Electronic Commerce Research, 4: 83–111 (2004)

Towards Pseudonymous e-Commerce

MARC RENNHARD
Swiss Federal Institute of Technology, Computer Engineering and Networks Laboratory, Zurich, CH

SANDRO RAFAELI
Lancaster University, Computing Department, LAI 4YR, Lancaster, UK

LAURENT MATHY
Lancaster University, Computing Department, LAI 4YR, Lancaster, UK

BERNHARD PLATTNER
Swiss Federal Institute of Technology, Computer Engineering and Networks Laboratory, Zurich, CH

DAVID HUTCHISON
Lancaster University, Computing Department, LAI 4YR, Lancaster, UK

Abstract

The lack of privacy is one of the main reasons that limits trust in e-commerce. Current e-commerce practice enforces a customer to disclose her identity to the e-shop and the use of credit cards makes it straightforward for an e-shop to know the real identity of its customers. Although there are some payment systems based on untraceable tokens, they are not as widely used as credit cards. Furthermore, even without buying anything, a customer is already disclosing some information about who or where she may be by just connecting to the e-shop's web server and leaving behind an IP-address. In this paper, we present novel components that enable secure pseudonymous e-commerce. On the one hand, these components allow a customer to browse through an e-shop, select goods, and pay the goods with her credit card such that neither the e-shop operator nor the credit card issuer nor an eavesdropper is able to get any information about the customer's identity. On the other hand, it is guaranteed that none of the involved parties is able to act dishonestly during the credit card payment. Such a system could greatly enhance trust in e-commerce since it overcomes the customers' privacy concerns.

Keywords: privacy, anonymity, pseudonymity, e-commerce, trust, Mix-networks, pseudonymous payment, pseudonymous credit cards

1. Introduction

During the past years, people have become more sensitive regarding privacy issues in the Internet. They realised that they leave all sorts of traces when surfing the Web or exchanging e-mail messages. Encryption provides the means to protect the privacy with respect to third parties in the sense that eavesdropping becomes very difficult, and the popularity of tools such as Pretty Good Privacy (PGP) [Zimmermann, 29] underlines its need. However, there are situations where the protection of privacy should go even further, and that is when anonymity comes into play. The requirement for anonymity in the Internet is completely justifiable since many situations in the real life are anonymous, e.g., traditional stores offer

a certain degree of anonymity for their customers if they pay with cash. Therefore, it is desirable that online shopping offers this anonymity as well.

Generally, the process of buying goods at an e-commerce store consists of several parts: (1) a customer (Alice) browses through the e-shop, looks at information about products, chooses the products she wants, and fills them into her shopping cart; (2) she proceeds to the checkout and provides a credit card, which is checked by the e-shop; (3) the e-shop may contact Alice via e-mail to inform her about the situation of the order; (4) after Alice's credit is cleared out, she has access to the products she has paid for.

Traditionally, none of these parts is anonymous. When browsing through the e-shop, IP-packets are sent from the customer's machine to the e-shop and vice versa. These packets contain the sender and receiver's IP-addresses. Note that most modern browsers employ advanced security options and give the user the option to disable Java applets, JavaScripts, and cookies. In addition they can establish encrypted and authenticated connections to protect the data exchanged between the browser and the web server, but none of these measures gives the user anonymity since each packet still carries the IP-addresses of both sender and receiver. An eavesdropper or the e-shop can easily derive Alice's identity or at least the name of the computer she is using from those packets. When Alice accesses her e-mail server to check if there are messages from the e-shop, she also leaves behind traces to her identity. More seriously, the messages sent by e-shops are rarely encrypted, and hence, can be easily accessed by eavesdroppers. Finally, when Alice has to submit information about her credit card to the e-shop, she discloses even more of her identity.

The main problem in the online world is that Alice has absolutely no control over what happens with the traces she leaves in the Internet. Merchants can sell data about her shopping habits to others. Administrators of web servers can give away their server logs to other companies. Somebody can collect a lot of data and prepare an extensive profile on Alice's personal preferences and sell this information to others. Alice has no way to know if this is happening. In the real life, Alice has much more control about the information she is willing to give away. If she wants to keep her privacy in a bookstore, she does not tell anybody her name and simply pays with cash. This is not possible in the online world, as we have explained above. We strongly believe that this lack of privacy is one of the main reasons that hinders a faster growth of e-commerce, since it limits trust in online shopping by its potential customers.

The World-Wide Web Consortium (W3C)'s Platform for Privacy Preferences (P3P) project [Reagle and Cranor, 18] is certainly a step in the right direction to protect the privacy of web users. The goal of P3P is to clarify a web site's privacy practices to its users. When a web site or e-shop is contacted, the user receives the privacy practices of that site. The user can accept these practices, reject them and ask for an alternative proposal, or send another proposal herself. If an agreement between user and web site is reached, the communication continues, otherwise it is terminated. However, P3P only specifies the protocol for exchanging structured data to reach an agreement, but it cannot do anything to enforce the privacy practices a web site has proposed: even if an e-shop has promised not to give away information about the user to third parties, there is no way for the user to check if the e-shop complies with the rules. In addition, P3P cannot do anything to hide the user's identity from the e-shop.

An anonymous record or transaction [Clarke, 7] is one whose data cannot be associated with a particular individual, either from the data itself or by combining the transaction with other data. Examples of anonymous transactions are casting a vote in a ballot or a cash payment. A pseudonymous transaction [Clarke, 7] is one that is identified by a pseudonym and cannot (in the normal course of events) be associated with a particular individual. This means that a transaction is pseudonymous in relation to a particular party if the transaction data contains no direct identifier for that party. However, if a specific piece of additional data is available, then the transaction data can be linked to that party. This piece of additional data could be an entry in an index that maps a party to its pseudonym. To be effective, a pseudonymous mechanism must involve legal and technical protections, such that the link can only be made (i.e. the index can only be accessed) under certain circumstances [Pfitzmann et al., 17]. Particularly in the context of e-business, anonymity may lead to fraud. For a trusted platform, pseudonymity is therefore a requirement.

In this paper, we describe novel pseudonymity components that enable secure pseudonymous e-commerce in the Internet. The components enable a customer to browse through an e-shop, select goods, and pay the goods with her credit card such that neither the e-shop operator nor the credit card issuer nor an eavesdropper is able to get any information about the customer's identity. E-commerce based on these components could greatly enhance trust in e-commerce since it overcomes the customers' privacy concerns.

The *Pseudonymity Network* (PN) is the first component and enables browsing the Web anonymously. It is not limited to e-commerce, but can be used for accessing any web server in the Internet. It is based on a set of distributed proxies that are operated by independent institutions. It allows users to browse the Web such that neither the web server nor any eavesdropper nor the independent operators of the proxies can find out where those users are, who they are, and where they are going.

In this paper, we present the design and the internal functionality of the PN. In addition, we provide some performance measurements to determine optimal parameters of the PN-internal protocol. We believe that we have found an innovative and efficient way of operating the system that makes it very resistant against attacks without introducing too much overhead.

The *Pseudonymous Secure Electronic Transaction* (PSET) payment protocol is the second component. Credit cards are very popular in e-commerce and this is not likely to change very soon. Therefore, the idea is that there could be financial institutions that would not require a user to reveal her real identity to obtain a pseudonymous credit card. This pseudonymous credit card could then be used to make pseudonymous credit card payments. In this paper, we present the PSET protocol, which is a novel protocol that enables pseudonymous credit card payments over the Internet. We also include an analysis of the protocol, which shows that it is not possible for anybody involved in the payment to act dishonestly.

Figure 1 shows how the two pseudonymity components fit into the e-commerce scenario: (1) The pseudonymity network is an overlay network in the Internet and hides the communication between the customer and the e-shop's web server. This communication includes the selection of goods by the customer and the initiation of the PSET payment between customer and e-shop. (2) The e-shop authorises the payment at the credit card

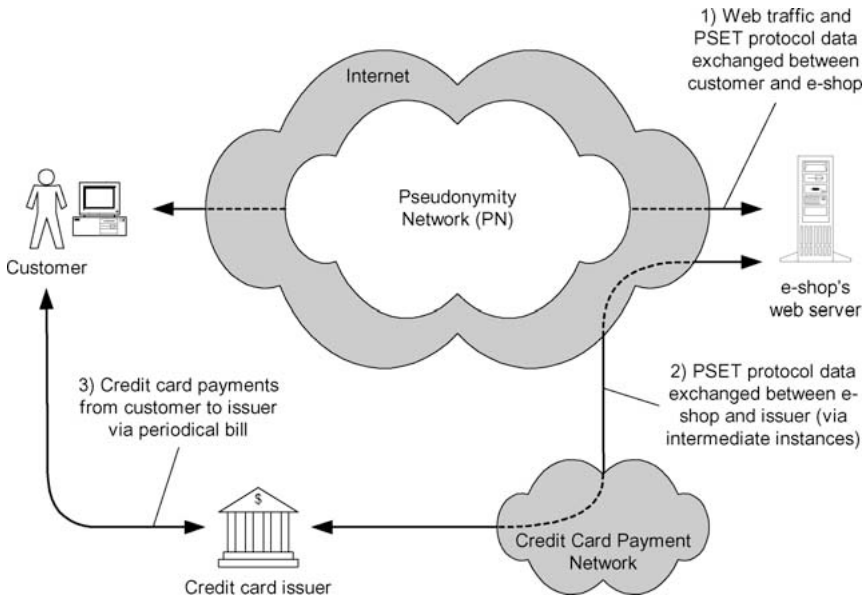

Figure 1. E-commerce and the two pseudonymity components.

issuer via intermediate instances (see Section 4.2) using the PSET protocol. This communication is done over the Internet and over today's credit card payment network. The protocol guarantees that the e-shop does not learn the identity of the customer and that the issuer does not learn the identity of the e-shop. (3) Finally, the customer is charged by the issuer via her normal monthly bill.

The remaining of this document is unfolded as follows. In Section 2, the requirements for pseudonymous e-commerce are defined. We describe the pseudonymity network in Section 3 and the pseudonymous secure electronic transaction protocol in Section 4. We present the limitations and give a few recommendations in Section 5 and conclude our work in Section 6.

