Primitive-Based Payment Systems for Flexible Value Transfer in the Personal Router

by

Xavier F. Brucker

Diplôme d'Ingénieur, Ecole Polytechnique, France, 1999
Diplôme d'Ingénieur, Ecole Nationale Supérieure des Télécommunications, Paris, France, 2001

Submitted to the Engineering Systems Division
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Technology and Policy

at the

Massachusetts Institute of Technology
June 2002

Signature of Author……………………………………………………………………………………….
MIT Technology and Policy Program, Engineering Systems Division
May 10, 2002

Certified by...……………………………………………………………………………………………
Sharon Gillett
Research Associate, Center for Technology, Policy and Industrial Development
Thesis Supervisor

Certified by...……………………………………………………………………………………………
John Wroclawski
Research Scientist, Laboratory for Computer Science
Thesis Supervisor

Accepted by.........…………………………………………………………………………………………
Daniel Hastings
Professor of Aeronautics and Astronautics and Engineering Systems
Director, Technology and Policy Program
Chairman, Committee for Graduate Students

Primitive-Based Payment Systems for Flexible Value Transfer in the Personal Router

by

Xavier F. Brucker

Submitted to the Engineering Systems Division on May 10, 2002
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Technology and Policy

ABSTRACT

The Personal Router is a mobile communication device developed by the Advanced Network Architecture group at the MIT Laboratory for Computer Science. The Personal Router is able to select and negotiate connectivity with local providers for different kinds of services and interfaces. It needs payment procedures to support these services. As this device is designed to be used in many distinct unpredictable contexts, it cannot implement a single payment system. The complexity of existing payment systems has to be mapped into this new environment. A different payment system must be chosen each time, depending on many variables such as costs, environmental constraints, privacy, user and provider's needs and preferences.

Privacy is a major issue for this device. In effect, getting wireless and mobile service everywhere will possibly leave an easily traceable trail; moreover, using this device supposes negotiating with many different untrusted providers and paying for the service. This can create huge potential threats for privacy and personal data management if this issue is not included in the early stage of the design.

Legal requirements and user preferences and expectations for privacy in electronic transactions are therefore explored. Past attempts to enhance privacy in different environments are examined. Reasons why most of them have failed and some of them are struggling to stay alive are analyzed. New privacy threats faced by the Personal Router are considered.

A new approach based on building blocks is made. Payment systems are split into primitive operations; each of them implements one step of a transaction. The combination of these building blocks replicates a payment protocol. The characteristics of a payment system can then be derived from the analysis of the implementation of each of these primitives. Users' preferences are defined by attributes. Payment systems can then be compared through their primitives and even slightly modified to be closer to users' ideal system by altering the primitives. The modular approach makes this easier. This framework is successfully tested on three major electronic payment systems. Several limitations of this approach and open issues related to the Personal Router are exposed.

Thesis Supervisor: Sharon Gillett
Title: Research Associate, Center for Technology, Policy and Industrial Development

Thesis Supervisor: John Wroclawski
Title: Research Scientist, Laboratory for Computer Science

Acknowledgements

This thesis was completed with intellectual and financial support from the Center for Technology, Policy and Industrial Development (CTPID). I gratefully acknowledge the support of the industrial sponsors of the MIT Program on Internet and Telecoms Convergence (ITC), listed at http://itc.mit.edu. This thesis was written thanks to the generous contribution of time, advice, patient guidance and encouragements of my supervisors Sharon Gillett and John Wroclawski. The discussions I had with both of them helped me to navigate to the end of this work. I would like to add special thanks to Sharon for her trust, her dynamism, for the time she always found to help and guide her students and for these two years I spent with ITC. I would also like to thank the entire team of ITC, fellow ITC research assistants and the students from TPP for their support and friendship.

I want to thank the Advanced Network Architecture group at the Laboratory for Computer Science for their warm welcome throughout the year. In particular, I am grateful to George Lee, who helped me draw the framework described in this thesis, and Dina Katabi, who accepted me in her office in NE43 during this year.

My experience here at MIT has been wonderful. I had the chance to meet people from all over the world and to live in a very stimulating environment. I will leave a lot of friends here in Boston, and I hope we will stay in touch.

Finally, Family and friends provided an invaluable support throughout my time at MIT. I want to acknowledge the constant encouragements I received from France, in particular from my mother Catherine, and from my two brothers Benoist and Matthieu. In Boston, all my friends have been very encouraging and supportive, particularly my roommates Vincent, David and Guillaume who spent the year with me and with whom I traveled as far as our good old Oldsmobile Delta 88 Royale would take us.

I would also like to give special thanks to Carol for her patience and for letting me spend these two years in Boston.

TABLE OF CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

1.1 Context

1.1.1 The Personal Router

The Personal Router (Clark and Wroclawski, 2001) is a mobile communication device currently under development at the Advanced Network Architecture group (ANA) at the Laboratory for Computer Science (LCS).

The aim of this device is to study a new paradigm and a new market model for mobile communications. The motivation for this project comes from two main observations on today's mobile markets and technologies. First, there is no middle point between a fixed wireless network, local to a physical entity, and a national or regional mobile communication system such as cellular networks. There is enough space in between for other market models, such as local providers offering service locally with short-time contracts. The other observation is that today's devices are limited by a restricted set of interfaces and wireless modules. Cell phones, 802.11b wireless, pagers, use different frequencies and interfaces. Technically and contractually, they cannot interact with each other. Therefore, the different markets for each of these devices are locked and do not compete. Opening interfaces and devices could create a new paradigm and more competition.

The Personal Router is the result of these considerations. Technically, it is able to interact with a wide variety of incompatible wireless services and interfaces. It can access these different networks on behalf of other personal digital devices carried by the user; these devices would all be linked together by a personal area network such as Bluetooth. For instance, a PDA trying to download emails would do it through the Personal Router, which would be in charge of finding the best network available for this task.

This responsibility requires tools to match different kinds of services (defined in terms of price, bandwidth, quality of service, delay) with the requirements and rules of the application and of the user. The Personal Router would then be able to negotiate and to make a payment for the chosen service. The user and the provider would subsequently be bound by a very short-term contract.

The goal of this research led by the Advanced Network Architecture group is dual. The first is to show that a new wireless communications market paradigm is technically possible. The second is to develop a framework for access to wireless services and automated negotiation for these services.

The Personal Router has the potential to change the market of mobile telecommunications. As we have seen in the past, it is almost impossible to anticipate perfectly the new direction that this market will follow after the introduction of this innovation. Therefore, the goal is not to foresee the future, but to make everything possible for it to happen. Therefore, the design process if the Personal Router is intentionally open, to allow unpredictable orientations of the technology and of the market.

Today’s forms of the wireless market are still possible in this new environment. Contracts could last longer than a few seconds or minutes, and provider can be larger than a few antennas in a local area. For instance, cellular networks operator could still provide long-term contracts and wide area coverage for a flat rate. The aim of this project is to create new business opportunity and to offer a new paradigm for the market of mobile communications.

1.1.2 Payment issues

The Personal Router is able to select and negotiate connectivity with local providers for different kinds of services and interfaces. It will buy an intangible good: connectivity and network services. The goal is therefore to support a wide range of payment models for these services. The Personal Router model also implies that a very large number of providers will interact with a very large number of users. These interactions will be primarily network services. The Personal Router needs payment procedures to support these services. As this device is designed to be used in many distinct unpredictable contexts, the possibility to use a fixed number of predefined payment systems is likely to limit the spectrum of possible usage and the overall flexibility of the system.

In the “real world,” users have different payment systems available to them, like cash, credit card, checks, or money order. These payment systems are not interchangeable. They have diverse properties, and that is why a particular user will prefer to pay by cash or by credit card for a particular good. This choice is made consciously for real objective reasons, or maybe unconsciously, because of habits.

It is very likely that this behavior will repeat itself in the electronic world. Depending on the context, users will have various sets of requirements to conduct a transaction.

This is why diversity and flexibility of the payment process is highly desirable for the success of the Personal Router.

1.1.3 Privacy

Privacy is a major component of the choice of a system or another. In effect, the Personal Router sits exactly at the intersection between several major technologies. First, it is a mobile communication device. Second, it needs to conduct electronic transaction very often. Finally, it provides access to the Internet and to all the services available on the Internet.

Getting wireless and mobile service everywhere will possibly leave an easily traceable trail. While the user is walking down the street or driving his car, the Personal Router is continuously scanning for available services and negotiating, depending on requests from the personal devices of the user. The providers with which the device is interacting are not known and cannot be trusted. This constant activity could leave a trail that could be traced back to the actual user if this issue is not addressed.

On the e-commerce side, negotiations lead to transactions and to the providing of services. Making a transaction can leak information, and this must be controlled.

The service provided can also lead to information leakage. The traffic (voice and/or data) is transiting through an unknown and untrusted provider. This provider could use the easily accessible data going through its network to get personal information about its customers. Providers in the world of the Personal Router are likely to be much more numerous, making it much more difficult to enforce regulations. On the other hand, the large number of providers greatly increases the difficulty for them to collude and to build an accurate profile of users.

The fact that the Personal Router intersects with those three issues of privacy makes the global problem even more difficult. A party accessing personal information of the user could obtain this information if released to other parties. For instance, if the user is going to pay with her credit card and leave her name, it may be useless to make the mobile and the Internet aspects anonymous if parties collude and correlate the different pieces of information.

Moreover, privacy is a complex problem in itself, because it has many different levels. People have many different privacy expectations, depending on context, culture, habit, or other variables quantified with difficulty. Users could also use their private information as a component of the transaction: for instance, they may choose to give their name or some other personal details in order to get a cheaper service in some specific circumstances.

To address this issue, chapter 3 gives some privacy prospective by defining privacy and examining privacy in the US and European law. It then describes the specific privacy issues in mobile communications, electronic commerce and Internet and gives a few examples of solutions that have been tried.

1.2 Approach to payment systems

1.2.1 Definition of the problem

The Personal Router model implies that a very large number of providers will interact with a very large number of users. This model will be difficult to put into service if only a fixed number of predefined payment systems are available.

To be able to compare and select any payment system, a comprehensive framework is needed. This framework must allow parties to select and negotiate payment systems to conduct a particular transaction. This will be done according to predefined constraints and preferences from both sides.

The framework should be able to automatically compare payment systems. It needs to be able to negotiate about the properties of the system, and possibly be able to slightly modify it to be closer to the ideal characteristics. It should also define fairly high-level variables that users could set up and that would represent important properties of payment system.

1.2.2 Related work

Most approaches to payment systems compare payment properties as a whole. This has a number of drawbacks. First, it requires a complete examination of a payment scheme to extract its properties. Payment schemes are very complex and are not easy to analyze. This is therefore a high level task, not easy to perform automatically. Secondly, there are many ways of building a payment scheme; therefore, two payment schemes will not be easy to compare. Finally, changing one property of a payment system is likely to require rebuilding a whole new system.

Other articles, like (Jackobson, Mraihi, Tsiounis and Yung, 1999), (Pfitzmann and Waidner, 1996), (Asokan, Janson, Steiner, and Weidner, 1996), or (Lee, Yu and Kuo, 2001), give different ways to analyze payment systems. They give precise definitions of the properties of payment systems. Depending on their standpoint, they chose different properties. These properties are important and constitute an accurate framework for describing payment system. This helps to build taxonomy of payment system, but is not close enough to implementation to allow automated comparisons and selections of payment schemes. None of these articles suggested a way to formalize comparisons and selections of payment systems depending on user preferences and context. While it is important to be able to define and classify payment system, the context of the Personal Router require a way to select a specific payment system according to a changing context and the unpredictable result of a negotiation.

From this standpoint, the approach demonstrated in (MacKie-Mason and White, 1997) is more interesting. The method is to choose thirty criteria. Each payment system is characterized along these criteria, and compared by a decision maker algorithm. This thesis builds on this approach and tries to simplify the way payment systems are described and built. The attributes chosen are important to give a good description of payment systems. The selection process presented in this thesis is an adaptation of the decision maker algorithm suggested in this article. The decision maker proposed in (MacKie-Mason and White, 1997) uses a form of prioritization characteristics. The idea is that a small number of high priority characteristics are often sufficient to make a decision, without the need to include smaller priorities. This simplifies the algorithm, but does not provide a way to slightly modify the chosen scheme to go closer to users' preferences.

Another interesting approach is the one chosen by P3P[1], developed by the World Wide Web Consortium (W3C). P3P lets a client and a server negotiate on what information will be recorded and how it will be used. However, P3P is designed for the web, and not really suited for electronic payment. The other issue is that P3P relies on trust: there is no enforcement.

1.2.3 The approach chosen

Instead of studying payment systems as a whole, this thesis approaches payment systems by decomposing them into primitive operations. A primitive operation is defined as an operation that performs a basic function during the payment procedure. The range of payment systems is wide, but this thesis proposes to extract six primitive operations: authentication, authorization, transfer, record keeping, aggregation, and timing. The

[1] http://www.w3.org/P3P/

combination of these primitives creates an entity that replicates the original payment system. The comparisons are made through a set of attributes.

Attributes represent a sub-part of the properties of payment systems presented in chapter 2. They are the main characteristics that are important for selecting a payment system in particular situations. They are not design properties, which are important to accurately describe a payment system, but which are not useful when comparing payment systems in a specific environment and context. The attributes selected are divided in two groups. The first group is a group of physical requirements due to the environment, such as the type of user device necessary, the time available, or the communication bandwidth. The second group is a group of preferences set by users, which could change depending on context, such as privacy, security, or obtrusiveness.

Primitives are elementary operation representing a portion of a payment. A payment can be decomposed into these primitive operations, and can be modeled as a construction of these primitives.

Primitive are only a small part of a much bigger problem. To achieve a complete framework, several additional pieces would be needed. First, a good decision maker algorithm would be necessary to pick the right payment systems among the choices available. Then an efficient user interface is mandatory to help users go through all the possibilities they have and to help them understand what decisions are taken on their behalf. Finally, an effective negotiation protocol must be implemented in order to find the most suitable payment system for all the parties involved.

This model of payment systems has several advantages over the traditional approach. First of all, instead of defining a taxonomy of complete payment systems, we can define a taxonomy of these primitives. Although it is true that each primitive will be implemented in a different way depending on the payment system, it is possible to observe classes of primitives acting almost identically and having the same types of properties. Then, by compiling the properties of each primitive involved, we can derive the properties of the payment system. Therefore, a comparison of two payment schemes is done very easily by comparing their primitive elements.

Secondly, once a payment system is decomposed into these primitive elements, it becomes very easy to modify its properties by interchanging one primitive. For this purpose, primitives are further broken down into general classes. Each of these classes describes a way to implement the primitive, and can be implemented for real. These classes perform a similar operation; it is therefore possible to exchange an implementation for another. As a result, the resulting payment system has modified properties.

Finally, the combination of these two advantages creates a favorable framework for negotiation. Starting from pre-defined payment systems, it is very easy to change their properties by changing their primitives. These modifications can takes place as the result of requirements from one or more parties involved, and can evolve depending on negotiation of the properties of the payment.

A potential application of this capability is the automated negotiation of payment systems. The device will be able to negotiate not only for services, but also for the payment system to be used and for the information exchanged. These three negotiations will be conducted together; for instance, the provider may charge less is the payment is made with electronic cash, compared to a payment through the credit card system. Instead of only negotiating for the type of payment to be made, it may be possible to modify a payment scheme so that it would fulfill the requirements of both the user and the provider. In this case, the negotiation would also include the construction of this corresponding scheme.

1.2.4 Results

Chapter 5 gives three examples of the application of the framework of primitives. Three payment systems are chosen: DigiCash, the credit card system, and PayPal. These three systems, chosen for their properties and their popularity, are decomposed into primitives. The attributes are characterized for each of these primitives. Some modifications of these primitives, creating alteration of the complete payment system, are examined.

The primitive are demonstrated to work for at least those three payment systems. They even provide a relatively easy way to slightly modify them. However, the proposed framework does not seem to be powerful enough to be able to produce deep changes in payment schemes. This is due to the fact that in the present framework, the shape and the interfaces of the payment system have to be preserved.

