is high, to increase the probability of authorisation and therefore maintain customer satisfaction. There may be leeway in provisioning $\mathsf{M}_{\mathtt{Ps}}$ from the viewpoint of the Issuer, since the minimal amount of Merchant data needed is not standardised.

5.2 Mobile-EMV Tokenisation

Mobile payment applications such as ApplePay [15], Google Pay [11] *etc.* allow registering plastic cards to pay via a mobile application. The onboarding requires an authorisation from the card's issuing bank, and therefore KYC is observed. The Payer can then use the app for contactless payments. When a payment is made, the card's long-term PAN is *tokenised*, and the payment transcript between the phone and the Merchant's PoS looks different from one made with the physical card, with PAN-related data replaced by tokenised values. Mobile-payment transcripts (see the extended version of this manuscript [58] for an example) include the following payer-identifying data relevant to us:

• *one-time tokenised PAN* – an ephemeral account number that changes with each payment and each app: each payments made with card C through mobile app A_1 or app A_2 will each generate a different number.

• *long-term PAR* – a fixed value that is shared amongst various/all payment apps $A_1, A_2 \dots$ to refer to *any/all* payment made based on the same physical card C; the PAR was introduced at the request of the Merchants and payment networks, so that mobile payments made with one card C, though showing varying tokenised PANs, can all be linked together.

The tokenised PAN and PAR are sent by the Merchant onto the payments networks, just as the "plastic" PAN was. However, before these reach the Issuer, the tokenised PAN is de-tokenised by entities in the EMV system who transform it back to the associated PAN. All the rest of the backend part of payment processing is as described in Section 5.1. Details of payment tokenisation and use cases are given in [46, 48]. So from the viewpoint of the Merchant, mobile EMV payments achieve a form of $\mathsf{P}_{\mathtt{Ps}}$, via the PAN, but do not achieve Unlnk, due to the PAR.

Takeaway Message. In our designs, we carefully combine the ideas explained above: (1) from mobile payments – namely, we will make the PAN/PARs be one-time/removed (to achieve payer pseudonomyity), and make all their long-term data randomised for each transaction (to achieve payment unlinkability); (2) from payment-authorisation – Curve already sends selective payment-authorisation data to the Issuer; we will further filter or remplace that data through a proxy to achieve merchant pseudonymity. On top, astute orchestration between a proxy and the Issuer, as well as careful protocol design, will lead our designs to achieve EMV-compliance and abiding by AML, KYC and SCA regulations. This intuition is developed further in the extended version of this manuscript [58].

[5]This is for debit cards. For credit cards, this differs slightly.

6 Anonymous EMV In-Shop Payments

Now, we propose two constructions, compatible with EMV contactless payments, providing privacy as per $\mathsf{P}_{\mathtt{Ps}}$, $\mathsf{M}_{\mathtt{Ps}}$, Unlnk, with provable guarantees, all the while being compatible with the aforementioned law and regulations. Legal frameworks may vary locally, our solutions may require adjustments to accommodate varying regulatory environments.

At the core of our first construction called `PrivBank`, given in Figure 1, there is a privacy-friendly issuing bank who provisions $\mathsf{P}_{\mathtt{Ps}}$ and Unlnk for its customers. To do this, this bank strongly partners with a Proxy who mediates and curates customers' payments providing $\mathsf{M}_{\mathtt{Ps}}$. Meanwhile, at the heart of our second construction called `PrivProxy`, given in Figure 2, there is no longer a bank, but rather a pseudonymity-friendly Proxy which aims to provide $\mathsf{P}_{\mathtt{Ps}}$, $\mathsf{M}_{\mathtt{Ps}}$ and Unlnk of its own accord and at its own risks, to Payers who bank with whoever they chose to, independently of the Proxy.

The crux of our designs is to compose several standard, nonprivate EMV-payments or parts thereof, such as to obtain one mobile, contactless EMV payment which attains $\mathsf{P}_{\mathtt{Ps}}$, $\mathsf{M}_{\mathtt{Ps}}$, Unlnk. We realise this via the design and use of proxies (also called instant escrow), without modifying EMV elements in the original payments, and without cryptographic additions. As such, all the cryptography used in our schemes and all EMV building blocks can be treated as black-boxes inherited from EMV, and our only focus is going to be the design of the proxied systems, from an engineering perspective alone. Indeed, our proofs w.r.t. $\mathsf{P}_{\mathtt{Ps}}$, $\mathsf{M}_{\mathtt{Ps}}$, Unlnk follow from the proxied construction, and the cryptographic or inner protocol details (e.g., Visa, Mastercard variations) are irrelevant therein, as they are in the descriptions that follow.

We will now describe the functionality of our proposals and answer our first research question RQ1 by describing the main aspects and intricacy of `PrivBank` and `PrivProxy`.

6.1 Construction `PrivBank`

In `PrivBank` (Figure 1) a Proxy, contractually committed with the bank, intermediates all in-store transactions done

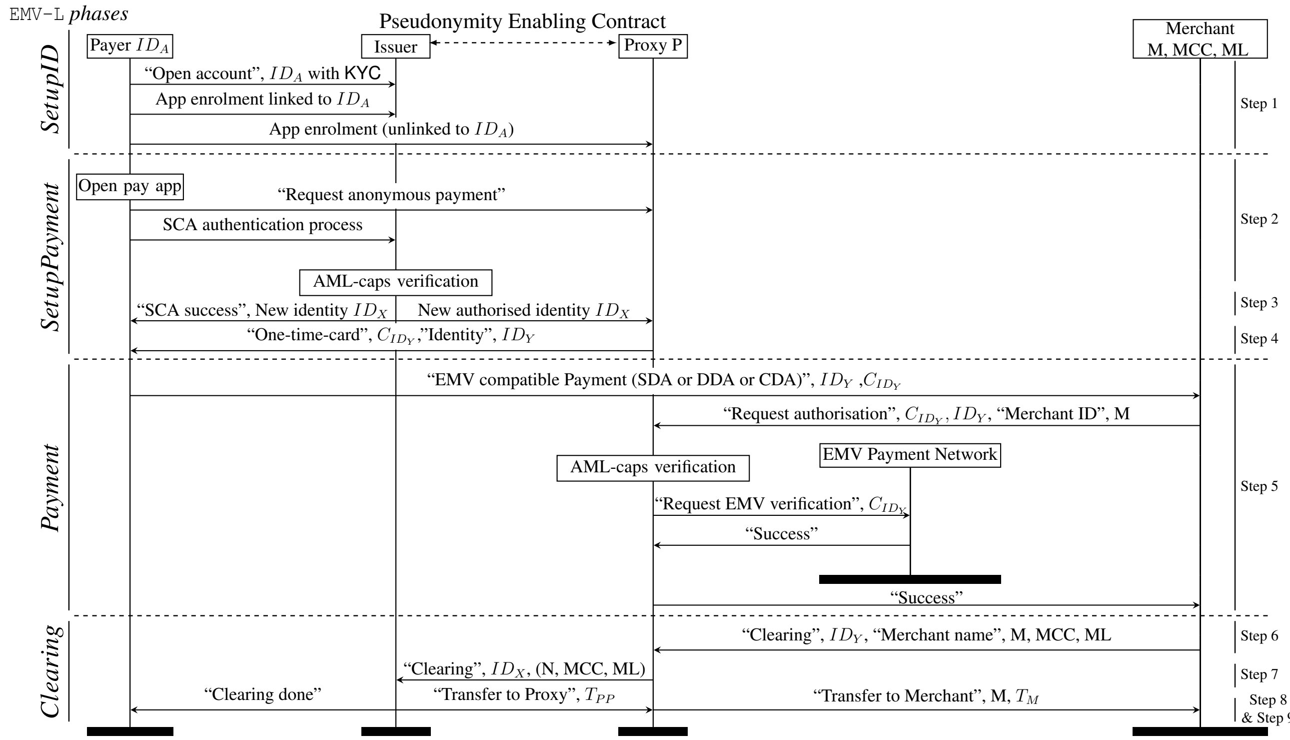


Figure 1: Protocol/Implementation flow of `PrivBank`. *All communications apart from the payment from the Payer to the Merchant are assumed to be executed on a secure channel (encrypted and authenticated communications).*