2. Pseudonymity requirements

It is very difficult to predict how Internet-based e-commerce will work in a few years. The current procedures are simple and have not changed much during the last couple of years. The biggest security concern of today's e-commerce is the credit card payments, and it is widely believed that the industry has to adopt a more secure payment method in order not to limit the growth of e-commerce. Besides that, however, it is unlikely that the basic procedures are going to change. People will still access the e-shop via a web browser, pay with their credit card, and exchange information via e-mail. We use this scenario to design our pseudonymity components: e-commerce basically works as it does today, with

the addition that it will employ the Secure Electronic Transaction (SET) protocol [SET, 25] to process credit card transactions.

The main goal of our pseudonymity components is to enable pseudonymous e-commerce based on the scenario described above. Pseudonymous electronic commerce should have the following properties:

Property 1. *The e-shop is not able to find out any information about the customer's identity, neither during the exchange of web traffic nor when sending or receiving e-mail messages. By any information, we do not only mean the customer's identity, but also other data that could give hints at a customer's identity such as the IP-address of the computer the customer is using to access the e-shop. Similarly, an eavesdropper should not learn the identities of both the customer and the e-shop even if he performs sophisticated traffic analysis attacks.*

Property 2. *The e-shop is not able to acquire knowledge about the customer's real identity from credit card information provided.*

Property 3. *The credit card provider is not able to link the payment from a customer to an e-shop.*

Property 4. *The risk for the merchant or any of the other entities involved in the payment process is not greater than that in non-pseudonymous payments.*

Our pseudonymity components are operated by third party entities and, as such, require trust. The question is how much trust a customer is willing to have in a third-party service. One could argue that there is a single instance offering such a service, which means that nobody except that single instance can link a user's pseudonymous activities back to her. The Anonymizer [Cottrell, 8] is an example for a service offered by a single third-party to browse the Web pseudonymously. However, this requires much trust in that single instance and not every customer may accept that.

At least two entities should be in the middle of any pseudonymous communication: one closer to the client and the other closer to the server. We refer to these entities as pseudonymity entities. The entity closer to the client knows about the client's location, but does not know its destination. It only knows the other pseudonymity entity. On the other hand, the entity closer to the server knows who the server is, but does not know who is connecting to it since it only sees the entity closer to the client. The information about the entire end-to-end connection is divided between the pseudonymity entities, and they have to collude to recover the whole path. Each entity knows just a piece of the entire information. Furthermore, including more pseudonymity entities in the chain between client and server can diminish the level of trust on every party, but one must bear in mind that such an increased trust is achieved at the expense of increased operational overhead. We will make use of this basic approach in both pseudonymity components we will present.

3. The pseudonymity network

As seen in Section 2, a customer must not send data directly from her computer to the e-shop's web server since this would expose her computer's IP-address. At first glance, this might not be a big problem since most home-users today get a temporary IP-address assigned by their Internet Service Provider (ISP), which means that deriving the customer's identity from that address is not so simple (without cooperation from the customer's ISP, for instance). However, deployment of permanent access technologies (e.g., ADSL) also means that more and more home-users will possess a permanent IP-address, which is also the case for many office and university computers. Therefore, we want a solution that does not enable the server to derive any relevant information, such as name, country or IP-address about the user from the IP-packets it receives. This means that the IP-packets going from the user's side to the destination must be relayed by at least one intermediate proxy server. If one proxy server is used between customer and e-shop, then the e-shop's web server sees the proxy's source address in the received IP-packets and has no way to derive the customer's IP-address. However, for the reasons given in Section 2, there should be at least two independent proxy servers between the customer and the e-shop.

The Pseudonymity Network (PN) is a general mechanism that enables pseudonymous web browsing. It is not bound to e-commerce, but can be used for any browsing-activity on the Internet. It is based on a set of distributed proxies that should be operated by independent institutions. It enables users to browse the Web in a way such that neither the web server nor eavesdroppers and not even the independent operators of proxies can find out where those users are, who they are, and where they are going. This means that the goal of the PN is to provide sender anonymity and relationship anonymity [Pfitzmann and Köhntopp, 15]. The PN is based on Chaum's idea of a Mix-network [Chaum, 6] for e-mail messages but has been enhanced in an innovative way such that it efficiently supports near real-time data such as web traffic as well. The PN is similar to two other approaches (see Section 3.5), Onion Routing [Reed et al., 19] and the Freedom System [Boucher et al., 4], but we believe that our system is much more resistant against attacks without introducing too much overhead.

We first explain the basic functionality of the PN [Rennhard et al., 22] and then present measures to make the PN resistant against attacks.

3.1. Basic functionality

The PN consists of two or more *Pseudonymity Proxies* (PPs). The idea is that all communication between the customer and the e-shop is relayed by PPs.

Figure 2 depicts a communication between Alice and an e-shop, where the traffic is relayed via three PPs (PP_1, PP_2 and PP_3). Bob is also communicating with the e-shop via three PPs (PP_2, PP_3 and PP_4). The PPs operate on the application layer and a TCP-connection is used to connect a pair of PPs. Note that if multiple users use the same link between a pair of proxies (PP_2–PP_3 in Figure 2), then their traffic is transported within the single TCP-connection between the PPs. The sequence of PPs used in an anonymous connection from the user to the web server is named the *chain of PPs*.

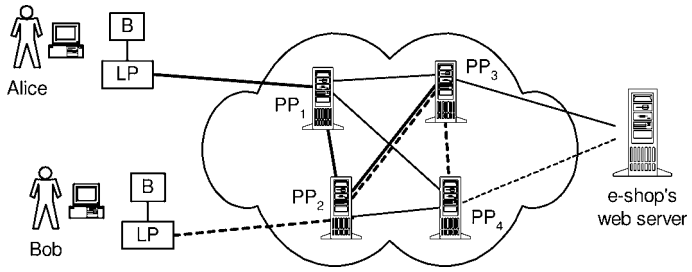

Figure 2. Pseudonymity network.

A user accesses the PN via a *Local Proxy* (LP). The local proxy is a program running in the user's computer and is accessed by the web browser (B) in the same way as a normal web proxy. Access to the PN is therefore transparent to the browser, and hence does not require any modifications. The LP has several important tasks. When it is started, it sets up the user's *anonymous tunnel* from the LP to the last PP in the chain. It then waits for requests from the web browser and uses the PN-internal protocol to forward them through the anonymous tunnel via the chain of PPs to the web server. The LP also sanitises the http-request it receives from the web browser and removes all data that could potentially identify the user, such as the operating system, the browser, or the web page where the user has initiated the request. The replies from the web server are sent back to the web browser via the chain of PPs (in reverse order) and the LP.

One cannot forget about traffic analysis. Any of the involved parties (the proxy servers and the e-shop) or any other party could do traffic analysis trying to expose the end-to-end connection. Traffic analysis means that the eavesdropper monitors the incoming and outgoing links of a proxy and tries to correlate the packets. If this is done at all intermediate proxies between customer and e-shop, it could be possible to find out an end-to-end connection. To protect the PN against traffic analysis, we use the approach of a Mix-network [Chaum, 6]. A Mix-network complicates traffic analysis using the following measures: (1) all traffic between a pair of proxies is encrypted (we name these encryptions *link encryptions),* this prevents that an attacker can successfully perform traffic analysis by simply looking at the content of packets; (2) all packets exchanged between the proxies have exactly the same length, this defeats traffic analysis based on correlating packets by their length; (3) a PP reorders incoming packets from different links such that the outgoing sequence of packets is not related to the incoming sequence. This makes traffic analysis based on timing very difficult because for an eavesdropper, each of the outgoing packets can correspond to each of the incoming packets with the same probability; and (4) dummy packets are sent from one proxy to another when this is required to maintain a certain level of anonymity.

However, these measures are not enough to protect the anonymity from internal attacks. Although the data sent through the PN is encrypted on each link between two proxies, each PP sees the same data after decrypting a packet. For instance, this means that the first PP in the chain would know the web server the user connects to since it sees the http-request. Furthermore, a set of PPs can collude to examine the data packets they received. If any

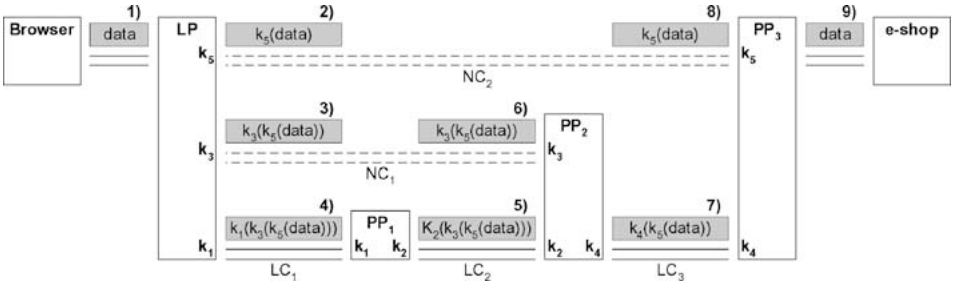

Figure 3. Layers of encryption.

two PPs have seen the same decrypted data, then they know that they were used in the chain of the same anonymous tunnel. We employ *nested encryptions* in addition to the link encryptions described above to defeat attacks by colluding PPs.

Figure 3 shows how the link encryptions and nested encryptions are used to protect the anonymity:

1. Alice's LP connects to PP_1, they perform a key-exchange algorithm which results in a common key k_1 and a secure channel LC_1 between LP and PP_1. LP then issues a request to PP_1 to connect to PP_2.
2. If PP_1 does not have a secure connection to PP_2 yet, they establish it now, which results in a common key k_2 to protect the secure channel LC_2.
3. LP performs a key-exchange algorithm with PP_2, which results in a common key k_3 and a secure nested channel NC_1 between LP and PP_2. Note that there is no direct connection between LP and PP_2, since all data (including the key-exchange itself) sent between them are relayed by PP_1. LP then issues a request to PP_2 (via PP_1) to connect to PP_3.
4. Similar as in step 2, PP_3 and PP_3 establish a secure channel LC_3 protected with a common key k_4. LP then performs a key-exchange algorithm with PP_3 (via PP_1 and PP_2), which results in a common key k_5 and the secure nested channel NC_2 between LP and PP_3.