In the conclusion, after a brief overview of key findings, several open issues are pointed out. Among them, privacy issues, the design of a powerful user interface and negotiation algorithm, and potential enhancements of the framework that would allow deeper modifications of payment systems are the most important.

2 THE RICHNESS OF PAYMENT SYSTEMS

The aim of this chapter is to understand better the complexity of the problems raised by the Personal Router in the fields of payment systems. The Personal Router needs different systems for different usages. The notion of payment system is complex and requires the understanding of the multiple characteristics of payment systems. Therefore, this chapter gives an understanding of and the numerous properties and the diversity of payment systems that the Personal Router should be able to support.

After a few definitions relevant to payment schemes, various characteristics of payment systems are presented.

To get a deeper overview of payment systems, see (O’Mahony, Peirce, and Tewari, 2001) and (Camp, 2000).

2.1 Definitions

In this section, we define some of the terms we will use intensively thereafter.

2.1.1 Transactions

A transaction is a transfer of goods and money between two entities. It usually involves four steps, some of which may be optional:

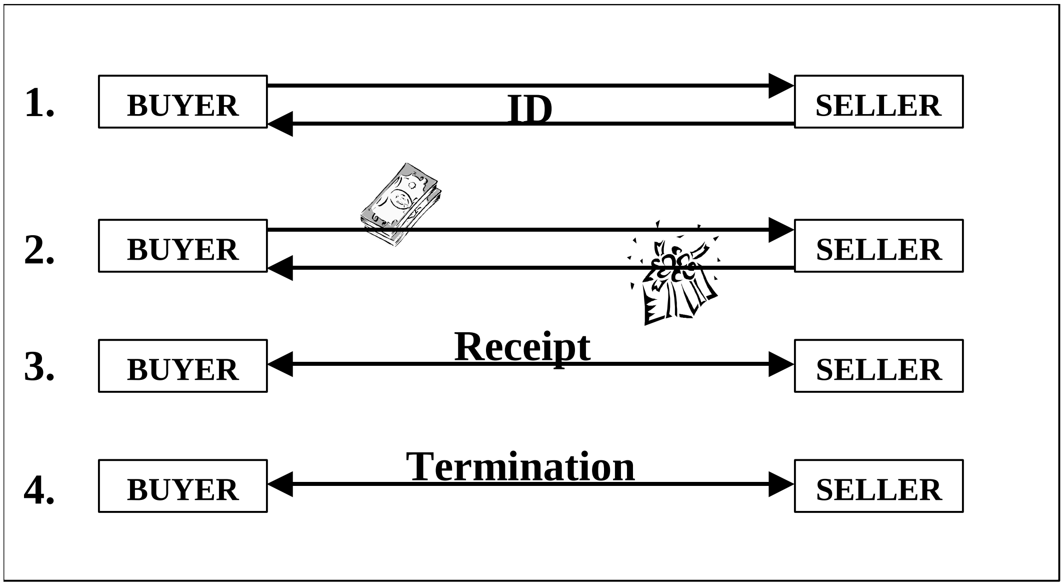

Figure 2-1: Definition of a transaction in four steps

1. Authentication: Each party identifies itself and shows proof of identification to make sure that they are authorized to conduct this transaction, and that they really are the entity they claim to be.
2. Trade: During this step, the payer gives money to the payer, and the payee gives the goods to the payer. This step may involve third parties certifying the validity of the exchange.
3. Acknowledgement: Each party sends a receipt that acknowledges the reception of the payment and of the goods.
4. Conclusion: If all parties agree on the outcome of the exchange, the transaction terminates. In the case of disagreements, dispute resolution mechanisms must be applied.

Examples

Retrieving a web page can be viewed as a very simple transaction, which involves neither exchange of money nor receipt. First, the browser connects to the distant server (and if necessary shows credentials in case of restricted pages). Then the server sends back the requested files. Finally, the connection ends.

A more complex transaction is, for instance, buying beer. In effect, buying beer can be described as a four steps transaction:

1. First, the buyer shows proof (e.g. ID) that he is over 21, thus being authorized to buy alcohol,
2. Then the buyer pays for the beer and the barman serves it. The buyer can check that the drink served is a beer, and the barman can check that the bill is not false.
3. The barman gives a receipt to the buyer, showing that the beer was paid.
4. All parties agree on the happy conclusion of the deal, and the buyer can drink the beer.

2.1.2 Stakeholders

Merchant and the merchant's bank

The merchant is the entity that sells goods, either physically or electronically, in exchange for money. A fraudulent merchant would try to get the money without delivering the goods. The merchant is sometimes called the <u>seller</u> or the <u>payee</u>.
The merchant has an account in a bank, which can also be called the <u>acquirer</u>. This bank is capable of converting electronic money into external forms of money.

Buyer and the buyer's bank

The <u>buyer</u> is the entity that buys goods from the merchant by using money. The buyer is sometimes called the payer, the customer or the client. A fraudulent buyer would try to get goods with paying for them.
The buyer has an account in a bank, sometimes also called the <u>issuer</u>, the <u>broker</u> or the <u>billing center</u>. The bank has the ability to convert external forms of money such as cash or checks into electronic money.

Broker

Some systems use a broker as a central authority for delivering electronic money to buyers and collecting it from merchants. In some cases, a single broker can play the roles of both the <u>issuer</u> and the <u>acquirer</u>.

The broker plays two important roles. The first one is to allow parties that are unknown to each other to complete a transaction. There is a transfer of risk from the payee to the broker. The second role is the role of an aggregator. A broker has a large number of relationships with players in the market, and can therefore benefit from strong economies of scale during the different steps of a transaction.

Trusted third-party

A trusted third-party is an entity trusted by both the buyer and the seller. It can assume many different roles depending on the system. It can either be a broker, a certification authority, a central authority, a clearing center (mediating transfers of money and goods), an arbiter, or an anonymizing / pseudonymizing server. It can also play different roles for law enforcement and tax purposes, or dispute resolution.

Observers

Observers are entities monitoring the transmissions between the customer and the merchant. Observers cannot read encrypted information, but can read all other information transmitted while making the transaction.

Governments

In the case of law enforcement with warrant, governments can obtain records from banks and merchant. However, the amount of information that can be obtained depends on the electronic payment system. The aim of governments is to prevent fraudulent actions such as tax evasion and money laundering.

2.2 Characteristics of payment schemes

In this section, we define the main properties of payment systems. These characteristics are the basic variables that can be used to describe a payment system in depth. They help to classify payment systems, and to define the requirements of a payment system to be used in a particular context. The following characteristics come from a number of articles, in particular (Mackie-Mason and White, 1996), (Ferreira and Dahab, 1998), (Pfitzmann and Waidner, 1996), (Asokan, et Al., 1996), and (Lee, et Al., 2001). This is not a compilation of design characteristics of payment models; the following section aims at exploring the spectrum of payment systems and pointing out the most important properties.

2.2.1 Exchange model

The model of exchange describes the way value is changing hands during a transaction. Historically, there have been three main models for exchanging value: barter, tokens and notational money.

Barter vs. money

Barter is the oldest way of exchanging value. It has since been replaced by money, which is a much more efficient way to exchange value. Barter requires that the two parties agree on two assets they would be willing to exchange. This way of exchanging value is not efficient, because this system needs to create correspondences between all exchangeable goods.

On the contrary, money requires only a correspondence between all exchangeable goods and money. Barter is replaced by a two-step process: first, sell the first good to convert it to money, and then buy the second good.

Money can also play two other important roles in the economy. The first one is to provide a standard of value, which facilitates the evaluation and the comparison of goods and services. The second one is a store of value. Wealth is much more difficult to store, exchange and transport in the form of cattle or fields than in the form of money.

Token based

In a token-based system, transactions are made by exchanging tokens of predefined values, bought from a central authority before the transaction, like coins and banknotes. The stream of bits is itself valuable. These systems are also called cash-like, or token money. A large sub-category of this model is pre-paid systems, in which the buyer buys token from the merchant before the transaction, and uses them to get the goods. One example is pre-paid phone cards.

The big issue raised by token-based schemes is double-spending. Double-spending occurs when users are able to spend twice or more the same token. In the case of electronic money, duplicating tokens can be fairly easy if additional protections are not created, as a token is nothing more than a stream of bits.

This can happen either in the case of fraudulent action of either the buyer, the seller or observers. The buyer could spend several times the same electronic token; a seller may try to redeem the same token several times; an observer wiretapping communications could copy the token and spend it instead of the buyer.

Notational

Notational systems exchange documents that enable money transfer between two accounts managed by a central authority. The central authority is responsible for keeping records of the transactions and of the accounts. The best example is bank checks or credit/debit cards. In notational systems, the information transmitted is not valuable by itself; it is valuable because of the transfers of money between accounts resulting from this information.

Value is actually transferred by debiting the payer’s account and crediting the payee’s account.

Relations between token and notational money

Token and notational money are just two extreme positions of payment systems. There exist different payment systems that can share properties of both systems at the same time. An example of these is cashier checks. Cashier checks presents several similarities with a token money, although there are closer to being considered as notational.

There are also many interfaces between notational and token money, such as ATM.

2.2.2 Hardware

Specific

Some systems require a specific hardware to process transactions. For example, credit cards, smart cards, are a specific hardware, and they need a card reader to complete the transaction. Hardware specific schemes usually assume that some part of the hardware is tamper-proof (for instance, the chip of a smart card is supposed to be inaccessible and tamper-resistant, even by the card holder). In this case, the hardware can certify that the user is not over-spending money. It can also help to block fraudulent actions. The best example of this is Chaum’s device (Chaum, 1992).

Non-specific / software only

Non-specific hardware can be any general-purpose hardware like a computer, a PDA, a cell-phone, etc. In this case, the transaction scheme is software only. The system must have additional security features to prevent users from obtaining any benefit from tampering with the software, the data or the communications exchanged.

2.2.3 Authorization

Out-band authorization

The central authority sends a message to the payer and asks her to approve or deny the transaction off-line using another communication channel. For instance, this channel can be post mail or a phone call. An example of this on the Internet is First Virtual, which sends a demand of approval to the payer via email.

Authorization by shared secret

The central authority requires that the transaction include a shared secret known only to the payer and the central authority. This secret can be a password or a PIN, or any cryptographic variation of a shared secret.

Authorization by Signature

The central authority can require that every transaction include a signature from the payer. This signature can be either electronic with the use cryptographic tools, or handwritten in the case of checks.

2.2.4 Utilization model

Account / registration

Most of the electronic payment schemes require that users register first and use an account. This account may be set either with the vendor or with a broker or a bank. This may limit the base for vendors and/or customers using this payment scheme, therefore limiting universality.

Storable electronic money

Some systems allow customers to store their electronic money on their personal device. It means that transaction between the bank and the users can be asynchronous. Users have to contact the bank only occasionally. This is often used by micropayment schemes, as it limits communication costs.

2.2.5 Central Authority

Off-line

Off-line systems allow users to make transactions without contacting the system's central authority at the moment of the transaction. Only the payer and the payee have to be in contact at the moment of the transaction.

This kind of payment must include a way of verifying the validity of the transaction. Examples of off-line systems are cash (the seller can verify that the banknote is a real one), checks (the seller can verify the ID of the customer and her handwritten signature). The main issue with off-line systems is that they need a way to ensure that users are not spending more money than they actually possess. These systems, like credit cards, usually implement limits on how much the user can overspend.

On-line

On-line systems require one of more of the parties to be on-line at the moment of the transaction and contact the central authority. The central authority can then authorize the transaction. This implies relatively high communication costs, and assumes that the central authority can always be contacted. If the network goes down, no transaction can be accepted. In the case of network congestions, an on-line transaction may be subject to time delays.

Examples of on-line systems are debit-card systems, and Internet based schemes.

2.2.6 Transaction costs

Transaction costs are the represent the marginal cost of a transaction. Processing a transaction creates some costs. For instance, it could be the cost of connecting to a central server, the cost of computing a cryptographic signature, a processing fee charged by a bank, or any other incremental cost.

High transaction costs

Transaction costs are typically high when the transaction involves manual processing. This causes huge overheads and long delays. An example of a scheme having high transaction costs is the check system.

Medium transaction costs

Payment system will incur an intermediate level of transaction costs if they do not use manual processing, but do use strong cryptography for online authentication and clearance. Therefore, they need high computational capabilities and most of them require large communication expenses. They often have high setup costs.

Low transaction costs

The cost of a transaction can be reduced further by eliminating the strong cryptographic features and reducing the overheads due to communications and computations. The transaction costs can be reduced to a few cents, and sometimes to fractions of a penny.

Financial risks

The risks of a financial loss can be borne by one or several parties. These risks may increase the costs of a transaction. The risks borne by the customer can be limited by limitations of the maximum amount that can be transferred. Some other systems, like credit cards, use a maximum customer liability in the case of fraudulent transactions ($50 in the case of credit cards).

The risk can be borne by the merchant or by the central authority, depending on the design of the scheme.

2.2.7 Payment Values

Each payment system is usually designed for a specific range of payment only, and may not be really well suited for other ranges. This is mainly due to a trade-off between the different characteristics of the scheme, in particular transaction costs and the level of security offered by the scheme. For instance, a scheme implying high transaction costs will be uneconomical for small financial transactions, and a scheme offering only poor security capability will be too risky for big transactions.

Large payments

Large payments require a more regulated framework to record the payment amounts and sometimes the parties. This is required for governmental inquiries and for dispute resolutions. Moreover, consumers need to have transaction records and dispute resolution mechanisms. They also need adequate security capabilities for transmission of funds. However, transaction overheads like processing costs, time or communication costs are of less importance because these costs tend to be really small in regard to the transaction itself, and large transactions are relatively infrequent. Also, anonymity is not always required, as a trail is usually required by law and makes dispute resolution easier.

Large payments are usually payment of more than a few hundreds of dollars. These systems often require dedicated networks (for instance inter-bank networks) offering greater security than Internet can provide today.

Micropayments

Micropayments are payment of a small value, typically less than $5. Some of the micropayment schemes are able to conduct transaction of tenths or even hundredths of a cent. The general property of these payments is to have a very low marginal cost, so that they can stay profitable even for small amounts.

In this case, dispute resolution is less important, because the amounts of transaction are very low. The exception to that statement is when micropayments are aggregated.

Therefore, most micropayment schemes provide fraud detection only when it is done on a large scale. Nevertheless, these systems must be simple to use, efficient, quick, should have a very low computation and communication overhead.

They should also be as anonymous as possible, so that users cannot be easily traced. However, anonymous systems require higher computational costs, and are therefore not easy to implement.

Small to medium payments

Small to medium payment fill the gap between large payments and micropayments schemes. They can have all varieties of properties of micropayments and large payments.

2.2.8 Elemental payment properties

ACID properties

ACID means Atomic, Consistent, Isolated, and Durable.

Atomic

An atomic transaction cannot be split into sub-parts. An atomic transaction either succeeds or fails completely. It conserves money: no money can be created (for instance, both accounts are credited) nor lost (both accounts are debited). For example, either the money is in the payer's wallet, or it is on the merchant' account. The goods are either delivered to the customer and the sender has a proof of the delivery, or they are not.

There are two different kinds of atomicity: money-transfer atomicity and goods-transfer atomicity. A transfer of good is atomic if the money and the goods are linked and transferred atomically.

Consistent

A transaction is consistent if all parties agree on the facts of the transaction. For instance, if a customer pays $10 to a merchant, the payer, the merchant and the banks all have to agree that $10 was transferred from the customer's account to the merchant's account.

Isolated

A transaction is said to be isolated if there are no overlap or interference with other transactions. Transactions executed at the same time should be separable. A customer should not be charged for someone else's transaction. If a customer makes two transactions at the same time, she should not be charged the sum of both transactions for the first transaction and nothing for the second.

Durable

A transaction is durable if it can be restored to its previous consistent state at any time.

Non-repudiation

ACID properties provide non-repudiation of the transaction. A transaction whose sub-steps are non-repudiable is non-repudiable itself.

Reliability

A system is reliable if it is able to recover from failures, attacks, memory losses, etc, to a previous consistent state.