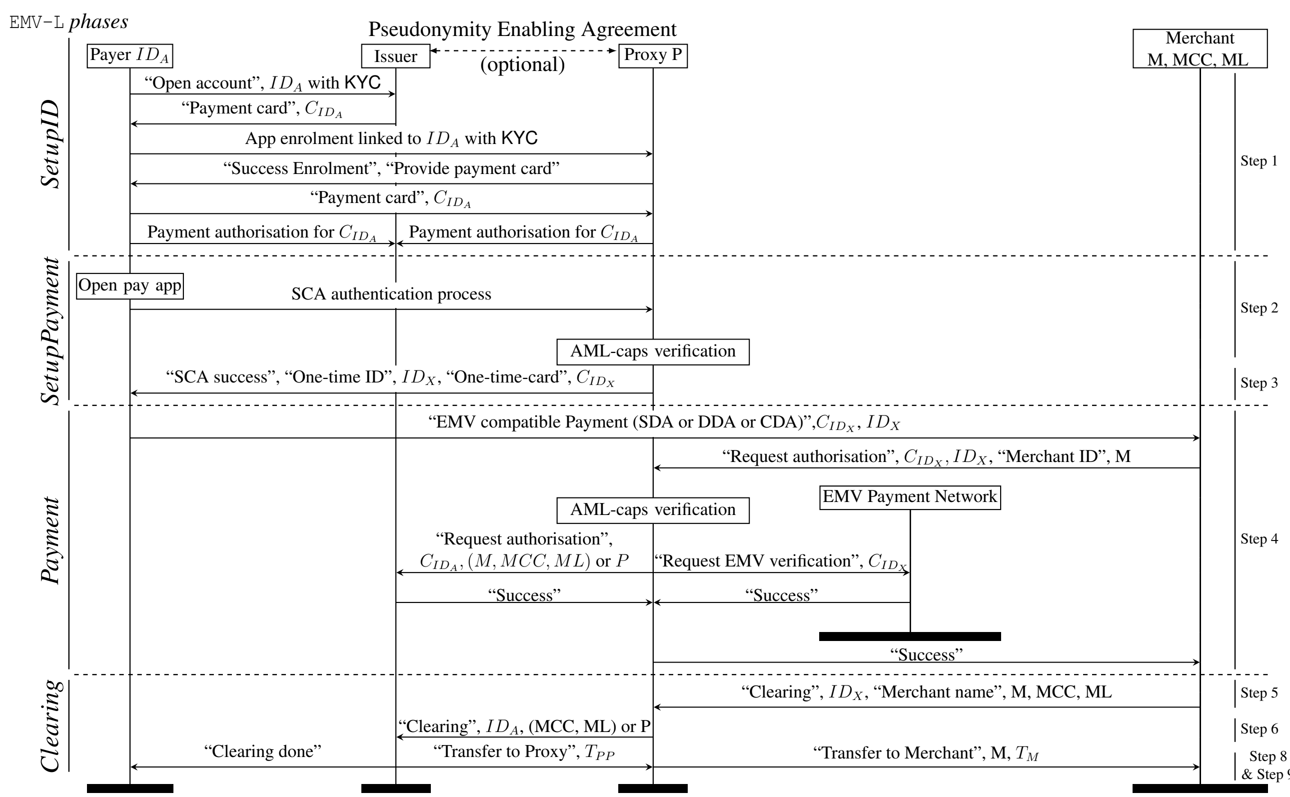


Figure 2: Protocol/Implementation flow of `PrivProxy`. *All communications apart from the payment from the Payer to the Merchant are assumed to be executed on a secure channel (encrypted and authenticated communications).*

by a Payer with a Merchant. The Proxy gets to know who the Merchants are but not the long-term identifiers of the Payers, whereas the bank knows who the Payers are but not the Merchants. As is required by AML regulation, the two entities can work together to recover full knowledge on any transaction. They have an agreement in place, thus sharing risk and liability. It's important to note that an agreement can create civil liability, e.g., to indemnify the bank for a fine if the Proxy fails to do something it has promised but it cannot apportion criminal liability or liability to pay a fine. Civil liability to the customer would only be by the bank. Payments done via `PrivBank` are supported via one mobile app[6] provided in partnership between the Issuer and the Proxy. We now describe, step by step, how a payment is made possible as well as carried through via `PrivBank`. These steps are also highlighted in Figure 1.

Step 1 (Registration, see *SetupID* in Figure 1). The Payer, whose identity is "ID_A" opens a bank account with the Issuer. This bank account comes with a "premium" option of support for privacy à la P_{Ps}, M_{Ps}, Unlnk, which is achieved via `PrivBank`. The Payer's banking account and their `PrivBank` account are with the Issuer, which handles KYC authentication, not the Proxy. The Issuer does not share the Payer's identification details with the Proxy. To allow this, the contract stipulates certain terms and conditions (T&C) as we will detail below.

As an account holder with the Issuer, the Payer accesses the `PrivBank` app, which connects the Payer, Issuer, and Proxy from an engineering standpoint. However, the app is provided by the Issuer and links only the app-store identifier to the Payer's identity ID_A on the Issuer's servers. The Proxy and other third parties identify the Payer through their app-store account, not the Issuer's method.

Under AML regulations, and similarly to other payment methods, the amounts spend using `PrivBank` may be capped (*e.g.,* EUR/GBP1000 per month); we call these the *AML caps*. The T&Cs set this limit, which the Issuer and Proxy enforce.

Step 2 (Authentication, see *SetupPayment* in Figure 1). When a payment is to be made by the Payer to a PoS of a Merchant's, the Payer ID_A opens the `PrivBank` app. The opening of the app prompts both the Issuer and the Proxy on secure (*e.g.,* HTTPS) channels:

(a) App-Triggers on the Proxy's Side: At this stage, the push by the app to the Proxy only says that someone, *with no specific identity revealed*, is intending making a payment.

(b) App-Triggers on the Issuer's Side: The SCA and AML checks are triggered through a request to the Issuer. SCA is a two-factor authentication, ensuring that Payer ID_A is making the payment. If needed, the Issuer also verifies that ID_A has not exceeded the AML caps for `PrivBank`. If the caps are reached or SCA fails, the protocol halts.

Upon successful SCA and AML checks, the Issuer creates a one-time virtual identity ID_X, pseudorandom and statistically independent of ID_A and any existing long-term card C_{ID_A}.

Step 3 (Pseudo-identity issuance, see *SetupPayment*). On the back-end (*i.e.,* not via the `PrivBank` app), the Issuer sends the identifier ID_X to the Proxy, which, in the light of Steps 1 and 2, only knows that an account holder with the Issuer using `PrivBank` wants to make a payment.

The Proxy expects this push[7], having received an alert in Step 2 that someone intends to make a payment.

Step 4 (One-time card issuance, see *SetupPayment*). At this stage, the Proxy issues – for whom it knows as Payer ID_X – a one-time, virtual, EMV-compliant card C_{ID_Y} with all aspects (PAN, certificates *etc.*) being freshly generated for one-time use, including an attached one-time, virtual card-holder name of "ID_Y". ID_Y/C_{ID_Y} are pseudorandom and statistically independent of ID_X and of ID_A/C_{ID_A}. The card issued is "loaded" onto the app.

Step 5 (Payment, see *Payment* in Figure 1). The Payer pays with it at the Merchant's PoS. Here, SCA authentication may have to be carried out offline using the CDCVM tag in the case of preloaded payment methods ID_Y/C_{ID_Y}. The Proxy (alternatively the Issuer) checks that it would not reach the AML caps through this specific amount being paid via `PrivBank` and executes the AML scrutiny. If any of these conditions fail, the protocol stops. AML caps were checked in Step 2 of `PrivBank`, but those checks did not account for the current payment.

Liability Shifts & Fraud-protection. Under its partnership with the Proxy, the Issuer accepts controlled *shift of liability* with respect to fraud protection. To this end, for *selected stores* – that are nominated based on MCC, MRI and ML, *etc.*– the partnership allows that the Issuer receives from the Proxy sanitised information. In practice, the *list of selected stores* can be large (*e.g.,* all Merchants in a country with given MCCs), as is the case for the "*Ticket Restaurant*" services with Edenred [10] or Up-one [18]. The sanitised information does not reveal the original Merchants' full identity, instead it contains what we call *pseudo-merchant identities*. These are prescribed, such that the Issuer can check the Proxy's compliance to the agreement[8].

In terms of fraud-detection disputes, the Proxy and the Issuer have to come together to resolve this, and the Proxy has to disclose to the Issuer the full Merchant data. This is reflected in the T&C of the contract that the Payer has with the Issuer on using the `PrivBank` product, *i.e.,* the Payer knows that it can use `PrivBank`, in selected stores.

Step 6 (Merchant clearing operation, see *Clearing* in Figure 1). Using standard EMV mechanisms, the (Acquirer of the) Merchant begins to resolve the payer's payment by

[6] For compliance with banking regulations, this app may require a Trusted Execution Environment (TEE), a chip designed for secure storage and cryptographic operations.

[7] The time between any push by the app in Step 2 and receiving ID_X is capped at 2 seconds due to EMV security constraints.

[8] E.g., a Merchant M in the country, with Merchant Location and Merchant Category Code getting pseudonymised as a fixed pseudo-merchant identity.

contacting the Proxy, which is the Issuer of C_{ID_Y}.

Step 7 (Proxy clearing operation, see *Clearing*). If the Merchant M is not on the "pre-selected" list, the protocol stops. Otherwise, using the PrivBank's back-end, the Proxy goes to the Issuer to resolve a payment for Payer ID_X, and provides a *pseudo-merchant identity N* instead of the true identity of the Merchant M.

Step 8 (Balance adjustment, see *Clearing*). The Issuer checks if Payer ID_X can pay to a pseudo-merchant N via PrivBank as per the pre-agreed list of merchants and as per the rules of PrivBank. Then, the Issuer further checks that Payer ID_X has funds to pay.

Step 9 (Balance adjustment, see *Clearing*). If step 8 went through, the payment is resolved towards the Proxy and then from the Proxy to the Merchant.

6.2 PrivBank Law & EMV Compliance

PrivBank align with the specification and all regulations applicable to the banking system. Firstly, note that when a payment is made with the PrivBank app, the transcript is the same as in one made with an EMV contactless payment card. The CDCVM is the only different aspect to standard EMV, and we detail this in the four paragraphs below; these also answer our research questions RQ1 and RQ2.

On Compliance with SCA. Strong customer authentication is checked by the Issuer when Payer ID_A intends to pay at Step 2 to provide ID_Y/C_{ID_Y} to Payer ID_A. Note that an SCA authentication (that may be carried out offline using the CDCVM tag) may be required for the payment if it is not executed within a few seconds after the first one.

