

RADBOD UNIVERSITY NIJMEGEN

MASTER THESIS

Threesomes on the Internet
Investigating the security of the iDEAL payment
system

Supervisor:
dr.ir. E. Poll

Author:
Willem Burgers
s0814830

Second supervisor/reader:
prof. dr. E.R. Verheul

Third supervisor:
J. de Ruiter, MSc



November 13, 2014

Abstract

There are many online webshops that sell all sorts of products. In order to pay for these products, people often use online payment systems. In the Netherlands the most used online payment system is iDEAL [19, 6]. Fraud with online transactions has been an issue in using online payment systems.

Even though fraud on online payment systems seems to be on its retreat in the Netherlands[6, 23], a new attack on the iDEAL payment system surfaced, which is explained in a recent news article[4]. This attack makes it possible to manipulate a client into paying for some product he does not want and even send it to an attacker. Because the attacker is disturbing the relation between client and merchant and is buying a different product with the money of the client, the news article spoke of a triangular transaction. Hence the name of this thesis is: Threesomes on the Internet.

Currence, the company behind the iDEAL payment system, responded swiftly by making the merchants (webshop owners), who sell 'anonymous' products, implement a countermeasure. By 'Anonymous' products we mean products that have an equivalent value in money and are easily transferable without registration of a transfer.

Unfortunately the countermeasure is not implemented by every webshop that sells anonymous products. It is hard to control every merchant. Only upon reports of fraudulent transactions can merchants be forced to implement countermeasures. Apparently, the scale of the attack was small and therefore not every merchant was contacted.

In this thesis we want to find a better countermeasure for this attack by examining the iDEAL protocol and request details. In order to find the root cause of the problem, the attack is described in detail. By comparing iDEAL with other (well known) payment systems, other countermeasures for both the iDEAL protocol and the request details can be determined. In particular, the description field of the iDEAL protocol is not properly used by most merchants. This thesis also discusses counterattacks on the proposed countermeasures. Finally other measures are examined that are not related to the iDEAL protocol or the request details. The conclusions of this thesis include recommendations for the different parties involved in iDEAL. The recommendation easiest to implement is for the banks to show the domain part of the *returnURL* field in the iDEAL protocol on the transaction approval screen. Furthermore should merchants fill the description field to contain product information instead of order numbers.

Keywords and phrases: online payments, security protocols, iDEAL.

Acknowledgements

Writing this thesis was quite a struggle for me. First and foremost I would like to thank my supervisor Erik Poll, without whom this thesis would be a great mess. Erik has always been patient with me, even when I did not make as much progress as he would like me to make. He always saw the bigger picture, when I was focused on the commas. This applies to both the structure of this thesis as well as the contents. Erik provided me the topic of this thesis.

I would also like to thank Eric Verheul for being my second supervisor and second reader for this thesis as well as for taking the time to review this thesis and giving me feedback. His knowledge of both the banking world and the iDEAL payment system provided for an in depth review.

I would like to thank Joeri de Ruiter for being my third supervisor. His technical view on this topic has been very helpful. Joeri always provided great examples of payment systems during the discussions about this thesis. He uses many payment systems and often told a funny anecdote about the systems.

Additionally, I would like to thank Arjan Lamers at First8, who found the time to give me an introduction to the world of online payment systems. He showed that most systems work very similar and he showed the general concepts of online payments. He also gave his visions on the improvement of iDEAL.

I also want to thank Jan-Paul Ciere, Monique van der Horst and Oscar Covers at Betaalvereniging Nederland for reviewing a draft version of this thesis. Currence is still the legal company to own the iDEAL payment system, but Currence is now part of Betaalvereniging Nederland. Jan-Paul, Monique and Oscar took the time to meet with me and Erik to discuss this thesis.

Furthermore I want to thank my family for reviewing a draft version of this thesis and for supporting me.

Finally I like to thank Maja Vasić for supporting me throughout the writing process and for pushing me to keep on going, whenever I did not feel like writing.

Contents

Acknowledgements	2
1 Introduction	4
2 The iDEAL payment system	6
2.1 Parties involved	6
2.2 The iDEAL protocol	6
2.3 Bank specific transactions	8
2.4 CPSPs	14
2.5 <i>TransactionRequest</i> details	15
2.6 Courtesy refunds	17
3 Attack	19
3.1 Bitcoin transaction in the background	20
4 Root cause analysis	24
5 Similar systems	26
5.1 PayPal	26
5.1.1 The PayPal protocol	26
5.1.2 Differences between iDEAL and PayPal	26
5.2 Credit Card payment	27
5.2.1 SecureCode or 3-D Secure	29
5.2.2 Differences between iDEAL and credit card	29
6 Countermeasures	33
6.1 Countermeasure A Register client's bank account number at merchant	33
6.2 Countermeasure B IP addresses	34
6.3 Countermeasure C Shared cookies	35
6.4 Countermeasure D Give more semantics in the transaction details	36
6.5 Countermeasure E Address details via the bank	37
7 Trustmarks	38
7.1 Effectiveness of trustmarks	38
7.2 Finding invalid trustmarks	39
7.3 European trustmark	42
8 Privacy	43
8.1 Client information sent to the bank	43
8.2 Client information sent to the merchant	44
9 Related work	45
10 Future work	47
11 Conclusions	48
11.1 Specific recommendations	50

1 Introduction

Online shopping is still increasing in popularity today. Many websites offer all different kinds of products for a client to buy. Merchants (owners of webshops) can choose which payment systems they will accept in their webshop. In the Netherlands, one payment system for doing online purchases is most often used by clients[6]. This payment option is iDEAL. The iDEAL payment system is released by Currence in 2005. This payment system builds on the online banking application of the different banks in the Netherlands. The online banking applications with multifactor authentication were already popular, so this was a logical step forward in paying using banking transactions[11]. The iDEAL payment system is based on a four-party model: A client, a merchant, the merchant's bank and the client's bank.

Early 2014, an attack method involving iDEAL was reported by a news website and a consumer television program. These attacks were of a new form and therefore newsworthy. The consumer television program MeldPunt! by Omroep MAX gave a broader view of online webshop fraud and how clients can easily be misled[15], but the news article by ComputerWorld described the new attack in more detail[4]. An attacker was able to perform a man-in-the-middle (MITM) attack such that clients were paying for a transaction that the attacker created instead of paying for a product they chose. The attack in short works as follows:

1. The attacker creates a fake webshop with attractive products
2. Clients go to the fake webshop to buy a product
3. The client selects to pay using iDEAL
4. The attacker goes to a legitimate merchant that sells 'anonymous' products
5. When the client is ready to pay, the client pays for the 'anonymous' product

By anonymous products we mean products that have an equivalent value in money and are easily transferable without registration of a transfer. Examples of anonymous products are: calling credit, game credit, gift cards and bitcoins. Because the attacker is disturbing the relation between client and merchant and is buying a different product with the money of the client, the news article spoke of a triangular transaction. Hence the name of this thesis is: Threesomes on the Internet. The attack allows the attacker to create a transaction for an anonymous product, while the client is browsing the attacker's fake webshop. An attacker is interested in anonymous products, because he can easily get away with these products, without leaving a trail.

Currence immediately took action against this attack by forcing merchants who sell anonymous products to implement a countermeasure[4]. The merchants have to verify the bank account number of the client before the client is able to buy the product. This should make the attack impossible. This specific attack

on iDEAL is no longer reported in the media, so the countermeasure succeeded in stopping the attack. Unfortunately the countermeasure is not implemented by every webshop that sells anonymous products. It is hard to control every merchant. Only upon reports of fraudulent transactions can merchants be forced to implement countermeasures. Apparently, the scale of the attack was small and therefore not every merchant was contacted. The countermeasure itself is still not the best possible solution for the problem. In this thesis this specific attack is examined. In order to do that, this thesis focuses on the factors that enable the attack to work and what can be done about it. The research question posed in this thesis is:

What are the differences between iDEAL and other payment systems that allow a client who pays with iDEAL to be manipulated into buying a product they did not order themselves and how can this attack be prevented?

To answer this question, the iDEAL protocol is further examined in Section 2. The attack is described in more detail in Section 3. The root causes of the problem are described in Section 4. In Section 5, other well known online payment systems are examined to determine if the same attack is possible. Section 6 discusses the possible countermeasures against the attack. Section 7 discusses other methods to provide the client with trustworthy information about webshops. In Section 8, the privacy issues of iDEAL transactions are discussed. Section 9 and Section 10 discuss related work and future work respectively. Finally conclusions are drawn and recommendations are specified for the different parties involved in the iDEAL payment system in Section 11.

The recommendation easiest to implement in order to mitigate this attack is for the banks to show the domain part of the *returnURL* field in the iDEAL protocol on the transaction approval screen. The field is already used in the protocol, so the protocol itself does not have to be changed. Banks only have to add the domain part to the transaction approval screen such that the client is able to verify that the domain indeed matches the domain of the webshop where the client went to buy his products. This single countermeasure can already clear up a lot of uncertainties about the recipient of the transaction. It is not expensive to implement but will benefit the payment system.

2 The iDEAL payment system

With more and more online shopping websites, in 2005 the Dutch organisation Currence came up with an online payment system. This system was based on the already existing Internet banking applications by the different banks. Using this system is not only easier for merchants than using banking transactions, but also more familiar for clients. The client gets redirected to the website of his or her own bank. The user interface of the bank is the same for online banking as for this new transaction system called iDEAL. Any client with a payment account at a Dutch bank can use iDEAL.

2.1 Parties involved

There are several parties involved in an iDEAL payment transaction[12]. First of all we have the client (Client *A* in Figure 1) who wants to buy a product from the webshop. The administrator or owner of the webshop is called a merchant in the iDEAL payment system (Merchant *B* in Figure 1). The merchant interacts with his own bank, also called the *acquiring bank* (Merchant's bank *C* in Figure 1) and finally there is the bank of the client, also called the issuing bank (Client's bank *D* in Figure 1). So-called Collecting Payment Service Providers (CPSPs) can act in the name of an acquiring bank. More information on CPSPs in the iDEAL payment system can be found in Section 2.4.

2.2 The iDEAL protocol

When a client wants to buy a product at a webshop and the iDEAL payment method is chosen, the iDEAL protocol is initiated. The steps of the protocol mentioned below match the steps in Figure 1. The interaction between the merchant *B* and the merchant's bank *C* is specified in the merchant integration guide provided by Currence[12]. The interaction between the client *A* and the merchant *B* is not provided in the merchant integration guide, but can be derived from usage and the merchant integration guide. The interaction between the merchant's bank *C* and the client's bank *D* is proprietary. Some details can be derived from the merchant integration guide. The steps of the protocol are:

0. The client *A* indicates he wants to pay using iDEAL.
1. The first action is for the merchant *B* to ask his bank *C* which other Dutch banks are able to participate in an iDEAL transaction in a '*DirectoryRequest*'. These are the so-called *issuing banks*.
2. The merchant's bank *C* provides the merchant *B* with a list of possible issuing banks.
3. The merchant *B* lists these issuers on his website such that the client *A* can select it using a form.

4. The client *A* selects his or her own bank from the list.
5. Now the merchant knows the issuing bank of the client *D* and sends a ‘*TransactionRequest*’ to the merchant’s bank *C* by requesting the Application Programming Interface (API) URL at the merchant’s bank *C* given in the merchant integration guide. This request includes the identifier of *D*, a *returnURL* that returns the client to the webshop after payment, an *entrance code* (*ec*) as well as the details about the order at the webshop, such as total amount and possibly an order number. Whether or not the order details are included and which order details are included in the *TransactionRequest* is up to the merchant. For more details about the data in the *TransactionRequest* see Section 2.5.
6. Upon a new *TransactionRequest*, the merchant’s bank *C* will negotiate a new transaction with the issuing bank *D*. This part of the protocol is proprietary. Only the parties that have a license or a certificate (see Section 2.4) from Currence can get access to this part of the protocol. We do not have access, so we don’t know what exactly goes on here. We believe the merchant’s bank *C* creates a *transaction id* (*trxid*) in this step.
7. It is trivial that the issuing bank *D* will reply with at least the *issuerURL*, the link where the client *A* can pay for the order.
8. The merchant’s bank *C* then sends a transaction object to the merchant *B* which includes the (*trxid*), the same *issuerURL* and a signature over the important data. This is the response to the *TransactionRequest*. The merchant *B* checks the data for validity using the signature.
9. The merchant *B* then redirects the client *A* to the provided *issuerURL* of the issuer bank *D* with an HTTP redirect.
10. The client *A* requests the *issuerURL* at the client’s bank *D*.
11. The client’s bank *D* responds with the transaction approval page.
12. The client *A* can now verify and approve the transaction at his or her own bank *D* with the same mechanism the bank has for doing online banking. Some banks use *TAN-codes* provided over the mobile network to the client’s phone and others use a separate hardware device or token to approve the transaction using the client’s bank card.
13. If the approval of the transaction succeeds, the money is transferred.
14. When the transaction is either approved or cancelled or when an error occurred, the client’s bank *D* will redirect the client back to the *returnURL* of the webshop with the *trxid* and the *ec*.
15. Upon the call of the *returnURL* with the *trxid* and the *ec*, the merchant will know that the transaction is completed with an unknown status (either verified, cancelled or an error occurred). The *trxid* and *ec* are included to make sure the client can only return after a completed transaction.

16. To get the status of the transaction, the merchant B will send a ‘*StatusRequest*’ to the merchant’s bank C .
17. The merchant’s bank C responds with a ‘*StatusResponse*’, which includes the status and the amount payed and a signature over important data. It is up to the merchant B to verify if the data is correct and if the amount is equal to the amount he asked for. If the status of the transaction equals ‘*Success*’, the money is now guaranteed to be in possession of the merchant B . He can now proceed to ship the product to the client.

On the iDEAL website by Currence, there is a promotion video that shows the steps of the protocol¹.