Alice can now use her anonymous tunnel to send requests to the e-shop's web server, as illustrated in Figure 3. The browser sends data to LP (1). LP encrypts the data, first with k_5 for PP_3 (2), then with k_3 for PP_2 (3), and then with k_1 for PP_1 (4). The resulting data is sent to PP_1, which decrypts the packet with k_1, encrypts it with k_2 (5), and sends it to PP_2. PP_3 decrypts the packet with k_2 (6), decrypts the result with k_3, encrypts it with k_4 (7), and sends it to PP_3. Finally PP_3 decrypts the packet with k_4, decrypts the result with k_5 (8), and sends the resulting data to the e-shop (9). The reply from the web server follows the inverse path back to Alice.

This anonymous tunnel is now secure against internal attacks. Assume that PP_1 and PP_3 collude. After removing the link encryption with k_1, PP_1 knows $k_3(k_5(data))$. PP_3 knows $k_5(data)$ after removing the link encryption with k_4 and *data* after removing the nested encryption with k_5. However, they have no idea if $k_3(k_5(data))$ and $k_5(data)$ are the same data, since they do not know k_3, which is known only to LP and PP_2. In general,

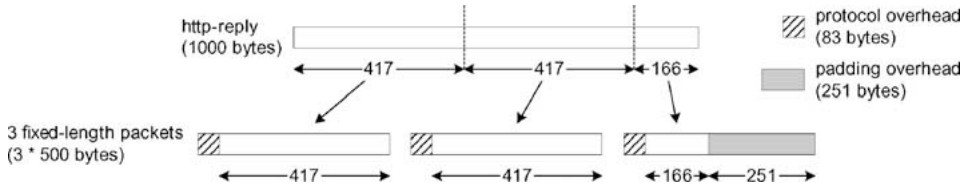

Figure 4. Padding overhead introduced by fixed-length packets.

no coalition of fewer than all of the involved PPs in an anonymous tunnel can break the anonymity, if nested connections are used.

Setting up a secure channel between two PPs in steps 2 and 4 is only needed if the two PPs do not have established a secure channel earlier. Since these secure connections are longstanding, their establishment is usually only needed after a PP has been started. Similarly, once the anonymous tunnel has been set up, it is used by the user for all her anonymous connections until the tunnel is torn down.

Furthermore, authentication plays an important role in the PN. In order to avoid unauthorised PPs to join the PN, two PPs authenticate each other when setting up the secure channel. In addition, the LP authenticates each PP in its anonymous tunnel when performing the key-exchange algorithm. The authentication of PPs is guaranteed by certificates based on the X.509 standard [Houselyc and Polk, 13].

All of the encrypted channels use the Secure Socket Layer (SSL) protocol [Dierks and Allen, 9]. The SSL protocol easily integrates the certificates we use to authenticate the PPs. The Diffie–Hellman (DH) key exchange protocol [Diffie and Hellman, 10] is used to establish the keys between two PPs and an LP and a PP.

An important property of the PN is that an anonymous tunnel is completely transparent for the web browser and the e-shop. This means that neither the browser nor the e-shop's web server needs to be adapted in any way. The PN also supports https-connections: when the browser connects to a secure web site (which is usually done when submitting payment information), the *data* shown in Figure 3 is now simply encrypted (between browser and web server). For the anonymous tunnel, it does not matter if the payload it transports is encrypted or not.

3.2. Fixed-length messages

To make the PN more resistant against traffic analysis, all packets exchanged between any pair of proxies should have the same length. Independently of the packet length, the PN protocol has some overhead: 20 bytes per packet are needed to route the packet correctly along the anonymous tunnel, and 21 bytes are needed per SSL-layer (5 bytes SSL-header and 16 bytes for the MD5-hash used by SSL). Since the number of SSL-layers corresponds to the number of PPs in the anonymous tunnel (Figure 3), there is an overhead of $20+n\cdot 21$ bytes per packet, where n is the number of PPs used in the anonymous tunnel.

Fixed-length packets introduce additional overhead, as Figure 4 illustrates. Assume that the length of the packets is 500 bytes, and that 3 PPs are used in an anonymous tunnel. This

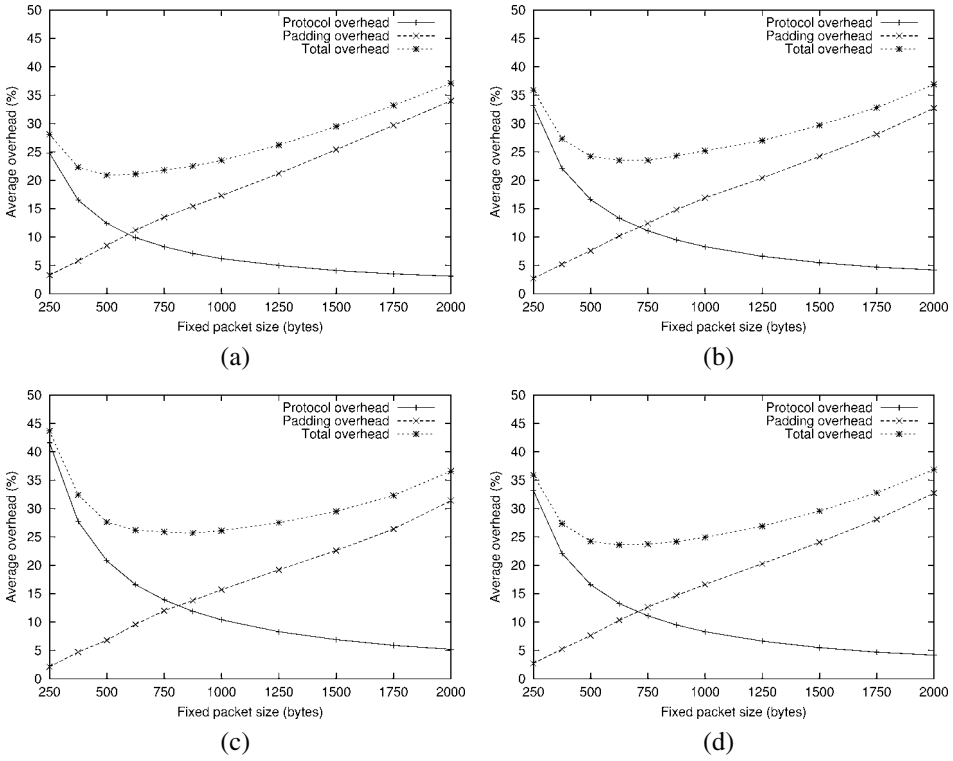

Figure 5. Protocol and padding overhead depending on the number of PPs. (a) Overhead when using 2 PPs. (b) Overhead when using 3 PPs. (c) Overhead when using 4 PPs. (d) Average overhead.

means that there are $20 + 3 \cdot 21 = 83$ bytes protocol overhead per packet, which leaves 417 bytes per packet for payload. If a user now makes a request for an html-document with a length of 1000 bytes, then three of the 500 bytes packets must be sent from the web server through the PN to the user. Since $1000 = 2 \cdot 417 + 166$, only 166 bytes of the payload of the last packet are used, which means that the remaining 251 bytes are unusable and filled with random data. We name this overhead *padding overhead*.

We have run some tests with our prototype implementation, in order to measure the optimal packet length. The probability distribution of the lengths of real http-replies follows the Pareto distribution [Feldmann et al., 11]. Therefore, we generated http-replies with lengths according to this distribution to make sure that the results are realistic. We ran the test using anonymous tunnels with 2, 3 and 4 PPs.

Figure 5 illustrates the results of the tests, which show clearly that protocol overhead is dominating for small and padding overhead is dominating for large packets. In between, there is a range (from 500–1000 bytes) where the total overhead is more or less constant. Figure 5(d) shows the average of the other three figures, since users are free to select any number of PPs they wish, whereas we believe that any number of PPs between 2 and 4

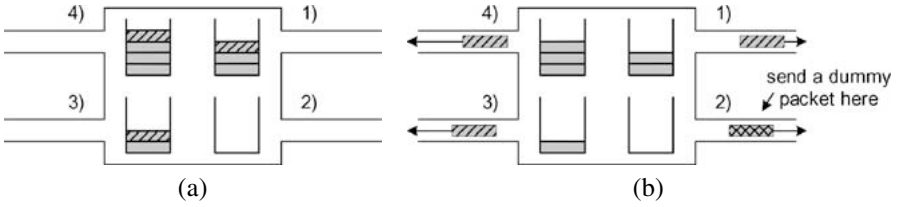

Figure 6. Mixing and the usage of dummy packets.

is reasonable. So purely from the padding overhead, we can say that a fixed-length in the range of 500–1000 bytes is best.

3.3. Mixing and dummy traffic

Another measure to make the PN stronger is by reordering the packets in a PP and using dummy traffic if needed. The goal of this is to make it very hard for an attacker to correlate incoming and outgoing packets at a PP based on the sequence the packets arrive at and leave the PP. Our basic approach makes use of the fact that at any moment, a PP is connected to some other PPs and gets packets from and sends packets to these other PPs. The idea is now that if the PP sends a packet to another PP, then it also sends one packet to each of its other connected peers. The order of the links on which the packets are sent out is always the same and therefore completely unrelated to the order of incoming packets. For an attacker, it is then not possible to find out on which link an incoming packet has been forwarded since one packet has been sent out on each link.

Assume that a PP is connected to n other PPs. A naive approach to achieve the effect described above could work as follows: when a packet arrives on a link, the PP generates $n - 1$ dummy-packets. Then it goes through all of the n links in a fixed order and sends out the real packet if it is the link the packet should be forwarded on, or a dummy packet if it is another link. Although this approach works, it requires $n - 1$ dummy packets for each real packet. If each PP were connected to 10 other PPs on average, 90% of the packets in the network would be dummies.

However, by modifying this approach a bit, it can be made much more efficient. Instead of forwarding an incoming packet right away, we buffer it for a short time in the PP. During the time it is buffered, many other packets arrive at the PP, which means that there is a good chance that at least one packet is waiting on many of the outgoing links. If one packet is now sent out on each link in a fixed order, then we have achieved the same as above: the attacker cannot correlate any incoming packet to one of the outgoing packets. Instead of using many dummy packets, a PP now uses the other real packets to hide a particular packet. On the outgoing links where no packet is waiting, we still have to send dummy packets.