On Timing Compliance w.r.t. the EMV System. EMV-compliant payments are required to set a maximum general processing time for each transaction. Allowed timings range between a few hundred milliseconds to a few seconds. In the stages defined in PrivBank, Step 1 corresponds to an initial setup independent of any payment. Steps 2 to 4 involve SCA authentication, up to the point where the card is loaded into the app. This process can be executed ahead of time for one or several one-time virtual cards. Steps 5 to 9 exactly correspond to a timed EMV process of payment. Over all, PrivBank results in a processing time within the range of current payment standards and similar to already deployed solutions such as tokenization or Curve. Thus, from a technical point of view, PrivBank complies with the standard.

On Compliance with KYC. KYC regulations are fulfilled via the Issuer, who checks Payers' identification documents upon them opening a bank account. A contract in between the Issuer and the Proxy mandate the Issuer for the identity verification. On the other side, there is a liability shift towards the Proxy on the verification of the Merchant's identity.

On Compliance with AML. The T&Cs of PrivBank subscribers are such that the amount of payments that any Payer ID_A makes via PrivBank will be capped to values as per AML regulations (*e.g.,* EUR/GBP1,000). Upon the AML verification done by the Proxy, if a breach is found by the Proxy, then an alert would be sent to the Issuer. The Issuer's officers would investigate and generate a so-called "Suspicious Activity Report (SAR)" [55] would send this to the authorities, since ultimate AML liability in PrivBank sits with the Issuer. The latter would need to collaborate with the Proxy to solve the case, and –for this– any Proxy and Issuer in PrivBank would need an initial legal agreement. The privacy of entities will be reverted in case of a SAR investigation.

6.3 Construction PrivProxy

PrivProxy (Figure 2) is based on the same three main parties, but while at the core of PrivBank there is a "pseudonymity-friendly" Issuer, now, in PrivProxy, it is a Proxy who provides a service to add pseudonymity on top of EMV payments. Note that there could be many such Proxies.

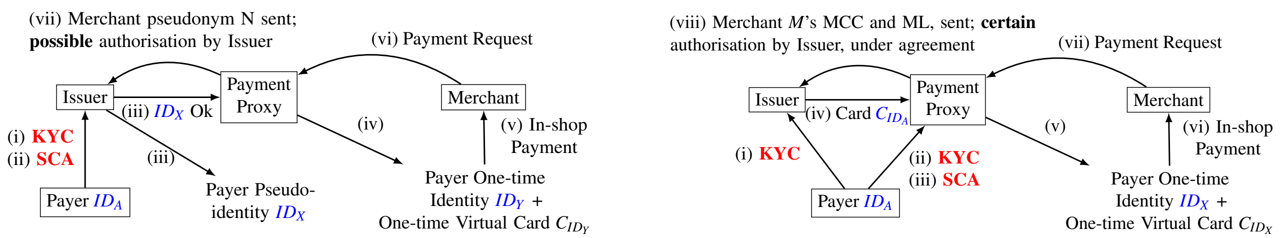
Step 1 (Registration, see *SetupID* in Figure 2). The Payer is known via their identity as "ID_A" by the Issuer and the Proxy, and it holds accounts with both. During the onboarding process, the Payer ID_A links the bank account they hold with the Issuer with the user account they hold with the Proxy.

EMV Compliance. At onboarding with PrivProxy, banking pre-authorisation is conducted, where the Proxy can take up to a fixed total from the Payer's bank account. In line with the EMV rules, this *pre-authorisation cap* can be a maximum of x amount per year/month/day (*e.g.,* EUR/GBP1000/month). The Payer gets access to the PrivProxy app as a Proxy-provisioned service and has to comply with KYC procedure via their identity ID_A to use it. This time, the app, from an engineering perspective, has nothing to do with the Issuer.

Step 2 (Authentication, see *SetupPayment* in Figure 2). When a payment is to be made by the Payer ID_A to a Merchant with identity M, the Payer opens the PrivProxy app. The opening of the app prompts the Proxy to do the SCA process, see *SetupPayment* in Figure 2. Upon successful SCA checks, the Proxy checks that ID_A has not reached their AML-cap. If they have, the protocol stops.

Step 3 (One-time card issuance, see *SetupPayment* in Figure 2). At this stage, for Payer ID_A, the Proxy creates a one-time virtual identity and a one-time EMV-compliant card, shown as ID_X and C_{ID_X} in Figure 2; they are pseudorandom and statistically independent of ID_A and C_{ID_A}. The card issued is automatically loaded into the PrivProxy app. The transcript produced between the app and the Merchant's PoS, when paying with this app, is that of the EMV physical card except for CDCVM.

Step 4 (Payment, see *Payment* in Figure 2). The Payer goes to pay with its one-time EMV-compliant virtual card by the Merchant's PoS. Accounting for the value of the current payment, the Proxy checks that the Payer's pre-authorisation caps are not reached, and nor are the AML-caps and executes



(a) `PrivBank`: EMV-compliant payments with pseudonymity provisioned by privacy-friendly Issuer, using a third-party Proxy.

(b) `PrivProxy`: EMV-compliant payments with pseudonymity provisioned by third-party Proxy.

Figure 3: Contrasting our proposals, especially on KYC, SCA, and identities.

the AML scrutinising.

Step 5 (Merchant clearing operation, see *Clearing* in Figure 2). Using standard EMV mechanisms, the (Acquirer of the) Merchant begins to resolve the payer's payment by contacting the Proxy, which is the Issuer of C_{ID_X}.

Step 6 (Proxy clearing operation, see *Clearing* in Figure 2). The Proxy behaves differently if the Merchant is on their pre-vetted selected stores list to use `PrivProxy` or not. In either case, the Proxy aims to provide Merchant pseudonymity and proceeds as described below.

If the Merchant is on the "selected-stores" list, then, using the payment networks' back-end, the Proxy matches ID_X with the payer's identity ID_A and asks the Issuer to resolve a payment for Payer ID_A. It does not declare Merchant identity M but its own identity P as being the Merchant.

If the Merchant is not on the "selected stores" list, the Proxy still does not declare the Merchant's identity M to the Issuer. Instead it provides restricted information (*e.g.,* ML and MCC) using the payment network's back end. In this case, the Proxy takes on risks in terms of their authorisation rates (*i.e.,* it is possible that the Issuer will not approve the payment due to too little information on the Merchant).

Liability & Fraud-protection. In terms of fraud-detection disputes, the Proxy takes on the liability. This is stipulated in the contract it has with the Payer, meaning it will have to reimburse the payer in some cases as it is the payment provider. Should the Issuer need to be involved, this is done entirely by the Proxy, with no legal obligation on the Payer.

Step 7 (Balance adjustment, see *Clearing*). If the payment is not authorised by the Issuer (which is unlikely for Merchants on the selected-stores list), the protocol stops. Otherwise, the Issuer checks if Payer ID_A has funds to pay and approves the transaction if it does.

Step 8 (Balance adjustment, see *Clearing*). If all went through the payment is resolved towards the Proxy and then from the Proxy to the Merchant M.

6.4 `PrivProxy` Law & EMV Compliance

`PrivProxy` also align with the EMV specification (answers RQ1) and all regulations applicable to the banking system (answers RQ2). Arguments similar to those set out in Section 6.2 apply. We detail below.

On Compliance with EMV Specifications. EMV-compliant one-time cards are issued and delays are managed in the same way as in `PrivBank`. Here, it is also possible to pre-load the pseudo-identity ID_X and the card C_{ID_X} in order to carry out EMV contactless payments with CDCVM authentication.

On Compliance with KYC. KYC regulations are fulfilled by the Issuer and the Proxy, who check Payers' identification documents upon them opening accounts with each.

On Compliance with SCA. Since the Proxy did KYC on-boarding of the Payer, the Proxy can do the SCA step and checks that it is indeed Payer ID_A attempting to pay.

On Compliance with AML. Like in `PrivBank`, here we have AML Caps and "Suspicious Activity Reports (SAR)" [55] if they are broken. But, unlike in `PrivBank`, the Proxy has all the payer's data and has ultimate responsibility in terms of AML; they will raise a SAR directly with the authorities, but the Issuer will be involved, as bound by KYC and other banking regulations. The privacy of entities will be reverted in case of a SAR investigation.

6.5 Comparing `PrivBank` & `PrivProxy`

Different Designs. The best way to see differences in designs between `PrivBank` and `PrivProxy`, as well as in the most-relevant protocol steps is Figure 3. A comparison between key properties of the two designs is given in Table 2.

Varying Features. For instance, `PrivBank` requires additional assumptions to enhance pseudonymity, such as a legal agreement of collaboration between the Proxy and the Issuer (Table 2, row 7). Conversely, `PrivProxy` could be offered by the Proxy independently of an Issuer, though legitimate payments may be rejected (row 10) due to the lack of such an agreement and the Proxy curating Merchant-related information of their own accord. Furthermore, `PrivBank` allows for multiple Proxies as partners of the Issuers, offering Payers a choice of providers; if it is the Payers who pay for the Proxy service, the balance of trust may shift, potentially making `PrivBank` more appealing to some Payers. Also, liability is shared in `PrivBank`, but not in `PrivProxy` (rows 3 & 4).

Criteria	PrivBank	PrivProxy
1. Payer pseudonymity	w.r.t. the Proxy and the Merchant ☺	w.r.t. the Merchant ☹
2. Merchant pseudonymity	w.r.t. the Issuer ☺	w.r.t. the Issuer ☺
3. Liability for legal compliance	Issuer and Proxy	Proxy
4. Liability for economic risk	Shared between Issuer and Proxy	Proxy
5. Payers' identities distributed	Yes ☹	No ☺
6. Payer's trust	In the Proxy and the Issuer	In the Proxy
7. System's assumptions	Trust between the Proxy and the Issuer ☹	None ☺
8. Security assumptions	Issuer's app not leaking identities ☹	None ☺
9. Feasibility	Privacy-friendly Issuers may be rare ☹	Immediately feasible by some companies ☺
10. False rejection rate	None ☺	Risks of low payment authorization rates ☹

Table 2: `PrivBank` and `PrivProxy`: Pseudonymity-provision, Advantages and Disadvantages.