Divisibility

Divisibility is a property that allows users to exchange arbitrary fractions of a previous transaction. All account-based systems have this property, as any fraction of the balance can be withdrawn. Cash, and more generally token-based systems, are not divisible: it is not possible to give half a one-dollar bill as a payment for 50 cents!

Some schemes go around this issue by issuing tokens whose values are the successive powers of two. E-cash, for instance, issues tokens worth 1 penny, 2 cents, 4 cents, 8 cents, etc, so that any amount of money can be transferred.

Transferability – Role inversibility

<u>Transferability</u> is a property that allows people to transfer funds between each other without contacting the bank or a central authority. This property is obvious for cash; however, it is very difficult to implement electronically because of security issues.

Role inversibility means that the payer and the payee can play each other’s role. In the case of cash or checks, roles are interchangeable. Each user can play different roles. In the case of credit cards, roles are not interchangeable; each user has a predefined role (buyer or seller). This property is also called “two-way” or “peer-to-peer” transaction.

2.2.9 System security

Data transmission

Isolating the transmission infrastructure

One secure but expensive way of achieving transmission security is to use a dedicated channel of communication. This solution is used by banks to process funds transfer within and between banks, and by smart cards schemes, which use a secure channel, the card reader, to communicate with the chip of the smart card.

Using Encryption

Another way of achieving security is to use encryption of the communication link over insecure channels. This allows conducting financial transaction over the Internet or the public communication networks. Most of the systems use public-key encryption. The encryption key is publicly available, while the decryption key is kept secret. The sender needs only to know the receiver’s public key to send encrypted messages.

Authentication

Electronic payment systems may require that the payer and/or the payee be authenticated at one point during the transaction. For instance, the bank needs to identify the payer before delivering money, and the seller needs to be authenticated to verify that the payment is made to the real payee; this is particularly important to prevent the “man-in-the-middle” attack. Payment systems display a wide range of authentication methods, including signatures, passwords, and biometrics.

Handwritten signature

Handwritten signatures are the oldest authentication scheme. They are used to secure check and credit card transactions. However, This scheme is very expensive, as it requires an expert to tell whether the signature was forged. In effect, signatures vary and cannot be easily recognized by electronic devices. Furthermore, they do not really protect from the misuse of stolen checks or cards.

PIN / Passwords

The most common way of authenticating is by a Personal Identification Number. This system is used to authenticate simple systems like a magnetic card in an ATM. Passwords are just an evolution of this scheme. The number of possible combinations is much higher than with a PIN-based scheme.

However, PINs and passwords suffer from the same defaults. People have difficulties remembering their password, they tend to write them down, or use very simple passwords such as their birthday date.

Electronic signatures

Electronic signatures use cryptographic technologies to certify that a document was created by a particular person. The document, or the hash value of the document, is encrypted with the private key of the user. Anybody can verify that the document was created by this person and was not modified by decrypting this value with the public key and checking that the hash value is the same as the one obtained by hashing the document.

The weakness of this scheme is that the secret key has to be stored in an electronic form. It can either be stored on a personal computer, but in this case anyone that can access this computer can sign a document on the behalf of the real user, or it can be stored on a smart card, but in this case we need a way (PIN or password) to gain access to the smart card.

A public key can be checked by having it signed by a central authority recognized by everybody.

Biometrics

Biometrics uses measurable biological characteristics such as fingerprints, voiceprints, eyeprints, face recognition, etc, to authenticate the user. These systems are not widely used yet, but they should develop in the near future as techniques improve.

2.2.10 Costs

Costs can be differentiated between four categories:

- Computation
- Communication
- Fraud
- General and administrative costs

Although it can be fairly easy to normalize computation and communication costs (e.g. number of elementary operations to perform, bandwidth), it seems much more difficult to normalize fraud costs and administrative costs.

Computational costs

Different types of computational costs

Computational costs are the costs incurred by computation required by the payment scheme. There are three main types of arithmetic operations required: authentication, encryption and hash functions.

Authentication

The most used way of identifying someone is to use a Public Key Infrastructure (PKI). This requires the generation of public/private keys pairs, and large computations to compute and to verify signatures. Moreover, verifying a signature is often not enough, especially if the public key of the user has not been previously verified. In this case, the signature has to be signed by a certificate authority, whose signature has also to be verified. This creates a chain of signature verification along the certification authority (CA) hierarchy, which goes up to the certification root.

Some systems (usually micropayment schemes like PayWord, MicroMint or PayTree) use a much less computation-intensive way to verify the identity of a user. Instead of computing public key signatures, they compute the hash function of a number given by the payer. If the result of the hash function gives a pre-determined value, then the user is authenticated.

Encryption

Encryption is very important to keep the data between the parties secret and to avoid eavesdropping. This can be a part of the security features of the system, to prevent eavesdroppers to learn secret codes or coins, and/or this can be a feature designed to keep the transaction private. Two kinds of encryption exist: the symmetric encryption and the asymmetric encryption. In the symmetric encryption, both users use the same cryptographic key. In the asymmetric encryption, each user has a private/public key pair. Asymmetric encryption is much more computation-intensive.

Schemes like SSL allow to establish contact via asymmetric encryption, and then to exchange a session symmetric key, which will be used to encrypt communications.

Hash function

Hash functions are used a lot by micropayment schemes. A hash function is a function that produces a number from a message in such a way that it is extremely unlikely that some other text will produce the same hash value. It means that finding another message whose result is the same by the hash function is extremely difficult. Commonly used hash algorithms are SHA, MD5, MD6, etc.

They are used to digest messages and to ensure integrity, and sometimes as authentication system (see *authentication*).

Type of user’s device

Depending on the type of device making the computations, computational cost can be more or less important. If the computations are done on the user’s PC, this should not be a problem, since most machines are capable of computing several hundreds of 1024 bits RSA signatures per second. However, if the payment system is to be used on a mobile device with limited computational power, or on a smart card with only a few kilobytes of ROM and RAM memory and a slow 8-bits processor, this may be a huge problem. Special algorithms have to be designed in this case.

On the server's side

On the server's side, the scaling of equipment with the number of user has to be reasonable. In particular, the bottlenecks can be the communication link, the number of digital signature to sign or verify per transaction and the number of computation in general. This is particularly the case in micropayment schemes, where the equipment has to be leveraged on a very large number of transactions to become profitable.

Some payment systems, like MicroMint, take advantage of the large number of coins to be prepared; in fact, the marginal cost of minting a new coin is decreasing exponentially. Other systems do not scale so nicely.

Communication costs

These costs are especially high in the case of mobile commerce. If the transaction must occur over a mobile link of usual current mobile networks such as GSM, communications are slow and expensive. Therefore, payment systems designed for this type of application must implement special features to limit the amount of data to be transfer as well as the time of connection. As an example, SET developed a specific mobile-oriented payment system called mobile-SET, which is an evolution of the SET protocol specifically designed for mobile handsets.

Administrative costs

Administrative and general costs are the costs of maintaining the system, keeping records of different operations, registering users, searching for fraudulent transactions, charging users, etc.

For instance, DigiCash has to keep records of all coins spent to prevent double spending. Other systems have high registration costs. Some systems have to send bills to customers, which is very expensive.

Fraud costs

Fraud costs occurs everywhere. Each of the previous costs is a trade-off with fraud. Reducing administrative, computation or communication costs will often increase fraud costs.

Fraud: two opposite approaches

Some systems try to prevent fraud by design. Fraud is made impossible by the use of private credential, or by other means. For instance, to forge a check, we would have to imitate someone else's signature, which is supposedly impossible. In the electronic world, we would have to show that we know the private key of someone else. PayPal and Yahoo! PayDirect use a link to the real world to ensure the identity of their customers and their real-world bank account. When a new user is registering, the system makes two small deposits of a random amount between 1 cent and 99 cents, and if the user can tell the amounts transferred, then she has access to the real-world account and she is authenticated as the real owner of the account.

On the other hand, others schemes acknowledge that small-scale fraud is possible, and try to prevent only large-scale fraud. This is often the case in micropayment schemes. The main approach in this case is to make sure that forging a coin is much more expensive than the expected value of the coin. Therefore, there will be no economical incentive to cheat, although it is technically possible.

Detecting fraud before or after the transaction

Another design characteristic is to know whether the system tries to detect fraud before or after it was performed. It is obviously much more expensive to detect it during the transaction, because it involves much more real-time communications and computations. If the system is designed to detect fraud after the transaction, it can be done off-line during the idle time of the server.

Micropayment often choose the second solution. If a fraudulent action by the user is detected, this user can be blacklisted and rejected from the payment system.

Who is supporting the fraud costs?

In case of fraud, at least one of the actors of the transaction has to lose some money. Depending on the system, fraud costs are in fact supported by the financial institution, the payee, the payer, or a combination of those three actors.

Note that in the special case of credit cards, the customer is usually only liable for a small part of a fraudulent transaction. The customer usually has to pay the first $50 in the US. The credit card network must pay the rest.

This cost takes a very important part in the financial equilibrium of the payment system.

2.2.11 Active / passive payment

A basic definition of an active payment is a payment in which the user has an active role. A passive payment is a payment that does not need any input from the user. As we will see, the spectrum is more or less continuous between these two extreme positions.

Active payments

In most design of current payment systems, users have been supposed active. For example, paying with current payment systems as cash or credit card requires a very active part from the user.

This is sometimes desirable. For example, when a customer buys an airline tickets on the Internet for $400, she would like to be involve in the payment process at some point to make sure everything is in order. This is usually done by having to enter the credit card number on the checkout page.

Similarly, most designs of electronic payment systems have at least a popup window to ask the user if she agrees on the payment to be made. Although the user's perception of the payment in this method is less active than the previous one (one must only push the "OK" button), this is still active in the sense that the user is involved in the payment process.

The gray zone between active and passive

However, this "active" process may become annoying, especially if this happens too often and if the amounts paid are too small. That is why we need methods where the user's perception of the payment is more passive.

Here are different typical models of use. There are obviously a lot of systems sitting between these three models.

Active set-up, active action, passive transaction

In this case, the user prefers to set up the payment system in advance for all subsequent payments, and let everything work on its own.

A good example of this type of utilization is EZ-Pass. The user registers once for all with EZ-Pass, and can then avoid paying directly traffic tolls. By taking an active step, all future payments are passive: the user is not involved in the payment process at the time of the transaction. This is also the model of a cell phone registration.

Still, this is still active in the sense that the payment is activated by an action of the user. The user knows that when passing a toll zone, she will be charged for it.

This is also the business model of micropayments. A user sets up a payment system, either with a monthly bill or with a pre-paid system. When reading a newspaper on the Internet, users know that they will pay a certain amount per article viewed. However, no popup message is there to tell them.

Active set-up, passive action, passive transaction

In this case, the user sets up the payment system before using it. However, she does not initiate directly the transaction. This could be a case of use of the Personal Router: the user is walking down the street, and walk into a "hot spot" without have any knowledge of it. The Personal Router decides autonomously to check the user's email, and to conduct the transaction on its own. The user has no idea this is happening.

However, this is not a full-passive mode, because the user set up the payment system and configured the device before the transaction, so that it would act like this in a hot spot. Although the user is really not involved in any part of the transaction, this transaction is still a result of her will.

Passive payment

In a full passive mode, the device would decide everything, from the payment system to the decision of negotiating and using services. Therefore, it does not seem realistic to imagine that a device could decide to pay for something on its own. The payment process is always the result of the will of the user, expressed in a way or another during the set up phase.

2.2.12 Visible / invisible payment

The user perception of the payment is not the same if the good bought is perceptible or not. This characteristic is one of the difficulty coming with the Personal Router is that connectivity is invisible. Most payment systems are designed to deal with visible goods, real or electronic. But the Personal Router's user would buy connectivity, which is hard to define and is totally seamless for the user. Connectivity is characterized by delay and bandwidth. These two variables are much too low level for the user. The user just wants to check emails or download a file. Therefore, the user perception of this payment is fundamentally different from other payment paradigms. Thus, it makes sense to have some sort of passive mode, as it is much easier for a computer to understand the definition and the quality of the connectivity provided than it would be for a human being.

2.2.13 Privacy

Privacy issues will be examined in depth in section 3. This section analyzes only the privacy properties directly related to payment systems. The analysis method with privacy tables presented at the end of this section ill be used extensively in the chapter 4.

Anonymity and untraceability

Privacy requires both anonymity and untraceability.

Anonymity means that the user's identity cannot be uncovered during the transaction. This requires that information that could link to the user's real identity, like account ID or credit card number, be not used.

Untraceability means that the transaction cannot be linked to other transactions involving the same user. Traceability means that all transactions of a user can be identified. Traceability facilitates profiling and marketing, and worries privacy advocates, while untraceability is an issue for legal authorities, as it is easier to use such systems for tax evasion and criminal activities.

There are several degrees of traceability and anonymity. A transaction can be traceable, which means that a record is generated for each transaction, identifying data such as buyer, seller, amount, date, etc, as it is done in every credit card transaction. On the other hand, untraceable payments cannot be linked to each other, and does not generate records, as for instance cash transactions. The spectrum is large between those two extreme points. For instance, some systems offer pseudonymity, which provide anonymity (the user's real identity is never exposed) without untraceability. From this kind of system, it is possible to improve the untraceability by frequently changing of pseudonym.

Perfect privacy

Privacy is defined in paragraph 3.1.1. Schemes providing perfect privacy allow users to conduct transactions without leaking any personal information. This privacy is only limited by the capacities of the cryptographic techniques used. If cryptographic tools protecting the data are weak or can be broken, then the private information is exposed. Schemes like e-cash can convince the merchant and the bank that the information provided is correct and worth the value of the transaction, without revealing any information that could lead to the user's identity. The cryptographic technique used for e-cash is called "blinding;" the bank signs a coin, thus giving it value, without being able to see the serial number. As a result, the coin is valid (signed by the bank), but nobody can identify the buyer, not even the bank.

Another example of perfect privacy is cash, if the merchant cannot recognize the face of the customer.

No or limited privacy

Privacy can be provided by many other ways. For instance, pseudonymity can prevent third parties from knowing the real identity of a user. A third party, such as a broker or a central authority, can also conduct the transaction on behalf of the customer.

Finally, some systems, for instance credit cards or checks, do not provide any privacy at all.

However, in some cases, parties can learn only a small part of the information about the others. For example, in a credit card transaction, the seller can learn the buyer's name and credit card number, but not her address or phone number. An observer of a cash transaction can see the time of the transaction and the faces of both the seller and the buyer, but cannot know their name and address.

Revocable privacy

Privacy is needed in most transactions, but it can sometimes cause regulatory and/or legal issues. In particular, a system providing perfect privacy capabilities could potentially make money laundering, illegal sales, fraud, and other illegal actions easier. Some schemes are able to revoke anonymity under certain circumstances such as a court order.

Context

Privacy audits of a scheme must be conducted from the point of view of each of the stakeholders involved in the transaction. For example, cash provides perfect privacy towards governments and banks, and limited privacy towards the merchant and observers. Encrypted credit card transactions (for instance by using SSL) provide limited privacy towards governments and observers, and no privacy towards banks and the merchant.

Information considered

Privacy is provided for certain types of information only. A scheme can provide perfect privacy for information about the customer, and no privacy at all for the amount of the transaction, the date, and the item purchased.

This information can be summarized in a table. For instance, Table 2-1 lists the privacy characteristics of real cash.

Table 2-1: Privacy characteristics of a real cash transaction

			WHAT THEY KNOW				
			Identity of buyer	Identity of seller	Amount of transaction	Time and place of transaction	Nature of transaction
WHO KNOWS	Buyer	Identifiable	N/A	Face recognition	Everything	Everything	Everything
		Traceable	N/A	Nothing			
	Seller	Identifiable	Face recognition	N/A	Everything	Everything	Everything
		Traceable	Nothing	N/A			
	Bank	Issuer	Nothing	Nothing	Nothing	Nothing	Nothing
		Acquirer	Nothing	Nothing	Nothing	Nothing	Nothing
	Central authority		Nothing	Nothing	Nothing	Nothing	Nothing
	Observer		Face recognition	Face recognition	Nothing	Everything	Partial

This table lists in columns the type of information released, and in rows the party getting access to the information. For instance, the buyer knows the identity of the seller if the seller's face is recognized. Banks do not learn anything about the transaction. Observers know the faces of the buyer and the seller, and the time and place of the transaction, but do not know the amount.