Figure 6 illustrates how a PP operates. At each outgoing link, there is a queue. In Figure 6(a), there are 3, 0, 2 and 4 packets waiting on the four links to other PPs. When the PP decides to forward packets, it sends one packet out on each link, according to an internal numbering of the links (1–4). From each queue, the first packet is removed and

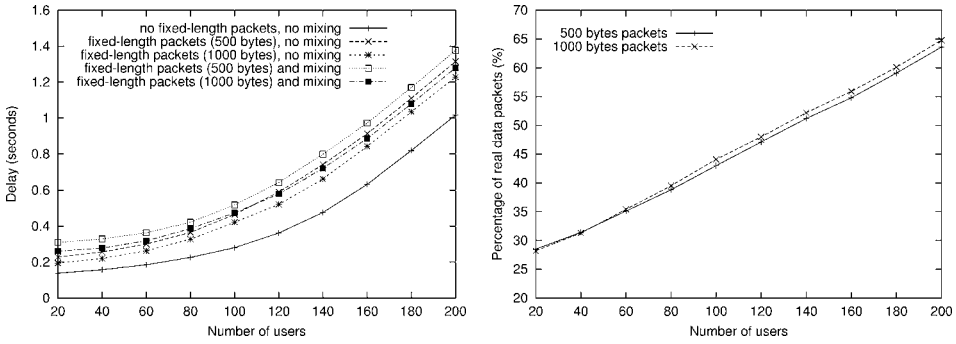

Figure 7. Overall performance of the pseudonymity network. (a) System performance. (b) Real traffic versus dummy traffic.

sent to the next PP (Figure 6(b)). Since there is no packet waiting in the queue on link 2, a dummy packet is sent out on that link. Note that since all packets are encrypted and have the same length, a dummy packet is indistinguishable from real packets for an external observer.

We can now analyse how well the PN performs. We use a simple test-bed with four PPs, which are fully interconnected. We simulate different numbers of users, whereas every user uses all four PPs in her anonymous tunnel. Each user makes an http-request, receives the corresponding http-reply, waits for an arbitrary time between 0 and 5 seconds, and issues the next request. The lengths of the http-replies follow again a Pareto distribution. We measure the time it takes from issuing an http-request until the document corresponding to the http-reply has been completely downloaded.

Figure 7(a) compares the download times for three fundamentally different cases: (1) only the layers of encryptions (Figure 3) are used, (2) fixed-length packets are added, and (3) mixing is added. For the two latter cases, message lengths of 500 and 1000 bytes are compared. Figure 7(a) shows clearly that the download times when using the PN are acceptable for a user. However, more interesting than the absolute figures is the conclusion that the addition of fixed-length messages and mixing does not harm the download times significantly. Using fixed-length packets of 1000 bytes, Figure 7(a) shows that the maximum absolute overhead when using fixed-length packets and mixing is about 0.27 seconds. The relative overhead is shrinking from about 90% for few users to about 26% for many users. The reason for this is the way a PP operates when mixing is used: it buffers a packet for a short while, hoping that other packets arrive such that fewer dummy packets have to be used. When there are only few users, the probability that many other packets arrive during the time a packet can be buffered is quite small, so the packet is usually buffered for a relatively long time (up to the maximum time a packet is allowed to be buffered in a PP). With many users, the buffers are filled rather quickly, which means that packets can be forwarded earlier.

Figure 7(b) shows that the more users there are using the PN, the fewer dummy packets are needed to protect their anonymity. The percentage of real packets in the network climbs

from below 30% to 65% as the number of users grows from 20 to 200. This is reasonable, since with many users, the probability that real packets are waiting at many of the outgoing links of a PP is higher, which means that fewer dummy packets are needed to mix the data. Although 70% of dummy packets when there are only few users looks like a lot of overhead, it is perfectly justifiable, since (1) the few data packets generated by these few users are not enough to confuse an attacker and to provide strong anonymity, which means that a lot of cover traffic has to be generated and (2) since the PPs do not have to handle many real data packets, they have enough computing power to process the dummy packets. Summarising, the amount of dummy traffic in the PN is adjusted by the number of real data packets.

Note that Figure 7(a) also shows that a fixed packet length of 1000 bytes gives better end-to-end performance than 500 bytes. The reason for this is that the PPs can handle fewer long packets better than many short packets. We therefore choose 1000 bytes as the length for the packets in the PN.

3.4. Prototype implementation

Our prototype is implemented in Java 1.3 and it currently provides support for http and https. Using the Just-In-Time (JIT) compiler option, the system provides adequate performance for experiments.

3.5. Related work

Anonymising web traffic has been tried before. We briefly present an overview of the best-known approaches and compare them with our system.

A system to anonymise ISDN-telephony [Pfitzmann et al., 16] via a MIX cascade provides anonymity among the subscribers connected to the same end-office. The approach makes heavily use of the synchronised telephony system in the sense that all subscribers are always sending data to the end-office so that real phone calls cannot be distinguished from the dummy data. An eavesdropper can trace a phone call only to the end-office, but he cannot tell which of the subscribers connected to this end-office is making the call. Although the idea can be realised very efficiently in the telephony world, it is questionable how well it is suited for the highly asynchronous Internet.

Onion Routing [Reed et al., 19] is a MIX network to enable anonymous use of near real-time Internet services. The system employs uniform message length and layered encryption of messages to complicate traffic analysis. This system was the first to deliver results on anonymising delay-sensitive Internet traffic via a MIX network, but it has been discontinued in January 2000. Although one result of Onion Routing was that Mix-networks are vulnerable to sophisticated traffic analysis attacks if no dummy traffic is used, its developers did not come up with a solution to tackle this problem. Since we have found an efficient way of operating the PPs using dummy traffic, we believe that our system gives much better protection from these attacks without significantly increasing the overhead. A second generation system of Onion Routing, which should offer better protection than

the first generation prototype, has been underway as of June 2000 [Syverson et al., 27], but no detailed information about how this goal should be achieved has been made public so far. It should be noted that the U.S. government was awarded a patent for Onion Routing on July 24th, 2001. It is uncertain what impact this patent has.

The Freedom System [Boucher et al., 4] was a commercial service provided by Zero-knowledge Systems to enable users to browse the Web anonymously. Its approach is very similar to that of Onion Routing, and the mixing components are called *Anonymous Internet Proxies* (AIPs). In the first version of the system, a route employed always three AIPs, all packets in the system had the same length, and dummy traffic was used to further complicate traffic analysis. In the second version of the system, the number of AIPs in a route was reduced to two and dummy traffic and fixed-length packets were eliminated. The system designers' argument is that the increased resistance against traffic analysis is not worth the bandwidth overhead. In addition, the AIPs do not really mix the traffic but forward the packets in a FIFO manner. As a result, Freedom is not very resistant against traffic analysis attacks [Back et al., 1]. Our system employs dummy traffic and the PPs mix the traffic and do not simply forward them in the same order as the packets arrive, hence we strongly believe that our system offers much better protection from attacks than Freedom. In addition, as we have shown in Sections 3.2 and 3.3, good protection from traffic analysis attacks does not imply significant bandwidth overhead. The Freedom network has been shutdown as of October 22nd, 2001, probably due to economical reasons [Zeroknowledge Systems, 28].

Web MIXes is another project [Berthold et al., 2, 3] that aims at providing anonymous access to near real-time services in the Internet. It assumes a very strong attack model and uses a MIX cascade to complicate traffic analysis. A prototype of this system is up and running[1] but does not yet provide the kind of resistance against attacks it is supposed to. Our trials of the Web MIXes system have shown an acceptable performance for web browsing, but at this time, we cannot judge how well it will perform under heavy load and how much the performance is going to be affected by the increase in protection from attacks.

Crowds [Reiter and Rubin, 20] makes use of a different approach to provide anonymous web browsing. It works by grouping web users in a crowd and this crowd performs web transactions on behalf of its members. When a user requests an URL, this request is forwarded randomly to another member in the crowd. Whenever a crowd member receives a request from another member, it makes a random choice to either forward the request to another crowd member (chosen randomly) or submit this request to the end server to which the request is destined. The reply from the server uses the same way back. If the crowd is large enough, then neither the other members nor the server nor any eavesdropper outside the crowd can tell which member in the crowd initiated the request. The system provides anonymity in the sense that any crowd member could have requested the web page. The overhead that Crowds introduces is relatively low compared to our system or those described above, as it does not make use of measures such as layered encryption and dummy traffic. On the other hand, the protection from attacks it offers is also significantly lower than that of our system. Crowds is vulnerable to internal eavesdroppers (who monitor

the traffic from within the crowd) and collaborating members (who exchange information trying to find the source of requests).

3.6. Conclusions and further work

We have designed and implemented a prototype of a service that allows browsing the Web and accessing e-shops pseudonymously while offering strong resistance against traffic analysis. We have shown that the addition of fixed-length packets and mixing does not introduce too much overhead and gives the user still very acceptable end-to-end performance. In contrast to the Freedom System's designers, we believe that this additional overhead is well worth the immense increase of the level of anonymity.

More research has to be done before such a service could be widely deployed. The round-robin method to send packets out on the links is not optimal yet, since this blocks a PP temporarily if a TCP send buffer is full. One approach to avoid this is to skip writing on a link if the underlying TCP send buffer does not accept data, and try again during the next round. However, it has also to be examined how much this approach would affect the degree of anonymity the PN offers. Another open issue is the protection from end-to-end timing attacks [Felten and Schneider, 12]. These attacks do not observe the PN at the PPs, but at the edges to correlate certain events. For example, a packet leaving an LP towards the PN and a packet exiting the PN somewhere else shortly after could be related. In its current state, the PN is vulnerable to such attacks, although such an attack could probably only be carried out by a very powerful adversary.

The PN also solves the problem that e-mail communication between Alice and the e-shop can give away information about Alice's identity. Alice simply uses the PN to connect to an http-based e-mail service such as Hotmail and Yahoo that do not require a user to register using personal data and opens an account such as `happy_customer@hotmail.com`. When she goes shopping at an e-commerce store later, she opens an account at the e-shop, using `happy_customer@hotmail.com` as her user-name. The e-shop can use this address to send e-mail messages to Alice, and the messages are stored on Alice's e-mail account at Hotmail. When Alice wants to send or read e-mail messages, she connects to Hotmail through the PN. Neither Hotmail nor the e-shop can find out who Alice really is. All the e-shop knows is that it exchanges e-mail messages with Hotmail and Hotmail cannot learn anything since Alice connects to it via the PN.