`PrivBank` boasts stronger decentralisation of knowledge, but requires collaboration between Issuers and Proxies for the management of the app, as it necessitates a robust agreement (row 7). Furthermore, in reality, pseudonymity-friendly Issuers may be rare. Ultimately, the choice between the two proposals depends on priorities and incentives driving system deployment. Thus, we continue to present and analyse both `PrivBank` and `PrivProxy`, as they cater to different markets and operate under distinct assumptions.

Achieving $\mathsf{P_{Ps}}$, $\mathsf{M_{Ps}}$, Unlnk. `PrivBank` and `PrivProxy` offer different pseudonymity guarantees (see Table 1). Most of the relevant comparative aspects between our two protocols are detailed in Table 2 with the relevant design choices. illustrate that neither `PrivBank` nor `PrivProxy` can be deemed superior.

`PrivBank` achieves $\mathsf{P_{Ps}}$ from the perspective of the Proxy, which is down to the Issuer and the Proxy having only partial access to identifying information. This is not the case in `PrivProxy`, so $\mathsf{P_{Ps}}$ cannot be achieved there. All such results recounted in Table 1 are formalised in Section 7. The information essential for pseudonymity is divided between the Proxy (payment information) and the card Issuer (identification information). This is necessary to comply with legislation.

On Legal Compliance. For an overview of conformance with KYC and SCA regulations, see Table 1. In `PrivBank`, a legal agreement allows the Proxy to rely on the Issuer for KYC and SCA; thus, payments can proceed only when the Proxy has received SCA approval from the Issuer. This is unlike the case of `PrivProxy` where SCA is no longer required from the Issuer, as the Proxy has done KYC and there is also a pre-authorisation by the Payer on some of funds made to the Proxy during enrolment. In terms of AML, in `PrivBank` the main responsibility is with the Issuer, and in `PrivProxy` it is with the Proxy; but, in both cases, they share responsibility in the case of an AML official alert [55] and this is stipulated in their initial contracts.

On Using Proxies. The use of proxies in the banking system, such as Curve, and Revolut, is well-established, widely accepted, and utilised by millions of customers worldwide. Additionally, our design leverages the principle that banks do not require all merchant data for payment authorisation. This ensures that both the Issuer and the Proxy can potentially monetise the service, supporting the acceptability and feasibility of our approach. Meanwhile, the Issuer (*resp.* Proxy) does not have access to the Payer's full payement information (*resp.* account information), promoting greater data decentralisation and encouraging potential client adoption.

On Implementation Aspects. Both solutions/apps generate one-time cards (*i.e.,* single-use PANs) and run contactless EMV as for physical cards with the PoS. Loading a card mandates use of TEE to comply with the AML regulations. Alternatively, the server could load tokens associated with one-time cards. Tokens do not require to be securely stored. This second scenario generates a single-use PAR, but this element would lose its traceability purposes. Thus, we prefer the first solution. Apart from these minor considerations, our two solutions can be implemented on the basis of an adapted version of the services provided by existing proxies.

On Additional Demands. Our proposed solutions impose greater demands on the network: more communication and production of large volume of virtual cards. They also imply new shifts in liability for economic risk. Consequently, this form of payment may entail higher costs, which could be absorbed by merchants, as is the EMV specification, and/or by users opting for a "premium" privacy-preserving service.

7 Formal Treatment of Payments Anonymity

We introduce a formalism that allows us to reason formally about the privacy notions $\mathsf{P_{Ps}}$, Unlnk and $\mathsf{M_{Ps}}$. This formalism is accessible and answers RQ3. In the extended version [58], we give a "traditional" cryptographic model and analysis of our schemes; we also show that the "simpler" definitions here fully capture the cryptographic definitions.

7.1 Execution Model

To study the security/privacy of (payment) protocols, we consider the parties in the protocol, denoted by $\mathcal{P}arties$: *Payers*, *Issuers*, *Merchants*, and *Proxies*. These represent machines, devices or humans, associated with long-term identifiers, well-defined PPT (Probabilistic Polynomial Time) algorithms to execute, and they may hold cryptographic material. There can be any number of such parties, they are all executed concurrently and outputs of some are inputs for others, in the way of Interactive Turing Machines (ITMs) [64].

7.2 EMV-L: A Language for EMV Protocols

To describe the main protocols executed by our payment parties, we define a language EMV-L that, similarly to an API (Application Programming Interface), describes the main procedures and sub-procedures of any EMV-compliant payment protocol. We omit the setup of the EMV system as it is already in place and focus on the registration and payment for the payer *ID*.

Definition 1 (EMV-L: A Language for EMV Protocols) *EMV-L is formed of the following procedures:* $SetupID(ID) \rightarrow (\lambda_{ID}, C_{ID})$*: sets up the* execution environment *denoted* λ_{ID}*, in which a Payer party with identity ID can make payments. The object* λ_{ID} *enables and encapsulates all payment information related to the Payer ID. It may also produce a long-term card* C_{ID}*.*

$SetupPayment(ID) \rightarrow C$*: based on a Payer's identity ID and its execution environment* λ_{ID}*, creates an EMV-compliant* payment device *C (*e.g., *a physical card, or a mobile device with a card registered to it) that can transact with an EMV-compliant PoS.*

$Payment((ID, C), M) \rightarrow \mathsf{pay}$*: based on an identity ID, a payment device C correctly set up as above, and a Merchant M, generates an EMV-compliant* payment transcript pay.

$Clearing(ID, M, \mathsf{pay}) \rightarrow T$*: based on an identity ID, a Merchant M and a transaction* pay *produced as per the above, finalises the payment by balancing the account of the participants, and returns the* terminating data *T*.

Each transcript *tran* resulting from these procedures is divided between the entities $\mathcal{E}$ taking part. We later consider the restricted output of, *e.g.,* the Merchant *M* in the payment pay as pay^M. A standard in-shop EMV payment, or an EMV-compliant one such as our PrivBank and PrivProxy using the EMV-L language, follows flow of EMV-L procedures:

$$SetupID \rightarrow SetupPayment \rightarrow Payment \rightarrow Clearing.$$

SetupID is executed once per Payer, the other procedures can be run multiple times. The instantiation of PrivBank and PrivProxy in these procedures is detailed in Figures 1 and 2, as well as in the extended version [58].

In this study, we examine the above operations from a network perspective rather than delving into cryptographic considerations. Indeed, actions like registering a Payer with the Issuer does not involve cryptography. Only *SetupPayment* and *Payment* can be viewed (partially) as cryptographic protocols, respectively card key generation and payment protocol [44]. Moreover, the clearing process (*Clearing*) occurs between Issuers and the network lacking any unique or public specifications; so, we treat these operations as black boxes.

7.3 Formalising Payments Privacy

Pseudonymity ${\mathsf{P}_{\mathtt{Ps}}}^{ID}$ holds if, when a Payer *ID* makes a payment *P*, the adversary considered by our threat model (Section 3.3) cannot build a relation of the type "payment *P* is related to a Payer *ID*". Similarly, we define notions on mathematical relations to later describe $\mathsf{P}_{\mathtt{Ps}}$, Unlnk, and $\mathsf{M}_{\mathtt{Ps}}$. Consider *R*, a binary relation on pairs (x, y).

One-Way Relation. We say that *R* is *one-way* if for all *x*, for all $y \in R_x$ (all *y* such that $(x, y) \in R$), given *y*, finding *x* is hard[9].

Class Hiding Relation. We say that *R* is *class hiding* if for all *x* and *y*, given *x* and *y*, it is hard to determine if $(x, y) \in R$.

Intuitively, we formalise our property as the ability to correctly break these properties according to the transcript a subset of all entities $\mathcal{E} \subset \mathcal{P}arties$ may have. The concept of a one-way relation is meant to reflect the inability to recover $x = ID$, the identity of a payer, from $y = \mathsf{pay}$, the transcript of a payment made by *ID*. Meanwhile, the concept of a class hiding relation formalises the idea that, for two payments $x = \mathsf{pay}_1$ and $y = \mathsf{pay}_2$, one cannot determine whether a link exists between them—in this case, whether they were produced by the same payer. To this end, we formalise the relation for which the above properties should apply.

Definition 2 (Payer/Payments/Merchant Relation) *Let* $[\mathsf{Alg}]$ *be the set of possible outputs of the algorithm* Alg. *Let* π *be an EMV protocol described in EMV-L. Let* ID *be a set of at least two long-term Payers' identifiers. Let* CARD *be a set of at least two long-term Payers' cards. Let* M *be a set of at least two Merchant identifiers. Let* PAY *be a set of transcripts outputted by Payment, generated by Payers with identifiers in* ID *toward some Merchants M in* M. *We define the relations:*

Payer Relation. *The* payer relation $R_{\mathcal{P}\mathsf{Idt}} \subsetneq \mathsf{ID} \times \mathsf{PAY}$ *consists of all pairs* (ID, pay) *such that there exists* $(\lambda, C_{ID}) \in [SetupID(ID)]$ *and* $C \in [SetupPayment(ID)]$*, where* $\mathsf{pay} \in [Payment((ID, C), M)]$*. Informally, a pair* (ID, pay) *is in* $R_{\mathcal{P}\mathsf{Idt}}$ *if payment* pay *has been made by Payer ID.*

Card Relation. *The* card relation $R_{\mathcal{C}\mathtt{Idt}} \subsetneq \mathsf{CARD} \times \mathsf{PAY}$ *consists of all pairs* (C_{ID}, pay) *such that there exists* $(\lambda, C_{ID}) \in [SetupID(ID)]$ *which encompasses a card* C_{ID} *and* $C \in [SetupPayment(ID)]$*, where* $\mathsf{pay} \in [Payment((ID, C), M)]$*. Informally, a pair* (C_{ID}, pay) *is in* $R_{\mathcal{C}\mathtt{Idt}}$ *if payment* pay *has been made with the long-term card* C_{ID}*.*

Payments Relation. *The* payments relation $R_{\mathcal{P}\mathtt{ayms}} \subsetneq \mathsf{PAY} \times \mathsf{PAY}$ *consists of all pairs* $(\mathsf{pay}, \mathsf{pay}')$ *such that there exist ID, M, M',* $(\lambda, C_{ID}) \in [SetupID(ID)]$*, and* $C, C' \in$

[9] No PPT algorithm can compute it with non-negligible probability.