2.3 Bank specific transactions

As said in Section 2.2, every bank has its own way of online banking. Some make use of hardware tokens and others require username and password credentials and a so-called *TAN-code* via SMS. The authentication method however is not really relevant for the attack that will be shown in Section 3. There are other differences between the banks that allow the attack to be performed. These differences are in the details shown on the transaction approval screen, being differences in terminology and differences in what details are shown on the page. Figures 2a to 2h give examples of the iDEAL transaction pages for two banks, namely ABN AMRO and ING. The screenshots in Figures 2a to 2h are all taken from the demo of the iDEAL website[5], so they are not from a live banking transaction. We don’t have an account for every bank, but these screenshots allow us to show a full transaction from start to end. The differences between banks are visible in these screenshots, and can therefore be used. The only difference with a live transaction is the ‘omschrijving’ field. In the demo screenshots it shows the name of the book that is bought. In a live transaction it mostly shows the order number (see Section 4 for more information). This is the *description* field in the iDEAL protocol, that is intended for product details, but many merchants just repeat the *purchaseID* field in the description (see Section 2.5 for information on the different fields).

As you can see in Figures 2a to 2h, there are small differences between the different banks. There are some terminology differences, like “leverancier” (supplier) in Figures 2b to 2d and “begunstigde” (recipient) in Figures 2f and 2h. This difference in terminology is strange, because the party that will supply the goods to the client might not necessarily be the direct recipient of the money. This is especially the case for CPSPs (see section Section 2.4). Recipient is the better term, because it is the recipient of the money, not the supplier of the goods. Different banks may choose to label the fields differently on the transaction approval screen, because the banks want the iDEAL interface to

¹<http://www.ideal.nl/ontvangen/video/>

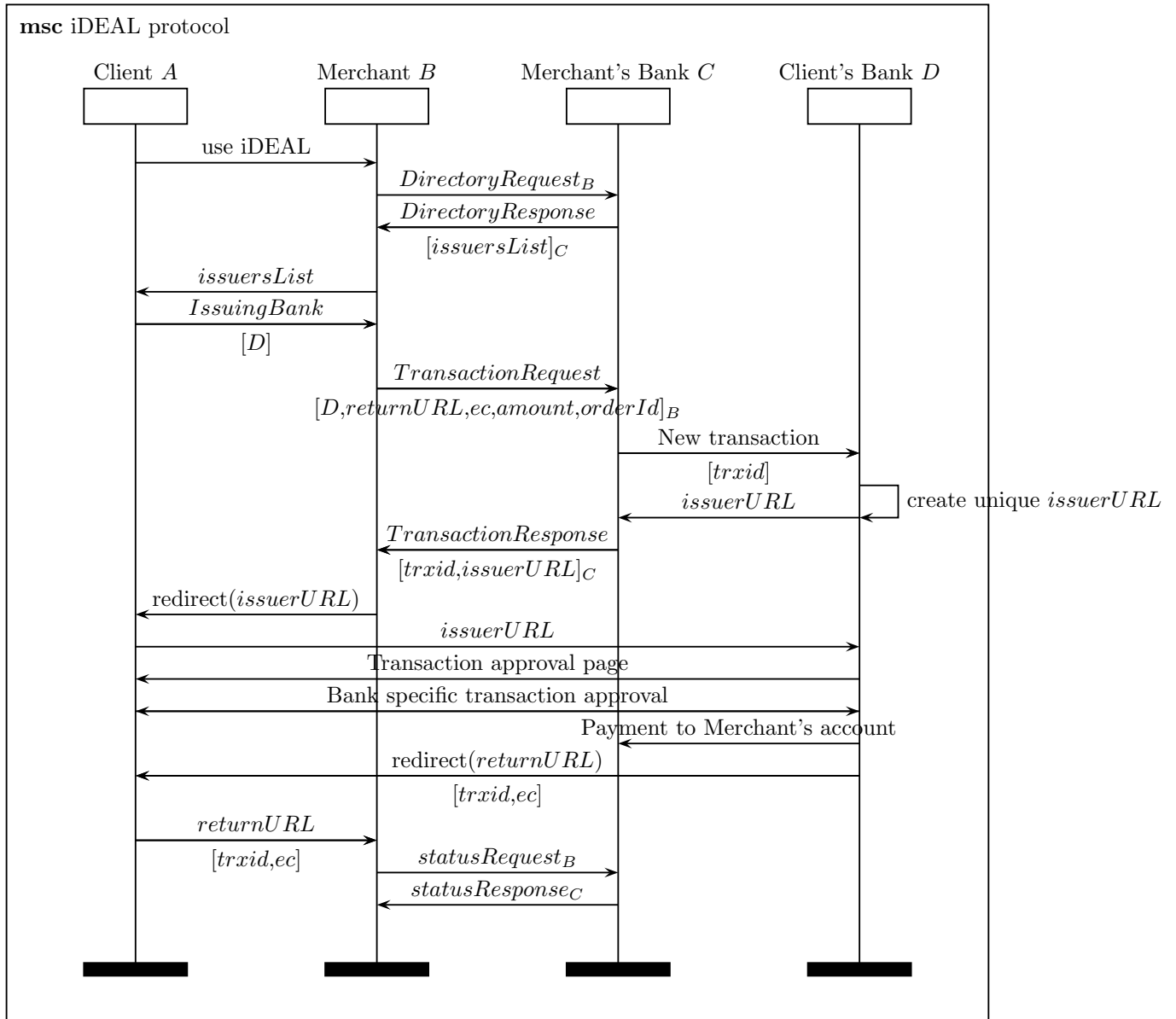


Figure 1: iDEAL protocol

iDEAL - Aankoop betalen



welkom bij ABN AMRO IDEAL

De webwinkel heeft voor u een betaalopdracht klaargezet. Controleer of het slotje in uw browser aanwezig is en u een [verbinding](#) hebt met ABN AMRO. U kunt nu betalen. Daarna keert u terug naar de webwinkel.

LET OP! Voortaan staat hieronder de nieuwe e.dentifier2 standaard geselecteerd! Wanneer u de andere e.dentifier gebruikt vergeet deze dan niet eerst te selecteren.

bestelling

leveragesier	RoekenGigant
kenmerk	Aankoop boek
datum	27-03-2014
omschrijving	Het Diner, Koch, Herman
bedrag	EUR 24,95

Invoeren rekening- en pasnummer

Welke e.dentifier hebt u? e.dentifier2 e.dentifier

rekeningnummer

pasnummer

Klik op "OK & Verder" of druk op ENTER Onthoud mijn gegevens

Wilt u betalen via een andere ABN AMRO rekening? Of hebt u een vraag over iDEAL? Raadpleeg dan de [help](#).

Controleer of het internetadres begint met <https://ideal.abnamro.nl>...

Controleer of het slotje in uw browser aanwezig is en u een [verbinding](#) hebt met ABN AMRO.

Lees meer over [veilig betalen](#) via iDEAL.

(a) ABN AMRO uses a hardware token to login and approve the transaction


look as much as possible like online banking. Therefore there is no uniform agreement on terminology of labels.

Another difference is that some banks show more information on the transaction approval screen than other banks. ABN AMRO shows a bank reference to the transaction, labeled 'kenmerk'. Every bank gives a unique reference to a transaction, but only ABN AMRO shows this in the transaction screen. Van Lanschot and ING are the only banks that show the iDEAL transaction id (*trxid*) in the transaction approval screen. ING and KNAB are the only banks that show both the *description* field and the *purchaseID* field.

After the approval, all banks give an overview that the transaction is indeed approved and the recipient and amount are shown again (for example in Figure 2d). All banks show more information in this overview. Namely information that was not available to the client before approving the transaction, like a transaction number. This is the 16 digit long iDEAL transaction number. Some also show a 'betalingskenmerk' (transaction reference). This is most likely the *ec* defined by the merchant or the transaction reference of the bank (like ABN AMRO displays before the approval). It is unclear why some banks do show this and why some banks do not. There is no added value to showing the *trxid* and *ec* to the client unless transactions could be refunded, which is not the case for iDEAL transactions. An example of the transaction number can be found in Figures 2d and 2h.

iDEAL - Aankoop betalen

betaal instructie


 Plaats met behulp van uw e.dentifier uw elektronische handtekening. Als u op 'Verzenden' klikt, wordt de betaling direct uitgevoerd. Hebt u meerdere betaalrekeningen? U krijgt de mogelijkheid om de rekening te selecteren, waarvan u het bedrag wilt laten afschrijven.

bestelling

leverancier: BoekenGigant
 kenmerk: Aankoop boek
 datum: 27-03-2014
 omschrijving: Het Diner; Koch, Herman
 bedrag: EUR 24,95

op uw e.dentifier2

- > Voer uw pas in
- > Druk op **2** Verzend opdr.
- > Toets uw pincode in
- > Druk op **OK**
- > Toets de volgende code in **7467 9654**
- > Druk op **OK**
- > Een response wordt getoond



Vul hier de response in

Klik op **"Verzenden"** onder in dit scherm. Uw betaling wordt direct uitgevoerd.

(b) ABN AMRO uses a hardware token to login and approve the transaction

iDEAL - Rekening selecteren

betaal instructie

 Van welke betaalrekening wilt u het bedrag laten afschrijven? Als u op 'Verzenden' klikt, wordt de betaling direct uitgevoerd. U ontvangt daarna een bevestiging van uw betaling.

bestelling

leverancier: BoekenGigant
 kenmerk: Aankoop boek
 datum: 27-03-2014
 omschrijving: Het Diner; Koch, Herman
 bedrag: EUR 24,95

Rekeningselectie

rekeningnummer	soort	tenaamstelling	saldo/waarde
12.34.56.789	PRIVEREKENING	J A Q0BFDSXF	1.234,56 EUR
BETAALREKENINGEN			
12.34.56.789	PRIVEREKENING	J A Q0BFDSXF	1.234,56 EUR
98.76.54.321	PRIVEREKENING	J A Q0BFDSXF	1.234,56 EUR
19.28.37.465	PRIVEREKENING	J A Q0BFDSXF	1.234,56 EUR

(c) ABN AMRO uses a hardware token to login and approve the transaction

iDEAL - Bevestiging

help | Deze pagina printen

U hebt betaald! Vergeet niet uw bestelling af te ronden onderaan de pagina.

bevestiging
Uw betaling is succesvol verwerkt.
Druk op "**Bestelling afronden**" om het bestelproces af te ronden bij uw leverancier. U kunt daar ook terecht met vragen over uw aankoop.

bestelling

leverancier	BoekenGigant
ten gunste van	NL13TEST0123456789
kenmerk	Aankoop boek
datum	27-03-2014
omschrijving	Het Diner; Koch, Herman
transactienummer	0030000151033669
bedrag	EUR 24,95

gegevens betaalrekening

rekeningnummer	12.34.56.789
tenaamstelling	A L XAEQSYN

[Bestelling afronden](#)

(d) ABN AMRO uses a hardware token to login and approve the transaction



Start iDEAL betaling (stap 1 van 4)

Om met iDEAL te betalen, logt u in op Mijn ING.
Controleer of het internetadres begint met <https://ideal.ing.nl/> en of u het slotje in de browser ziet..

Inloggen Mijn ING

Gebruikersnaam

Wachtwoord

Onthoud mijn gebruikersnaam

[Inloggen](#) [Annuleren](#)

[Gebruikersnaam en/of wachtwoord vergeten?](#)



Wat is iDEAL?
Veilig betalen met iDEAL

Help

(e) ING uses credentials to login and a TAN-code to approve the transaction



Selecteer betaalrekening (stap 2 van 4)

Selecteer de betaalrekening waarmee u de iDEAL-betaling wilt doen.

Bedrag	€ 24,95
Naam begunstigde	BoekenGigant
Datum	17-09-2012
Mededelingen	Omschrijving: Het Diner; Koch, Herman Ordernummer: 933816643272912099

Selecteer betaalrekening

Betaalrekening	€ 5.000,00
NL99 BANK 0123 4567 89	Hr C Monet

[Selecteren](#) [Annuleren](#)



► Help

(f) ING uses credentials to login and a TAN-code to approve the transaction



TAN-code invoeren (stap 3 van 4)

Controleer het aantal opdrachten, het volgnummer en totaalbedrag in het scherm. Zoek het volgnummer op in uw TAN-lijst en vul de bijbehorende TAN-code in.

Bedrag	€ 24,95
Van betaalrekening	NL99 BANK 0123 4567 89 - Hr C Monet

TAN gegevens

Volgnummer:	778
TAN-code:	<input type="text"/>

[Betalen](#) [Annuleren](#)



► Help

(g) ING uses credentials to login and a TAN-code to approve the transaction



Bestelling afronden (stap 4 van 4)

✓ U heeft betaald. Klik op 'Bestelling afronden' om terug te keren naar de webshop.

Afdrukken

Betalingsbevestiging	
Bedrag	€ 24,95
Begunstigde	NL81INGB0000012345 € BoekenGigant
Van betaalrekening	NL99 BANK 0123 4567 89 - Hr C Monet
Datum	17-09-2012 11:30
Transactienummer	9000-0000-0000-2300
Mededelingen	Omschrijving: Het Diner, Koch, Herman Ordernummer: 933816643272912099
Bestelling afronden	



Help

(h) ING uses credentials to login and a TAN-code to approve the transaction

Figure 2: Bank specific transaction pages examples

2.4 CPSPs

With the iDEAL payment system, not only banks are able to acquire money. There are many companies that implement multiple payment systems into a single API for the merchant. These companies are Payment Service Providers (PSPs). In the iDEAL payment system, only a single type of PSPs is specified. There are three types of PSPs according to an online knowledge base[3].