4. The pseudonymous secure electronic transaction protocol

Credit cards have been the dominant payment method in the Internet during the past years and this is not likely to change soon. Credit cards are widely accepted among customers and merchants, and its procedures are well understood. All attempts to replace credit cards in the Internet have failed so far, despite the superiority of some of the alternative proposals. The most prominent among them is maybe "untraceable electronic cash" [Chaum et al., 5], which allows for secure and anonymous e-cash payments. It never managed to be widely

accepted in the electronic marketplace, although credit card payments are much less secure and not anonymous at all.

In its pure form, credit card payments are very vulnerable to fraud. All an attacker usually needs is a valid credit card number, the associated name of the cardholder, and the expiry date, and he can purchase goods over the Internet on the cardholder's charge. The major credit card companies recognised this problem and joined with others in the industry to develop the *Secure Electronic Transaction* (SET) protocol [SET, 25]. SET combines credit card payments with strong public-key cryptography and certificates and thwarts attempts to credit card fraud. Today, it is not clear if SET will be widely used in the Internet in the near future. However, it is reasonable to assume that the future growth of e-commerce calls for secure payment methods. If credit cards should be one of them, then a standard such as SET or a variant of SET has to be widely deployed. It should be noted that while SET serves well to make e-commerce more secure, it does not much to hide the identity of the customer. All parties involved in a SET transaction learn the identities of both the customer and the e-shop. That means that we cannot simply use SET as the payment protocol for pseudonymous e-commerce.

Two main criteria influenced the design of our payment system: first, credit cards will be most likely the dominant payment method in the Internet in the years to come. Second, the payment system should offer a high level of security. Since SET already fulfils both criteria, it was a natural decision to develop our payment system as an extension to SET. This resulted in the *Pseudonymous Secure Electronic Transaction* (PSET) protocol, which is a novel protocol that allows for secure and pseudonymous credit card-based payments over the Internet. In this section, we describe the PSET protocol based on a simplified version of SET, similar as in technical textbooks [Stallings, 26]. While this simplified version leaves out many of the details of SET, it contains all the important security-relevant features. For a more detailed discussion of PSET, consult the corresponding technical report [Rennhard et al., 21].

4.1. PSET basics

PSET introduces a new participant, the *Pseudonymous Credit Card Provider* (PCCP). The main task of a PCCP is to provide *Pseudonymous Credit Cards* (PCCs) that cannot be easily linked to the cardholder's real identity. Note that PCCs are not physical credit cards like the pieces of plastic used in real life. A PCC is rather a representation of the credit card data used when shopping online. Therefore, a PCC is the combination of a credit card number, an expiry date, and the name of the cardholder, except that this name has no connection to the cardholder's real identity. To solve this, a PCC can use it's own scheme to assign names to the unknown cardholders, for instance random strings composed of letters and numbers to represent first and last name.

A user that wishes to obtain a PCC simply contacts a PCCP and asks for a PCC and a certificate that can be used for signing in the PSET protocol. Note that the user does not have to give away her real identity to get a PCC. The PCCP issues the pseudonymous credit card and a certificate without knowing the real identity of the receiver. The PSET protocol

guarantees that it is not possible to misuse the PCC, although the issuer of the PCC does not know the cardholder's identity. Since the PCC has to be obtained anonymously, the cardholder should contact the PCCP only via a traffic-anonymising network (Section 3).

It is important to note that although the PCCP issues credit cards without knowing the identity of the cardholder, this is not the same as an anonymous bank account (e.g., a numbered Swiss bank account) where the account holder has the possibility to deposit and withdraw money whenever she likes. As anonymous bank accounts can be abused for various kinds of things, e.g., to hide or launder money, international pressure could force more and more countries to no longer support such anonymous accounts. However, pseudonymous credit cards do not give their owners the possibilities of anonymous bank accounts since there is no bank account associated with them. All an owner of a pseudonymous credit card can do is using it during payments in e-commerce to protect her privacy. Furthermore, we will see later in this section that a payment with a pseudonymous credit card is always related to a payment with the user's real credit card. As a consequence, a pseudonymous credit card cannot be used to move large amounts of money since the spending limit of a pseudonymous credit card is limited by the spending limit of the user's real credit card (e.g., 5000 Euros per month). But the strongest argument why pseudonymous credit cards are very unlike anonymous bank accounts is that pseudonymous credit cards provide *pseudonymity* rather than *anonymity,* which means that the identity of the owner of a pseudonymous credit card can always be unambiguously resolved if this is requested by a court order.

Since the PCCP issues pseudonymous credit cards without knowing the applicant, we have to make sure that a PCC can only be used when it is guaranteed that the PCCP eventually gets the money back from the user. The basic idea how this is achieved in PSET is that we assume that the owner of the PCC also has a normal credit card. When paying with the PCC, the cardholder not only includes payment information to charge her pseudonymous credit card, but also payment information to charge her real credit card. However, PSET must guarantee that none of the involved parties in the payment process is able to easily learn who the owner of the PCC is. The basic approach in PSET is that the PCCP pays the merchant and the issuer of the cardholder's real credit card pays the PCCP. Therefore, it is still the issuer of the cardholder's real credit card that pays the merchant in the end, but we use the PCCP in the middle to separate the knowledge to link the cardholder to her real credit card.

4.2. Participants in PSET

The following participants are involved in PSET:

– The *Issuer* is usually a financial institution and issues credit cards. In addition, it issues digital certificates that link a public key to a credit card such that only the legitimate owner of a credit card (who knows the corresponding private key) can use the credit card.

– The *Cardholder* is the holder of a credit card and a digital certificate, and a pseudonymous credit card with a digital certificate. This means that the cardholder has two credit cards with two certificates, but only one credit card is linked to her identity.
– The *Merchant* sells goods or services to the cardholder. A merchant that accepts credit cards must have a relationship with an acquirer.
– The *PCCP* issues pseudonymous credit cards. It also issues digital certificates that link a public key to a pseudonymous credit card such that only the legitimate owner of a PCC (who knows the corresponding private key) can use the credit card. Note that the PCCP plays the same role as the issuer in the sense that it issues credit cards. However, since the PCCP will never validate a payment with the PCC before the payment is validated with the cardholder's real credit card, there is no financial risk for the PCCP. This is a very important property of PSET. Like the merchant, the PCCP needs a relationship with an acquirer.
– The *Acquirer* is the financial institution that establishes an account with the merchant or a PCCP and processes credit card payment authorisations and the payments. In PSET, there are two acquirers: one is the acquirer of the merchant and the other is the acquirer of the PCCP.
– The *Payment Gateway* is located between the merchant or the PCCP and an acquirer. The payment gateway interfaces between PSET and the existing payment networks for authorisation and payment functions. This means that the merchant or the PCCP is usually talking to the acquirer's payment gateway and not to the acquirer directly. There are two payment gateways in PSET, one for each of the two acquirers.

Note that the PCCP plays two roles: One is as an issuer of credit cards when talking to the merchant's acquirer. The other is as a merchant when talking to the payment gateway of its own acquirer.

Figure 8 shows the participants and the messages exchanged between them. It also illustrates clearly why PSET is an extension of SET: in SET, the merchant's acquirer contacts the issuer directly, while in PSET, the payment validation goes via PCCP, the PCCP's payment gateway and the PCCP's acquirer. The extension results in two additional messages: the *Nested Authorization Request* (NestAuthReq) and the *Nested Authorization Response* (NestAuthRes).

4.3. Protocol phases of PSET

Like SET, PSET makes use of public-key cryptography. We use the following notation: $\mathrm{PrK}_{S,x}$, $\mathrm{PuK}_{S,x}$ and $\mathrm{Cert}_{S,x}$ are the corresponding private key, public key and digital certificate used for digital signatures of party x. Similar, $\mathrm{PrK}_{KE,x}$, $\mathrm{PuK}_{KE,x}$ and $\mathrm{Cert}_{KE,x}$ are the corresponding private key, public key and digital certificate used for key exchange.

According to Figure 8, we will now explain the messages exchanged in the PSET protocol. We will always point out where exactly SET is extended.

4.3.1. Initiate Request (PInitReq) and Initiate Response (PInitRes) The cardholder sends a PInitReq message to the merchant. The payload of the message contains the brand

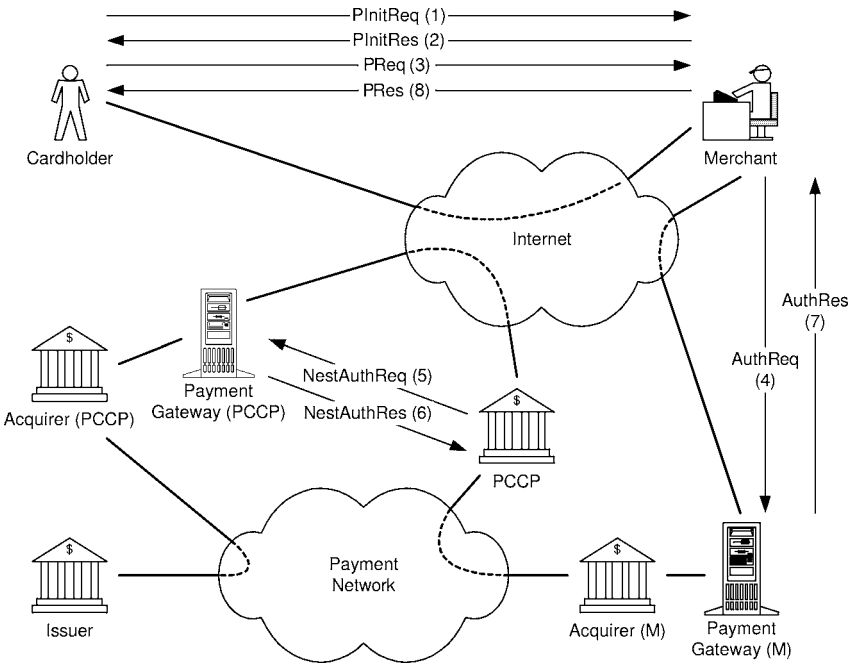

Figure 8. Participants in PSET.

of the credit card the cardholder wants to use. Since a pseudonymous credit card is used to pay the merchant, the payload specifies the brand of the pseudonymous credit card. The merchant gets the message and checks if he accepts the credit card brand. The merchant then generates a unique *Transaction ID* (TransID) to identify the transaction. The merchant signs the TransID and includes his certificate for signing ($\text{Certs}_{S,M}$) and his payment gateway's key-exchange certificate ($\text{Cert}_{KE,PG\text{-}M}$).