$[SetupPayment(ID)]$, *where* $\mathsf{pay} \in [Payment((ID,C),M)]$ *and* $\mathsf{pay}' \in [Payment((ID,C'),M')]$. *Informally, a pair of payments is in* $R_{\mathtt{Payms}}$ *if it was produced by the same Payer.*

Merchant Relation. *The* Merchant relation $R_{\mathsf{MIdt}} \subsetneq \mathsf{M} \times \mathsf{PAY}$ *consists of all pairs* (M, pay) *such that there exist ID,* $(\lambda, C_{ID}) \in [SetupID(ID)]$, $C \in [SetupPayment(ID)]$, *for which* $\mathsf{pay} \in [Payment((ID,C),M)]$. *Informally, a pair* (M, pay) *is in* $R_{\mathtt{Payms}}$ *if payment* pay *has been directed to Merchant* M.

Restricted Relations. *For each relation, we define the* restricted relation $R_*^{\mathcal{E}}$, *for* $\mathcal{E} \subset \mathcal{P}arties$, *by considering the restricted view of the payments,* i.e., *the restricted view of the transcripts in all the algorithms of Definition 1.*

Assuming that no two transactions within any of the protocol's views leads to identical data transcript (due to the timestamp amongst others), $\mathsf{PAY}^{\mathcal{E}}$ is in bijection with PAY and the elements in a relation $R_*^{\mathcal{E}}$ are in bijection with the elements in R_*. Hence, the restricted relations are well defined. By requiring intractability of properties of these relations, we define our three payments-privacy properties.

Payer Pseudonymity. This property entails that from all one-time payment-transcripts which an attacker sees, included from corrupted parties in a set $\mathcal{E}$, the attacker cannot link back to a correct long-term payer identity ID_A for a honest payer ID_A. Mathematically the payer relation $R_{\mathcal{P}\mathsf{Idt}}^{\mathcal{E}}$ formed by tuples $(ID_A, \mathsf{pay}^{\mathcal{E}})$ is one-way for the set $\mathcal{E}$ hence, the view of $\mathsf{pay}^{\mathcal{E}}$ of the payment pay. In the execution scenario considered the adversary also has access to the transcripts subsequent to the payment $\mathsf{pay}^{\mathcal{E}}$, *i.e.,* the clearing transcript and is an external observer of the other procedures. Payer's long-term cards C_{ID} can also be regarded as sensitive. The *pseudonymity w.r.t. the Payer's long-term card,* ${\mathsf{P}_{Ps}}^{C_{ID}}$, is defined similarly and can be considered against the same corruption set $\mathcal{E}$.

Definition 3 (Pseudonymity - ${\mathsf{P}_{\mathbf{Ps}}}^{ID}\&{\mathsf{P}_{\mathbf{Ps}}}^{C_{ID}}$**)** *Let* $R_{\mathcal{P}\mathsf{Idt}}$ *be a Payer relation defined by an EMV protocol described in EMV-L and* $\mathcal{E}$ *be a set of parties. We say that* ${\mathsf{P}_{Ps}}^{ID}$ *holds in front of a set* $\mathcal{E}$ *of parties if the relation* $R_{\mathcal{P}\mathsf{Idt}}^{\mathcal{E}}$ *is one-way for the payments made by an uncorrupted Payer ID,* i.e., *it is unfeasible to create ID from* $\mathsf{pay}^{\mathcal{E}}$ *and other transcripts seen by parties in* $\mathcal{E}$. *Similarly, pseudonymity* ${\mathsf{P}_{Ps}}^{C_{ID}}$ *is attained in front of the corruption set* $\mathcal{E}$ *if the relation* $R_{\mathtt{CIdt}}^{\mathcal{E}}$ *is one-way* i.e., *it is intractable to yield* C_{ID} *from* $\mathsf{pay}^{\mathcal{E}}$ *and other transcripts seen by parties in* $\mathcal{E}$.

Unlinkability. *Unlinkability* refers to the capacity to ascertain whether two payments originate from the same Payer. This property is formalised by saying that the attacker cannot form the relation whereby the pairs are two payments emitted by the same Payer; our notion for this is called 'class-hiding', *i.e.,* the attacker cannot form equivalence classes over payment transcripts, hence requiring the class-hiding property for the relation $R_{\mathtt{Payms}}^{\mathcal{E}}$, given again for a corruption set $\mathcal{E}$.

Definition 4 (Unlinkability - Unlnk**)** *Let* $R_{\mathtt{Payms}}$ *be a payments relation defined by a EMV protocol and* $\mathcal{E}$ *be a set of parties. We say that the protocol* attains unlinkability Unlnk for a set $\mathcal{E}$ of parties *if* $R_{\mathtt{Payms}}^{\mathcal{E}}$ *is class hiding for uncorrupted Payers.*

Merchant Pseudonymity. Merchant pseudonymity is defined similarly to Payer pseudonymity $\mathsf{P}_{\mathtt{Ps}}$. It is also based on the one-way property, but, this time, of a *Merchant relation* R_{MIdt}.

Definition 5 (Merchant Pseudonymity - $\mathsf{M}_{\mathbf{Ps}}$**)** *Let* R_{MIdt} *be a Merchant relation defined by a payment protocol and* $\mathcal{E}$ *be a set of parties. We say that the protocol* attains Merchant pseudonymity $\mathsf{M}_{\mathtt{Ps}}$ for a set $\mathcal{E}$ of parties *if the relation* $R_{\mathsf{MIdt}}^{\mathcal{E}}$ *is one-way for payments made to an uncorrupted Merchant.*

7.4 Provable Anonymity

We state our properties $\mathsf{P}_{\mathtt{Ps}}$, Unlnk, and $\mathsf{M}_{\mathtt{Ps}}$ against `PrivBank` and `PrivProxy`. As per our threat model, any observer between the Payer and the Merchant is at most as strong as a corrupt Merchant. Similarly, someone who breaks the back-end channel between the Merchant and the Issuer is at most as strong as a corrupt Issuer. So, we state our results below only w.r.t. corrupt parties.

We start with the pseudonymity of the Payer, which differs between our constructions. Indeed, a KYC procedure is required in `PrivProxy`, where the pseudonymity-friendly Issuer provides a one-time identity for the Payer to present to the Proxy in `PrivBank`. This KYC-based difference indirectly leads to:

Proposition 1 (`PrivBank` respects ${\mathsf{P}_{\mathbf{Ps}}}^{ID}$**)** *Consider an arbitrarily picked honest Payer with identifier ID, and* `PrivBank` *in the threat model given, where the Issuer which gives service to Payer ID is not corrupted. Then,* `PrivBank` *attains* ${\mathsf{P}_{Ps}}^{ID}$ *in front of the Proxy and the Merchant.*

No long-term card C_{ID} is ever produced by $SetupID(ID)$ in `PrivBank`. Hence, the pseudonymity ${\mathsf{P}_{\mathtt{Ps}}}^{C_{ID}}$ cannot be defined for `PrivBank`. Pseudonimity of cards is still guaranteed for any long-term card C_{ID} held by a Payer outside of `PrivBank`. Indeed, neither this card nor any data related to it would ever need to be provided by the Payer in `PrivBank`.

Proposition 2 (`PrivProxy` – ${\mathsf{P}_{\mathbf{Ps}}}^{ID}$ **and** ${\mathsf{P}_{\mathbf{Ps}}}^{C_{ID}}$**)** *Consider an arbitrarily picked honest Payer with identifier ID, and* `PrivProxy` *in the threat model given, where the Issuers and Proxies which give joint service to Payer ID are not corrupted. Then,* `PrivProxy` *attains* ${\mathsf{P}_{Ps}}^{ID}$ *and* ${\mathsf{P}_{Ps}}^{C_{ID}}$ *in front of the Merchant.*

We move to the attainment of payment unlinkability. In `PrivProxy`, the Proxy can link payments by virtue of controlling all sides of identifiers of the payers. In `PrivBank`, even if the Proxy does not know the Payer's ID_A, the Proxy can link their payments by using the Android/Apple account on the phone of the Payer for which is receives requests. We change this w.r.t. the app (especially since it is provisioned by the Issuer); however, this would complicate the resolution/authorisation of payments by the Proxy towards the Issuer. This leads us to the result below.

Proposition 3 (`PrivBank` and `PrivProxy`– Unlnk) *Consider an arbitrarily picked honest Payer with identifier ID, and `PrivBank` and `PrivProxy` in the threat model given, where the Issuers and Proxies which give joint service to Payer ID are not corrupt. Then, `PrivBank` and `PrivProxy` attain* Unlnk *in front of the Merchant.*

We move to Merchant pseudonymity. Our result is:

Proposition 4 (`PrivBank` and `PrivProxy` – M_{Ps}) *Consider an arbitrarily picked honest Merchant with identifier M, and `PrivBank` and `PrivProxy` in the threat model given, where the Proxies and Payers which jointly pays to Merchant M are not corrupted. Then, `PrivBank` and `PrivProxy` attain* M_{Ps} *in front of the Issuer.*

All proofs are included in our extended version [58].