1. *Distributing Payment Service Providers (DPSPs)* can provide an interface for multiple payment systems to the merchant. The merchant must collect the money from the different systems himself.
2. *Collecting Payment Service Providers (CPSPs)* are allowed to collect the money for the merchant and can pay the merchant in bulk. The money from different payment systems is collected for the merchant. The merchant still has contractual relationships with the different payment systems.
3. *Aggregating Payment Service Providers (AgPSPs)* provide multiple payment systems in a single contract with the merchant. The aggregating PSP collects the money for the merchant like a CPSP.

The iDEAL payment system makes use of AgPSPs. PSPs that want to accept iDEAL payments can also accept other payment systems and the merchant only needs a contractual relationship with the PSP. However, Currence calls them CPSPs[5] in the iDEAL payment system. Because this thesis is about iDEAL and the iDEAL protocol, PSPs in the iDEAL payment system are referred to as CPSPs in the rest of this document.

In order for a webshop owner to accept iDEAL payments, he can choose to either get a subscription at his own bank or go to a CPSP. CPSPs more often have a pay per transaction system, whereas the banks use a subscription fee. CPSPs can therefore be more affordable for a starting webshop with few transactions. Merchants only need to have a contract with the PSP and do not need to have a contract with the acquiring bank. Companies that want to be an iDEAL PSP must go through a certification process, to get a certificate from Currence. Acquiring banks must get a license from Currence in order to accept iDEAL payments. Certificates cost less than licenses. As a CPSP is not a bank, it has to have an account at a bank in order to accept money. The CPSPs use the same iDEAL protocol as the merchant uses for their communication with their own bank. So if a merchant does a *TransactionRequest* at the CPSP, so will the CPSP at the acquiring bank. Upon registration at a CPSP, the merchant has to provide a legal name of the company. This legal name is then registered at the acquiring bank of the CPSP. Banks then display the recipient as: <name of CPSP> inzake <webshop company B.V.> (or translated to English: <name of CPSP> in the name of <webshop company Ltd.>).

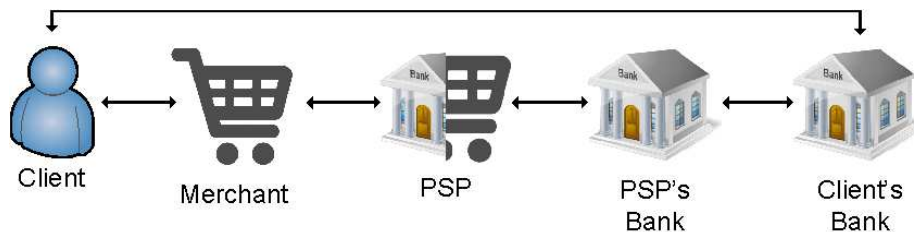


Figure 3: Communication with a PSP involved in the iDEAL protocol

As can be seen in Figure 3, when a PSP is involved in the iDEAL protocol, the communication from merchant to bank goes via the PSP. The merchant can work with the same iDEAL protocol that the banks use to communicate with the PSP. From the perspective of the merchant, the PSP acts as a bank. From the perspective of the bank however, the PSP acts as a merchant, because the PSP has a bank account and wants to acquire money with the iDEAL protocol. Therefore the visualization of the PSP is half bank, half merchant.

2.5 *TransactionRequest* details

The *TransactionRequest* is the most important request in the iDEAL protocol. The merchant specifies the important information about a payment in this

request. There are several fields that should be included in a *TransactionRequest*[12].

- `createDateTimestamp`: A timestamp of when the *TransactionRequest* is created.
- `issuerID`: The chosen issuer. The `issuerID` is the Bank Identification Code (BIC) of the bank.
- `merchantID`: The ID given to the merchant by the acquirer. This field has a maximum length of 9 digits.
- `subID`: The merchant can have multiple webshops with the same `merchantID`. To identify which transaction came from which webshop, a `subID` can be used. CPSPs also use the `subID` to determine the different clients of the CPSP. The CPSP uses the `merchantID` to identify himself at the bank and each client gets a `subID`. This field has a maximum length of 6 digits.
- `merchantReturnURL`: The URL to which the client is redirected after the transaction is approved at the bank. This field has a maximum length of 512 characters.
- `purchaseID`: Unique identifier within the system of the merchant to identify orders. The `purchaseID` will be displayed on the bank statement of both client and merchant. This field has a maximum length of 35 characters.
- `amount`: the amount of euros. This field has a maximum length of 12 numbers including 2 numbers after the decimal separator.
- `currency`: The currency of the transaction. iDEAL only supports EUR at the time of writing.
- `expirationPeriod`: (Optional) The merchant can set the moment when the transaction expires and the client is no longer able to pay.
- `language`: Language of the transaction approval screen. Some issuers may not support other languages than Dutch.
- `description`: Description of the ordered products or services. This field has a maximum length of 35 characters.
- `entranceCode`: The `entranceCode` is used to continue the session when the client returns to the webshop after the transaction is approved at the bank. This field has a maximum length of 40 characters.
- `SignedInfo`: This field contains information about the digital signature on all of the fields above. It also describes the algorithm the merchant used to create the signature.
- `SignatureValue`: The signature of the merchant on the fields specified in the `SignedInfo` field.
- `KeyInfo`: An identifier of the certificate to use to validate the digital signature.

Field relevance

As said in Section 2.3, only some of the fields shown on the transaction approval screen are relevant for the client. The most relevant fields in the *TransactionRequest*, shown on a transaction approval page, are: *recipient*, *currency*, *amount*, and *description*. Where *recipient* is determined from *merchantID* and *subID*. With these fields, the client is able to verify the transaction. These fields are underlined in the explanation of the fields above.

In Figures 2a to 2d and 2f to 2h the *recipient* and *amount* fields are the only fields marked by a circle to show their relevance for the client. The *currency* field is also relevant to the client, but not marked by a circle, because iDEAL only supports euros and is only used in the Netherlands. Therefore, the currency is always the same and not really relevant to check for correctness by the client. The *language* field can change the language of a transaction approval screen (if the client's bank supports the language). The language field can be relevant for the client, but similarly to the *currency* field, other languages may not be supported. The *language* field has influence on other fields and is not visible as a separate field on the transaction approval screen. Therefore it is not marked by a circle. The *description* field is also not marked by a circle, because as said in Section 2.3, most merchants use the *description* field to repeat the *purchaseID* instead of filling it with item descriptions. The *purchaseID* is a random number and therefore very often not relevant for the client. As said in Section 2.4, the *recipient* may be some CPSP in the name of a company name. When the company name is not the same as the name of the webshop, this can be unclear for the client. Even though it is unclear, the *recipient* that is shown, is always the right one. Unlike the *description*, which may be random, the *recipient* can be checked even if it is unclear. Therefore it is marked by a circle.

There is one more field in the *TransactionRequest* that can be relevant to the client. The *merchantReturnURL* is a URL which is similar to the domain of the webshop at which the client ordered his products. The client went to this domain to select the products. The domain part of the *merchantReturnURL* is known to the client. The rest of the fields is either unknown to the client or the client does not need to know them.

Furthermore, the *purchaseID* and *transactionID* (*trxid*) are shown at the transaction approval screen of some of the banks. To the client these are just random numbers. These numbers are not relevant for the approval of the transaction.

Please note that the recipient name of the merchant is determined by either the bank or CPSP. The merchant has a contract with a bank or a CPSP and needs to register his company name. This name is shown in the transaction approval screen.

2.6 Courtesy refunds

When clients are the victim of fraudulent online banking transactions, the Dutch banks often refund the client out of courtesy or leniency. Since the first of January 2014, there are uniform rules for online banking. If Dutch consumers

keep to those rules and still are victim of a fraudulent transaction, the bank will refund out of courtesy. These rules are determined by the Nederlandse Vereniging van Banken (NVB or Dutch Banking Association DBA) and the consumentenbond (consumers association)[24]. The rules are:

1. Keep your codes and passwords to yourself.
2. Never let anyone else use your bank card.
3. Make sure you have proper protection on the devices you use for online transactions.
4. Check your bank account and bank statement.
5. Report incidents to your bank immediately and follow the instructions of your bank.

These rules also apply to iDEAL transactions, so when a client buys a product from a fraudulent webshop and the merchant does not send the products, the client will likely get a refund from his bank. This is however not part of the iDEAL payment system and the client should contact his own bank for the possibilities. If the client's bank decides to refund the client, the bank will also contact the bank of the merchant to report the transaction. According to an online news article, a third of financial institutions think that the cost of implementing a countermeasure against fraudulent transactions are higher than just handling refund requests by clients[14]. In the case of the attack as described in Section 3, the webshop of the attacker is not fraudulent, but fake. Both the client and the merchant are a victim in this case. The transaction between client and merchant was not fraudulent. Therefore the client's bank can not report the transaction. A refund for the client may still occur, because the client kept to all above rules, but still did not get a product.

3 Attack

This section discusses a man-in-the-middle (*MITM*) attack on the iDEAL payment system. The attack is first described on a news website[4]. In this news article, it is stated that this attack is relatively new. An attacker who wants to make money builds a fake webshop. He lures clients to his webshop by asking a lower price for his products and by getting a high page rank in the search results and product comparison websites. Another way to lure clients is to make the webshop look just like a well known legitimate webshop. The tempted client orders a product from the fake webshop of the attacker. From this point on there are several possibilities for the attacker to make money. Below is a list of the options. In this thesis, the focus is on the fourth attack. Please note that the first and second option are generic for any payment system and not specifically for the iDEAL payment system. Also note that the bitcoin transaction is just an example of an alternative way of payment.

1. **no product sent:** The first possibility is probably the most simple one to manage. The attacker uses his own bank account for receiving the money and does not send any product to the client. For this attack to work with iDEAL, the attacker needs a contract with a bank or a CPSP. This attack however is also very easy for law enforcement to track down. The bank, that manages the account of the attacker, can work together with the law enforcement to block the account and provide details on the whereabouts of the attacker.
2. **money mule:** The second possibility is for the attacker to use a bank account of someone else and to give that person money for this operation. This is a so called *money mule*. The iDEAL contract with the bank or CPSP is also in the name of the money mule. The money mules are often threatened or extorted by the attacker or criminal, or they are short on money and would like to make some quick money. When the law enforcement tracks the money mule, they are happy to get out of the deal with the attacker.
3. **naive client manipulation:** The third possibility is for an attacker to trick or manipulate naive clients to pay in an alternative way. On the webshop payment page, an attacker can state that the client must go to another website to pay. The attacker then directs the client to a bitcoin website for example, where bitcoins can be bought using iDEAL. The (naive) client thinks he is paying for his product using iDEAL, but in fact he is giving bitcoins to the attacker.
4. **background transaction setup:** The fourth possibility is for the attacker to set up the transaction for anonymous products in the background, while the client does not leave the fake webshop. The attacker can for example choose bitcoins. This attack does not require such a naive client, because it can only be detected at the bank specific transaction approval. The

bank shows the recipient on the transaction approval page. It is up to the client to verify the recipient. This final possibility is a MITM attack and this is the attack that is considered in this thesis.

3.1 Bitcoin transaction in the background

This section discusses the attack in more detail. Suppose that an attacker has built a malicious webshop and is able to attract clients to his webshop. When the client has chosen his products and is at the payment step in the checkout process, the attacker can set up a connection to another website in the background. If the attacker creates another transaction at a webshop that sells bitcoins, he might be able to trick the client into paying this bitcoin transaction. The bitcoins go to the attacker and the client does not get his products, because the attacker already specified where the bitcoins should be sent to. To clearly show how the attack works, we edited the protocol description from Figure 1. Figure 4 shows the attacker E as a man-in-the-middle between the Client A and the legitimate merchant B . A simple visual representation of the parties involved and which party talks to which other party can be seen in Figure 5. The steps of the protocol mentioned below match the steps in Figure 4.

0. As stated, the client A can select products from a fake webshop.
1. The client A indicates he wants to pay using iDEAL.
2. The attacker E tells the legitimate bitcoin merchant B that he wants to get bitcoins for the same amount of euros, the client A has to pay. The attacker E also specifies the so-called ‘wallet’ where the bitcoins should be sent.
3. The attacker E proceeds to the payment, selecting to pay with iDEAL.
4. The legitimate bitcoin merchant B now does a ‘*DirectoryRequest*’ at the merchant’s bank C .
5. The merchant’s bank C provides the merchant B with a list of possible issuing banks.
6. The merchant B lists these issuers on his website.
7. The attacker E parses the list from the merchant B ’s website and presents them on his own website.
8. The client A selects his or her own bank from the list.
9. The attacker E forwards the choice of the client to the merchant B .
10. The merchant B creates a transaction request and provides the transaction details to his bank.

11. Upon a new *TransactionRequest*, the merchant's bank *C* will negotiate a new transaction with the issuing bank *D*. This part of the protocol is proprietary. Only the parties that have a license or a certificate (see Section 2.4) from Currence can get access to this part of the protocol. We do not have access, so we don't know what exactly goes on here. We believe the merchant's bank *C* creates a *transaction id (trxid)* in this step.
12. It is trivial that the issuing bank *D* will reply with at least the *issuerURL*, the link where the client *A* can pay for the order.
13. The merchant's bank *C* then sends a transaction object to the merchant *B* which includes the (*trxid*), the same *issuerURL* and a signature over the important data. This is the response to the *TransactionRequest*. The merchant *B* checks the data for validity using the signature.
14. The merchant *B* then redirects the attacker *E* to the provided *issuerURL* of the issuer bank *D*. The attacker has full control over his own machine and can therefore intercept the *issuerURL* and use it to redirect the client.
15. The attacker *E* redirects the client *A* to the provided *issuerURL*.
16. The client *A* requests the *issuerURL* at the client's bank *D*.
17. The client's bank *D* responds with the transaction approval page.
18. The client *A* can now verify and approve the transaction at his or her own bank *D* with the same mechanism the bank has for doing online banking. Some banks use *TAN-codes* provided over the mobile network to the client's phone and others use a separate hardware device or token to approve the transaction using the client's bank card.
19. If the approval of the transaction succeeds, the money is transferred.
20. When the transaction is either approved or cancelled or when an error has occurred, the client's bank *D* will redirect the client back to the webshop of *B* with the *trxid* and the *ec*.
21. Upon the call of the returnURL with the *trxid* and the *ec*, the merchant *B* will know that the transaction is completed with an unknown status (verified, cancelled or an error occurred). The *trxid* and *ec* are included to make sure the client can only return after a completed transaction.
22. To get the status of the transaction, the merchant *B* will send a '*StatusRequest*' to the merchant's bank *C*.
23. The merchant's bank *C* responds with a '*StatusResponse*', which includes the status and the amount payed and a signature over important data. It is up to the merchant *B* to verify if the data is correct and if the amount is equal to the amount he asked for. If the status of the transaction equals '*Success*', the money is now guaranteed to be in possession of the merchant

B. He can now proceed to ship the product to the client. In this case he will send the bitcoins to the address given by the attacker.