2.3 Transaction Models for the Personal Router

2.3.1 Introduction

This section tries to list as many models of transaction occurring in the context of the Personal Router as possible. Of course, this list cannot be exhaustive and it should not limit the design of the transaction system implemented in the Personal Router. The aim of this list of models is to get a more precise idea of the possible types of transactions that could occur between a Personal Router user and a service provider. This can be particularly useful in order to have a better understanding of the flows of money and information between the different parties during a transaction. These models are not at the same level as payments; in fact, the payment could occur as a part of a transaction, and many other data exchanges could occur in addition. Therefore, these models lead to a better appreciation of the potential data leakages that are not due to the payment.

2.3.2 Business opportunities

There are different business incentives for an established business to install a Personal Router base station. The first one is obviously to sell connectivity. Users passing by have the opportunity to get access to the Internet through this paying access point.

Another incentive is to use the Personal Router as a device transferring value and money. If the payment capabilities of the Personal Router are well developed, this device will be capable not only of buying bandwidth and network access, but also of buying any good available.

Finally, another important aspect of the Personal Router is the “bribe.” By providing access, it is possible to stimulate a business. It could even be for free (“come buy a pizza in my pizza shop, you’ll get 15 minutes of free access”).

These three models are particularly interesting for convenience stores. They could provide “hot spots” of high bandwidth Internet access, could easily take advantage of a new way of paying electronically, and could stimulate their business by having cars, for instance, coming to their gas station because drivers can have high speed Internet access while refueling.

2.3.3 Definitions

Four possible entities may be involved:

- The user is the owner of the Personal Router, trying to access wireless services through his Personal Router.
- The provider is the operator of the Personal Router Base Station (PRBS). It can be either an individual or a company operating several PRBS. Another entity could also be included in these models: the upstream provider, which provides the upstream communication link to the provider.
- A third party is a distinct party involved in the transaction.
- The broker is a third party that may be needed, generally to participate in the exchange of money.

Three different links are considered in these simple models: Authentication, Payment and Relationship.

- A payment is a transfer of money between two entities
- An authorization may be needed to ensure that a user has the right to access a service of other entities. Credentials may be a form of authorization.
- A relationship is a particular association between two entities that goes beyond the communication itself (e.g. employee, hotel guest, etc)

2.3.4 Direct transaction between the user and the provider

In this scheme, the user is doing a transaction with the provider only. Figure 2-2 shows the three different possibilities.

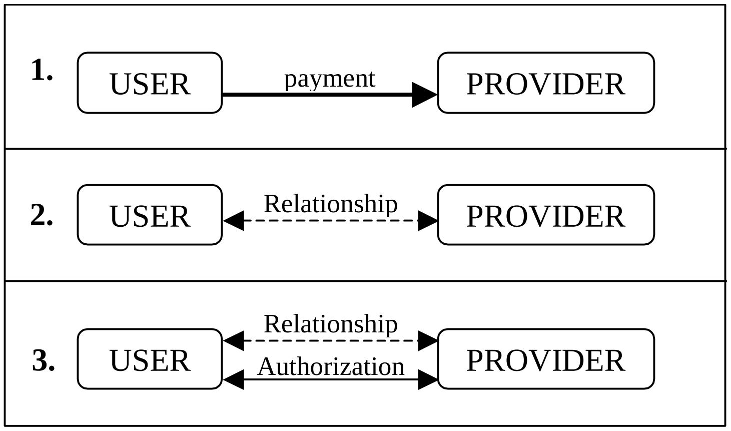

Figure 2-2: Direct transaction between the user and the provider

Direct payment

The first possibility is that the user pays directly to the provider. A third party may be involved, only if it is necessary to conduct the payment; it depends on the payment scheme. These payments may be done for example through micropayment schemes, or by a long-term relationship with the provider.

Simple relationship

Another option is when the user can have a simple relationship with the provider. For instance, a user could get a free wireless services just for being inside a mall.

Special Relationship

Finally, the user can use her company's wireless network, a personal Airport station at home or any other private network. Authentication may be used to control access to the network; however, no payment is required.

2.3.5 Transactions through brokers

Possible roles of the broker

The broker can play two different roles in a transaction.

Relationships

The first value of a broker is to bring in value added by relationships. Effectively, brokers are more likely to be already engaged in a relationship with different parties than a single player in the market. They are therefore more likely to successfully conduct transactions with parties that do not know each other.

Aggregator

Another important role of a broker, coming from the fact that they have a large number of relationships with players in the market, is to aggregate payment from different users and directed to different providers.

This results in a transfer of risk from the provider to the broker; in effect, the broker now takes on the risk that the user may not pay or may make fraudulent payments.

Aggregation also allows the broker to benefit from strong economies of scale for billing. This role is very close to the one of credit card companies.

Aggregation is a source for profit, because aggregators are able to get a margin from their role in the market. For instance, they can buy large amounts of connectivity wholesale and sell it retail. They also benefit from economies of scale, because they are able to set up a more efficient service, able to process a large number of transactions, than individual players in the market.

Simplified model: one broker

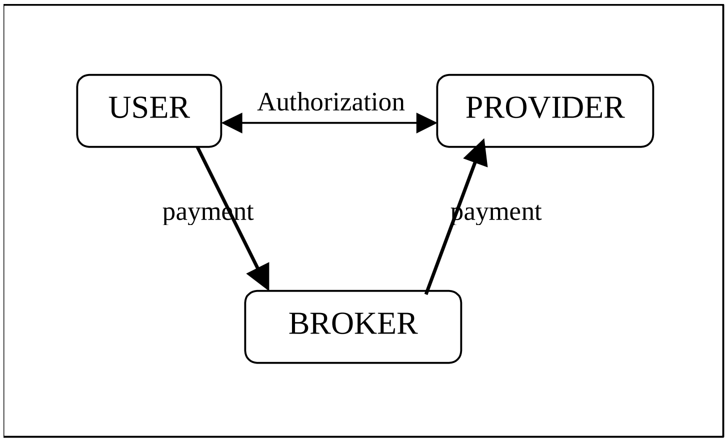

Figure 2-3: Simplified model: one broker

As shown in Figure 2-3, the user pays the broker, who then pays the provider. If desired, blinding techniques can be used to make these payments anonymous and untraceable so that neither the provider nor the broker can track the user.

The relationship between the user and the broker or between the broker and the provider can be either short term or long term.
If it is a short-term relationship, both parties are connected only for the duration of the communication. Payments can be settled by micropayments.
If it is a longer-term relationship, from a few hours to years, parties are bound by some kind of contract (e.g. subscription, pre-paid communications, credit card billing, etc).
The broker is aggregating usage and therefore is transferring some part of the financial risk from the user and the provider to itself.

This model is not dependent on the payment system. Payments could be made off-line: for instance, the user can pay the broker through a monthly billing, the provider can be credited on a monthly basis, and the fund transfer can be aggregated so that the payments are not too small. In this case, there is no need to design a specific payment scheme.

In many senses, this scheme is very close to credit card transactions: the user is making a payment to the provider through a broker, whose role is close to the role of Visa or MasterCard in a credit card transaction.
It is also very close to roaming agreements between mobile operators: The provider is the distant operator, providing services, and the contracted operator is acting like a broker, providing only financial services.

Several brokers

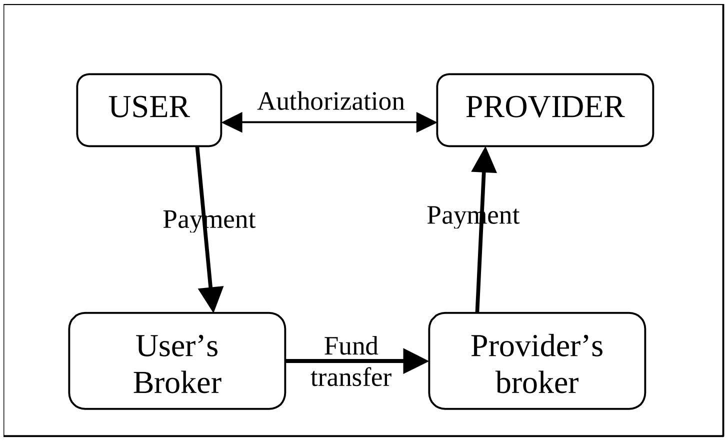

Figure 2-4: Several brokers

More generally, we can consider a network of brokers connected to one another, as shown in Figure 2-4. The exchanges of money works exactly as in the previous model. In this scheme, there are an infinite number of providers and an infinite number of brokers.

However, if there are too many brokers, they will not be able to aggregate enough payments to make profitable money transfers.

2.3.6 Bribes

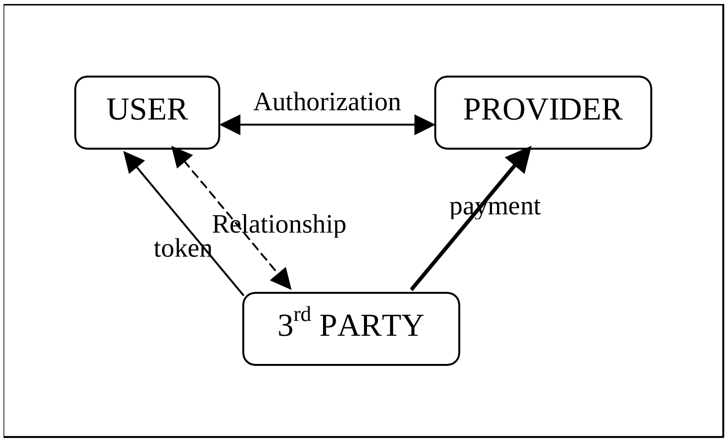

Figure 2-5: Bribes

In the case of bribes, as shown in Figure 2-5, the user is involved into some kind of relationship with a 3rd party. For instance, the user may have bought a coffee in a Starbucks Coffee, and get 20 minutes of free wireless access.

The 3rd party, Starbucks in this example, has to pay back the communication provider.

Another possibility (which is not really a “bribe” model anymore) is that the user is an employee of a company providing a wireless network to its employee through a distinct provider. The company does not have to install and maintain a wireless network, but it has to pay the provider for the services.

2.3.7 Distinct groups of users

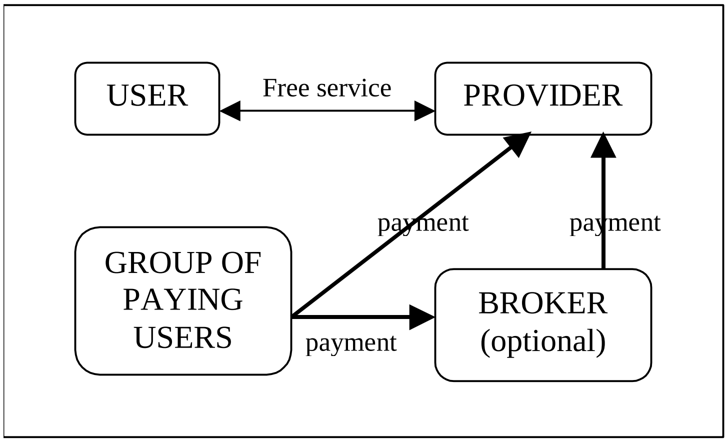

Figure 2-6: Distinct group of users

A group of user may pay to get services in an area. This model is described on Figure 2-6. For example, a neighborhood association could pay for services all around the neighborhood, and let anyone in this place access it without paying. Another example is the Advanced Network Architecture group who installed (and paid for) an 802.11b wireless network in the Laboratory for Computer Science building, and lets anyone in the building access it.

This scheme could also provide a special service to those who paid (e.g. faster or more reliable service versus best effort service). We could include a broker between the paying users and the provider, or the provider could actually be the paying users, who installed and manages the network themselves, as in the example of the ANA network.

2.3.8 Summary: transaction models

These different models provide an insight of the wide spectrum of transaction models possible with the Personal Router. Although this list covers many different types of models, it is likely that new unpredictable models will emerge as the device develops. These models go from simple models, where the user only interacts with the provider, to more elaborate models involving several other parties such as brokers and other types of third parties.

We can see from these models that payments will be only a small part of all information flows that will occur through the Personal Router. It is important to consider these possible models when using payment systems in the Personal Router, to avoid blocking possibilities of other transaction models.

2.4 Summary

Payment systems are very complex procedures that are difficult to describe. In this chapter, Definitions of the actors and the characteristics involved in payment systems have been given. The different axes along which it is possible to describe payment systems have been presented.

One of the most important characteristics is privacy. It is also one of the more complex to represent. A table is used to describe privacy's two main dimensions: to whom the information is revealed, and what kind of information is considered.

The different values that a payment scheme can take along these axes define its different characteristics, performances and behaviors.

To understand to complexity of the world surrounding the Personal Router, different transaction models have also been exposed. When implementing new payment system in the Personal Router, we must be careful not to rule out any possible payment system that could be needed in the future.

3 PRIVACY

Privacy is a key issue raised by the Personal Router. In effect, the Personal Router combines three worlds: mobile communications, electronic transactions and the Internet. Each of these different layers raises separate issues. The aim of this chapter is to give a background in privacy in the context of the Personal Router. Some of these issues are the same as in the Internet or in mobile networks today. Personal router users, as well as mobile communication users, will face three main threats. First, they will access networks through mobile network operator, who will have the ability to learn a lot of information about their customers, and in particular their location. Then users will go over insecure networks, and will possibly leave a trail. Finally, they will transact over these networks with untrusted providers, thus raising privacy threats again.

Others issues will be more specific to the Personal Router, such as the difficulty that service providers are unknown, untrusted and numerous compared to today's structure of mobile communication providers: most users have a contract with only one of them. It is much more difficult to make sure that all providers are complying with their privacy agreement if they are a lot of providers in the system as it is in the case of the Personal Router. Also, the Personal Router will require constant negotiation of payment system and services, which is likely to leave a trail easier to follow than in usual communication systems.

Most of these interactions will happen without the user knowing it or even initiating it. For instance, the device could try to check and download emails; it will begin by looking for providers, negotiating with them, selecting one of them, paying and downloading emails.

The information released can even be a part of the negotiation. But the user has to be careful not to leak information on any of these three parts. If there is a leak, and if other parties can correlate the information, then privacy is compromised in all three parts.

In this chapter, we will first define privacy and give a quick background over some regulatory aspects of privacy. Then we will go through privacy issues on the web, in electronic commerce and in mobile communications, and through some examples that have been tried to solve these problems.

In chapter 4, we will consider only privacy from the payment system standpoint. This chapter draws the larger picture of privacy issues.

3.1 Definitions

3.1.1 A definition of privacy

Privacy is the individuals' right to manage data held by third parties about them, to add, modify, or delete entries, and to know and control what use is made of this information.

Personal information is any information related to an individual. It can be a name, an address, a phone number, or a national identification number. It can also be a pseudonym or any information that could allow third parties to create a profile of this particular user. These parties can record the preferences of this particular user, like the kind of web sites she likes and the kind of payment she uses. They could also record the general behavior of this user, like where she likes to get information on a product, on what kind of banner she clicks, what kind of advertising she is sensitive to.

Privacy

Leakage of information occurs when information that should be kept private is released and can be obtained by a third party. It is different from information willingly given to a third party.

As Agre and Rotenberg explain in (Agre and Rotenberg, 1997), “the debates about privacy are structured as a series of tradeoffs and by the assertion of abstract rights (dignity, autonomy, association, self determination) against specific encroachments (each accompanied by a compelling justification).”

These abstract rights are very difficult to define in words, and even more difficult to define for a computer program. Moreover, they differ depending on the location, cultural environment, context, time, etc. For instance, health data are private, except for doctors and health insurances, and a worker may hide his salary in certain countries but not in others, depending on cultural context. Parents may not want to give any information about their children, except to school officials or doctors.