7.5 On Privacy Treatments

Our privacy properties are specific to payment systems; moreover, pseudonymity is a restricted form of anonymity. However, our notions follow commonplace privacy definitions in the cryptographic setting. In the extended version of this manuscript [58], we indeed show that our definitions can be re-cast and re-proven via standard game-based, cryptographic proofs [61]. Concretely, P_{Ps} and Unlnk are both recast, in the game-based formalism, via indistinguishability-style games. As such, their nature is to protect identities strictly regarding protocol data and fields. Orthogonally, well-known studies [40] link payments via behaviour (time, GPS, shopping patterns), using data science. Our notions, even if lifted to temporary identifiers, cannot distinguish such "hidden" links, as they focus solely on protocol transcript-based relations. Profiling attacks [40] are not addressed by our protocols.

Our formalism resembles [33], where privacy is defined using attacker-infeasible relations between identifiers and secure-layer messages. Unlike our one-wayness for P_{Ps} and M_{Ps}, [33] assumes direct relation formation. Privacy relations in [33] use ephemeral identifiers, like protocol data. Similarly, we could redefine our privacy by targeting short-term payer identifiers (ID_X or ID_Y). [33] aligned relational privacy on ephemeral IDs with *weak unlinkability* and the work on long-term IDs with *strong unlinkability* by Arapinis *et al.* [26].

Our attacker is any PPT algorithm, which is known to be stronger than a Dolev-Yao attacker [22]. So, any Dolev-Yao attack translatable to our PPT relational definition applies here. In some cases, an attacker's "if-then-else" tests cause errors, enabling them to mount privacy attacks, by distinguishing to whom a message belongs; this is akin to reverting payment relations and break P_{Ps} (or stronger ephemeral-ID versions). For example, [37] show (an implementation of the) e-passport failing privacy, due to unmasked errors. However, if `PrivBank` and `PrivProxy` randomise all transcripts as specified, no such leakage occurs, and P_{Ps} holds against such attacks. Orthogonally, some attacks do not apply here: counter-based one (*e.g.,* [33] on IoT), as all counters are randomised

8 Discussions and Conclusions

On Legal Adoption. A main hurdle around adoption of privacy-enhancements in EMV is that they must not infringe the laws and regulations that are relevant: AML, KYC, PSD2 and SCA. Regarding KYC, PSD2, SCA, the concerns are around dealing with "common" impersonation fraud; for this, in our proposals, the agreements put in place between the Issuers and the Proxies would have to abide by laws such as the Proceeds of Crime Act 2002, hinging on their shared/sole liability; this is legally straightforward. Managing regulatory enforcements for anti-money laundering at scale is more complex. In Sections 6.2 and 6.4, we discussed how each proposal will deal with this, in accordance to the AML regulation and the guidance by the Financial Conduct Authority.

On Practical Adoption. Perhaps surprisingly, there is no technical or legal barrier to adoption. The practical hurdle is the sheer volume of virtual cards to be inserted in the payment systems; in fact, if PrivBank or PrivProxy would be entirely feasible, if offered only to a select few. To this end, a product director at Curve states: *"PrivBank and PrivProxy appear feasible in practice. These schemes make considerable progress on how to deal with AML compliance, and it seems feasible to make agreements with banks on how to deal with liabilities and the exact merchant data that proxies would send to issuing banks to balance 'merchant anonymity' vs. good payment-authorisation rates. The key practical challenge for Curve would be the significant cost of* managing *the high volume of one-time virtual cards, especially while complying with the business rules and regulations, across different countries and their Payment Services."*

So, we proposed PrivBank and PrivProxy: EMV-compliant, law-abiding solutions achieving payment pseudonymity, unlinkability, and merchant pseudonymity. We formalised and proved these, using a new privacy model. Our proposals are industry-implementable, and we will pursue practical studies.

Acknowledgements. We thank Robin Savage, Head of Payment Products at Curve, for useful insights into real-life payments systems and valuable feedback on our constructions.

Ethical Considerations. We rigorously adhere to ethical standards; for instance, here we used no sensitive or personal financial data. We focused solely on publicly available information and theoretical models. We carefully considered the potential impact of our findings on security and privacy, striving to enhance the payment ecosystem responsibly without introducing risks to users or financial institutions.

Open-Science Considerations. While our research does not involve the generation of data, code, or related materials, we remain committed to the principles of open science. We will ensure that our findings and methodologies are thoroughly documented and openly shared through the publication, including on open repositories, in some format. By doing so, we aim to contribute valuable insights to the community and facilitate further research and discussion in the field.

References

[1] The electronic money regulations 2011, 2011. https://www.legislation.gov.uk/uksi/2011/99/2016-03-01, Last accessed on 11-16-23.

[2] Central Bank Digital Currency: functional scope, pricing and controls, 2021. https://www.ecb.europa.eu/pub/pdf/scpops/ecb.op286~9d472374ea.en.pdf, Last accessed on 11-16-23.

[3] Cental bank accounts are dangerous and unnecessary, a critique of two papers, 2022. https://taler.net/papers/accounts-dangerous-2022.pdf, Last accessed on 11-16-23.

[4] The ECASH Act, 2022. https://ecashact.us/, Last accessed on 11-16-23.

[5] Overview of EMVCo, 2022. https://www.emvco.com/about/overview/, Last accessed on 11-16-23.

[6] Paypal: Account limits, 2022. https://www.paypal.com/uk/smarthelp/article/faq1253, Last accessed on 17-05-22.

[7] Visa prepaid reloadable personal cards, 2022. https://www.visa.co.uk/content/VISA/usa/englishlanguagemaster/en_US/home/pay-with-visa/cards/prepaid-cards/all-purpose-reloadable.html, Last accessed on 01-06-22.

[8] Curve: Rule your money, 2023. https://www.curve.com/en-gb/, Last accessed on 11-16-23.

[9] ebay payment terms of use, 2023. https://pages.ebay.co.uk/payment/2.0/terms.html, Last accessed on 11-16-23.

[10] Edenred, 2023. https://www.edenred.fr/ticket-restaurant, Last accessed on 02-24-24.

[11] Google pay help: United kingdom: Supported payment methods, 2023. https://support.google.com/pay/answer/7351534, Last accessed on 11-16-23.

[12] Lyf, 2023. https://www.lyf.eu/en/, Last accessed on 02-23-24.

[13] Mastercard prepaid | just load and pay, 2023. https://www.mastercard.co.uk/en-gb/personal/find-a-card/general-prepaid-mastercard.html, Last accessed on 11-16-23.

[14] Otac: A new paradigm for user authentication and device authentication, 2023. https://www.swidch.com/technology/otac, Last accessed on 11-16-23.

[15] Set up apple pay, 2023. https://support.apple.com/en-us/HT204506, Last accessed on 11-16-23.

[16] Shop with amazon pay, 2023. https://pay.amazon.co.uk/using-amazon-pay, Last accessed on 11-16-23.

[17] Spend safely online with single-use cards, 2023. https://www.revolut.com/cards, Last accessed on 11-16-23.

[18] Up-one, 2023. https://up-one.up.coop, Last accessed on 02-24-24.

[19] Mastercard payment account reference inquiry, 2024. https://developer.mastercard.com/payment-account-reference-inquiry/documentation/, Last accessed on 08-29-24.

[20] Takepayments: What are contactless payments and how do they work?, 2024. https://www.takepayments.com/blog/product-information/what-are-contactless-payments-and-how-do-they-work/, Last accessed on 08-29-24.

[21] Visa scan to pay, 2024. https://developer.visa.com/innovation-corner/example-projects/scan-and-pay, Last accessed on 02-27-24.

[22] M. Abadi and P. Rogaway. Reconciling Two Views of Cryptography (The Computational Soundness of Formal Encryption). *J. of Cryptology*, 20(3):395–395, 2007.

[23] UK Public General Acts. The money laundering and terrorist financing (amendment) regulations 2019. https://www.legislation.gov.uk/uksi/2019/1511/contents/made, Last accessed on 08-30-24.

[24] UK Public General Acts. Proceeds of crime act 2002. https://www.legislation.gov.uk/ukpga/2002/29/contents, Last accessed on 08-30-24.

[25] Nicholas Akinyokun and Vanessa Teague. Security and privacy implications of nfc-enabled contactless payment systems. In *Proceedings of the 12th international conference on availability, reliability and security*, 2017.

[26] Myrto Arapinis, Tom Chothia, Eike Ritter, and Mark Ryan. Analysing unlinkability and anonymity using the applied pi calculus. In *2010 23rd IEEE computer security foundations symposium*, pages 107–121. IEEE, 2010.

[27] Financial Conduct Authority. Financial crime guide:a firm's guide tocountering financialcrime risks (fcg). https://www.handbook.fca.org.uk/handbook/FCG.pdf, Last accessed on 09-02-24.

[28] David Basin, Ralf Sasse, and Jorge Toro-Pozo. The emv standard: Break, fix, verify. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021.

[29] Mike Bond, Omar Choudary, Steven J Murdoch, Sergei Skorobogatov, and Ross Anderson. Chip and skim: cloning emv cards with the pre-play attack. In *2014 IEEE Symposium on Security and Privacy*. IEEE, 2014.