4.3.2. Purchase Request (PReq) During this phase, the cardholder prepares the necessary information for the merchant, the merchant's payment gateway, the PCCP and the PCCP's payment gateway.

Figure 9 shows the initial operations the cardholder performs. The *Order Information Data* (OIData) contains information about the order. OIData contains the TransID and the *Hashed Order Description* (HOD). The HOD is a hash of the actual description of the order. Although the format of this description is not part of SET or PSET, it is usually a listing of the ordered goods and their prices and is sent form the merchant to the cardholder before the payment starts. The cardholder also computes a hash HOIData of OIData.

The cardholder builds two *Payment Information Data* (PIData) blocks. The first ($\text{PIPData}_{PG\text{-}M}$) is prepared for the merchant's payment gateway and contains an identifier of the merchant (MerchantID) and information about the pseudonymous credit card (PCC). The other ($\text{PIData}_{PG\text{-}PCCP}$) is generated for the PCCP's payment gateway and contains an identifier of the PCCP (PCCPID) and information about the real credit card (CC). In addi-

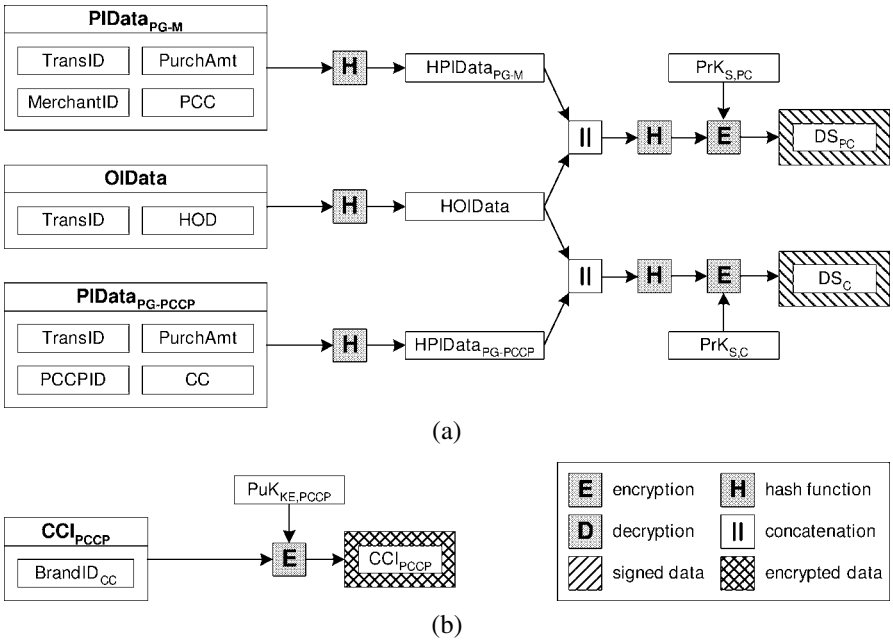

Figure 9. Computations performed by the cardholder.

tion, both blocks contain the TransID and the amount of the purchase (PurchAmt). As with OIData, the cardholder computes the hashes of the two PIData and gets HPIData$_{PG\text{-}M}$ and HPIData$_{PG\text{-}PCCP}$. In contrast to PSET, there is only one PIData (PIData$_{PG\text{-}M}$) needed in SET, which contains the id of the merchant and the real credit card information.

A central concept to assure security in SET is the *Dual Signature* (DS). A dual signature is the same as a normal digital signature, but it signs two data blocks together and therefore links them unambiguously. The main reason for the dual signature in SET is that it links an order to a payment, such that neither the cardholder nor the merchant can falsely claim that a payment was intended for another purchase. In SET, there is one dual signature while in PSET, we need two dual signatures since there are two payments. The first dual signature (DS$_{PC}$, Figure 9(a)) is generated by signing the concatenation of HOIData and HPIData$_{PG\text{-}M}$, using the private key PrK$_{S,PC}$ that corresponds to the pseudonymous credit card (owned by the 'pseudonymous' cardholder (PC)). The second dual signature (DS$_C$, Figure 9(a)) is made over the concatenation of HOIData and HPIData$_{PG\text{-}PCCP}$, using the private key PrK$_{S,C}$ that corresponds to the real credit card (owned by the 'real' cardholder (C)).

Since the PCCP has to know the brand of the cardholder's real credit card to forward the payment information to the correct payment gateway (which handles that particular brand of credit card), the cardholder builds the *Credit Card Information* (CCI$_{PCCP}$, Figure 9(b)). It contains an identifier of the cardholder's real credit card and it is encrypted for the PCCP. Note that whenever data has to be encrypted in SET and PSET, a one-time symmetric key

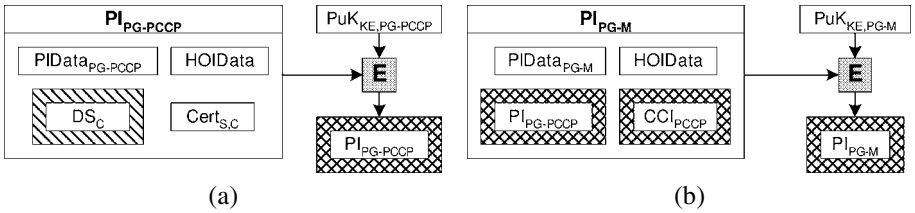

Figure 10. The payment information for the PCCP and the merchant.

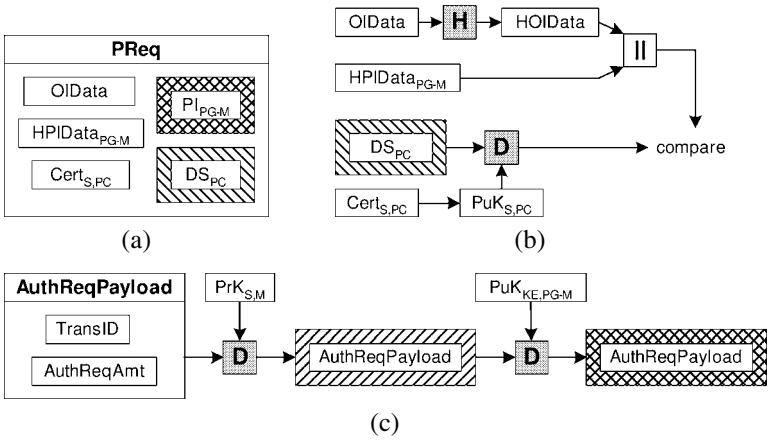

Figure 11. The PReq message and its processing by the merchant.

is generated. This key is used to encrypt the data and the symmetric key itself is encrypted using the recipient's key-exchange public key.

Figure 10 shows the *Payment Information* (PI) blocks the cardholder builds for the PCCP and the merchant. The payment information for the PCCP's payment gateway ($PI_{PG\text{-}PCCP}$, Figure 10(a)) contains $PIData_{PG\text{-}PCCP}$, HOIData, DS_C and $Cert_{S,C}$ and is encrypted for the PCCP's payment gateway. Similarly $PI_{PG\text{-}M}$ (Figure 10(b)) is encrypted for the merchant's payment gateway and contains $PIData_{PG\text{-}M}$, HOIData, the encrypted $PIData_{PG\text{-}PCCP}$ and the encrypted CCI_{PCCP}.

The cardholder now generates the PReq message (Figure 11(a)) and sends it to the merchant. PReq contains OIData, $HPIData_{PG\text{-}M}$, the encrypted $PI_{PG\text{-}M}$, DS_{PC} and $Cert_{S,PC}$.

On receipt of the PReq message, the merchant checks if $Cert_{S,PC}$ is valid and applies $PuK_{S,PC}$ on the dual signature DS_{PC} (Figure 11(b)). If the result is the same as a self-computed value Hash(Hash(OIData), $HPIData_{PG\text{-}M}$) (using OIData and $HPIData_{PG\text{-}M}$ from PReq), then the signature was made by the owner of the pseudonymous credit card and OIData has not been changed in transit. The merchant also checks if TransID (in OIData) corresponds to a previously requested transaction id. If this test is OK, then the merchant knows that this payment corresponds to a previously initiated payment. In a final test, the merchant compares HOD (in OIData) with a self-generated version based on the locally stored information about the cardholder's order. If this test is OK, then the mer-

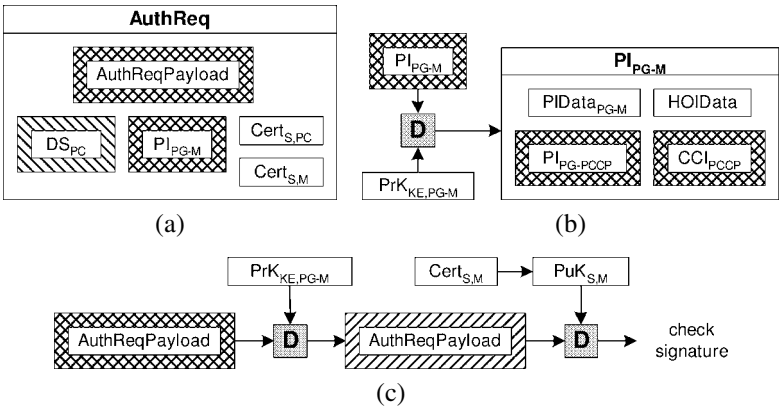

Figure 12. The AuthReq message and its processing by the merchant's payment gateway.

chant knows that the cardholder is indeed paying for the order they agreed on earlier, and is paying the correct amount.