Protecting privacy requires also a very good definition of the needs. For example, a pub does not need the name, the address, and the age of a customer to serve alcohol: it needs only to be sure that this customer is over 21.

The problem of privacy is even more serious with new technologies. Users are not necessarily aware of information leakage because it is more difficult to see. They do not understand what information they are conveying, how permanent it is, who can access it, and what the intentions of those using this information might be. Users cannot see the trail they leave when they apply for services, make payments, browse the Internet.

The problem is to know whether privacy can be efficiently protected, despite several articles such as (Meeks, 1999) claiming “privacy in the digital age is dead”, or Sun Microsystems CEO Scott McNealy’s provocative answer to a journalist in 1999: “You have zero privacy anyway. Get over it.”

3.1.2 Stakeholders

As explained in (Camp, 1999), parties in e-commerce and privacy systems have different goals, which are often incompatible. Some of these parties are much more interested in what the others do in specific cases. For instance, as soon as money is exchanged, governments have a special interest in the transaction.

Law enforcement

Government needs information to accomplish its legitimate purposes such as detection of illegal transactions. They fight fraudulent activities, like money laundering, terrorist networks, etc. At the same time, they must protect individuals' right to privacy. Increasing data availability makes it easier to detect patterns of illegal activity and pursue the appropriate parties.

Businesses

Companies collect information for uses both primary (their own) and secondary (somebody else's). They have the ability to record information about users, or to ask them for personal data. Users do not always benefit directly from these practices. Customers prefer a personalized service, but they could also value privacy. Companies are not always clear about their intents.

Businesses are also motivated to record profiles of user because of a profitable secondary market for consumer information.

System designers and administrators

They need to collect detailed information on system usage to tune and improve it as much as possible. But that information can also be used to spy or invade the privacy of a web user.

An observer

An observer may be spying the communication and/or the transaction between a client and a server, either to gain access to hidden data, to get private data (what the customer is buying, at what price, what is the customer's name, etc), or to get sensitive information such as a credit card number. That is why transactions have to be protected, for instance by encryption.

Banks

In most cases, banks are the final step to the completion of a transaction. They can often access to sensitive data over their customer, which they should not know. Nevertheless, banks must know a minimum amount of information on the transaction in order to be able to process the transfer of money and avoid money laundering.

Users

Users would like to preserve their privacy. At the same time, they appreciate to receive a personalized service. Moreover, their behavior is not always rational. Many surveys have shown that privacy is one of their top concerns when buying something on the Internet[2]; yet they seldom take action to fight leaks of privacy, and are often not even aware of these leaks.

Summary of conflicts between players

Businesses and system designers would like to know as much information as possible about users. This allows them to provide personalized service and targeted advertising, and also to design their systems according to predicted users' behavior.

Governments and banks need to control some information about transactions in order to be able to respond to warrants and to detect illegal patterns.

Observers do not really need to know anything, but they can try to get their hands on private information that could let them access payment tools on behalf of users and to spy on other users.

Users would prefer to keep their personal information as private as possible. They can decide to trust one or more of the other parties.

[2] See for instance March 2000 Business Week/Harris Poll
http://businessweek.com/datedtoc/2000/0012.htm

3.2 Background: Privacy in the Law

The Personal Router is confronted by a complex background in Privacy laws. This section gives an overview of several laws in the US and in Europe, and their interconnection via the Safe Harbor agreement. The aim of this section is not to be extensive on this difficult field, but to point out the complexity of regulations on privacy. A comprehensive collection of US and international privacy laws can be found in (Rotenberg, 2000).

There are strong disagreements and differences of point of view between the United States and the European Union on how to manage citizens' privacy. A good analysis of these divergences can be found in (Agre and Rotenberg, 1997) and in (Rotenberg, 2000). In the US, the debate centers on privacy concerns, whereas in the European Community the debate concerns data protection. In effect, European regulators assume that they have to protect citizens against bad behaviors of corporations; while in the US it is the government that is considered as being likely to act badly. Consequently, U.S. regulations focus more on governmental agencies' privacy policies, while European directives are more aimed at misuse of personal information by companies.

In particular, in the US, once a person freely chooses to disclose information for a specific purpose, this information is no longer considered private and can be disclosed generally (secondary use), while the European Union regulation prevents secondary use of data. Information collected for a specific purpose cannot be used for another purpose without authorization.

3.2.1 Privacy in the U.S. constitution

The first thing to be noted is that the word "privacy" does not appear anywhere in the US constitution. There is no real definition in the constitution of what privacy is and implies.

The first and the fourth amendment

The 1st and the 4th amendment are the most relevant to privacy issues. The 1st amendment ensures the meeting right, while the 4th amendment forbids unreasonable searches.

The first amendment states:

> "Congress shall make no law respecting an establishment of religion, or prohibiting the free exercise thereof; or abridging the freedom of speech, or of the press; or the right of the people peaceably to assemble, and to petition the government for a redress of grievances."

The first amendment implies privacy, as people under surveillance are not likely to assemble and to express views that could be disapproved.

The fourth amendment states:

> "The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no warrants shall issue, but upon probable cause, supported by oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized."

The fourth amendment creates a region of privacy inviolable by government except in constrained circumstances.

3.2.2 The Privacy Act of 1974

The Privacy Act of 1974[3] is the first US text to explicitly address privacy issues. It sets fair information principles, and regulates federal government agency record keeping and disclosure practices. The Act allows most individuals to seek access to federal agency records about themselves. There should be no secret record keeping databases. All records must be kept accurate, complete, timely, and must be relevant to the purpose for which they are to be used. There should be limits on the collection of personal

[3] http://www.usdoj.gov/04foia/privstat.htm ; Public Law 93-579

information. Information collected for one purpose may not be used for another purpose without notice to or the consent of the subject of the record. Finally, the record-keeper should assure a reasonable level of security, and is accountable for compliance with the other principles.

An important amendment was made in 1988 to regulate the “matching” of personal data among different databases coming from different agencies tying together their computer systems. It is possible to do it only if a number of conditions are fulfilled, and cannot remain in effect more than 18 months.

There are also several sector-specific privacy regulations, for instance the Right to Financial Privacy Act (1978)[4], providing confidentiality for financial records, the Electronic Communications Privacy Act (1986)[5], regulating in particular wiretapping, the Video Privacy Protection Act (1988)[6], dealing with video rentals records, the Telephone Consumer Protection Act (1991)[7], restricting use of automatic telephone dialing systems, the Driver’s Privacy Protection Act (1994)[8], prohibiting the release and use of certain personal information from State motor vehicle records, or the Children’s Online Privacy Act (1999)[9], regulating practices in connection with the collection and use of personal information about children on the Internet.

Despite numerous proposed legislative initiatives, there has been no general framework in the US since 1974.

3.2.3 Privacy in Europe

Evolution of privacy policies in Europe since 1970

The following description of this evolution is inspired from an analysis by Viktor Mayer-Schönberger in the 8th chapter of (Agre and Rotenberg, 1997).

[4] Public Law 95-630
[5] Public Law 99-508
[6] Public Law 100-618
[7] Public Law 102-243
[8] Public Law 103-322
[9] Public Law 105-277

The first generation data-protection laws originated in 1970s to address the fear of Big Brother, that is a single centralized data bank that knows everything about everybody. Special institutions were set up to supervise compliance with substantive data-protection regulation. The right to access and the right to modify one's personal data were introduced.

The second generation, at the end of the 70s, comes from a different danger: the fear is not a potential Big Brother anymore, but lies now in dispersed data processing by thousands of computers across the world. Data protection in the second generation focused more on individual privacy rights of the citizen: right to be left alone, right to delimit one's own personal life (telemarketing), etc.

The third generation, in the 1980s, addresses a different question. It is not whether one wanted to participate in societal processes, but how. This guarantees the ability of the individual, to decide in general for himself the release and use of his own personal data.

The final generation (early 1990s) tries to equalize bargaining position by strengthening the individual's position vis-à-vis the generally more powerful information-gathering institutions. Moreover, legislators have subjected former data-protection norms to mandatory legal protection. This approach reflects the understanding that some areas of informational privacy must be absolutely protected and cannot be bargained for individually.

The 1995 European Union Directive on Data Protection reflects this general evolution.

The 1995 European Union Directive on Data Protection

As several member states of the European Union have since the 1970s passed legislation protecting the rights of individuals to privacy from abuses resulting from the processing

of personal data, Directive 95/46/EC[10] asked members states of the EU to harmonize protection of the right to privacy by the end of 1998.
These laws apply to any identified or identifiable individual (name, telephone number, national identification number, etc).

Data protection laws provide for a series of rights for individuals such as the right to receive certain information whenever data are collected, the right to access the data, and if necessary, the right to have the data corrected, and the right to object to certain types of data processing. Moreover, these laws includes the obligation to use personal data for specified, explicit and legitimate purposes only, the obligation to guarantee the security of the data against accidental or unauthorized access or manipulation and in some cases the obligation to notify a specific independent supervisory body before carrying out all or certain types of data processing operations. Data should be accurate and kept up to date. Data controllers are required to take any reasonable step to ensure the rectification or erasure of inaccurate data. Finally, data should be kept in a form that permits identification of individuals for no longer than it is necessary.

Stricter rules apply for sensitive data, such as ethnic origin, political opinions, religious or philosophical beliefs, union membership and data concerning health or sex life.

These laws normally provide for certain safeguards or special procedures to be applied in case of transfers of data outside the EU. They prohibit the transfer of personal data to non-European Union nations that do not meet the European standard for privacy protection.

The United States is one of these countries. The U.S. Department of Commerce in consultation with the European Commission developed a "safe harbor" framework, which

[10] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data Official Journal L 281, 23/11/1995 p. 0031 – 0050
http://europa.eu.int/eur-lex/en/lif/dat/1995/en_395L0046.html

was approved by the EU in July 2000[11]. US companies can be certified by applying Safe Harbor Principles to their privacy policy and personal data processing. Certified US companies avoid interruptions in their business with the EU and/or prosecution by European authorities under European privacy laws. About 160 US companies have signed up so far (as of March 2002).

3.3 Privacy issues raised by the Personal Router

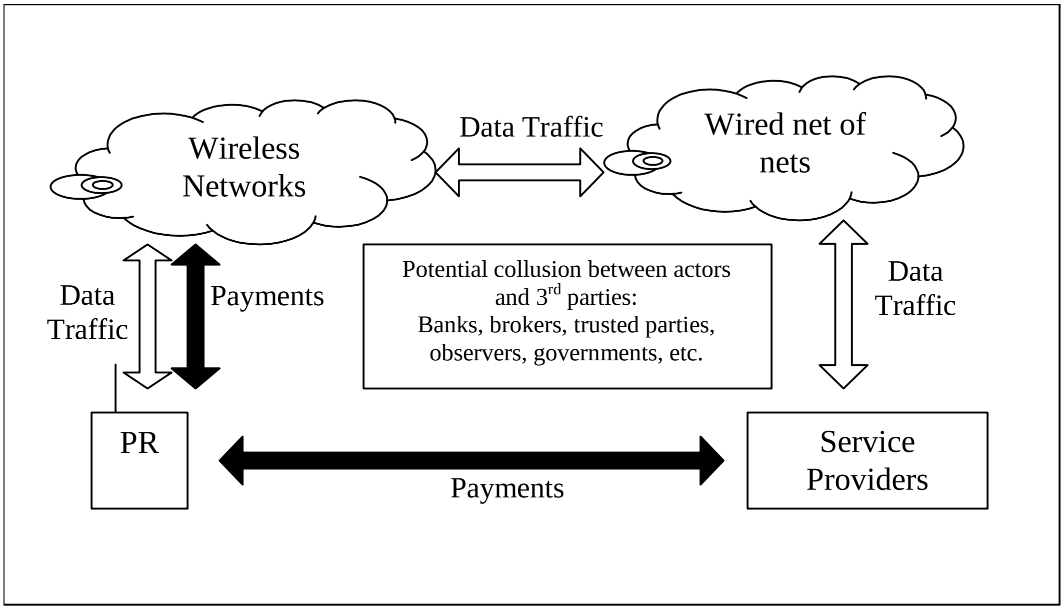
Wireless Networks
Data Traffic
Wired net of nets
Data Traffic
Payments
Potential collusion between actors and 3rd parties: Banks, brokers, trusted parties, observers, governments, etc.
Data Traffic
PR
Payments
Service Providers

Figure 3-1: Different levels of interaction with the Personal Router

The Personal Router raises numerous difficulties regarding privacy. As shown in Figure 3-1, it interacts with three different levels: The mobile network, the Internet and the service providers. It is important to observe that these threats are neither new nor different than what exists today. The information released in each level depends on the level; it can be either location information at the level of the mobile network, personal information at the level of Internet and service providers, and payment-specific information at the levels of mobile networks and service providers. The additional issue raised by the Personal Router lies in the fact that the three levels are stimulated at the same time and get information on users. The danger occurs when it becomes possible for

[11] US site on Safe Harbor (department of Commerce) http://www.export.gov/safeharbor/
List of companies adhering to the Safe Harbor framework:
http://web.ita.doc.gov/safeharbor/shlist.nsf/webPages/safe+harbor+list

the parties to collude and to merge the partial information they have to create a complete profile of the user. These dangers already exist in 3G networks. Nevertheless, as explained in section 3.3.3, it is complicated by the fact that there are many mobile access providers.

In this section, we will go over three threats faced by the Personal Router. First, we will see the issue of the trail left by users when going over insecure networks. Then we will examine the threat coming from transacting with untrusted service providers. Finally, we will analyze the threat coming from the fact that users will be mobile and use wireless networks.

3.3.1 Threat of leaving a trail over insecure networks

The Internet has revolutionized the information world and exchange practices. New techniques, such as data-mining or matching databases, are now not only possible, but also cheap and easy to perform. Collecting data is easier than ever. The profiling of potential customers has always existed. Nevertheless, it has grown to an unknown dimension with the development and the generalization of the use of Internet. The Personal Router faces similar kinds of issues. Users will benefit from tools already existing in the world of the World Wide Web. In this section, we will go over specific issues relative to the web and some examples that have been applied to solve them.

Profiling of users

There are two types of behavior from companies trying to get personal information on users. The first one is when users reveal some part of personal information and the companies do not access anything else. The second one is when users do not actively reveal personal information but companies find out ways to obtain this information anyway.

Information can therefore be collected in two different forms. The first one is profiling. Profiling is when users are observed and monitored, and information about them is collected only by passively recording their actions. On the other hand, personalization is when users actively set their preferences and provide information to third parties.

Depending on the use of information collected, different parties could benefit from these data collections. In some cases, users can get more personalized services from the information they gave away. In other, companies use, rent or sell the data collected as valuable customer data without much compensation for users.

Users are now scrutinized in depth. Companies such as DoubleClick are able to tell what kind of sites a user is likely to visit, what kind of article she is likely to buy, what kind of links she may click on (in terms of colors, location on the web page, objects advertised) and what are her main interests.

Moreover, auction sites like eBay.com are able to build a very large database on users, revealing for instance reservation prices for many different goods, which is highly valuable information.

Given these databases, advertising can be optimized and personalized. Users get a benefit, because they receive advertisements much more personalized, and therefore better directed to their needs and wishes. However, this profiling may be too expensive for them in terms of invasion of their privacy, compared to the services provided.

The problem comes from the fact that users cannot entirely decide what information is leaked and who gets it. In an ideal world, either there would be no leakage, or the information released would be revealed in agreement with the user. This goal is very difficult to achieve, because users leave an easily traceable trail, and companies are not always clear about their intentions. Several tools try or have tried to help users solve this issue.

This information can be revealed by two kinds of tools: tools acting at the IP and HTTP levels and cookies.

The IP/HTML level

When a computer makes requests on the Internet in HTML, it gives to web sites a lot of personal information[12] such as IP address, OS, web browser type, plug-ins available,

[12] See http://www.privacy.net to check all the information potentially collected by web sites

whether Java is enabled, the size of the screen, domain name, computer name, where users prefer to click, what they like, from what site they are coming, etc.