The setup of the transaction happens in the background, while the client *A* thinks he is buying a product at a webshop. When the transaction is finished, the client *A* is returned to the returnUrl of the legitimate merchant (step 18 in the protocol steps above). So in the example of the bitcoin transaction, the client *A* is returned to the bitcoin website and not the fake webshop where he bought his product. The client *A* should know now that something is wrong. Unfortunately, the transaction is already approved and the client *A* lost his money.

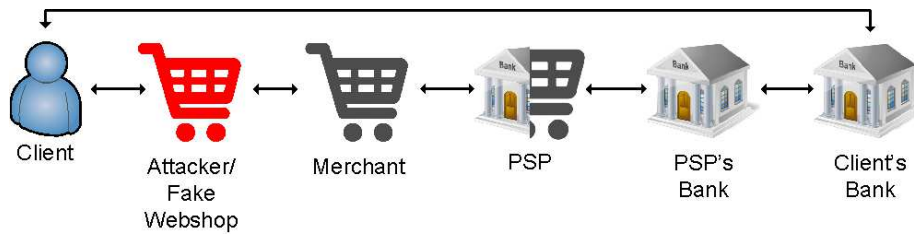


Figure 5: The parties involved in the attack. The attacker orders the wanted products at the legitimate merchant and lets the client pay for the products

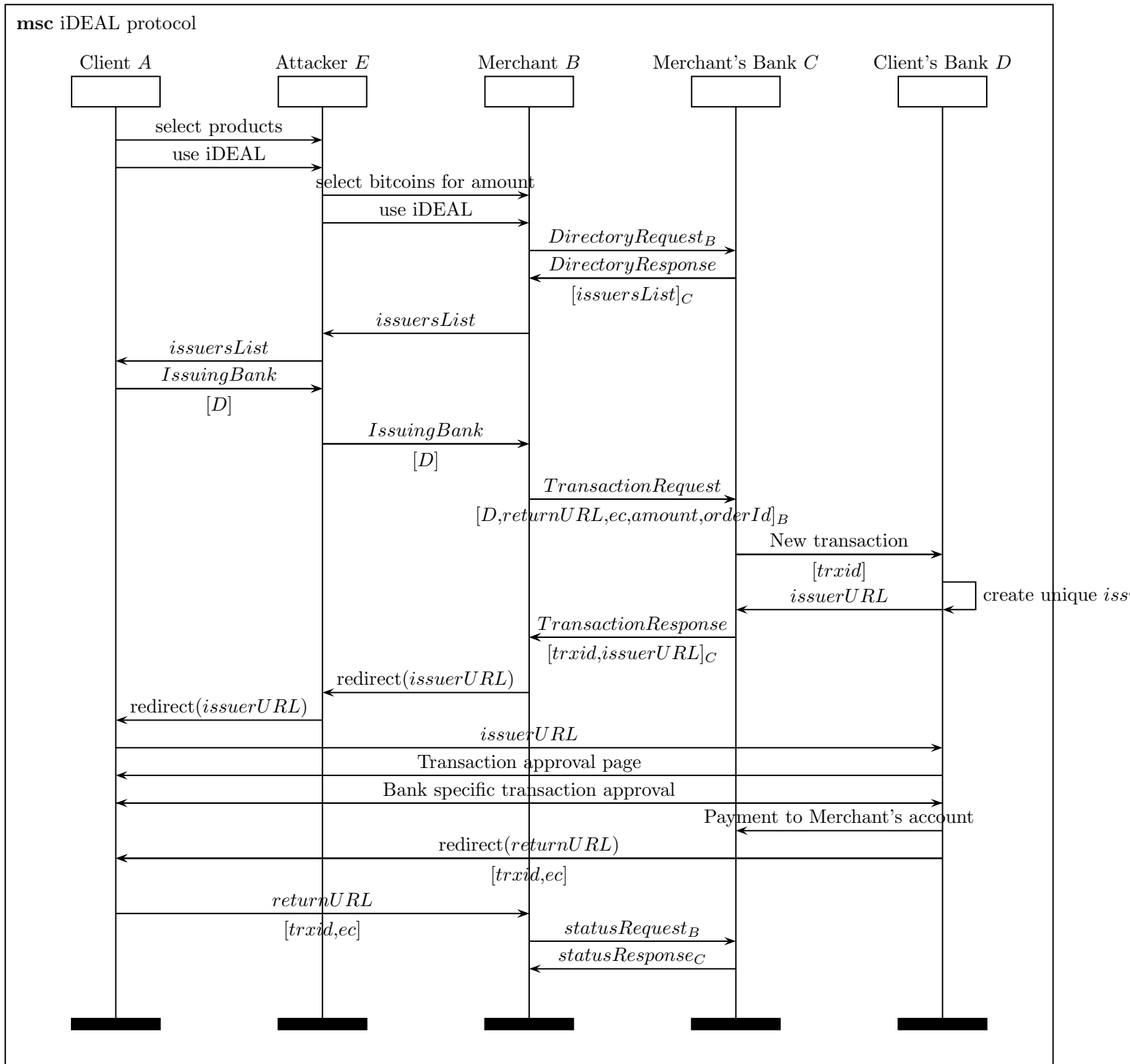


Figure 4: iDEAL protocol with an attacker

4 Root cause analysis

The attack as described in Section 3 is possible due to some problems in the iDEAL protocol described in Section 2.2.

1. The first problem is that the recipient shown on the transaction screen of the bank might not be the intended recipient of the client. Especially when a CPSP is involved. Often the company name of the webshop is shown instead of the webshop name, because the bank is required by law to use the legal name of the company. A single company can have multiple webshops, so the name of the company is often very different from the actual webshop name. The name of the company may be unknown to the client, so the client might be tricked into sending money to a different recipient than the intended webshop when sending money to <name of CPSP> in the name of <webshop company Ltd.>. The merchant must state his chamber of commerce number and company name on the webshop site, so the client is able to find the name of the company and verify that name. The client can only see the recipient of the transaction at the transaction approval screen, so the client would have to open a new tab or screen in the browser, or cancel the payment and go back to the webshop to check if this is indeed the intended recipient.
2. The second problem is that the client does not always know what he is paying for. As stated in Section 2.5, there are two fields to describe the purchase, the *purchaseID* and the *description*. The *purchaseID* should be the identifier for the merchant for the entire order. So if multiple items were purchased in one transaction, the order number is used to see which products relate to that order. The *description* field can contain a description of the products bought by the client. This field has a maximum length of only 35 characters, because the *description* field is also used on the bank statement, which has limited space. Even if only the make and model number of a product are put in this field, not many item descriptions will fit. Therefore, many merchants repeat the *purchaseID* in the *description* field. This is not the proper use of the description field as it should contain a description of the items purchased. The client only gets to see the purchaseID, which is some (random) order number. The order number does not mean much to the client at the time of the transaction. Some webshops show the order number before proceeding to the transaction approval, but also many webshops do not show the order number. An example of a transaction description without any information is displayed in Figure 6. When buying a train ticket for an international train ride at the Dutch Railway (Nederlandse Spoorwegen), the transaction description displays a ‘.’. This ‘.’ is just to fill the *description* field with something. It does not provide any information about what the client is buying.
3. The third problem is that there is currently no way to verify if the person

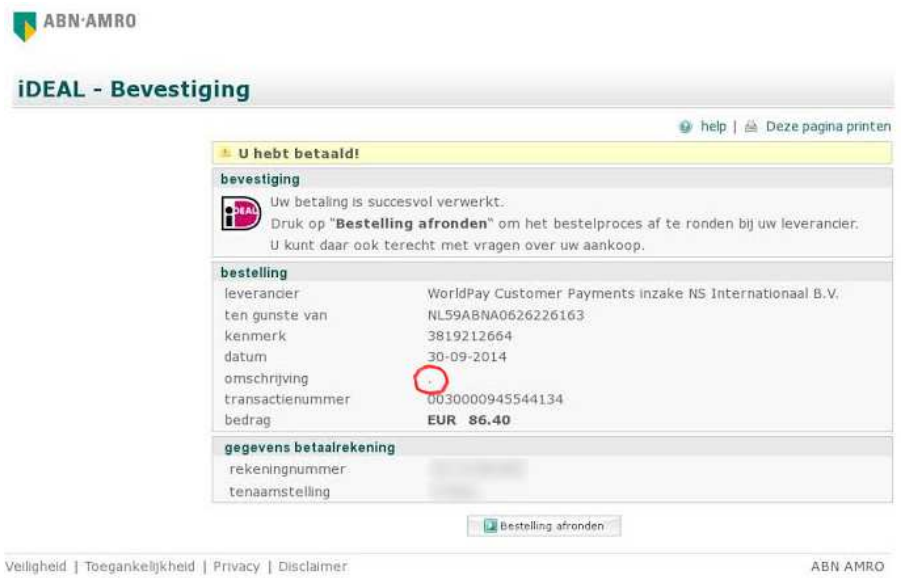


Figure 6: Example of a description without any information of the purchase

who selected the products at the webshop is the same person as the one approving the transaction. In the bitcoin example in Section 3, the attacker selected the amount of bitcoins he wanted. Then he sent the client to the transaction approval screen. So the attacker who selected the products is not the same person as the client who approves the transaction. The iDEAL protocol should include a way for both the merchant and the client's bank to know that they are indeed connected to the same person or client and there is no man-in-the-middle.

So for the client it may be unknown who he is paying and what he is paying for. An attacker can use this to his advantage and manipulate the client into paying a different webshop for a different product. There is no verification that the client who selected the products at the webshop, is also the one paying for the items. The attacker can select which products he wants and redirect the client to the issuerURL so that the client will pay for him. The legitimate merchant and the bank should verify that the person who selects the products is the same person who approves the transaction.

5 Similar systems

There are many online payment systems. One of the best known systems is PayPal. With PayPal, people can send each other money using just their email address to identify the recipient and sender. Another often used system is credit card payments. The credit card has been around for many years. Online credit card payments are therefore easy and familiar for the client.

5.1 PayPal

PayPal is an online payment system founded in 1998[18]. PayPal went public in 2002 and was sold to eBay later the same year. It became popular as online payment system in the United States due to its integration with eBay.

5.1.1 The PayPal protocol

PayPal uses a REST API that is in some ways similar to iDEAL. PayPal is a system built on top of existing systems like credit cards and bank transfers. Therefore, there is no need for a *DirectoryRequest* like in iDEAL, because there is only one PayPal. The only identifier necessary for clients and merchants to transfer money is an email address. The merchant can call the PayPal API to create a new payment transaction. In this API call, he specifies the payment details. The details include the amount of the transaction and a description of the items that will be bought[21, 20]. This *PaymentRequest* is similar to a *TransactionRequest* in iDEAL. In Figure 7, an overview of the steps in the PayPal protocol is given.

5.1.2 Differences between iDEAL and PayPal

Protocol and transaction details

Because the PayPal protocol is very similar to the iDEAL protocol, the attack as described in Section 3 will probably also work on PayPal, but it requires more naivety from the client. The main reason is that PayPal does not restrict the length of descriptions to be sent with the *PaymentRequest* as much as iDEAL does. A description does not have to fit on a bank statement and can therefore be more than 35 characters. The limit of a description is currently set to 127 characters. Furthermore PayPal has the ability to include an array of items. Each item can have a name with a maximum of 127 characters and a description of the same maximum size. All of this information is shown on the payment approval screen of PayPal (see Figures 8a and 8b). The client is able to check if the information on the screen matches what he wants to buy and from who. Therefore, a client can't be easily manipulated into buying something he does not want and sending it to the attacker.