4.3.3. Authorization Request (AuthReq) The merchant now asks his payment gateway to validate the payment. He forwards the payment information received from the cardholder, and includes an *Authorization Request Payload* (AuthReqPayload). Figure 11(c) illustrates how AuthReqPayload is built: it contains the TransID and the amount requested (AuthReqAmt). The AuthReqPayload is signed by the merchant and encrypted for the merchant's payment gateway.

The merchant builds the *Authorization Request* (AuthReq) message using the encrypted AuthReqPayload, the encrypted $PI_{PG\text{-}M}$, DS_{PC}, $Cert_{S,M}$ and $Cert_{S,PC}$ (Figure 12(a)) and sends it to his payment gateway.

On receipt of AuthReq, the payment gateway first checks the certificates for their validity, decrypts $PI_{PG\text{-}M}$ (Figure 12(b)), decrypts AuthReqPayload, and checks the merchant's signature (Figure 12(c)). It compares if TransID in AuthReqPayload corresponds to TransID in $PIData_{PG\text{-}M}$ to check if the cardholder is indeed paying for the transaction the merchant requests. The payment gateway also makes sure that AuthReqAmt in AuthReqPayload is the same as PurchAmt in $PIData_{PG\text{-}M}$, which assures that the cardholder pays enough and that the merchant does not charge too much. Like the merchant, it decrypts the dual signature DS_{PC} and compares the result with the self-computed value Hash(HOIData, Hash($PIData_{PG\text{-}M}$)). If the two values are equal, then the signature was made by the owner of the pseudonymous credit card and $PIData_{PG\text{-}M}$ has not been changed in transit. Then it checks if the MerchantID in $PIData_{PG\text{-}M}$ is the same as the MerchantID in the merchant's certificate for signing. This guarantees that the merchant cannot use payment information intended for another merchant.

The merchant's payment gateway then uses PCC and PurchAmt from $PI_{PG\text{-}M}$ to authorise payment through existing payment card financial networks. Note that this means an authorisation of the payment at the issuer of the cardholder's credit card. Since the issuer of the pseudonymous credit card is the PCCP, the authorisation goes to the PCCP. The dif-

ference to normal credit card payment authorisations is that the authorisation now not only includes the credit card information and the amount, but also $PI_{PG\text{-}PCCP}$, which the merchant's payment gateway received from the cardholder (via the merchant), the TransID, and the credit card information CCI_{PCCP} for the PCCP.

Here we have to extend the original SET protocol quite a bit. In SET, the issuer would now check if the cardholder is indeed credit-worthy by looking at internal data about the cardholder. However, the PCCP cannot answer this, since we defined earlier that the PCCP never directly authorises credit card payments. Therefore, the PCCP verifies the real credit of the cardholder via the PCCP's payment gateway and—if the cardholder is creditworthy—only then authorises the payment back to the merchant's payment gateway. From this point of view, the PCCP acts similar as the merchant: it receives payment information and relays it to its payment gateway for further processing. Before the PCCP can do this, it extracts the brand of the cardholder's real credit card from CCI_{PCCP}. This tells the PCCP which payment gateway to contact.

4.3.4. Nested Authorization Request (NestAuthReq) The PCCP sends a *Nested Authorization Request* (NestAuthReq) message to its payment gateway. NestAuthReq looks similar to AuthReq (Figure 11(a)), but does neither contain DS_{SP} nor $Cert_{S,PC}$, since we do not want the PCCP's payment gateway to learn anything about the PCC. NestAuthReq contains an encrypted NestAuthReqPayload, which is built in the same way as AuthReqPayload (Figure 11(c)).

On receipt of Nest AuthReq, the PCCP's payment gateway performs the same checks as the merchant's payment gateway above: it decrypts the encrypted data, verifies the authenticity of PCCP's signature on NestAuthReqPayload, checks if TransID in NestAuthReqPayload is the same as TransID in $PIData_{PG\text{-}PCCP}$, and that the requested amounts in NestAuthReqPayload and $PIData_{PG\text{-}PCCP}$ are the same. The payment gateway checks the validity of $Cert_{S,C}$ and the dual signature DS_C (both in $PI_{PG\text{-}PCCP}$, in the same way and for the same reasons as the merchant's payment gateway did as described above. Finally, it checks if the PCCPID in $PIData_{PG\text{-}PCCP}$ is the same as the PCCPID in the PCCP's certificate for signing. This guarantees that the PCCP cannot use payment information intended for another PCCP.

The PCCP's payment gateway can now use CC and PurchAmt from $PIData_{PG\text{-}PCCP}$ to authorise payment through existing payment card financial networks. This is where the payment with the cardholder's real credit card is authorised at the issuer of that credit card. Eventually, the payment gateway gets the response from the issuer if the payment was approved or not. The authorisation is sent from the PCCP's payment gateway to the PCCP in a *Nested Authorization Response* (NestAuthRes) message. This message is the same as the *Authorization Response* (AuthRes) message in SET. Since the PCCP can now be sure that it will get the money of the payment with the pseudonymous credit card, it can answer to the authorisation request of the merchant's payment gateway (via the merchant's acquirer) with an AuthRes message. Finally, the merchant sends a *Purchase Response* (PRes) message to the cardholder and the PSET payment is complete.

4.4. Analysis of PSET

In this section, we analyse briefly the security and pseudonymity aspects of PSET. For a more detailed discussion, refer to the technical report [Rennhard et al., 21].

PSET consists of two SET payments. Since SET is widely believed to be secure and since the first payment works exactly like SET, there is no way for the cardholder to act dishonestly (for instance by using a listing of more expensive goods than she actually wants to pay for in the OIData). Therefore, it remains to be shown that (1) the identity of the cardholder is indeed protected and (2) the cardholder can unambiguously link his real and pseudonymous identities if she has been tricked by any of the involved entities.

Looking at the protocol, then we can see that none of the involved parties can learn both the identity of the cardholder and the e-shop. The merchant, the merchant's payment gateway, and the PCCP learn the identity of the e-shop, whereas the PCCP's payment gateway and the issuer learn the identity of the cardholder. The anonymity of the cardholder can be broken if at least one party of each of these two sets collude and share their knowledge.

The dual signatures guarantee that the two payments can be linked, because the cardholder uses the same OIData as input for both dual signatures. This means that the cardholder indeed made both payments for the same order, which links the payments unambiguously to that particular order. Since it is not possible for an attacker to find another, meaningful OIData that hashes to the same HOIData [Schneier, 23], this also means that the two payments are unambiguously linked.

Note that if the cardholder uses a hash of another OIData than the real one as input in the dual signature for the PCCP's payment gateway, then this will never be noted by any party, if no dispute arises. This is not an issue, since linking the two dual signatures based on the same input HOIData is only required in the case of a dispute (as described above). It is the cardholder's own fault if she uses different hashes as inputs to the dual signatures, since she cannot prove the linkage of the two payments anymore if a dispute arises (for instance if the merchant refuses the service the cardholder paid for).

The cardholder gets a PCC completely anonymously at a PCCP. Since this procedure is simple, it could even be used for one-time usage of a PCC. One-time PCCs are interesting, since a potential problem with PSET is that when a user has been deceived by a merchant, she is usually forced to reveal her identity if she wants to expose the merchant. This relates all former payments made with the same PCC to that customer. A merchant could actually use this attack to find out the real identities of his customers: simply refuse delivery of goods and wait until the customer complains. Although this attack will greatly harm a merchants reputation over time, it has to be assumed that it could be carried out. A countermeasure is to change the pseudonymous credit card frequently. An extreme measure would be to throw away a PCC after each payment the card was involved. This would still allow the attack by the merchant as described above, but would not link previous payments to the customer.

4.5. Related work

As mentioned previously, the best-known anonymous payment method is "untraceable electronic cash" [Chaum et al., 5]. Alice can go to a bank, withdraw some digital coins from her account, and spend the money at an e-shop. Later, when the merchant takes the money to the bank, the bank checks if it has issued the coin earlier, and if the coin has not been spent before. The trick is that the bank cannot link the coin it receives from the merchant to Alice, since the coin was blinded by Alice when the bank issued it, using a technique called blind digital signatures. Untraceable electronic cash never was widely accepted, probably because it is too different from the well-known credit card payments. Another problem is that it is maybe too anonymous. The biggest problem for kidnappers today is receiving the ransom. Untraceable digital cash would make this much simpler [Schneier, 23]: Bob kidnaps Alice's baby. Instead of requesting real money, he prepares blinded coins, sends them to Alice, and tells her to have the coins signed by a bank and publish the results in a newspaper, otherwise he would do harm to the baby. The next day, Bob just buys a newspaper and unblinds the coins. He checks if the coins were correctly signed by the bank and releases the baby. Since nobody except Bob can unblind the money, there is no way to recognise the money as Bob spends it later.

Anonymous Credit Cards (ACC) [Low et al., 14] makes use of a complex protocol to provide anonymous payments for a customer. The idea is that a customer has two accounts in different banks. The first bank knows the person's identity whereas the second does not (e.g., a numbered Swiss bank account). Since the first bank knows the person, it is willing to grant her credit. The second bank is not willing to grant the person credit, as it does not know her. However, on the person's request, the first bank agrees to put credit into the anonymous account at the second bank. When the person pays, she uses a credit card for her account at the second bank. The bank checks the person's credit and—if she is creditworthy—authorises the payment. Eventually, the second bank sends a bill to the first bank, which sends a bill to the person. The protocol has many similarities to ours in the sense that it uses intermediate instances to separate the information such that no party knows the identities of both customer and merchant. One main difference is that in the ACC protocol, the authorisation of the credit card payment is done by the second bank (since the first bank has put credit into the anonymous user's account) whilst in PSET, the authorisation is done by the issuer of the real credit card. This implies that ACC is not well suited for a scenario where a user has several pseudonymous credit cards issued by different institutions or where she changes her pseudonymous credit card frequently, as described in Section 4.4. The explanation for this is that in the ACC protocol, the user must explicitly tell the first bank to put credit into her anonymous accounts. As a bank is probably not willing to grant a user unlimited credit, the user must know in advance the approximate amount she plans to spend with each of her anonymous credit cards and carefully share the credit among her anonymous accounts. In PSET, no such assignment is needed in advance and credit is only assigned during the authorisation phase of a payment process. Consequently, pseudonymous credit cards can be requested and used right away without informing the issuer of the real credit card.