This information can be retrieved and stored by any web site without the explicit knowledge or consent of the user. The user is not actively releasing this information, and unwillingly leaves a trail when surfing on the Web. Customizing web browsers and checking the personal information released can reduce the amount of data available. Nevertheless, it will not block everything, and requires rather strong computer skills.

Cookies

Cookies are very small pieces of information left by web sites on the hard drive of individual users. They can store personal information such as name, address, credit card number, and/or they can simply store a user ID used to personalize a user visit to a website.

They are three main uses of cookies. First, targeted marketing allows sites to build a profile on where individuals go while on the Internet and on their primary interests. The second use is web tracking, which means spying the users' behavior on a particular web site and counting exactly how many people have visited a site. Finally, site personalization allows users to define preferences, for instance on news, areas of interest, or presentation of information. Only the third use is really useful for the user.

Today's browsers offer simplistic ways of dealing with cookies. It is possible to either turn them off, restrict them to the owner site, or to turn them on. But this requires to understand what cookies are and what they are used to. In addition, some sites condition access to the use of cookies. Users have then the choice between turning on cookies and no accessing certain web sites.

Fighting information leakage

There are numerous solutions to fight information leakage on the web. They range from basic anonymous web browsing to advanced pseudonymous services. A description of five typical services researched or currently offered on the Internet follows. These services have been chosen to show the spectrum of possible privacy tools accessible.

These different models correspond to a new potential market: the market of "Privacy Service Providers" or PSPs. They are classified into three categories:

- Negotiation services (e.g. P3P), allowing a web site and a user to agree on private information stored and used by the web site. The privacy threat is solved by negotiation and trust that parties will respect the agreement concluded at the end of this negotiation.
- Basic anonymous services (e.g. anonymizer.com), providing a simple way to be anonymous, but without personalized services. In particular, these services systematically block cookies, and block direct access to users' computer via IP and HTML.
- Advanced services (e.g. zeroknowledge.com's freedom), providing anonymity through tailor-designed pseudonymity, and allowing users to get personalized services. Cookies can be accepted in certain cases, and some information can be passed to the distant web site via HTML. But these services provide strong pseudonymity, and can change the apparent identity of users very often. Therefore they can let users access personalized services without much information leakage.

Negotiation services: P3P

The Platform for Privacy Preferences (P3P) Project[13] is developed by the World Wide Web Consortium[14]. The aim is to provide a simple and automated way for users to gain more control over the use of personal information on Web sites they visit. P3P is a description of privacy policies and preference in a machine-readable format.

On the one hand, the web site visited presents a set of propositions on how it handles personal information about its users. On the other hand, the user defines a set of privacy preferences. A P3P-enabled web browser compares the proposition of the web site to the preferences of the user, and decides whether there are compatible.

However, there is no enforcement, and the web site does not have to act as declared; so the user must trust that the web site is telling the truth.

[13] http://www.w3.org/P3P/

[14] http://www.w3.org

Basic anonymous services

Basic anonymous services use anonymity as a strategy for protecting privacy. They usually offer little or no services apart from providing anonymity. Two services are presented here: anonymizer.com, providing a proxy service, and crowds, a research project developed by AT&T.

Anonymizer.com

Anonymizer.com[15] blocks cookies, Java, JavaScript, and other tracking methods. It changes URLs on web pages to make sure links go through its servers.
The user can then browse the Internet without giving information to any web site he is visiting. Anonymizer.com makes the requests to the requested server for the user, and then redirects replies to the user.

Nevertheless, the transmission between the anonymizer.com server and the user is not secure, and eavesdropping or wire-tapping on the user's line is possible. Moreover, it is only a proxy service; the server knows perfectly who users are, and what they are doing. Moreover, Anonymizer.com is not designed to support private transactions such as electronic payments.

Crowds

Crowds[16] was developed by AT&T in the mid 90's as a research project. It was used only on a small scale. This is a good example of a distributed peer-to-peer privacy enhancer. The approach is based on the idea of hiding one's actions within the actions of many others. To execute web transactions in this model, a user first joins a crowd of other users. The user's initial request to a web server is first passed to a random member of the crowd. That member can either submit the request directly to the end server or forward it to another randomly chosen member, and in the latter case the next member independently chooses to forward or submit the request. When the request is eventually submitted, it is submitted by a random member, thus preventing the end server from identifying its true initiator. Even crowd members cannot identify the initiator of the

[15] http://www.anonymizer.com
[16] http://www.research.att.com/projects/crowds/

request, since the initiator is indistinguishable from a member that simply passed on a request from another.

There are two major privacy and security issues.
The first risk is that each computer sends requests that the user did not initiate, which may raise some problems. A second risk of running Crowds is that by routing requests through others’ machines, Crowds may increase the risk of disclosure of the data in the request and corresponding reply.

This last issue implies that this system is intended more for browsing than shopping. In fact, this system completely fails if it is used for shopping, as all the information involved in the transaction (credit card number, address, name, etc) would be known by every member forwarding the request (Crowds doesn’t support encryption).

Advanced services: pseudonyms

More advanced services use pseudonymity instead of pure anonymity. This allows the user to get personalized services without revealing personal information. The user can use different pseudonyms while browsing on the Internet, which cannot be connected to each other. Services considered here are LPWA developed by Lucent Technologies as a research project, and Freedom.net, a virtual private network providing privacy developed by zeroknowledge.com.

LPWA

The Lucent Personalized Web Assistant[17] (LPWA) was a technology demonstration by Lucent Technologies. This project has now ended. It was created in 1997.
The aim of LPWA was to provide personalized and anonymous access to web servers, including emails. It would allow users to get personalized services on web sites like Yahoo!, and get emailed newsletter without giving personal information. It uses aliases. An alias is a username, a password and an email address. Different aliases are presented to different web sites, making aggregating data impossible. They cannot be correlated.

[17] http://www.bell-labs.com/project/lpwa

These aliases are secure, consistent, and are not stored on LPWA's servers.

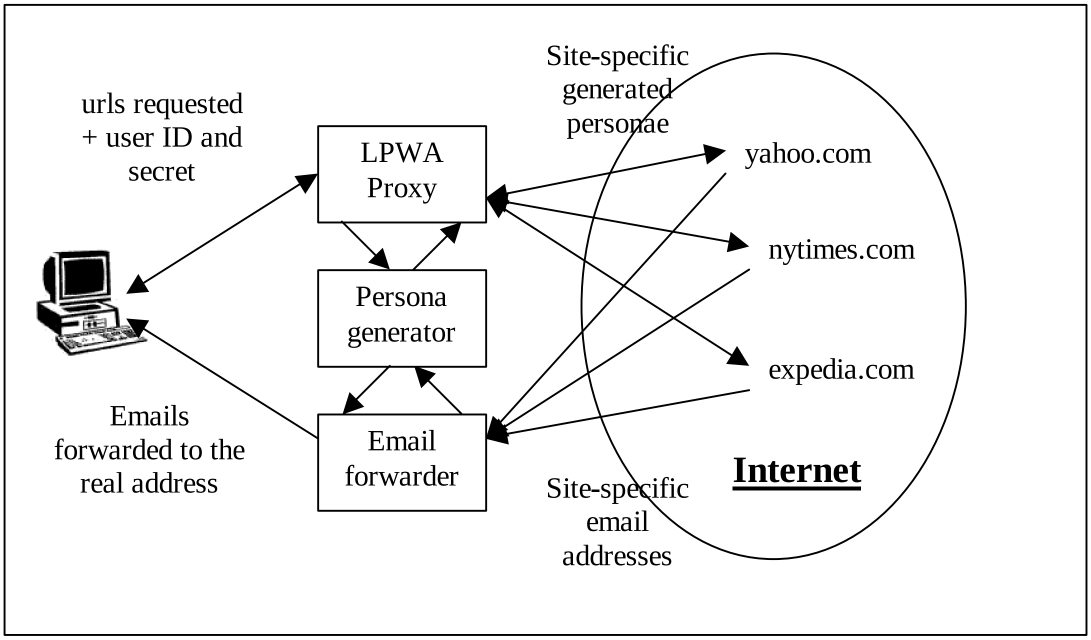
urls requested + user ID and secret
LPWA Proxy
Persona generator
Email forwarder
Emails forwarded to the real address
Site-specific generated personae
Site-specific email addresses
yahoo.com
nytimes.com
expedia.com
Internet

Figure 3-2: A description of the LPWA scheme

As shown in Figure 3-2, LPWA consists of three functional components: The persona generator, the browsing proxy, and the email forwarder. The persona generator generates a unique, consistent site-specific persona on demand by a user. The user provides a real email address and a secret. Using a cryptographic operation with the user's email address and secret, and the destination web site domain name, the persona generator computes a persona for this user, which is consistent to this web site and does not need to be stored on the server; in fact, it is recomputed each time the user access the web site.

The browsing proxy indirects connections on the TCP level and filters headers on the HTTP level. The email forwarder forwards mails addressed to a LPWA persona to the real user.

When a web site asks a user for her username, password or email address, the user simply types the appropriate escape sequence (\u for username, \p for password, and \@ for email address). The LPWA proxy then recreates the user's persona for this particular web site, and replaces escape sequence by the correspondent element.

This system does not allow users to conduct anonymous transactions. It provides only highly personalized services and anonymity. Personalization and anonymity are then reconciled.

Freedom

Zero Knowledge[18] is a Canadian company working on privacy issues. It used to provide a free piece of software, Freedom, and to sell pseudonymous identities to be used with this software. Zero Knowledge has now changed its business, and sells privacy tools to companies and different web protecting tools and personal firewalls to individuals.

Freedom[19] protects privacy by proxying the various supported protocols, and sending those proxied packets through a private network before they are deposited on the Internet for normal service. This private network is operated by Zero Knowledge and consists of a set of Freedom server nodes that make up the Freedom Network, and the Freedom Core Servers that provide basic services. The Freedom network is a virtual network using encrypted tunnels over public Internet. Server nodes are either operated by Zero Knowledge or by third party operators running a Freedom daemon on their servers. The network transports encrypted IP traffic from one node to the next. The number of nodes used in a route is chosen by the user by setting her security level in the Freedom client. The server nodes themselves are not linked by a fixed topology, instead, they can communicate with any other server node on the network, as requested by a client when creating a route. Freedom also prevents server operators from knowing who is using Freedom and what messages say. In effect, link encryption is applied between node pairs in order to hide the nature and characteristics of the traffic between them. Nevertheless, increasing the number of servers used in a route raises issues of latency and scalability. Nyms are the identities (pseudonyms) that Freedom users assume on the Internet. A nym is defined an identity (e.g. John Smith), by a unique email address at freedom.net and the associated digital signature key.

[18] http://www.zeroknowledge.com
[19] http://www.freedom.net

A nym is created when a user sends a nym creation token, a signature verification key and an encryption public key to the nym server. These can be purchased independently to ensure untraceability. The Freedom network is supposed to be unable to trace back nyms. Incoming mail for nyms is received by a freedom mail gateway. Then they are encrypted and delivered to the nym user through the Freedom network.

Private transactions are not supported, but it provides secure channels such as SSL.

Conclusions: Privacy on the Web

There are real privacy issues on the Internet today. They have roots in personalization and profiling. Both are necessary to provide the consumer with a tailored service. Nevertheless, hidden costs paid by users in terms of privacy invasion may be considered as too high compared to the benefits. Information leaking out of each click users make on the Internet is a real asset for companies.
That is why many start-ups were created on this potential privacy market, trying to sell products protecting users' privacy. Almost all of them have failed or have changed their business, like Zero Knowledge, the most promising of them, did.
Different surveys show that Internet users are still very concerned about their privacy. For instance, a March 2000 Business Week/Harris Poll[20] survey showed that users feel privacy invasion as a growing threat over the Internet. 78% of the users who shop and 94% of those who do not are somewhat or very concerned about their privacy online.

Nevertheless, the failure of most the privacy enhancing technologies show either that the right way to do it has not been found yet, or that most consumers are not willing to pay to protect their privacy.

3.3.2 Threats in electronic transactions

Transacting over a network and over Internet in particular raises several issues that are close to ones encountered by the Personal Router. However, the Personal Router raises issues that are slightly different. In effect, the Personal Router e-commerce transactions

[20] http://businessweek.com/datedtoc/2000/0012.htm

will often be limited to a simple payment for connectivity, which does not carry all the burden of conducting a transaction for a real good over the Internet. In effect, there is a big difference between transaction for tangible goods (e.g. buy a book) and intangible goods (e.g. network services). Much more information is needed to complete a transaction on tangible goods, such as name, address, and usually phone number.
The difficulty of e-commerce comes from the fact that in addition to all the information exchanged before and during the transaction, financial data is exchanged. This is a very sensitive piece of information, which implies a special treatment.

Electronic commerce includes electronic payment. Electronic payments are designed and optimized for a specific use. They also have different privacy characteristics. Privacy has a cost, as it implies implementing additional computations and interactions between the different entities taking part in the transaction. Therefore, electronic payment schemes offer usually little or no privacy if the amount to be transferred is too small (e.g. micro-payment schemes such as Millicent or MicroMint). The marginal cost of the transaction is too high compared to the amount to be transferred.

On the other hand, DigiCash[21] is totally anonymous and uses blinding techniques to provide this anonymity. Nevertheless, this anonymity may raise issues, as the transaction is not consistent; for instance, if there is a system failure in the middle of the transfer of the product sold, the merchant has no way to recognize the real buyer. There is no mechanism for conflict resolution[22].

Privacy is therefore a sensitive issue in commerce. The Personal Router will face the same kind of problems. However, they will be simplified most of the time by the fact that the good bought is connectivity, which is much easier to sell privately than real goods. In effect, there is no shipping, thus no need to exchange shipping specific information such as name or address.

[21] http://www.digicash.com

[22] Nevertheless, L. Jean Camp, Michael Harkavy, Bennet Yee, and J. D. Tygar, in (Camp, et al, 1996), have proven that it is possible to create an anonymous payment system while providing strong ACID (atomic, consistent, isolated, durable) transactional properties.

In this section, two very different approaches to privacy in electronic commerce are discussed. The first one is Microsoft Passport, who is collaborating and trusting third parties. This approach is sometimes called Personal Data Provider. The other approach is a scheme to conduct anonymous transaction over the Internet with real goods.

Microsoft Passport

Microsoft Passport[23] is an example of a personal information manager. Microsoft passport offers two services. The first one is a "single sign-in" service, which enables users to have one name and password at all participating websites. This is the .NET passport. The second service is the ".NET Passport express purchase." The user stores sensitive personal information such as personal address, credit card number, etc in a ".NET Passport Wallet." This service provides a faster checkout by using the information stored once for all in this .NET passport Wallet.

This approach let Microsoft manage all personal information of users. Users have to trust that Microsoft will keep this information secure and will not share with third parties without the consumers' consent. As this is a collaborative organization, users have also to trust that third party web sites will not misuse the information they can access to about .NET Passport users.

Microsoft Passport is a good example of a "Personal Data Provider." A Personal Data Provider can be seen as a bank: people give their personal data to store, and data banks are supposed to keep them secure. This service has a price, that the users pay by letting this bank conduct a business with this personal information.

Examples of anonymous transaction systems

In the case of real goods (e.g. a book or a computer) bought on the Internet, the good has to be shipped. This implies that the merchant has to know the name and the address of the buyer. A few schemes, like Iprivacy and Incogno SafeZone, can be totally anonymous (the merchant cannot learn anything on the buyer). The buyer chooses the goods she wants to buy and checks out through the server of the trusted third party. In return, she

[23] http://passport.com

gets a certificate that she paid. She gives this certificate and her name and address encrypted with the public key of the transportation company to the merchant. Although he merchant knows which good was bought, he is unable to identify the buyer. The merchant then gives the product to the transportation company, with only the encrypted message. The delivery company can decrypt the buyer's name and address, but is unable to find out what goods were bought. The buyer has purchased and received goods without revealing personal information to the e-merchant.

Privacy and transactions

Privacy in electronic transaction is examined in depth in chapter 2. However, the threat comes from the fact that a lot of personal information is exchanged in order to complete the transaction. In the context of real goods, this information includes address and other shipping information. The Personal Router does not have to deal with this particular information, but it has still to find a way to transact privately. The two previous examples are interesting because they show two very different approaches to do that with two different philosophies.