Refunds

PayPal has a policy to protect both clients and merchants. If the client never

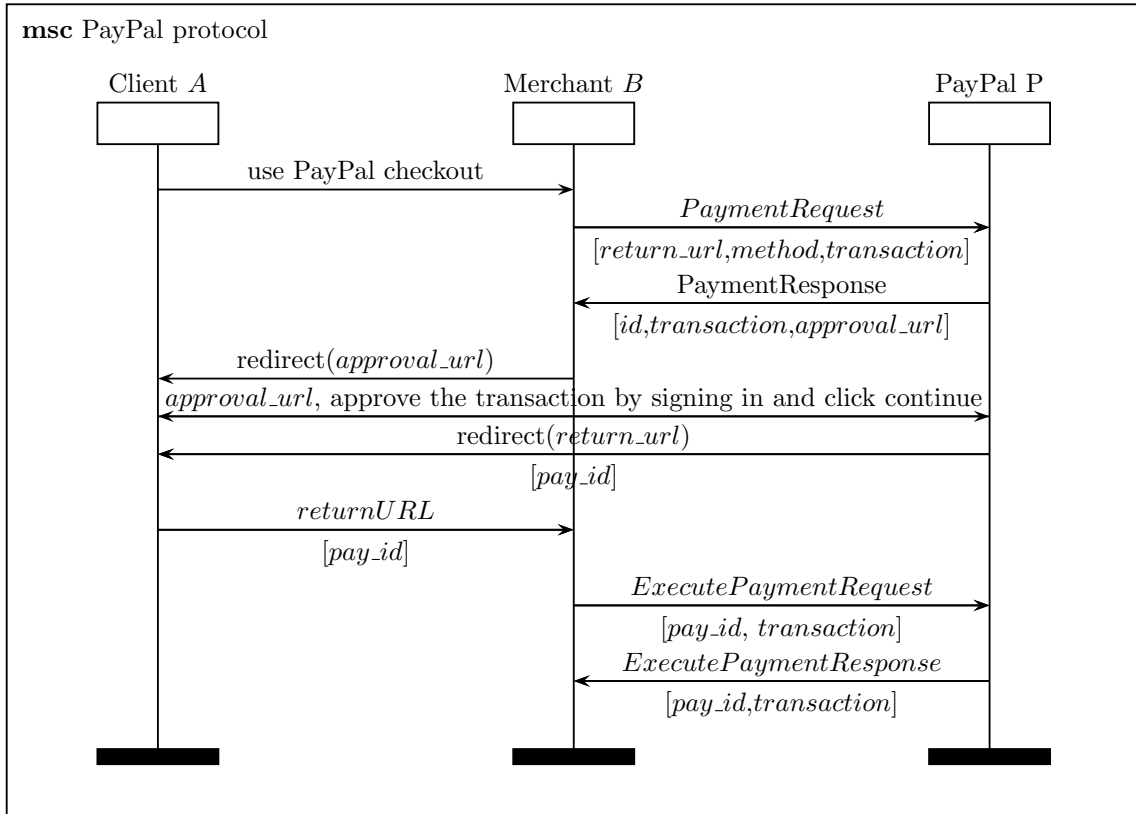


Figure 7: PayPal protocol

received the product, he can reverse the payment and get his money back. The merchant in this case has to prove that he indeed did send the product by mail using a track and trace code of the parcel. If it checks out that the merchant did indeed try to send the product, the merchant can keep the money for the product. If the merchant can't prove that the product was indeed sent, PayPal can take money from the merchants account. These policies only apply to physical goods that are sent via the mail. Virtual goods and services are not covered[22]. The PayPal API also specifies a refund request. When the client canceled the order after approving the transaction, the merchant can refund the money with a refund request.

5.2 Credit Card payment

Another well known system for online payments is credit cards. Credit cards have been around for a long time, so when online shopping over the Internet started, the credit card companies were quick to implement a method for paying online.

DX dealextreme
GREAT GADGETS, PRICE & !

Uw besteloverzicht

Beschrijvingen	Bedrag
Samsung Replacement Back Camera Mod.: Objectnummer: 176228 Objectprijs: \$27,38 Hoeveelheid: 1	\$27,38
Totaal object	\$27,38

Totaal \$27,38 USD

Een betaalwijze kiezen

Betalen met mijn PayPal-rekening PayPal

Log in op uw rekening om te betalen

E-mail
[input field]

PayPal-wachtwoord
[input field]

Dit is een privé-computer. Wat is dit?

Inloggen

[Uw e-mailadres of wachtwoord vergeten?](#)

Geen PayPal-rekening?
(Optioneel) Open een PayPal-rekening om in de toekomst sneller te betalen

[Annuleren en terugkeren naar dealextreme.](#)

(a) Enter account details

DX dealextreme
GREAT GADGETS, PRICE & !

Uw besteloverzicht

Beschrijvingen	Bedrag
Samsung Replacement Back Camera Mod.: Objectnummer: 176228 Objectprijs: \$27,38 Hoeveelheid: 1	\$27,38
Totaal object	\$27,38

Totaal \$27,38 USD

Uw gegevens controleren

Doorgaan PayPal

Verzendadres [Wijzigen](#)

Willem Burgers

Betaalmethoden [Wijzigen](#)

Directe overschrijving: Rabobank Bankrekening (Bevestigd) x-7734 €21,04 EUR

PayPal-wisselkoers vanaf 26 jul 2014: 1 euro = 1,30134 Amerikaanse dollar

Contactgegevens

Doorgaan

U bent bijna klaar. U bevestigt uw betaling op dealextreme.

[Annuleren en terugkeren naar dealextreme.](#)

(b) approve the transaction

Figure 8: PayPal approval screens

The credit card companies like Visa and MasterCard, only provide the network for transaction handling. They do not make cards themselves. Credit cards are often manufactured into a bank card. Since the credit card companies do not allow a merchant to accept credit cards directly in a webshop, a PSP is needed. Some PSPs allow the merchant to have the entry fields for credit card payments on their webshop payment page. This way the merchant is in control of the data of the client. The merchant collects this data before sending it to the PSP. Other PSPs let the merchant create a transaction and redirect the client to the payment page. This way, the merchant does not know anything about the credit card details of the client. The details on the payment approval screen therefore depend on the API of the PSP. If the PSP allows the merchant to have credit card fields on the payment approval page of his own webshop and later send them to the PSP, the merchant can choose what to display on the payment approval page. An example can be seen in Figure 9a, where the payment approval screen shows the product being bought while the client stays on the website of the merchant. When a PSP handles the input of credit card details on his own page, less payment information is available. A PSP may or may not allow the merchant to send information about the items or products the client wants to buy. Even if item descriptions are available in the API, the merchant can choose not to use these fields, similar to the description field in the iDEAL protocol. An example of a webshop with a payment screen hosted by the PSP can be seen in Figure 9b.

5.2.1 SecureCode or 3-D Secure

Merchants are able to choose whether to accept MasterCard's SecureCode and Visa's 3-D Secure or not. This code is a password set by the client to provide extra security during a transaction. After filling in the card details at either the merchant's webshop or the PSP, the client is redirected to a secure and trusted website. Credit card suppliers implement the SecureCode or 3-D Secure in different ways. The page where the client has to enter his SecureCode or 3-D Secure only shows the recipient of a transaction and does not show what the client pays for. It does however show a personal greeting that has to match the one the client set up along with his password. An example of a SecureCode can be seen in Figure 10.

5.2.2 Differences between iDEAL and credit card

As said, there are multiple ways to implement credit card processing for online payments. A PSP is required and depending on the API of the PSP, the merchant may be able to ask for credit card details on his own webshop page, or the merchant should redirect the client to a transaction approval page of the PSP.

When a merchant is able to ask for credit card details on his own webshop page, the client will have to trust that the merchant is indeed displaying the right information on the page. There is no way of knowing whether the merchant

WOW HD ALLES ALTIJD GRATIS BEZORGD

Betalingsvoortgang

1. Bekijk winkelmandje 2. Log in 3. Verzenden **4. Betaling** Bestelling compleet

Verwerk betaling

Voer je kaartgegevens in om de betaling af te ronden.

Jouw ordergegevens

E-mailadres:

Afleveradres:
Willem Burgers

Factuuradres:

Winkelmandje:

Produkt	Hoeveelheid	Prijs
Transit Insurance	1	€ 0,75
Script - No Sound Without Silence co	1	€ 9,99
Totaal:		€ 10,74

Jouw betalingsgegevens

WOW HD beveiligde betalingservice wordt ondersteund door GlobalCollect

Betaalwijze:
Visa

Betalingsdetails:

Kaartnummer (geen spatie, puntjes of komma's) *

Vervaldatum
Maand ▼ Jaar ▼ *

Veiligheidscode [Wat is een veiligheidscode? *](#)

Verder

(a) Credit card processing on the domain of the webshop

MultiSafepay The Payment Professionals

1 2 3 4 Verwerken van transactie van **EUR 417,06** voor **Trendy Computers** via **Maestro**.

Transactiegegevens

Controleer hier de gegevens van uw transactie.

Ontvanger **Trendy Computers**

Bedrag **EUR 417,06**

Beschrijving **Order ORD13827**

Betaalmethode **Maestro**

Totaal bedrag **EUR 417,06**

Betalingsgegevens

De volgende gegevens zijn nodig om uw transactie te voltooien.

Kaartnummer

Naam op kaart

Vervaldatum

Annuleer Betaal

Copyright © 2014 MultiSafepay. #30 RECHTEN VOORBEHOUDEN.

(b) Credit card processing on the domain of the PSP

Figure 9: Two examples of credit card processing

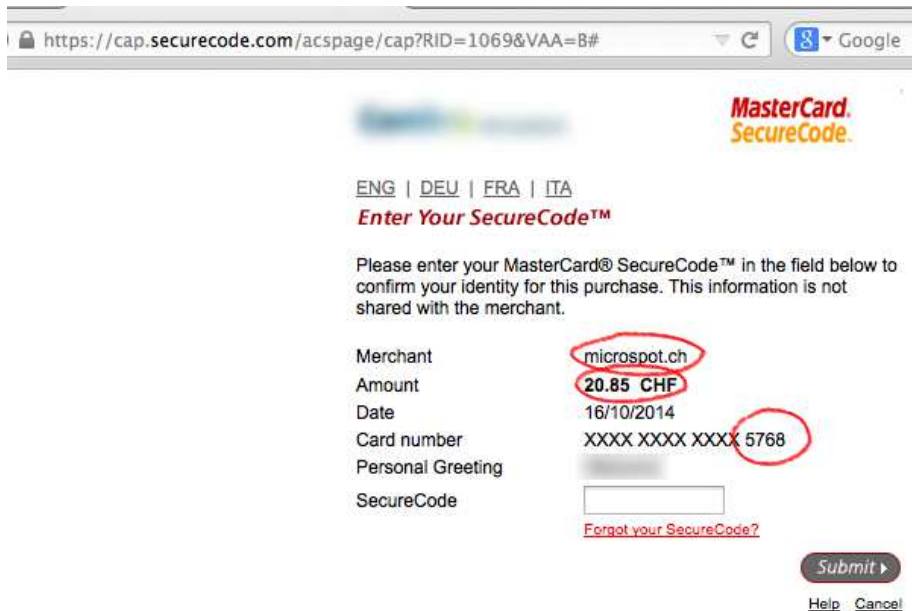


Figure 10: An example of a SecureCode credit card transaction

is displaying the right information or not. On the other hand, when clients pay with credit card and their products are not shipped to them, they can complain at the credit card companies to get a refund. The credit card companies will then swiftly force the merchant to comply to the rules of the credit card processing and refund the client. Credit card companies can even charge the bank account of the merchant after a notification or warning[25]. Therefore, the information on the merchant's webshop page is probably correct.

When a merchant is not able to ask for credit card details on his own webshop page, but has to redirect the client to the PSP transaction approval screen, the information about the transaction, will be displayed on the page of the PSP. For credit card transactions, the PSP is chosen by the merchant and not the client. Therefore the environment provided by the PSP to fill in the credit card details is still not really trustworthy. The responsibility for refunds is again forced upon the merchant, such that the client has no risk in paying with his credit card.

When the merchant accepts SecureCode and 3-D Secure the client is indeed sent to a trusted website. This website is hosted by the card manufacturer. In the Netherlands, most banks give out a credit card. The SecureCode or 3-D Secure page is then hosted by the bank. The client has to do a similar authentication as with iDEAL when he has a credit card given out by his bank. This is not the case for cards given out by other manufacturers. SecureCode and 3-D Secure payments are most similar to iDEAL payments, because the

client is taken to a trusted website on which he has to enter the code. This transaction approval screen does not give many details about the transaction. Only the recipient and the amount are shown as relevant details about the purchase. There is no item description. Therefore credit card transactions are still not very safe. The attack as described in Section 3 will also work on credit card transactions, even if SecureCode or 3-D Secure is used in the payment.

6 Countermeasures

To prevent the attack as described in Section 3, some countermeasures can be taken. In this section these countermeasures are described. The countermeasures do not consider privacy and are only meant to mitigate the attack. The privacy of the protocol as well as the countermeasures is considered in Section 8

6.1 Countermeasure A Register client's bank account number at merchant

Countermeasure A tries to prevent the attack as described in Section 3 from the merchant side by trying to solve root cause 3, as described in Section 4. Currence, the company that takes care of the iDEAL payment system, knows about the attack. According to the news article in Computerworld, Currence immediately set up a countermeasure in order to prevent the attack[4]. Merchants who sell anonymous products, like prepaid calling credit, credit for online games or bitcoins, must use some form of identity verification. The merchant is free to choose how to verify the identity of the client. Merchants can implement mobile number verification, IP-verification or bank account number verification. In order to mitigate the attack, none of the options are very effective, because the merchant or webshop is the only entity to validate the identity of the client. This still does not guarantee that the person selecting the products from the webshop is the same person paying at the bank. For another method of IP verification, see Section 6.2. The best countermeasure that Currence wanted merchants to implement is bank account number verification. When a client wants to buy anonymous products, he should first tell the merchant his bank account number. The client is instructed via email or SMS to transfer a few cents to the merchant. Only after the bank account is validated can the client order products using his webshop account or bank account.

From a conversation with an owner of a bitcoin webshop, we learned that Currence (via the banks of the merchants) only forced bigger merchants who sell anonymous products to implement this countermeasure. There are more merchants who sell the same products, but are less known to clients and attackers. These webshops have less traffic and are therefore not interesting enough. Banks have contracts with many merchants, so it is hard to verify the webshop of every merchant. When a merchant is reported for fraudulent behaviour, the bank will take action. All merchants that sell anonymous products should implement this countermeasure in order to fully prevent fraudulent transactions.

Counterattack against countermeasure A

Although the attack as described in Section 3 will be a lot harder to perform and most clients will get suspicious when receiving an email or SMS from an unknown webshop, the attack is not completely mitigated. The attack can be extended to work around countermeasure A. Suppose that an attacker implements the same registration procedure in his fake webshop. The client would register his bank account number at the webshop of the attacker. The attacker can just as

easily create a new account for the client at the merchant who sells anonymous products. The attacker can choose to send the verification email or SMS to the client or to send the instructions to himself and then forward the instructions while modifying some details. This way, the attacker can manipulate the client into registering his bank account. After this registration, the client is able to buy anonymous products from the legitimate merchant. This way, the attacker can still perform the attack, just like before. Not all merchants implement the countermeasure the same way. On some webshops this extended attack will still work.