[30] Ioana Boureanu, Liqun Chen, and Sam Ivey. Provable-security model for strong proximity-based attacks: With application to contactless payments. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, 2020.

[31] Ioana Boureanu, Tom Chothia, Alexandre Debant, and Stéphanie Delaune. Security analysis and implementation of relay-resistant contactless payments. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, 2020.

[32] Christina Brzuska, Nigel P Smart, Bogdan Warinschi, and Gaven J Watson. An analysis of the emv channel establishment protocol. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013.

[33] Ksenia Budykho, Ioana Boureanu, Stephan Wesemeyer, Daniel Romero, Matt Lewis, Yogaratnam Rahulan, Fortunat Rajaona, and Steve Schneider. Fine-grained trackability in protocol executions. In *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*. The Internet Society, 2023.

[34] Jeffrey Burdges, Florian Dold, Christian Grothoff, and Marcello Stanisci. Enabling secure web payments with gnu taler. In *Security, Privacy, and Applied Cryptography Engineering: 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings 6*. Springer, 2016.

[35] Sergiu Bursuc, Ross Horne, Sjouke Mauw, and Semen Yurkov. Provably unlinkable smart card-based payments. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023.

[36] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology*, Boston, MA, 1983. Springer US.

[37] Tom Chothia and Vitaliy Smirnov. A traceability attack against e-passports. In Radu Sion, editor, *Financial Cryptography and Data Security*, pages 20–34, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[38] Véronique Cortier, Alicia Filipiak, Jan Florent, Said Gharout, and Jacques Traoré. Designing and proving an emv-compliant payment protocol for mobile devices. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2017.

[39] Veronique Cortier and Ben Smyth. Attacking and fixing helios: An analysis of ballot secrecy. In *2011 IEEE 24th Computer Security Foundations Symposium*, pages 297–311, 2011.

[40] Yves-Alexandre de Montjoye, Laura Radaelli, Vivek Kumar Singh, and Alex "Sandy" Pentland. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science*, 347(6221):536–539, 2015.

[41] Joeri De Ruiter and Erik Poll. Formal analysis of the emv protocol suite. In *Theory of Security and Applications: Joint Workshop, TOSCA 2011, Saarbrücken, Germany, March 31-April 1, 2011, Revised Selected Papers*. Springer, 2012.

[42] Florian Dold. *The GNU Taler system: practical and provably secure electronic payments*. PhD thesis, Université Rennes 1, 2019.

[43] Danny Dolev and Andrew Yao. On the security of public key protocols. *IEEE Transactions on information theory*, 1983.

[44] LLC EMVCo. Emv integrated circuit card specifications for payment sys- tems, book 2, security and key management, version 4.3, 2011.

[45] LLC EMVCo. Emv next generation. next generation kernel system architecture overview., 2014.

[46] LLC EMVCo. Emv payment tokenisation specification technical framework v2.3. *EMVCo: Foster City, CA, USA*, 2022.

[47] LLC EMVCo. Emv annual report 2023: Enhancing emv technologies to supporting emerging payments. *EMVCo: Foster City, CA, USA*, 2023.

[48] LLC EMVCo. Emv payment tokenisation – a guide to use cases. *EMVCo: Foster City, CA, USA*, 2023.

[49] EUR-Lex European Union. Regulatory technical standards for strong customer authentication (sca) and common and secure open standards of communication. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02018R0389-20230912, Last accessed on 02-21-24, 2018.

[50] EUR-Lex European Union. Directive on payment services in the internal market. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02015L2366-20151223, Last accessed on 02-21-24, 2021.

[51] EUR-Lex European Union. Directive on the prevention of the use of the financial system for the purposes of money laundering or terrorist financing. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02015L0849-20210630, Last accessed on 02-21-24, 2021.

[52] L.-A. Galloway and T. Yunusov. First contact: New vulnerabilities in contactless payments. In *Black Hat Europe*, 2019.

[53] Ross Horne, Sjouke Mauw, and Semen Yurkov. Unlinkability of an improved key agreement protocol for emv 2nd gen payments. In *2022 IEEE 35th Computer Security Foundations Symposium (CSF)*. IEEE, 2022.

[54] ICANN (Internet Corporation for Assigned Names and Number). Information for Privacy and Proxy Service Providers. https://www.icann.org/resources/pages/pp-services-2017-08-31-en, 2004.

[55] AML Legislations. Suspicious Activity Report Documentation. https://www.lawsociety.org.uk/topics/anti-money-laundering/suspicious-activity-reports.

[56] Steven J Murdoch, Saar Drimer, Ross Anderson, and Mike Bond. Chip and pin is broken. In *2010 IEEE Symposium on Security and Privacy*. IEEE, 2010.

[57] Satoshi Nakamoto and A Bitcoin. A peer-to-peer electronic cash system. *Bitcoin.–URL: https://bitcoin.org/bitcoin. pdf*, 2008.

[58] Charles Olivier-Anclin, Liqun Chen, Tom Chothia, Anna Clee, Andreas Kokkinis, Pascal Lafourcade, Chris Newton, and Ioana Boureanu. Who pays whom? anonymous emv-compliant contactless payments. 2025. https://hal.science/hal-04917364.

[59] Andreea-Ina Radu, Tom Chothia, CJ Newton, Ioana Boureanu, and Liqun Chen. Practical emv relay protection. In *Proc. 43rd IEEE Symp. Security Privacy*, 2022.

[60] Rosehaslina Razali. Overview of e-cash: Implementation and security issues, 2002. https://www.giac.org/paper/gsec/1799/overview-e-cash-implementation-security-issues/103204, Last accessed on 11-16-23.

[61] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. http://eprint.iacr.org/2004/332, 2004.

[62] Aleksei Stennikov Timur Yunusov, Artem Ivachev. New vulnerabilities in public transport schemes for apple pay, samsung pay, gpay. *White Paper*, 2021.

[63] Visa. Minimum data requirements for merchants, 2021. https://www.elavon.ie/content/dam/elavon/en-gb/documents/customer-centre/customer-news/merchant-best-practice-infographic.pdf, Last accessed on 02-21-24.

[64] Peter Wegner. Interactive foundations of computing. *Theoretical Computer Science*, 1998.

[65] Timur Yunusov. Hand in your pocket without you noticing: Current state of mobile wallet security. *Black Hat Europe*, 2021.

Auxiliary Supporting Material

A Acronyms

In this paper, we frequently use various acronyms to streamline the presentation of complex terms and concepts. For the reader's convenience, we provide a list of these acronyms along with their full forms below in Table 3.

B Traditional Payment Systems & Their Attainment of Payment-Privacy

The following are well-known, "traditional"[10], in-shop payment solutions[11]. We look at them and consider aspects linked

[10] They do not use crypto currencies.

[11] Online payments are included here as they may give some insights into what might be possible

EMV	Europay Mastercard Visa
EMV-L	EMV language
EUR	Euro
AML	Anti-Money Laundering
KYC	Know Your Customer
SCA	Strong Customer Authentication
PSD2	Payment Services Directive (version 2)
MCC	Merchant Category Code
MN	Merchant Name
MRI	Merchant Risk Index
ML	Merchant Location
PAN	Application Primary Account Number
PAR	Payment Account Reference
TEE	Trusted Execution Environment
CVM	Cardholder Verification Method
CDCVM	Consumer Device CVM
CBDC	Central Bank Digital Currency
PSP	Payment Service Provider
T&C	Terms and Conditions
CBDC	Central Bank Digital Currency

Table 3: List of the Most Used Acronyms.

to their provision of KYC of pseudonymity for the payer. A summary of their properties has been given in Section 4, Table 1. In this section, we provide a description of each of them and give reasons for the claimed properties.

1. Cash (banknotes and coins). This works for in-shop purchases only. There is no KYC and strong pseudonymity; although shops may use of security cameras and subsequent review may allow the payer to be identified, this is beyond the scope of the payment methods. Banknotes might be traced, based on their serial numbers; but that requires a complicated set of step taking by different banks and the Merchant, which again makes the matter beyond into a complex type of identifiability. Coins which do not feature serial numbers. Thus, we can say that cash has no KYC and provides pseudonymity and unlinkability (within reasonable/normal measures). For the same reason, namely that cash cannot be linked to the payer, the SCA cannot be required before payments.

2. Cheques. They provide a payment mechanism which involved banks as intermediary between the payer and the merchant or the entity paid. In addition, cheques require the name of the payer and the merchant to be written on them, as well as an bank account number of the payer. The account number is also divulging to the bank issuing the cheque to the payer. This method therefore has KYC in place, and does not provide pseudonymity in any of the possible ways. In many countries, it has become common practice to require proof of identity in order to make a payment. In such cases, the payer is strongly authenticated and SCA holds.

3. E-cash. This was originally proposed by Chaum in his 1983 paper on "Blind Signatures for Untraceable Payments" [36]. Payers have an e-wallet topped-up with e-cash, from which they spend e-cash like they would real cash, in principle; limits on the expenditure exist though, e.g., for AML reasons. Thus, this digital mechanism aims to provide the user with the same level of pseudonymity that they achieve when using cash. If the link between the e-cash wallet and the owner's bank account is just for applying limits to the amount of e-cash stored on a wallet, then the merchant should not obtain information about the customer and even the bank should gain little information. How it is regulated and implemented will determine the outcomes which may differ from Table 1.

Now, we give further details on e-cash, since it is an interesting variant to what we do. It needs a separate network to EMV, but it has supporters, still, today. Chaum commercialised this idea, founding DigiCash in 1994. This and other early e-cash developments are described in [60]. Although none of these have been commercially successful, there is still interest in developing e-cash systems [34].