3.3.3 Threats in wireless communications

Potential issues

In addition to all the other privacy issues which exists on other communications systems such as telephones or the Internet, wireless communications add one of the most sensitive personal data: **Location**. This information, aggregated with other sensitive information like names, phone numbers dialed, online activities, etc., constitutes a very accurate profile of anybody monitored for some reason.

Two other problems may come from identification and transmission. The network needs identification. In effect, the service provider must identify the user (up to a point) for billing and for directing calls and communications. In effect, if a customer is called or receives an email, the provider must be able to direct the communication up to the user, and therefore must be able to identify and locate the user inside a cell. Moreover, the transmission medium is the air, which is even easier to listen to than a wire. As a result, the transmission must be protected as much as possible (using strong cryptographic techniques) in order to provide privacy and to prevent anyone from using someone else's registration. Finally, future mobile network (2.5G and 3G networks) will launch mobile commerce services, implying transactions and payments. This will be close to the problems faced by the Personal Router; the difference will be that users will interact with only one provider.

A particularity of the Personal Router is that users are likely to interact with a large number of providers. This implies that it will be much more difficult to trust a provider. In effect, when large companies like Verizon provide mobile services, law enforcement is relatively easy, because there are few companies providing this services and a large number of customer for each company. Therefore, these companies are likely to act according to regulations. However, in the case of a market with a large number of small providers, it will not be easy to make sure that every single provider complies with laws. Thus, the Personal Router needs to be able to provide this trust without any pre-existing trust relationship with the provider.

This problem is reduced by the disaggregation of providers. The large number of providers makes it much more difficult for them to collude and to exchange information to build a more complete profile of a particular user.

Location services

Location services constitute a potential market for mobile services. The first application, and the first incentive, is the enhanced 911. In effect, under Phase II of E911 implementation, wireless communications carriers must be able to locate 911 callers within one-tenth of a mile (a 125 meter radius) in 67% of all cases. Implementations of Phase II have begun since October 1, 2001. Some technologies use triangulation techniques, other take advantage of the GPS.

The communication carriers are investing to upgrade their network and to meet FCC's requirement for this enhanced 911. In addition to these requirements, location-based services constitute potentially a huge market. Although it is very difficult to foresee what the killer application of location-based services will be, it is more likely that there will be many different applications created by network operators. They could provide the right service and the pertinent information at the right time. Useful services could include navigation, reservation, ordering, home and local information, travel information, end-user assistance services, asset or people tracking, all based on user's actual location. The network could also automatically initiate services depending on the user's location, for example advertising services, location-based billing and filtering, etc.

These services should ensure the user's privacy. For instance, in the case of location-based advertising, Starbucks Coffee should be able to advertise by giving coupons without knowing anything of the potential customer walking near the shop. This will be done according to mobile companies' policies and agreements with users.

Services and applications: Mobile Commerce

With the forthcoming third generation mobile networks, the possible range of applications will be enlarged. The services that are the most concerned with privacy are of course m-commerce and location based services. Negotiating and paying for services (location-based or not) leave a trail easy to track if it is not protected.

If m-commerce is to become a leading driver of the success of future mobile networks, users trust the system, and have no concerns about the security and the privacy of the system and the methods of payment. If they have any fear that their personal information will not be protected, network operator might have a lot of troubles creating a demand for this market.

3.4 Conclusion: Privacy and the Personal Router

Privacy is a real concern for the communication world to come through all the coming new technologies. The issues raised are very similar in different fields and tools deployed to handle them usually lack of efficiency. The main flaw of these tools is that they are unable to take into account the context and the environment of the user, and the evolution of these parameters.

The question is the value of privacy. Customers may prefer to be served and to get personalized services. At the same time, they sometimes would like to be left alone by merchants. These personalized services have a cost, usually paid by the merchants, in exchange for a certain amount of personal information that allow them to optimize their advertising strategy and the design of their website. Therefore, there is a trade-off between personalization, privacy, and the price of the services provided. If customers do not want to release personal information, they will have to pay to get personalized services.

Privacy is regularly cited as a major concern for consumers. However, they have little understanding of the risks and little inclination to change their behavior.
Instead of paying for additional privacy related services, it is more likely that consumers will demand that their tools automatically protect their privacy as much as possible. This is the path chosen for instance by Microsoft with the release of the sixth version of Internet Explorer, which is able to manage cookies and personal data much more subtly than previous releases. However, it is likely that very few users use these additional privacy features, because of their complexity. Privacy protection must be invisible for users. They should just have to trust the instruments they use to interact on the Internet and in the real world to guarantee privacy for them.

The Personal Router must face all these multiple aspects of privacy. Privacy is a very hard problem, and the Personal Router makes it ever more complex. In the framework described in chapter 4, we will try to address some of these issues in the case of electronic transactions.

4 PAYMENT SYSTEMS FOR THE PERSONAL ROUTER

The Personal Router has several important differences with other payment systems. Most payment systems are usually focused on the merchant's side. They are designed so that it offers financial guarantees and a certain ease of use for the merchant. Of course, consumers are also considered, they will not use the system if it is too complicated. But many payment systems are based on a model where the merchant is known and well established, and sometimes has to be registered as a merchant in the system.

In the Personal Router model, the relation is much more symmetric. The provider, assuming here the role of merchants, can be anyone with an 802.11b antenna on the roof, and may not provide services on a regular basis. Therefore, the system has to be very easy to enter for both providers and users. The barriers of entry have to be very low, and payment systems should not create additional difficulty. A close to ideal existing system is for example PayPal. This system is easy to enter, the payer and the payee can play symmetric roles, and the payee does not have to be registered as a payee.

Another requirement for the Personal Router comes from the fact that the system is based on decentralization. Therefore, there will be potentially a very large number of providers, and thus a very large number of payment options. This is strengthened by the fact that the values exchanged can be very small.

That is why most existing payment systems do not really fit well into the Personal Router model. In this chapter, we will first examine several possible business models for the Personal Router. Then we will introduce a decomposition of payment systems into a set of primitive components. This framework will simplify the evaluation and the construction of new payment systems.

4.1 Introduction to primitive operations

The Personal Router model implies that a very large number of providers will interact with a very large number of users. This model will be less likely to develop if only a fixed number of predefined payment systems are available. In effect, the diversity of transaction models that could exist in the Personal Router system, demonstrated in section 2.3, suggests that a large number of possibilities of payment are necessary. Users will negotiate and transact with very different providers, merchants, and brokers. The context will change in the short term as they are moving and changing their needs for service. Thus, limiting the number of payment systems to be used will reduce the flexibility of the Personal Router system and its ability to respond accurately to the needs of the parties.

To be able to compare and select any payment system, a comprehensive framework is needed. This framework must allow parties to select and negotiate payment systems to conduct a particular transaction. This will be done according to predefined constraints and preferences from both sides.

Most approaches to payment systems compare payment properties as a whole. This has a number of drawbacks. First, it requires a complete examination of a payment scheme to extract its properties. Payment schemes are very complex and are not easy to analyze. This is therefore a high level task, not easy to perform automatically. Secondly, there are many ways of building a payment scheme; therefore, two payment schemes will not be easy to compare. Finally, changing one property of a payment system is likely to require rebuilding a whole new system.

Instead of studying payment systems as a whole, this thesis proposes to approach payment systems by decomposing them into primitive operations. A primitive operation is defined as an operation that performs a basic function during the payment procedure. Six primitive operations are defined here: authentication, authorization, transfer, record keeping, aggregation, and timing. These primitives can be combined to create an entity that replicates the original payment system.

This model of payment systems has several advantages over the traditional approach. First of all, instead of defining a taxonomy of complete payment systems, we can define a taxonomy of these primitives. Although it is true that each primitive will be implemented in a different way depending on the payment system, it is possible to observe classes of primitives acting almost identically and having the same types of properties. Then, by compiling the properties of each primitive involved, we can derive the properties of the payment system. Therefore, a comparison of two payment schemes is done very easily by comparing their primitive elements.

Secondly, once a payment system is decomposed into these primitive elements, it becomes very easy to modify its properties by interchanging one primitive.

Finally, the combination of these two advantages creates a favorable framework for negotiation. Starting from pre-defined payment systems, it is very easy to change their properties by changing their primitives. These modifications can takes place as the result of requirements from one or more parties involved, and can evolve depending on negotiation of the properties of the payment.

In the next section, attributes are defined. Then, primitives are examined and their main classes are exposed.

4.2 Selection of a payment scheme: attributes

Users need a way to express their requirements and their preferences. Conversely, the main properties of payment systems need to be given in a standardized from, which would allow them to be compared more accurately. This is a difficult issue, because of three main problems. First, as shown in the previous chapters, there are many properties that can help understand how a payment system was designed and how it performs. However, these properties are not always relevant to a user. This may be because users do not even know their existence, or just that this is not important to them. The one that are important are the one that users perceive and are concerned about. Most users are not aware of these properties and do not want to be. They cannot define their preferences in terms understandable by a machine. Nevertheless, they are able to make judgments in terms of high-level characteristics such as privacy, delay, interface, security, complexity, etc.

In addition, depending on user preferences and environmental constraints, some characteristics become more or less important. For instance, for a user trying to pay via electronic payment with a GSM mobile handset, communication and computation costs will be much more important than it would be if this same user was working on her PC at home with a DSL connection.

That is why selecting a payment system means that the device must understand the limitations of the world around it, understand the high-level decisions of the user, and try to match the best payment system available.

4.2.1 Definition

First, we define the interesting attributes of primitives and payment systems. These attributes are the one that users can perceive and that are important to them.

We call “attributes” the values taken for each of the properties considered in the following sections. These values can be either a precise value (e.g. executing this primitive costs 5 cents) or a range (e.g. this needs at least 10 kbps).

A difficulty comes from the fact that attributes are not static. They are dynamic and interdependent. In effect, they could vary depending on external conditions and on values of other attributes. In the following definitions, these dependencies are described.

Most of these attributes can characterize properties of primitives and payment systems. However, some properties will be impossible to derive from the study of primitives, and only an analysis of the complete scheme can provide them.

We first define those attributes that will define primitives and payment systems. Attributes relative to complete payment systems only are examined afterwards.

4.2.2 Attributes of both primitives and complete payment schemes

We examine here the properties that will be used to set attributes of primitives. They can also be used to describe a complete payment system. From these attributes, it will be possible to infer the attributes of the complete payment system. Some of these properties will be meaningless for a few primitives; in this case, they will be left blank.

A payment system is a combination of primitives. Computing the value of an attribute for the payment system from the values of attributes of its primitives can be done via different basic operations, such as addition, minimum, maximum. The type of operation is specified for each attribute.

Environmental constraints

Environmental constraints are the constraints seen by the user interface making the payment scheme decision. These constraints limit what it is possible to do. The main environmental constraints are computation power, communication capacity, type of user interface, and time.

User device

Computation power

Some primitives may not work properly on devices having low computational capacities such as cell-phones or smart cards. This may incur large delays or even the impossibility to complete the transaction.

This attribute represents the minimum amount of computational power required, measured in operations/second.

The attribute for the payment scheme is the *maximum* of the attributes of the primitives. *Low* means that a device like a cell phone or a smart card with a low computation power is sufficient; *High* means that a device with superior computational power is required. This attribute trades off against time, because a device with low computational power may be sufficient if time is not an issue. It is also affected by the load of the device at the moment when the primitive is executed.

User Interface

The type of user interface is very important. It may limit the possibilities of choice. For example, if the UI does not have any keyboard or another device through which it is possible to enter a PIN, the user cannot enter a PIN code or a password. For instance, a microphone and a voice recognition system would allow the user to authenticate herself by spelling the PIN code.

This attribute reflects the minimum type of interface to conduct the operation.

The value of this attribute for the payment system is the *addition* of the attributes of the primitives. For instance, if one primitive needs a screen and another one a keyboard, then the complete system will need a screen and a keyboard.

Communication bandwidth

If communication bandwidth is low or expensive, payment systems requiring a high communication capacity may not work properly because of delays or because of economical constraints.

This property is defined as the minimum amount of communication bandwidth required, measured in bits/second.

The attribute for the payment scheme is the *maximum* of the attributes of the primitives.

Low means that a small bandwidth, like the one available on a device like a GSM phone (a few tens of kilobytes per second) is sufficient; *High* means that a device with superior communication capabilities, like a DSL or a cable line, is required.

Like computation, this attribute trades off against time, because a device with low communication capabilities may be sufficient if time is not an issue.

Time

Time may be critical in some cases, and be unimportant in other. An example where time is important is when a Personal Router is negotiating services with a new provider, while the user is on a phone conversation (roaming). The negotiation and the payment must be done very quickly to avoid interrupting the communication.

This property is defined as the minimum amount of time required, measured in second.

The attribute for the payment scheme is the *sum* of the attributes of the primitives.

Time is highly dependent on other attributes like computational power and communication bandwidth.

Three levels are set for this attribute: Quick, Medium and Slow. These levels are just comparative. The aim is to give orders of magnitude and to have relative values. To give an idea of their meaning, we give here a more precise definition with an evaluation of these values.

Quick means that the transaction can be completed in about one second or less; it means that users do hardly perceive the delay. *Medium* means that it can be completed in about one minute; users perceive the delay, but it is short enough for them to wait. *Slow* means that the transaction needs a few days to complete; the completion of the transaction is too slow; users have to wait a long time, and will move or do something else before the transaction is completed.

User preferences

These preferences are much too complicated for a user. Even so, they are still important for the user in one form or another. The following properties are the one seen by the end-user, as opposed to more internal properties seen only by designers. They are organized in three main categories: economical, technical and social aspects. A fourth category, regulatory aspects, could be added. Regulations could force users to choose one or the other of the properties.

Economical aspects

Fixed costs

This property represents the fixed cost of adopting a particular system for each of the primitives. This cost depends on registration costs, cost of required hardware, installation costs, etc. It is the cost in US dollars of the necessary equipment.

The attribute for the payment scheme is the *sum* of the attributes of the primitives.

This trades off against most of the other attributes. For instance, users could have a choice between buying a basic GSM cell phone with small computational power, poor communication capabilities, and a monochrome screen capable of printing only a few characters. On the other hand, they could choose a fancier device with GPRS or 3G capabilities, a big color screen capable of showing graphics and a processor sufficient to handle a graphical interface and high-speed computations.

The values for this attribute can be *none* if there is no special hardware required and no paying registration; *low* if these costs are of a few dollars, and high if they are over a hundred dollars.

Transaction cost

This property represents the marginal cost of the execution of the primitive, measured in US dollars.

The attribute for the payment scheme is the *sum* of the attributes of the primitives.

According to the previous chapters, these costs can be broken up into computation costs, communication costs, administrative costs and fraud costs.

Once again, this cost is highly dependent on other attributes. For instance, users may pay a premium for using a high-speed communication link to speed up the transaction; they could also pay a premium to reduce fraud risks.

Transaction costs can be zero if the user does not have to pay anything for the transaction.

It can be *small* if the marginal cost is a few cents, and high if it is above one dollar.

Fraud risk

This property represents the estimation of the fraud risk coming from each of the primitives. This is different from the fraud risk of the complete transaction, described in section 4.2.3.

Technical aspects

Transaction time

The attribute represents the length of time during which the user needs to act or wait while the operation is processing. In other words, this represents the user-perception of the time taken by the transaction to complete. It is measured in seconds.

The attribute for the payment scheme is the *sum* of the attributes of the primitives.

This is also dependent on other attributes, like computational power, user interface, or communication capabilities.

Time is highly dependent on other attributes like computational power and communication bandwidth.

Three levels are set for this attribute: Quick, Medium and Slow. They have the same meaning as the attribute Time defined in the previous section.

Non-refutable

The parties involved in the transaction have a proof that the transaction took place, and can prove the details of the transaction such as amount, date and other data relative to the transaction.

This property is true if the execution of the primitive is non-refutable, and false if not. The attribute for the payment scheme is the true if the attributes of all primitives are true, false if not.