Disadvantage of countermeasure A

Aside from the possibility of a counterattack, there is also a disadvantage to using this countermeasure. Waiting for bank account number verification is not beneficial for online shopping. The bank transfer from merchant to client of a few cents is just a normal bank transfer, which may take up to a day to arrive at the client. When clients want to buy something from a webshop that sells anonymous products, they can't order directly, but first have to wait up to a day. This means that ordering a product will become a tedious process, which must be done over the course of two days. Therefore this countermeasure is not beneficial for online shopping.

6.2 Countermeasure B IP addresses

In order to mitigate the attack described in Section 3, the merchant and the bank should know that they both have a connection to the same client. The client who selects the products at the webshop of the merchant is also the one approving the transaction at the bank. The merchant has to perform some form of identification of the client and send the result to the bank. By adding the IP address of the client to the *TransactionRequest* as a separate field, the bank can check if the approving client indeed has the same IP address. If the merchant and the bank know they are talking to the same client, by sending the IP address in the *TransactionRequest*, root cause 3 would be solved and the attack would no longer work.

Of course IP addresses can be spoofed, depending on the protocol used on the transport layer of the connection. For tcp traffic, it is not possible to spoof the IP address with the attack described in Section 3, because the attacker only lured the client to the fake webshop. The attacker does not control the network of the client. Spoofing will also not work for the iDEAL protocol, as the client still needs to be redirected to the bank. If the *TransactionRequest* is done using a spoofed IP, the merchant will send the redirect operation directly to the client, instead of to the attacker. The merchant is not able to connect to the client, because there was no previous tcp connection. Every message was relayed by the attacker. Therefore the redirect will fail. So by including the IP address of the client in the *TransactionRequest*, the bank is able to authenticate the same client while spoofing does not work.

A possible way to circumvent this IP address validation is by injecting JavaScript into the client's browser. With a setup as visualized in Figure 11, the attacker could try to let the client make a connection to the legitimate merchant via JavaScript. This will not work as JavaScript can not send HTTP POST requests to a different domain. Only if the target domain allows the POST request, can the request be handled. By default, a server does not accept such requests.

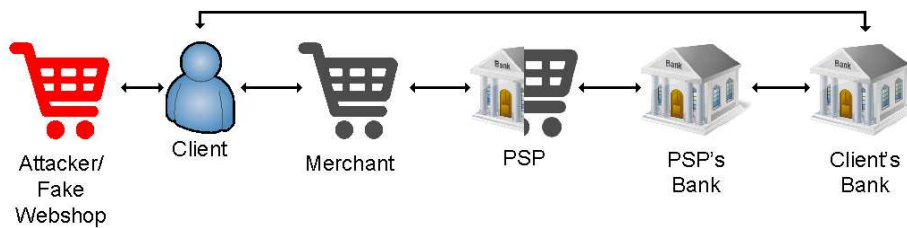


Figure 11: The client can visit the fake webshop where the attacker instructs the client to buy something at website of the legitimate merchant

6.3 Countermeasure C Shared cookies

Another method for banks and merchants to verify whether they have a connection with the same client or not, is to use some way to identify the browser of the user at both the merchant side and the bank side of the communication. Most web applications use cookies to identify their users. Due to the same origin policy, applications on different domains are not able to access the same cookies. To use cookies over multiple domains, one application can make an HTTP call to the other application to set the cookie in the domain of the second application. This way the cookie is set by the first application making it a shared cookie. In the example of iDEAL, the merchant can set a cookie by creating an iframe element that calls the API of the bank. The bank can later read the cookie value during the transaction and can know if the cookie is set by the merchant. This way the bank can verify if the client who is about to pay is indeed the client who selected the products. This method requires a new API at the bank. Therefore this method is less feasible than sending the IP address in the *TransactionRequest*. This method can also be reversed such that the merchant sets the cookie in his own domain and later the bank creates an iframe element that retrieves the cookie from the merchant's domain. This forces the heavy lifting of the method onto the merchant. Still it requires more effort for both the bank and merchant than sending the IP address.

6.4 Countermeasure D Give more semantics in the transaction details

There are many fields in the *TransactionRequest* (as described in Section 2.5). Some of these fields are just for use in the protocol or by either of the two parties (merchant and Bank). These fields are not shown in the transaction approval screen. If some of these fields were shown to the client, it would be harder to manipulate the client into approving a transaction he does not understand.

Transaction recipient

As described in Section 2.4, the client is not always able to verify if the money he is transferring is going to the intended recipient. Especially when a CPSP is involved. The client does know the URL of the webshop where he ordered the products, because the client went there himself. If the bank would show the *returnURL* on the transaction approval screen, the client would know which webshop is receiving the money, even though there is a company running the webshop which does not have a similar name. As stated in Section 3.1, the client was able to determine that something went wrong, because he was redirected to a different webshop than the one he ordered his products at. By including the *returnURL* as text on the transaction approval screen, the client can verify if this is the same webshop. This way the client is not only redirected to the *returnURL* after the approval of the transaction, but the *returnURL* also can be used to verify the transaction. The field gets more meaning or semantics, without altering the protocol. When this field (or at least the domain part of the *returnURL*) is shown on the transaction approval screen, it is more clear who the client is sending his money to.

Transaction description

As said in Section 4 the description field is not properly used by merchants. As described in Section 2.5, the description field can contain item descriptions for the ordered product(s). Unfortunately, this field has a maximum of 35 characters. If a single product is purchased, the description of that item might fit in 35 characters. When multiple products are purchased, the field is too small for multiple item descriptions. The field does not depend on the number of items a client can purchase. Therefore, the merchants use the *purchaseID* (see Section 2.5) to fill in the description. The client often only gets to see an order number that the client may have never seen before. The order number may be unknown to the client at the time of transaction, because the merchant did not show the order number during the ordering process. The client sees the order number, but does not know what he is paying for. Merchants should always fill the description field with as much item descriptions as could fit in 35 characters. The iDEAL protocol should be extended to either use a longer description or add multiple item fields. A longer description may not fit on the bank statement, but can be cut down to 35 characters to make it fit on a bank statement. This way at least the transaction approval page shows a proper

description. When multiple item fields are included, all items can be shown on the transaction approval page.

6.5 Countermeasure E Address details via the bank

Another way to provide transaction verification is by having the merchant send the recipient address to the client's bank to show on the transaction approval screen. For physical products, the merchant can insert the shipping address of the client in the *TransactionRequest*. For virtual products, like the anonymous products described in Section 1, the virtual address like for example the bitcoin wallet address can be included in the *TransactionRequest*. If the banks display the address, the attack as described in Section 3 would no longer work. The attacker sets up a transaction to have bitcoins delivered to his bitcoin address. If the client ordered for example a phone from the fake webshop of the attacker and the attacker sets up a different transaction, the bank would show the bitcoin wallet address to the client. This does not match the address filled in by the client. Even if both the client and the attacker want the same type of product (both want bitcoins for example), the wallet address does not match. The client should verify that the product is indeed shipped or sent to him.

far fetched ideas

Another method to make sure the shipping details are correct, is to ask the client for his address after the payment. The payment is often the final step in the ordering process. This is because an iDEAL transaction is non reversable. Both the merchant and client can't undo an iDEAL transaction. If the iDEAL transaction was reversable, the merchant or client could cancel the transaction even after the client approved. After the transaction, the client is redirected to the legitimate merchant. If the shipping details would be asked or at least be verified after the transaction, the merchant would know that he is always talking to the client instead of the attacker (because only the client can approve the transaction). This way, the product always goes to the client, because a client can fill in his own address. PayPal verifies the shipping details after the transaction approval.

The bank knows the street address of the client, so another way to make sure that the client gets the product is by letting the banks tell the merchant what the shipping address of the product is. This idea is far fetched, because the bank only knows the street address of the client, so this would not work for virtual goods. This countermeasure would also block the possibility of sending a product to a friend. Therefore this countermeasure is not really practically applicable. It would however protect the client.

7 Trustmarks

A totally different approach to provide confidence for a client that a webshop is legitimate, is a trustmark. There are several trustmarks for Dutch webshops, namely: Trusted Shops Keurmerk, Webshop Keurmerk, mkbOK Keurmerk, Thuiswinkel Waarborg, QShops Keurmerk, Thuiswinkel Keurmerk, Digikeur Keurmerk, Friendly Shop Keurmerk, Veilig Betalen Keurmerk, Safe2Shop Keurmerk, MKB Keurmerk, BrowseSafe Keurmerk, Webkeurmerk Nederland, ICTWaarborg Keurmerk, ICTRecht Keurmerk, Bettershop Keurmerk and WebwinkelKeur. A trustmark gives added value to a webshop of a merchant. Bigger webshops like bol.com or wehkamp.nl (these are big webshops in the Netherlands) do not really need a trustmark to convince clients that their webshop is legitimate. Still they want to show that they comply to Dutch and European law for eCommerce. For small and relatively unknown webshops, the added value of a trustmark can be greater. The client will trust the webshop more if there is a trustmark.

When a webshop has a trustmark, the client gets some guarantees. Most trustmarks make sure that the webshop complies to Dutch and European law for eCommerce. A trustmark can make a merchant comply with more criteria. For example, Thuiswinkel Waarborg wants to see the annual report of a webshop to ensure financial stability. Trustmarks also add to the dispute procedure. When a client has a complaint and the merchant does not want to comply, the client can go to the trustmark organization to mediate in the dispute. A merchant must meet all criteria of a trustmark in order to become a member. The trustmark organization checks the webshop in a certification process.

In order to verify the validity and legitimacy of the trustmark, the client is able to click on the trustmark logo in the webshop and go to the website of the trustmark, where the trustmark organization vouches for the integrity of the webshop in the form of a certificate. This way, the client can know that a webshop is legitimate. The effectiveness of a trustmark depends on whether the client knows the trustmark or not. Research by TNS NIPO shows that consumers know that trustmarks exist, but they generally can not name any particular trustmarks[17]. Only 9% of the respondents spontaneously mentioned ‘Thuiswinkel Waarborg’ as a trustmark. The research concluded that Thuiswinkel Waarborg is therefore the best known trustmark amongst consumers. It is good to note that this research was commissioned by Thuiswinkel Waarborg.

7.1 Effectiveness of trustmarks

As said, the effectiveness of the trustmark depends on whether the client knows the trustmark or not. If clients do not use the trustmarks in a proper way, the trustmark has no use. Certified websites get the right to use the logo of the trustmark in their webshop. If the client sees the logo and thinks everything is fine, the trustmark has no effect at all. It is easy for attackers to just copy the logo and place it on their fake webshop. Even the certificate itself can get copied. If the client clicks on the trustmark of the fake webshop, he is not taken to the website of the trustmark, but to a domain that looks like the

trustmark’s website. An example of a real certificate can be seen in Figure 12a. A fake certificate is shown in Figure 12b. The differences between the real certificate and the fake certificate are subtle. If the picture in the fake certificate would show the logo of the webshop, it is hard to notice the difference. The most notable difference is the URL of the certificate. Certificates given out by the trustmark organization are always hosted by the trustmark organization itself. The URL of the fake certificate shows that it is hosted by the fake webshop. trustmarks warn for these fake certificates by informing the client on their website. trustmarks offer the client more confidence, but clients can also be misled into believing that the fake certificate is real.

7.2 Finding invalid trustmarks

As stated in Section 3 attackers lure clients to a fake webshop. The clients should determine that a webshop is fake by looking at details given on the website. A trustmark is one of these details that can help determine if the webshop is fake. The trustmark organizations can warn against fake webshops, but not every client checks these warnings often. When clients find a webshop that looks suspicious and has a trustmark logo, the client can report this webshop at the trustmark website. Finding a webshop that has no valid certificate, but does have a trustmark logo proved to be relatively easy. A Google image search on a trustmark logo shows the many websites that use the logo. The results of such an image search can be split up in four categories:

1. Webshops with a valid trustmark logo and certificate
2. Webshops that haven’t paid annual contribution for their trustmark, making it no longer valid
3. Webshops that have no right to display the trustmark logo
4. News articles about trustmarks on a news website that displays a logo of a trustmark next to the news text

To find webshops without a valid certificate, we performed such image searches. The logo we searched for was the logo of Thuiswinkel Waarborg. To get the logo, a normal Google search for images was performed on the keywords ‘Thuiswinkel Waarborg’. This yields images in all shapes and sizes. We used several of these results as input for the image search. This yields all the websites that use this image. For a single image search there are around 135 results. Most results are in the first category. Webshops often sell more than one product and every product is on a dedicated page. The results of an image search often contain many pages for the same website. Legitimate webshops that have a valid trustmark and certificate are therefore the largest category. About 1% of the results are in the second category. Some of the results were of webshops that were previously a member of Thuiswinkel Waarborg. The webshop owner decided not to pay the annual contribution and therefore, the certificate expires. The logo remains on



**CERTIFICAAT
THUISWINKEL WAARBORG**

Thuiswinkel.org verklaart dat haar lid:



het Certificaat Thuiswinkel Waarborg mag voeren.

Dit betekent dat BCC is gecertificeerd door de Stichting Certificering Thuiswinkel Waarborg. De bij de certificering geconstateerde werkwijze en gehanteerde voorwaarden zijn in overeenstemming met relevante wet- en regelgeving en de Gedragsregels Thuiswinkel Waarborg van Thuiswinkel.org. De (her)certificering vindt jaarlijks plaats.