In the US, the "Electronic Currency And Secure Hardware Act" [4] proposes that an electronic dollar should be created with the same privacy properties as the dollar itself. If achieved there would be no *Static Data Authentication* (SDA) requirement and this could also be used for online purchases, although for full pseudonymity the purchaser may need to use a VPN to hide their IP address, use a separate e-mail account and a delivery locker for their purchase.

In contrast, the European Central Bank's document discussing e-cash (Central Bank Digital Currency, CBDC) [2] recommends that to avoid too much money being stored in consumer's digital wallets (and not in the banks) limits should be applied and to enable this the consumer would need to be identified and their CBDC wallet linked to their bank account. Others [3] opposed it, and it is not clear what might ultimately be decided.

4. Plastic/physical credit or debit cards. This are to be used in-shop or online, to make payments in ways we all know. In Section 5.1, we already explained what KYC, SCA and pseudonymity they provide and why.

5. Google Pay [11] and Apple Pay [15]. These are two of the most used methods of mobile payment, *i.e.,* payment via a mobile app "inside" which a physical card is registered. Like in physical cards, KYC and SCA are the norm here. In mobile-payments SCA, when making the payment, the customer may be asked to confirm their identity onto the payment device too, via PIN, fingerprint or face recognition. But, unlike payments by physical cards, making and authorising mobile payments need more intermediaries in the payment networks, and tokenisation (as outlined in Section 5.1). Due to this tokenisation mechanism, the merchant can link purchases by the same payer using their long-term PAR – created once during the onboarding of their card onto the app and used in all their payments thereafter. The PAR replaces the PAN, which is then hidden from the merchant, making payments anonymous but not unlinkable.

6. Pre-paid cards. These are cards which may have no bank account associated with them and one tops up with a set amount or one buys already topped up and uses. We divide pre-paid cards into two categories:

a) Top-up cards – cards offered with services such as those by Revolut [17]. Other providers exist: for example, UK pre-paid Mastercard cards' providers, listed on the Mastercard website, for general use and as gift all behave as per the below. The Payer may not have to undergo credit checks, but must satisfy identity and address checks and have money available in their account to cover any payments made. So, KYC is generally done; indeed, most of the issuers of such pre-paid cards act as electronic money institutions and are regulated by the Electronic Money Regulations [1]. Since these are cards, their pseudonymity properties are the same as plastic cards, or that mobile apps – if they are loaded therein.

b) Gift cards – card that can be bought is store cards, with set amounts preloaded onto them. There is generally no KYC done. Related to this, their use is restricted (to specific merchants) and the amount on each card is small, to satisfying the AML requirements [51]. If they are purchased with cash and not linked to a bank account (for re-charging, for example), then subject to the same caveats about IP addresses, e-mail accounts and deliveries, payer pseudonymity can be achieved.

7. Virtual or "one time" cards (VC) These provide pseudorandom card details (card number, expiry date, CVV), for each transaction. You can remain anonymous to the merchant, but the virtual card is linked to your 'real' card to enable payment to be made and so the issuer knows who the payer is and from whom they are purchasing. These are marketed for online use, in general. One example is a card offered the Revolut card [17]. Another company, Swidch [14] offer a range of services based on One Time Access Codes (OTAC) and this includes *ephemeral* cards; as for other virtual cards these are linked to a registered *real* payment card (requiring KYC). In this context, an intermediary may be acting as a payment proxy although how individual providers handle the payments differs from one company to another. In addition, unless strict usage limits are applied, SCA is required for all payments.

8. Payment service providers There are a number of providers in this category, for example:

PayPal PayPal offers a range of services. In terms of the discussion here, PayPal accounts can be used to make payments. Figure 4 shows how PayPal is used for making payments. The stages are:

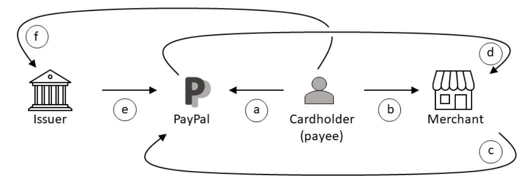


Figure 4: Making payments with PayPal.

1. The cardholder opens an account and registers a card to be used for payment. SCA for the Issuer is carried out at this point, but is not necessary afterwards. Unless the account holder confirms their identity and address (KYC) accounts are restricted and have limits placed on amounts that can be sent, received, or withdrawn [6]. PayPal is acting as the *payment service provider* (PSP) and knows who you are and gets to see who you are buying from and when.

2. The cardholder purchases an item from a merchant and pays using their PayPal account. Purchases can be made online, or in person.

3. The merchant receives their payment *from PayPal*. And not directly from the payer, thus the purchaser may use a pseudonyme and separate e-mail account and a delivery locker for their purchase to obtain full pseudonymity.

4. PayPal pays the merchant.

5. PayPal charges the issuer. The issuer knows that something was purchased, but not who from.

6. The cardholder pays the issuer.

In this case, payment or identity information has been provided to PayPal, but has not been made known to the merchant because PayPal has filtered most of it. Other information, such as e-mail, may be pseudonyms, not linked to the payer's identity.

Amazon Pay. Amazon Pay [16] offers a similar service and allows paying online with a credit card, debit card or by direct debit. They make it clear that the merchant does not receive your payment details: "We do not share your full credit card, debit card, or bank account number with sites or charitable organisations that accept Amazon Pay. The merchant only receives information that is required to complete and support your transaction. This information may include your name, email address, and shipping address.".

Curve. Curve [8] provides a payer with a card and a payment application. Curve users must satisfy the KYC rules. The payer registers multiple bankcards issued by one or several banks in the Curve app. When the payer pays with their Curve card or with the Curve app, one of their Curve-registered bankcards is charged in principle, but the payment is not "settled" entirely (*i.e.,* the payment onto the classical card is "pending"). Curve pays the merchant on behalf of the payer on the spot, but Curve also provides the payer a period of 30-days to potentially move the transaction to another of their registered, classical cards. So, there is an intricate process of what is called "payment authorisation" between Curve, the issuers of classical bankcards and other bodies (*e.g.,* Visa, Mastercard). During the payment authorisation, all the information required is shared without filtering, so that all the entities know each other's identities.

9. Online marketplaces. Examples of these are provided by Amazon and eBay. We view these are merchants here. So, from that viewpoint, clearly, here is generally no KYC or SCA needed to open accounts with them, as a payer for their goods. However, eBay, for example, states in their terms [9] that they may require "any other data about the buyer which the buyer's payment service provider or we may require". Aside from that, if goods bought from them are sent to a pickup locations, then some degree of pseudonymity can be achieved.

Further Payment Alternatives. Cryptocurrencies [57] are alien to EMV. But, non-EMV payments close to EMV exist. For instance, Lyf [12] and Visa [21] propose payment services which rely on their own payment network and QR codes. Or, a large-scale, EU-funded project tries to push new payments based on the GNU Taler initiative [34] and using the well-known e-cash idea by Chaum [36]. They perform online transactions and are compliant with online-payments' regulation; they do not use the card-to-PoS-merchant payment networks like us. Attaining privacy via online transactions is easier – *e.g.,* via one-time cards without the worry of "in-shop" SCA but relying on 3D secure, without having to share credentials over an app between different-domain entities.

C Our Solution At A Glance

We took inspiration from existing payment systems: plastic-card and mobile EMV, disposable EMV cards, proxying of EMV payments by Curve [8] and machinations during EMV-payment authorisation.

(A). On Payers' Pseudonymity and Payments' Unlinkability. The main inspiration for our designs here come from mobile EMV-payments and one-time/disposable cards for online shopping, and we bring the later into the space of "in-shop" payments. As a result of enhanced security, mobile payments are already more privacy-preserving than plastic cards as they hide the main identifier of the physical card, the *Primary Account Number* (PAN), via an ephemeral card-like number called *tokenised PAN*, which contributes to payers' pseudonymity. Yet, mobile-payments made via the same bankcard still contain the fixed card-identifying *Payment Account Reference (PAR)*; this leads to payments being linkable. All of our designs will revert to tokenisation and PAN-PAR-based constructions in mobile payments. Instead, our mobile apps will utilise one-time disposable cards which will produce transactions as per plastic cards, which is akin to having a one-time PAN.

(B). On Merchants' Pseudonymity. Here, we take inspiration from EMV-payment proxies such as Curve [8] (see Section 5). We add an intermediary in the interaction between the payer and the merchant, which also relays the payment to the issuing bank, but stripped of certain merchant-related data. In more detail, based on an agreement between the issuer and the proxy w.r.t., *e.g.,* certain categories of merchants with sufficiently low *Merchant Risk Indicators*, the proxy omits sending the merchant name to the issuer, while still providing the latter with some merchant identification data.

However, there is one last hurdle to our designs, chiefly the sets of regulations, as follows.
(1) AML and counter-terrorism financing regulations require payments' auditability by certain payment-system parties, therefore, for any transaction, the payer and the merchant must be traceable.
(2) SCA/*Payment Services Directive* (PSD2) require identification of payers prior to using a payment service, including opening bank-accounts and making payments.

So, we carefully combined the ideas in (A) and (B) above to achieve payers and merchants pseudonymity as well as payments unlinkability, while still achieving EMV-compliance and abiding by regulations (1) and (2) above.

To achieve this, some entities retained some of the identity information required and, when all combined, the systems obtained are in accordance with regulations (1) and (2) above.

D Proofs of our Main Results

This is in the extended version of this manuscript [58].

E Game Based Formalisation

This is in the extended version of this manuscript [58].

F Sample Real Card Traces

This is in the extended version of this manuscript [58].

G Sample Mobile Application Traces

This is in the extended version of this manuscript [58].