Recently, a new credit card payment service named SafeDoor [Securicor, 24] has been launched. Customers register with SafeDoor, giving it their credit card numbers. Then, when it comes to online payment, the retailer charges SafeDoor's account that on its turn charges the customer credit card. Thus, the retailer sees SafeDoor as the customer and the credit card issuer sees SafeDoor as the retailer. The drawback of SafeDoor is that the customer must rely entirely on SafeDoor to keep her anonymity (not mentioning the safekeeping of the credit card number).

4.6. Conclusions

With PSET, we have designed a novel protocol to enable secure, pseudonymous credit card payments over the Internet. PSET is an extension of SET and the protocol guarantees that the customer remains anonymous in the normal course of events, but also makes sure that (1) the customer cannot abuse her pseudonymous credit card to act dishonestly and that (2) the customer herself can proof that it was her who made the pseudonymous payment if she thinks she has been tricked by another entity involved in the payment process.

We have chosen SET as the basic protocol for our pseudonymous payment protocol, although it is not clear today if SET will be ever widely deployed. However, with similar methods as in PSET—separation of knowledge and inclusion of information that unambiguously links the two payments — it should be possible to extend any SET-like payment protocol such that it allows pseudonymous payments.

Using a payment protocol such as PSET gives the customer also the possibility to control the identity management the e-shop is able to do. If Alice wants to be advertised via e-mail messages and get personalised service at the e-shop, she simply uses the same anonymous e-mail account (Section 3.6) all the time, and the e-shop recognises that all purchases are made by Alice without knowing her identity. Conversely, if Alice does not want that her purchases can be linked, she can avoid this by presenting the e-shop a new anonymous e-mail address and a fresh pseudonymous credit card each time she makes a payment at the e-shop. Since the procedure to obtain a PCC is very simple (Section 4.4), this one-time usage of a PCC is acceptable for both customer and PCCP.

5. Limitations

Although the PN provides reliable pseudonymous web browsing, where no single entity has the total knowledge of the end-points, it suffers from some limitations. Java applets and JavaScripts pose as a serious problem when a secure channel between customer and merchant is used. Whilst a customer is browsing a web site using an open channel (not encrypted), the local proxy can remove http-headers and Java references from the html-files. However, when a secure channel is established between customer and e-shop (usually during checkout) the local proxy can no longer filter it. It means that some information about the customer can leak to the server. For further security, the customer should not enable Java applets and JavaScripts at least during checkout time.

Recently, Yahoo was legally required to prevent French citizens from accessing auctions of Nazi material. It seems that French users could easily infringe their law by using the PN to access those auctions. Yahoo would not be able to identify French users using the PN. Not even the PP providers would be able to thwart this situation. Ultimately, Yahoo, or other concerned sites would have to block completely access from customers using such anonymising systems.

Finally, e-commerce based on our components is initially limited to electronic goods, such as printable books or digital libraries access. The reason for that is that for material goods, a delivery address is required and a system would have to be devised for assuring customer anonymity. Although the customer could have access to an anonymous mailbox where the purchases could be delivered to, some merchants may not be willing to accept it and require a real address. We can visualise a solution using a pseudonymous delivery system employing a chain of pseudonymity delivery companies, but the price burden imposed by such solution would make it prohibitively expensive for customers.

6. Conclusions

Based on the use of two novel pseudonymity components, we have shown a solution for providing an e-commerce experience emulating the anonymity that can be achieved in traditional shops.

The properties we have stated about pseudonymous e-commerce in Section 2 are fulfilled by our pseudonymity components. Property 1 can be guaranteed by accessing the e-shop only via the PN and using an anonymous http-based e-mail account, as is offered by Hotmail or Yahoo. This e-mail account should also only be accessed via the PN. Properties 2, 3 and 4 are fulfilled by the pseudonymous secure electronic transaction protocol.

Although today's e-commerce procedures are not advanced enough to enable pseudonymous e-commerce based on our ideas, we can envision a future where our components can be integrated nicely. A realistic scenario for the future is that secure e-commerce will be based on a secure credit card payment protocol such as SET or a variant of SET. This payment protocol could be extended such that it provides pseudonymous payments as an option with only small changes to the original protocol, in a similar way as we have extended SET. The integration of the pseudonymity network to access the e-shop and the anonymous http-based e-mail account is much simpler, since they do not require an e-shop to change its procedures at all.

Besides security concerns, the lack of privacy is one of the main reasons that limits trust in e-commerce by its potential customers. Secure pseudonymous e-commerce based on components such as the ones we presented in this paper could overcome these problems and greatly enhance trust in e-commerce.

Acknowledgments

The work presented here was done within ShopAware—a research project funded by the European Union in the Framework V 1ST Programme (project 12361). Marc Rennhard

would like to thank also the Swiss Federal Office for Education and Science for his sponsorship.

Note

1. http://anon.inf.tu-dresden.de/index.html

References

[1] Back, A., I. Goldberg, and A. Shostack. (2001). "Freedom 2.1 Security Issues and Analysis." White Paper, `http://www.freedom.net/info/whitepapers/Freedom_Security2-1.pdf`.

[2] Berthold, O., H. Federrath, and M. Köhntopp. (2000a). "Project 'Anonymity and Unobservability in the Internet.' " In *Proceedings of the Workshop on Freedom and Privacy by Design, Conference on Freedom and Privacy 2000 CFP.* Toronto, Canada, pp. 57–65.

[3] Berthold, O., H. Federrath, and S. Köpsell. (2000b). "Web MIXes: A System for Anonymous and Unobservable Internet Access." In *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability.* Berkeley, CA, USA, pp. 101–115.

[4] Boucher, P., A. Shostack, and I. Goldberg. (2000). "Freedom Systems 2.0 Architecture." White Paper, `http://www.freedom.net/info/whitepapers/index.html`.

[5] Chaum, D., A. Fiat, and M. Naor. (1989). "Untraceable Electronic Cash." In *Advances in Cryprology—CRYPTO '88*, Lecture Notes in Computer Science, Vol. 403, Springer-Verlag, pp. 319–327.

[6] Chaum, D. L. (1981). "Untraceable Electronic Mail, Return Adresses, and Digital Pseudonyms." *Communications of the ACM* 24(2).

[7] Clarke, R. (1999). "Identified, Anonymous and Pseudonymous Transactions: The Spectrum of Choice." In *Proc. User Identification & Privacy Protection Conference*, Stockholm, Sweden.

[8] Cottrell, L. (1996). "The Anonymizer." `http://www.anonymizer.com`.

[9] Dierks, T. and C. Allen. (1999). "The TLS Protocol Version 1.0." RFC 2246.

[10] Diffie, W. and M. E. Hellman. (1976). "New Directions in Cryptography." *IEEE Transactions on Information Theory* IT-22(6), 644–654.

[11] Feldmann, A., A. C. Gilbert, P. Huang, and W. Willinger. (1999). "Dynamics of IP Traffic: A Study of the Role of Variability and the Impact of Control." In *Proceedings of SIGCOMM '99*. Massachusetts, USA.

[12] Felten, E. and M. Schneider. (2000). "Timing Attacks on Web Privacy." In S. Jajodia and P. Samarati (eds.), *7th ACM Conference in Computer and Communication Security 2000*, pp. 25–32.

[13] Housely, R. and W. Polk. (1999). "Internet X.509 Public Key Infrastructure." RFC2528.

[14] Low, S., N. Maxemchuk, and S. Paul. (1994). "Anonymous Credit Cards". In *Proceedings of the 2nd Annual ACM Conference on Computer and Communications Security*, pp. 108–117.

[15] Pfitzmann, A. and M. Köhntopp. (2001). "Anonymity, Unobservability, and Pseudonymity—A Proposal for Terminology; Draft vO.12." `http://www.koehntopp.de/marit/pub/anon/Anon_Terminology.pdf`.

[16] Pfitzmann, A., B. Pfitzmann, and M. Waidner. (1991). "ISDN-MIXes: Untraceable Communication with Very Small Bandwith Overhead." In *Kommunikation in verteilten Systemen 267*, 451–463.

[17] Pfitzmann, B., M. Waidner, and A. Pfitzmann. (1990). "Rechtssicherheit trotz Anonymität in offenen digitalen Systemen." *Datenschutz und Datensicherung* 14, 243–253, 305–315. (In German.)

[18] Reagle, J. and L. F. Cranor. (1999). "The Platform for Privacy Preferences." *Communications of the ACM* 42(2).

[19] Reed, M. G., P. F. Syverson, and D. M. Goldschlag. (1998). "Anonymous Connections and Onion Routing." *Journal on Selected Areas in Communications* 16(4).

[20] Reiter, M. K. and A. D. Rubin. (2000). "Crowds: Anonymity for Web Transactions." In *ACM TISSEC*.

[21] Rennhard, M., S. Rafaeli, and L. Mathy. (2001a). "From SET to PSET—The Pseudonymous Secure Electronic Transaction Protocol." TIK Technical Report Nr. 117, TIK, ETH Zurich, Zurich, CH.

[22] Rennhard, M., S. Rafaeli, L. Mathy, B. Plattner, and D. Hutchison. (2001b). "An Architecture for an Anonymity Network." In *Proceedings of the IEEE 10th Intl. Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises* (*WET ICE 2001*). Boston, USA, pp. 165–170.
[23] Schneier, B. (1996). *Applied Cryptography*. New York: Wiley, 2nd Edition.
[24] Securicor. (2000). "SafeDoor." `http://www.SafeDoor.co.uk/`.
[25] SET. (1997). "Secure Electronic Transaction Specification—Books 1-3." `http://www.setco.org/download.html`.
[26] Stallings, W. (2000). *Network Security Essentials.* Prentice Hall, 1st Edition.
[27] Syverson, P., G. Tsudik, M. Reed, and C. Landwehr. (2000). "Towards an Analysis of Onion Routing Security." In *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability.* Berkeley, CA, USA, pp. 83–100.
[28] Zeroknowledge Systems. (2001). "Shutdown of Freedom Network." `http://www.freedom.net/prem.html`.
[29] Zimmermann, P. R. (1995). *The Official PGP User's Guide.* Boston: MIT Press.