BCC werd voor het eerst gecertificeerd op 30 maart 2010
BCC biedt u ten minste de volgende 5 zekerheden:

1. Weten met wie u zaken doet en hoe u het bedrijf gemakkelijk kunt benaderen
U doet zaken met:

BCC

Vestigingsadres:	Bellsingel 61 1119 NT SCHIPHOL-RIJK	Postadres:	Postbus 75513 1118 ZN SCHIPHOL
Telefoonnummer:	0900 0555 (10 cent pm)		
E-mailadres:	info@bcc.nl		
Webadres:	www.bcc.nl		
KvK-nummer:	33156765		
Btw-nummer:	NL806471670B01		
Lid sinds:	14-10-2004		



2. Algemene voorwaarden overeengekomen met de Consumentenbond
Voor u zijn de duidelijke algemene voorwaarden van BCC direct toegankelijk en van toepassing. De Algemene Voorwaarden Thuiswinkel

(a) real certificate of bcc.nl



**CERTIFICAAT
THUISWINKEL WAARBORG**

Thuiswinkel.org
verklaart dat haar lid:



het Certificaat Thuiswinkel Waarborg mag voeren tot
21 juni 2015

Dit betekent dat BCC is gecertificeerd door de Stichting Certificering Thuiswinkel Waarborg. De bij de certificering geconstateerde werkwijze en gehanteerde voorwaarden zijn in overeenstemming met relevante wet- en regelgeving en de Gedragsregels Thuiswinkel Waarborg van Thuiswinkel.org. De (her)certificering vindt jaarlijks plaats.

BCC werd voor het eerst gecertificeerd op 21 juni 2012.
BCC biedt u ten minste de volgende 5 zekerheden:

1. Weten met wie u zaken doet en hoe u het bedrijf gemakkelijk kunt benaderen
U doet zaken met:

BCC

Vestigingsadres:	Amsterdamsweg 49 3812RP, Amerstoft	Postadres:	Amsterdamsweg 49 3812RP, Amerstoft
E-mailadres:	info@BCC.nl		
Webadres:	www.BCC.nl		
KvK-nummer:	33156765		
Btw-nummer:	NL833156765B01		
Lid sinds:	21-6-2012		



2. Algemene voorwaarden overeengekomen met de Consumentenbond
Voor u zijn de duidelijke algemene voorwaarden van BCC direct toegankelijk en van toepassing. De Algemene Voorwaarden Thuiswinkel zijn vastgesteld in overleg met de Consumentenbond onder toezicht van de Sociaal

(b) fake certificate of bcc-marktplaats.com

Figure 12: Real and fake certificate of a trustmark

the website. If the client clicks on the logo, a warning message is displayed at the website of the trustmark organization. The warning states that the webshop is no longer a member. It is unknown if the Thuiswinkel Waarborg trustmark organization orders merchants to take down the logo or not. About 2 to 3% of the results are in the third category. Several webshops can be found that have no right to display the trustmark logo. Sometimes, a merchant uses multiple domains for different shops. A Thuiswinkel Waarborg trustmark is only valid for a single domain. Some merchants use this trustmark on all of their domains. Other trustmarks may support multiple domains for a single membership. For example WebwinkelKeur allows five domains. Other websites that have no right to display the trustmark logo are no members of the trustmark organization. An interesting example is a website that indexes holiday last minutes. The website itself is not registered and only displays last minute holidays of other travel agencies. These other travel agencies are legitimate and well known travel agencies. The indexing website displays logos of Thuiswinkel Waarborg as well as ANVR and SGR, which are trustmarks for travel agencies. The indexing site is not a member of Thuiswinkel Waarborg, ANVR or SGR. Another interesting example is of a webshop that sells supplies to grow weed. In the Netherlands, it is legal to grow up to five plants of weed in sunlight. The supplies that the webshop sells help the client grow larger amounts with lights and climate control. The only thing this webshop does not sell is the seeds. Therefore, the supplies they sell are probably legal, because they can be used to grow other plants. Still this webshop is not a member of Thuiswinkel Waarborg and does not have a logo. Both the holiday indexing website and the weed supply webshop are reported at Thuiswinkel Waarborg for abuse of the logo. In a reply the Thuiswinkel organization said to address these webshops. About 3 to 4% of the results are in the fourth category. Some of the results are of news websites that have an article about trustmarks. These results are not interesting in finding fake webshops or webshops that don't use the trustmark logos properly.

This test for the use of trustmark logos can be automated. A program that checks all result links for the image search can flag a website if the logo is not properly used. This method will also flag news websites with a news article about trustmarks. These are false positive results, so a filter should be applied for news websites. It is unknown to us whether the trustmarks perform such scans themselves or not. We believe such a scan is a good way of finding fake webshops or webshops that don't use the trustmark logos properly. Webshops in the first and second category of results are already known to the trustmark organization. They know who their members are and which webshops were previously a member of their trustmark organization. This knowledge can speed up the process, because these domains can be skipped in the automated test.

There is a tool that claims to check the validity of trustmarks for webshops. *WebwinkelChecker* is a browser plug-in that shows a small information bar on the screen when a client visits a known webshop. The information bar shows the trustmarks of a webshop as well as reviews about the webshop written by other clients. This way, a client does not have to check the validity of trustmark logos and certificates himself. *WebwinkelChecker* does that for the client. The

plug-in does not work for every webshop, but new webshops can be added by contacting the author of the plug-in. This plug-in does not scan the webshops automatically, nor does it check for invalid use of a trustmark by doing an image search. It is however an extra validation for a client and automatically shows when visiting a webshop.

7.3 European trustmark

In September 2014, a new European trustmark was announced by the European Multi-channel and Online Trade Association (EMOTA)[10]. This European trustmark should be displayed alongside a national trustmark and is not handed out to merchants without a national trustmark. The EMOTA trustmark should give consumers some guarantees about quality of the products they order in a webshop that is not in their own country, but in another European country. The trustmark will also handle dispute resolution on a European level as national trustmarks do on a national level. In the Netherlands, EMOTA works with the WebwinkelKeur Foundation. Only this foundation can hand out EMOTA trustmarks for Dutch webshops. Not all European countries participate in this initiative. EMOTA will start their certification process early 2015.

8 Privacy

Privacy of the client has not been a consideration for the countermeasures described in Section 6. The main goal of this thesis is to find the differences between iDEAL and other payment systems that allow a client to be manipulated and to find a way to prevent this manipulation. Privacy is however an important issue to discuss. Therefore it is discussed in this section.

In an iDEAL transaction, several details of the client are shared with both the bank and the merchant.

- The bank knows where the client is doing his purchases, for what amount and - if the description field includes item descriptions - the items bought.
- The merchant knows the address details of the client to send the product (if the product is a physical product) and the bank account number of the client.

8.1 Client information sent to the bank

As described in Section 4, merchants often use the *description* field in the iDEAL protocol to repeat the *purchaseID*. The client can be misled into paying for a transaction he did not create. To mitigate this, the details of the transaction have to be clearer. iDEAL transactions are directly coupled to bank transactions. Therefore, iDEAL transactions can be seen on the bank statement. The client is able to see the transaction details of the iDEAL transaction, like the *description* field. If the transaction details become more clear for the client during the transaction approval by indeed including item descriptions or by extending the *description* field, it also affects the details shown on the bank statement. This may cause privacy issues for shared or managed bank accounts. Everybody with access to the shared account can see in full detail the items that are bought in an iDEAL transaction, instead of an order number for a transaction. A surprise gift for a spouse bought with iDEAL will not be a surprise anymore. The bank obviously also has access to the bank statements of the client. If item descriptions are included in iDEAL transactions, the bank knows which items the client buys at which stores. This is a goldmine of information for the bank. Dutch banks have already been experimenting with the analysis of client data. ING wanted to get offers for their clients by coupling client data with advertisers[8]. Consumer authorities as well as the House of Representatives have responded that this was not such a good idea. The Rabobank also wants to analyze client data, but they want to stay far away from selling the data to third parties[7]. The data should be used to further strengthen the relation between the bank and the clients.

In order to protect the privacy of the client, the client should be able to choose how an iDEAL transaction is displayed on his bank statement. Webshops that sell products that people can be ashamed of (related to sexual activities) already offer such a service. The client is able to choose how to display the

merchant name and description details in the iDEAL transaction. As it is now, the iDEAL fields are directly used in the bank statement. Therefore, such a service of the merchant affects both the iDEAL transaction and the bank statement. If the iDEAL protocol would have separate fields for the bank statement, the client is still able to verify the transaction in full detail, but the bank statement can contain less information. This would also be the case if the iDEAL protocol used separate fields for multiple items purchased, where each item has an item description. Banks still handle both the iDEAL transaction and bank statements, so this does not solve the privacy issue, but it can provide the client with more options. Furthermore, we believe that banks should always use an opt-in mechanism for data analysis rather than opt-out. This data analysis should not be forced upon clients.

Sending identifying information about the client - like the client's address for the countermeasure described in Section 6.5 - from the merchant via the merchant's bank to the client's bank may cause more privacy issues. The banks have to securely process this information according to the European and Dutch law. This might be a big hassle for the banks.

8.2 Client information sent to the merchant

The client has to supply a shipping address in order for his products to be sent to him. After a transaction, the merchant is also able to see the bank account number of the client. When the merchant also asks for the birth date of the client in the ordering process, the merchant has a lot of information to potentially harm the client. The merchant is able to call the bank and impersonate the client. He is able to answer security questions asked on the phone by the bank like: 'when is your birth date' and 'what is the zip code and street number of your home'. Such misuse of the information of the client is not likely to happen as the merchant is registered at several places, such as chamber of commerce and the bank. Getting the information necessary to impersonate the client can be much more easily acquired through phishing attacks. Furthermore a call to the bank to handle bank affairs is getting less and less common. Many affairs can be handled through online banking. Therefore this is not a feasible privacy issue.

9 Related work

It is very hard to find any related work about the iDEAL payment system. The only paper about the iDEAL payment system that can be found is a paper by employees of Currence. In this paper, S. Gonggrijp et al. (2013)[11] describe the success of iDEAL and explain why the iDEAL payment system got adopted by the Dutch consumers so fast. Online banking was already very popular in the Netherlands and credit cards are not often used. The iDEAL payment system makes use of the same online banking applications of the different banks. Gonggrijp et al. write: “Within three years of its launch in 2005, iDEAL became the leading payment method for Dutch web shops.” This paper shows that iDEAL is a unique payment method with a “four-party model”, where both the merchant’s bank and client’s bank play a role, whereas a three-party model only has a single platform which handles the transaction. PayPal and credit card payment systems are examples of a three-party model. According to the authors, the four-party model ensures both competition and cooperation, which benefits the consumer. The paper does not mention anything about the safety of the protocol or the way that clients are protected.

Another source that mentions the iDEAL payment system is a master thesis by A. Bouch (2011)[1], about the security of Visa’s 3-D Secure. The thesis discusses the advantages and disadvantages of the iDEAL payment system. Bouch poses that Visa’s 3-D Secure (and MasterCard’s SecureCode) adds an extra layer of security for transactions, but the solution is still not optimal. It is yet another password for the client to remember and if the credit card companies want to make so called ‘card not present’ transactions any better, the whole protocol has to change in order to improve the situation. According to Bouch, the 3-D Secure system as it is now will not remain for long. The weaknesses of using payment card data online have to be addressed. In a chapter about alternatives, the author mentions the iDEAL payment system as well as PayPal. Bouch states that these systems are built “as ‘e-commerce-only’ solutions” and that these systems do not have to “integrate or support a prior ‘real world’ payment system”. These protocols are therefore made with security in mind instead of making them backwards compatible. Based on the iDEAL and PayPal protocol, the author does an assessment of the security of the payment systems. Both iDEAL and PayPal perform fairly well in the test, although Bouch does not explain what a ‘fair’ rating really means. He compares both protocols with an ideal (not to be confused with iDEAL) payment system that has certain requirements to be secure. These requirements are: Confidentiality, Integrity, Authentication, Non-Repudiation, Availability, Implementation, Interoperability, Ease of Use and Scheme Protection. So overall iDEAL scores ‘fair’ on these points. Bouch did not include the transaction details as a requirement. Bouch mentions a lack of message-level assurances as a disadvantage of iDEAL. Because of this disadvantage, a phishing or man-in-the-middle attack is still possible. This would require an active attack against the customer. Messages between merchant and the merchant’s bank are however signed, so there is message-level

assurance. Bouch did not check the protocol thoroughly enough. The attack as described in Section 3 however is indeed an active attack against the client, as the attacker displays false information on the screen of the client by luring the client to a fake webshop. This attack fits the description of the author, but is probably not what the author had in mind. Bouch proposes a new payment system, but does not provide fixes for the current payment systems, as this thesis does. The new system Bouch proposes is a four-party model, without direct communication between client and the client's bank. Such a system does not prevent manipulation by an attacker like the attack on iDEAL described in Section 3.

Similar to iDEAL, it is hard to find papers about specific attacks on the PayPal payment system. G. Maatoug et al.[13] use formal models of payment protocol implementations to find attacks against the protocol. In their paper, G. Maatoug et al. use PayPal express as an example to find and successfully launch an attack on the protocol. The authors create an abstraction of the protocol. The messages sent over the protocol are not included in the model checking. Therefore, the attack on iDEAL as described in Section 3 can't be detected by such a formal model check.

Other papers on PayPal are about fraud detection and phishing. When PayPal first started, some attacks were still possible on the payment system. One of these attacks was a phishing attack on PayPal customers[9]. Spammers sent out fake emails to people to get them to create a PayPal account. All the information to register was entered into a fake website. This website created a real PayPal account for the client, but the attackers also had all the information of the client, because the client entered it on the website of the attacker. A similar attack used cross-site scripting (XSS) to inject a warning message into the official PayPal page. This redirected users to a fake help center to 'unblock' their account[16]. In 2007, PayPal introduced multi-factor authentication for its users to further prevent phishing and identity theft[2].

Attacks on PayPal similar to the attack on iDEAL described in Section 3 are not found in literature. We believe such an attack has never occurred for PayPal, because the PayPal protocol has clear descriptions of the items bought by the client. Therefore it is hard to manipulate the client into paying for a transaction the attacker created.

10 Future work

It would be interesting if clients could automatically be warned against fake webshops. *WebwinkelChecker* as mentioned in Section 7 looks like a good start, but it needs input from users to include new webshops. It only shows webshops that are trustworthy, which is not really helpful if clients want to order products from an unknown webshop. Scanning the web for trustmark logos and finding webshops that misuse the logo would be a good way of indexing fake webshops.

The implementation of some of the countermeasures as described in Section 6 is another interesting topic of research. These implementations should not introduce new vulnerabilities for the iDEAL protocol. Formal validation of the specific implementations may be needed in order to prove that the implementation indeed does not introduce a vulnerability.

Other questions that were raised during the writing of this thesis are:

- Will security of iDEAL payments on a mobile phone increase when instead of a redirect to the transaction approval page of the bank, the client is redirected to the mobile app of the bank? Some banks already provide this functionality in their mobile banking app.
- PayPal makes use of merchant logos to let the client know who the recipient of a transaction is. Does the use of logos for recipients improve the recognition of the merchant? Some banks already show logos in online bank statements.
- Will the difference between online banking and paying in a physical shop eventually fade? In Belgium a client is able to pay in a physical shop using online banking on his phone by scanning a QR-code. Something similar could be used for iDEAL.

11 Conclusions

In this thesis, we looked at the iDEAL payment system and the specification of the transaction details in the protocol. By comparing both the protocol and the transaction details of iDEAL to other payment systems, we found that a client can be manipulated into buying a product he did not order. The different payment systems we looked at are similar in protocol, but are distinct in which details are shown at which specific webpage. The systems also differ in where the client has to fill in his approval, namely the untrusted webshop versus the trusted bank environment or payment system environment. For credit card transactions there are fewer details about the product and recipient. For some webshops the credit card information even has to be filled in at the webshop or at a PSP, instead of a trusted website. The ‘Secure Code’ or ‘3-D Secure’ for credit card payments does have to be filled in in a trusted environment, but this extra step is not used for every transaction. The merchant may decide whether to require this for his transactions or not. The iDEAL payment system also shows few (relevant) details, but does show all the details on the transaction approval screen of the bank, which is a trusted environment. This transaction approval screen is also where the client approves of the transaction, so no bank account details other than the bank account number go to an untrusted merchant. PayPal shows relevant information on a transaction approval screen at the PayPal website, which is a trusted environment. PayPal also does not leak account details to an untrusted merchant. So credit card transactions are the least verifiable transactions for the client, whereas PayPal transactions are the most clear.

The three main reasons that an attacker is able to manipulate a client for the iDEAL payment system are:

- It is unclear to the client who the recipient of the payment is.
- It is unclear what the client is paying for.
- There is no validation that the person who selects the products in the webshop of the merchant, is the same person who approves the transaction on the transaction approval page of the bank.

Who the recipient of the transaction is

To determine who the recipient of the payment is, the iDEAL protocol uses the merchantID and subID. The bank or CPSP then use the legal name of the company that the merchant provided in the contract. The name of the company is not always the same as the name of the webshop. Therefore it can be unclear who the recipient of the payment is. To improve this, we propose that the banks show the domain name of the webshop on the transaction approval screen. There is no need to include an extra field for this, since the domain is obviously included in the *returnURL* field. Even if the name of the company is not the same as the name of the webshop, the client will then know if this is indeed the webshop where he wanted to buy his products.

What the client is paying for

The iDEAL protocol uses the *TransactionRequest* to communicate the transaction details from the merchant to his bank and the client's bank. The *purchaseID* field in this *TransactionRequest* is the identifier used by the merchant to refer to the order. This identifier is just a number that the client does not need unless he goes to the webshop and wants to see his order history. The *description* field in the *TransactionRequest* should be used to describe the products bought. The maximum amount of characters in the *description* field is 35. This is not enough if multiple different products are to be described. Merchants fill in the *purchaseID* in the *description field*. Therefore it can be unclear what the client is paying for. To improve this we propose that the description field should always contain the item description (or as much as fits of an item description). A longer description field could fit multiple descriptions of products. Another improvement would be to include a list of items like PayPal does. Each item in the PayPal protocol allows a description of 127 characters, such that there is no shortage of space to describe the entire purchase. The items are shown on the transaction approval screen in the PayPal payment system.

Validation of the client between merchant and client's bank

If both the merchant and bank could verify that they are talking to the same client, an attacker could no longer select products and let the client pay for them. This would make it a lot harder to set up the attack. We propose that the IP address of the client is included as a field in the *TransactionRequest*. It has no use for the attacker to spoof the IP address at this point in the protocol. The bank can check if the transaction approval page is indeed visited by the same client based on the IP address of the client.

Another method of client identification is by letting the client register his bank account number at the merchant. This method was proposed by Currence to prevent the attack. This method only delays the attack to make it less interesting. The attack is still possible depending on the implementation by the legitimate webshop. A disadvantage of this method is that it takes longer for the client to buy a product. The bank account first has to be verified at the merchant before a product can be bought. This introduces extra hassle for the client.

Address details via the bank Finally we described another verification for the client, where the merchant would add address details to the *TransactionRequest*, such that the client would know where his product is being shipped. If the attacker had a hand in setting up the transaction, the address would not be the same as the address the client supplied. If the address is unknown to the client, he can cancel the transaction. Some more far fetched variations of address verification could be implemented as well.

Trustmarks

In this thesis we also described some trustmarks in the Netherlands that should

provide confidence to the client. These trustmarks can provide an extra failsafe if clients know what details to look for to know the difference between a fake and a legitimate webshop. The client has to follow the link in the trustmark logo to go to the trustmark website. There he can verify if the certificate for the webshop is correct. These details can be subtle and are therefore not the best solution to detect fake webshops. Trustmarks do provide extra details if the client was already in doubt about the legitimacy of the website. Therefore trustmarks are a good addition in detecting fake webshops.

11.1 Specific recommendations

Below the recommendations per party, involved in the iDEAL payment system, are described. The recommendations are ordered best ‘return-on-investment’ first, per party.

for Currence

- A1 Increase the size of the *description* field of the iDEAL protocol, such that multiple item descriptions would fit.
- A2 Create a new *IP address* field in the iDEAL protocol that holds the IP address of the client and check if the person selecting the products is the same person that pays for the products.
- A3 Create a new *item* field in the iDEAL protocol that holds an amount per item and a description per item.
- A4 Create new fields for bank statement information to enhance privacy.

Recommendation [A1](#) and [A3](#) are not very different from each other. [A1](#) may cause problems in backward compatibility with bank statements. We still believe it gives bigger ‘return-on-investment’.

for the iDEAL banks

- B1 Show the domain part of the *returnURL* field of the iDEAL protocol on the transaction approval screen.
- B2 Ask for both the company name and the webshop name in the iDEAL contract with the merchant and show both on the transaction approval screen.
- B3 Force merchants to use the description field of the iDEAL protocol properly.

for PSPs

- C1 Ask for both the company name and the webshop name in the iDEAL contract with the merchant and make sure it shows on the transaction approval screen.
- C2 Force merchants to use the description field of the iDEAL protocol properly.

for Merchants

- D1 Always fill the *description* field of the iDEAL protocol with item descriptions as much as fits.
- D2 Show an overview transaction details to the client before redirecting the client to the bank.
- D3 Don't create many webshops under the same company name.

for Clients

- E1 Check all the details in the transaction approval screen before approving the transaction. A new Internet banking safety commercial could be used to inform consumers how to check their transaction.
- E2 Check the usage of trustmarks by the merchant. Click the trustmark logo to go to the webpage of the trustmark (and not the domain of the attacker).

for Trustmark organizations

- F1 Scan the web for usage of the images of the trustmark logo and check for invalid use of the logo.

We believe the attack as described in Section 3 could and should be prevented by showing more relevant details to the client at the time of transaction approval. The client should know what he pays for, who the recipient is and where his product is going. Once again the devil is in the (transaction) details.

References

- [1] Anthony Bouch. 3-D Secure: A critical review of 3-D Secure and its effectiveness in preventing card not present fraud. *Master Thesis*, Department of Computer Science, Royal Holloway University of London, 2011. http://www.58bits.com/thesis/3-D_Secure.pdf [Online; accessed 21-October-2014].
- [2] Steve Brunswick. eCommerce fraud—time to act? *Card Technology Today*, 21(1):12–13, 2009.
- [3] About-Payments.com B.V. Payment Providers, 2014. <https://www.about-payments.com/knowledge-base/providers> [Online; accessed 27-August-2014].
- [4] Computerworld. Oplichting door nepshops explodeert, 2014. <http://computerworld.nl/beveiliging/81210-oplichting-door-nepshops-explodeert> [Online; accessed 15-June-2014], Website in Dutch.
- [5] Currence. iDEAL.nl, 2014. <http://ideal.nl> [Online; accessed 25-July-2014].
- [6] Currence. Jaarverslag 2013, 2014. http://www.currence.nl/Downloads/Cu_CurrenceJVNL2013.pdf [Online; accessed 15-September-2014].
- [7] Het Financieele Dagblad. Rabobank wil grens opzoeken in analyse van betaalddata klanten, 2014. <http://fd.nl/ondernemen/20515/rabobank-wil-grens-opzoeken-in-analyse-van-betaalddata-klanten> [Online; accessed 7-November-2014].
- [8] Nederlands Dagblad. ING wil gegevens klanten verkopen, 2014. <http://www.nd.nl/artikelen/2014/maart/11/ing-wil-gegevens-klanten-verkopen> [Online; accessed 7-November-2014].
- [9] Tamara Dinev. Why spoofing is serious internet fraud. *Communications of the ACM*, 49(10):76–82, 2006.
- [10] EMOTA. A pan European network of e-commerce trustmarks, 2014. <http://www.emota.eu/#!european-ecommerce-trustmark/cijas> [Online; accessed 29-October-2014].
- [11] Stefan Gonggrijp, Max Geerling, and Piet Mallekoote. Successful introduction of new payment methods through co-opetition. *Journal of Payments Strategy & Systems*, 7(2):136–149, 2013.
- [12] Currence iDEAL B.V. iDEAL Merchant Integratie Gids, 2014. Available via Rabobank version 3.3.1 may 2014 https://www.rabobank.nl/images/20140317_ideal_merchant_integratie_gids_v3_3_1_nl_maart_2014_29542840.pdf [Online; accessed 27-August-2014].

- [13] Ghazi Maatoug, Frédéric Dadeau, and Michael Rusinowitch. Model-Based Vulnerability Testing of Payment Protocol Implementations. In *HotSpot'2014, 2nd Workshop on Hot Issues in Security Principles and Trust, affiliated with ETAPS 2014*, Grenoble, France, April 2014.
- [14] Info Security Magazine. De helft van alle financile instellingen vergoedt schade door cybercrime zonder onderzoek, 2014. <http://infosecuritymagazine.nl/2014/07/10/de-helft-van-alle-financiele-instellingen-vergoedt-schade-door-cybercrime-zonder-onderzoek/> [Online; accessed 5-October-2014].
- [15] Omroep MAX. Tijd voor MeldPunt! - Internetoplichting, 2014. <http://www.meldpunt.tv/home/uitzending/meldpunt-donderdag-30-januari-2014/> [Online; accessed 15-June-2014].
- [16] Paul Mutton. Paypal security flaw allows identity theft. *Netcraft*, 2006. news.netcraft.com/archives/2006/06/16/paypal_security_flaw_allows_identity_theft.html [Online; accessed 29-October-2014].
- [17] TNS NIPO. Onderzoek TNS NIPO naar thuiswinkelgedrag en de bekendheid van het Thuiswinkel Waarborgkeurmerk in Nederland, 2011. https://www.thuiswinkel.org/data/uploads/marktonderzoeken/bekendheid_thuiswinkel_waarborg/Onderzoek_Bekendheid_Thuiswinkel_Waarborg_2011.pdf [Online; accessed 19-October-2014].
- [18] Masoud Nosrati, Ronak Karimi, Kamran Makekian, and Mehdi Hariri. PayPal as the Most Loved Payment System among Merchants and Buyers in Online Transactions. *World Applied Programming*, 3(9), 2013.
- [19] Pay.nl. Stijging iDEAL gebruik blijvend?, 2014. <https://www.pay.nl/nieuws/stijging-in-ideal-gebruik-blijvend> [Online; accessed 15-September-2014].
- [20] PayPal. Explore further payment capabilities, 2014. <https://developer.paypal.com/webapps/developer/docs/integration/direct/explore-payment-capabilities/> [Online; accessed 8-September-2014].
- [21] PayPal. PayPal Developer: REST API reference, 2014. <https://developer.paypal.com/webapps/developer/docs/api/#create-a-payment> [Online; accessed 31-August-2014].
- [22] PayPal. Safety and Security, 2014. <https://www.paypal.com/webapps/mpp/paypal-safety-and-security> [Online; accessed 14-October-2014].
- [23] Nederlandse Vereniging van Banken. Fraude internetbankieren daalt met meer dan de helft, 2013. <http://www.nvb.nl/nieuws/2013/2105/fraude-internetbankieren-daalt-met-meer-dan-de-helft.html> [Online; accessed 5-October-2014].

- [24] Nederlandse Vereniging van Banken. Uniforme Veiligheidsregels Particulieren, 2013. https://www.veiligbankieren.nl/downloads/001029_uniforme-veiligheidsregels_1.pdf [Online; accessed 5-October-2014].
- [25] Visa. Chargeback management guidelines for Visa merchants, 2014. <http://usa.visa.com/download/merchants/chargeback-management-guidelines-for-visa-merchants.pdf> [Online; accessed 6-October-2014].