Security

This property measures whether the system is tamper-resistant, and is not easily stolen, and makes fraudulent action more difficult. It is measured as the time needed to break the system.

It depends on security measures taken by each of the primitives. The level of security of the total transaction is the *minimum* of all attributes of the primitives used (as an analogy, the strength of a chain is the strength of its weakest link).

Security could be a function of time constraints and computational power; if the device is capable of doing strong cryptographic operations in a short time, the security could be strengthened.

Social aspects

Unobtrusive

This attribute measures the *number of steps* the user has to take to complete a primitive. This property is particularly important for micropayment, because users do not want to be prompted each time they spend 2 cents.

The attribute of the payment scheme will be the *sum* of the attributes of the primitives used.

Privacy and anonymity

The privacy analysis is very important, and cannot be summarized in a single attribute. This property has multi-dimensional aspects, which often requires a deeper analysis.
For each of the primitives, it is possible to draw a table representing the privacy properties of the primitive, as shown in section 2.2.13. The attribute is this multidimensional table. Users would define, directly or through a smart agent and a user interface, what information they are ready to give and what information they would like to keep private for each box of the table. The decision maker algorithm would then have to select a matching primitive from the comparison of the user's table and the table of the primitive.
The attribute of the payment system is the *addition* computed for each box of the table for each of the primitives. For instance, if a primitive reveals the name of the payer and another reveals her address, then the resulting payment system will reveal the name and the address.

4.2.3 Attributes of the complete system

These attributes can only be defined at the level of the transaction, because they require a level of analysis higher than the one offered by primitives. The following attributes express fundamental mechanisms of payment systems, relative to others. Therefore, in the previous list, they would all appear in the section "economical aspects."

Interoperable

A payment system is interoperable if it is easy to convert to other payment systems.

Two-way payment

The system can be used to pay or to receive a payment; there is no registered merchant or customer status. This property is also called peer-to-peer payment.

Immediately respendable

In the case of token money, coins can be re-spent immediately; they do not need to be redeemed.

In the case of notational money, it is possible to use the money as soon as the transaction is completed.

Actual payment time

Depending on the payment system, the user's account can be debited/credited at a different time.

4.2.4 Use of the attributes

The attributes are interdependent, and in addition depend on external variables like the workload on the processor, or the available bandwidth on the communication link. Moreover, the attributes of a primitive depend on its particular implementation. Therefore, comparing primitives and attributes is not easy.

However, it is possible to compare relative values of classes of attributes. From the analysis of classes, it is possible to derive general values of attributes like "small" or "high." Thus, a first selection can be made, depending on the requirements of a specific transaction. Then, between the remaining implementations of primitives, a more precise analysis can be made on the fly at runtime to decide which one of them has the best properties. At this point of the process, it is possible to put actual numbers on the different attributes, and therefore it is possible to compare accurately different primitives.

4.3 Description of Payments with Primitive Elements

In this section, we define in depth each of the six primitive operations. Their attributes are highlighted for each specific case. We will first introduce three primitive objects, which will be used by the primitive operations.

4.3.1 Primitive objects

The idea is to create primitive objects, which could be viewed as classes in object-oriented programming.

These objects can be called and used by primitive functions. They hold different types of information, and can be access by primitive functions in read only or read write mode, depending of the relationship between the function and the object.

The following objects are defined here:

1. State
2. Identity / Credentials
3. Transaction

State

Definition

This object stores the current state of the payment system and its different parameters. For instance, it could store the identity of the other parties, the result of different negotiation that took place earlier, and other information relevant to the transaction.

Identity / credentials

Definition

This object symbolizes the possible identity of an entity taking part in the transaction. This object possesses all the different ways of authenticating a particular entity. It is able to use this data depending on the context. It knows all the personal information and the credentials of the entity it is working for. Such an object is delicate to handle. There are huge issues of security: should this object be stolen or broken, all personal information of the user would be revealed. This object may not have to exist, it may be only a way to model interactions.

Examples

This object would know the private/public key pair of the user in a PKI system, would store temporary identities created for specific transactions, would store hash trees for hash authentications, etc. It would also know a lot of information about the user, like name, address, etc.

Transaction

Definition

A transaction object defines the source and the destination of the transfer, as well as its parameters (e.g. encrypted or not, encryption key, encoding, etc). An identity object may be passed along to define some of these parameters. A transaction object contains all the data representing the actual transaction; it contains the coin if it is an e-cash transaction, and just data if it is a notational scheme.

4.3.2 Primitive operations

A primitive operation is an operation that performs a basic function during the payment procedure.

Six primitive operations are defined:

1. Authentication
2. Authorization
3. Transfer
4. Record keeping
5. Aggregation
6. Timing

The function described is high-level; therefore, there may be many implementations of it, and each of these implementations can act very differently. The attributes of these operations greatly depend on the implementation. It is possible to classify these implementations in different classes. Implementations in the same classes will have similar attributes.

For each of the primitives, the description of the different classes is followed by a table summarizing the attributes of these classes. Attributes are expressed here in fuzzy terms like “small,” “medium” or “high.” Each real implementation should be able to put actual numbers on them; this table is just there to give orders of magnitude.

Authentication

Definition

Authentication is the operation by which two entities exchange credentials to prove that they are who they say they are, or that they belong to a particular group. Authentication can be done without revealing personal data.

For example, I could use authentication to prove that I am John Doe, and/or that I am an MIT student. A payment server could prove that it is a genuine payment server.

The spectrum of authentication is wide. There may be several levels of authentication. It could go to "I am John Doe" to "I am the same person you met last week" or "I belong to this group."

From the perspective of a particular user, authentication can mean two very different things. It could be either authenticating herself to somebody else, by showing credentials proving her identity, or authenticating someone else by verifying the presented credentials.

Functional diagram

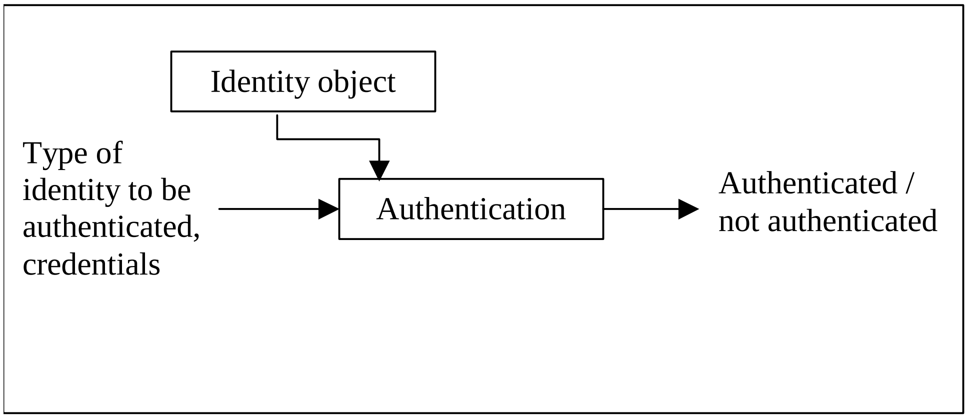
Identity object
Type of identity to be authenticated, credentials
Authentication
Authenticated / not authenticated

Figure 4-1: Diagram of primitive *Authentication*

Figure 4-1 shows the functional diagram of primitive *Authentication*.

Classes

The attributes of the following classes are summarized in Table 4-1.

"Human" authentication

In most cases of real-life payments, authentication is done by human beings. It could be physically authenticating a customer in a bank, certifying a bank note, or authenticating a written signature. This type of authentication requires no computation and no communication (except for the case of a check that has to be physically sent to the issuing bank).

Fraud is possible if a bank note or a written signature can be forged.

This method of authentication cannot be used on-line. However, it is possible to use it at an early stage of the scheme, for example during the initialization. For instance, users could have to present their driver's license to get access to the system.

Biometrics

Biometrics uses measurable biological characteristics such as fingerprints, voiceprints, eyeprints, face recognition, etc, to authenticate the user. It needs a special hardware device to verify that these biological characteristics match the user. Biometrics is not widely used, but represent an important way to authenticate users, and it could be potentially more used in the future.

User ID + Password authentication

Users are authenticated by presenting a user ID and a password. This method is commonly used on the Internet when accessing a web site. For instance, PayPal uses this method to authenticate its users. This is also the method used by credit card systems when users have to enter their Personal Identification Number (PIN). The user identification is given by the card, and users have to provide the PIN code.

Fraud occurs when other people gain access to the user id and to the password.

Digital signature: PKI

A user digitally signing a message first needs a public / private key pair, given by a Public Key Infrastructure (PKI). Then the message signed with the private key is sent to the receiver. The receiver has to check both the signature and the public key of the sender. This can be done either by storing the previously authenticated public key of the user, or by going along a certification path. A certification authority signs the public key of the user. The signature of this certification authority has to be signed itself by a higher certification authority, up to the root certification authority. Each of the signature verifications is computationally intensive, and going along the certification path incurs communication costs. There is a trade-off with administrative costs: it is possible to store all or part of the required public keys, so that it is not necessary to go up to the root certification authority.

Fraud implies that one of the PKI signatures has to be broken, which is supposed to be impossible given the size of the keys.

Hash function

There are other ways to authenticate a user. It is also possible perform an authentication with less computationally intensive tools, such as hash functions. Hash functions are used a lot by micropayment schemes. A hash function is a function that produces a number from a message in such a way that it is extremely unlikely that some other text will produce the same hash value. A hash function with good cryptographic qualities will make it very hard to generate two texts having the hash value. These functions are called one-way functions.

The protocol used for authentication is very simple (Schneier, 1996, p.53). There is an initialization phase where the user exchanges results of a hash function with a server; subsequently, each time the user can provide the number whose result by the hash function is the same as the one stored on the server, she is authenticated. Each number used is crossed out and cannot be used anymore.

This is very close to a userID – password scheme. The difference comes from the fact that passwords can be used one time only, and that the system has to be reinitialized once in a while.

Fraud implies that the one-way hash function has been inverted or that the number has been stolen.

Attributes

Table 4-1 lists the attributes of primitive authentication, and Table 4-2 lists the privacy properties. The design of Table 4-2 comes from the analysis of privacy in payment systems made in section 2.2.13.

Table 4-1: Attributes of primitive Authentication

	Human authentication	Biometrics	UserID + Password	PKI	Hash function
Computation	None	High	None	High	Low
User interface	N/A	Biometric device	Screen + device to enter code	None (device to access private key?)	None
Communication	N/A	None	Low	Medium	Low
Time	Slow	Medium	Slow (enter userID and password)	Medium	Quick
Fixed costs	None	High	None	High (if subscription to the PKI scheme)	None
Transaction cost	Low / High, depending on the scheme (e.g. checks)	Low	Low	Low	Low
Fraud risk	Low	Potentially none	High	Low	Low
Transaction Time	Slow	Medium	Quick	Medium	Quick
Non refutable	N/A	N/A	N/A	N/A	N/A
Security	N/A	N/A	Low, medium if communication link is encrypted	High	High
Unobtrusive	N/A	No	Enter user ID and password	Enter password	Yes

Table 4-2: Privacy characteristics of primitive Authentication

			WHAT THEY KNOW				
			Party authenticating	Party to authenticate	Amount of transaction	Time and place of transaction	Nature of transaction
WHO KNOWS	Party authenticating	Identifiable	N/A	Credentials	N/A	Everything	N/A
		Traceable	N/A	Credentials			
	Party to authenticate	Identifiable	Nothing	N/A	N/A	Everything	N/A
		Traceable	Nothing	N/A			
	Bank	Issuer	N/A	N/A	N/A	N/A	N/A
		Acquirer	N/A	N/A	N/A	N/A	N/A
	Central authority (if applicable)		Some	Depends	N/A	Time	N/A
	Observer		Depends	Depends	N/A	Time	N/A

Authorization

Definition

Authorization is the right to access a system or a device, or the right to perform a particular operation. It usually involves credentials and some sort of authentication. For instance, MIT students may be authorize to access the MIT bursar account system. A customer may be authorized to withdraw cash from an ATM, provided that she has enough money on her account.

Authentication and authorization may be different. Authorization often includes some type of authentication, but not necessarily. An example where we can have authorization but no authentication is a cash transaction: the bill or the electronic coin is checked (or authorized), but this is done without authenticating the payer.

Authorization may be performed by a bank or by a user.

Authorization could be passive: a user could authorize a system to perform transaction on her behalf if certain properties are verified (such as amount of transaction, context, etc).

Functional Diagram

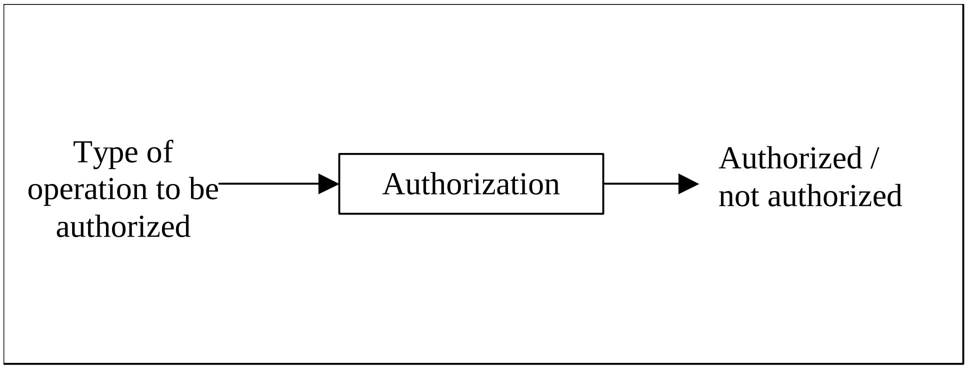
Type of operation to be authorized
Authorization
Authorized / not authorized

Figure 4-2: Diagram of primitive *Authorization*

Figure 4-2 shows the diagram of primitive *Authorization*.

Classes

The attributes of the following classes are summarized in Table 4-3.

Human authorization

Authorization via the "real world" is an authorization made by phone, mail, email, fax, etc. A human being needs to personally authorize the transaction. For instance, in the First Virtual payment system, the user receives an email asking to confirm each transaction. In the real world, a bank customer may have to be physically present to authorize certain transactions (e.g. large money transfers), or send a fax, or just confirm the order by phone. Some systems are very simple, and just prompt users with an “OK” dialog box to make sure they authorize the transaction.

Human authorization incurs no computation, but requires communications. Fraud can occur if the communication link is tampered with.

Automatic authorization

In this case, a machine is granted the right to confirm a transaction. The machine can check for example that there is enough money on the bank account, or that the credit card number is valid and is not on a blacklist, or that a token has not been spent yet.

There are two types of automatic authorization: First, the party authorizing the transaction has all the information needed locally. This happens for instance in the case of the payer, if the device is granted the right to conduct automatically the transaction, or in the case of the payee, in the case of an off-line scheme such as real cash or CAFÉ, all the information needed are stored on the device. The other possibility is when the user has to go on-line to authorize the transaction. This happens in the case of DigiCash, for example, where the token has to be checked in a central database against double spending.

Automatic authorization incurs some computation (compute the authorization), and communication (request and respond to the request). This system often uses a database. For instance, DigiCash authorize a payment by checking that the presented token has not been spent yet.

Again, fraud can occur if the communication channel or the machine is tampered with.

Attributes

Table 4-3 lists the attributes of primitive authentication.

Table 4-3: Attributes of primitive Authorization

	Human authorization	Automatic off-line authorization	Automatic on-line authorization
Computation	None	Some	Some
User interface	Depends on the interface (phone, dialog box, etc)	Screen + device to enter information	Screen + device to enter information
Communication	N/A	None	Low
Time	Slow	Fast	Medium
Fixed costs	Depends on the implementation	None	High if central database
Transaction cost	Low, high if communication are expensive	Low	Medium because of communications
Fraud risk	Low	Medium, easier to fraud	Low
Transaction Time	Slow	Quick	Quick
Non refutable	Yes, depending on the implementation	Yes, if a proof is given	Yes, if a proof is given
Security	Depends on the implementation	Depends on the implementation	Depends on the implementation
Unobtrusive	No	Yes	Yes

Privacy attributes:

In the case of off-line systems, privacy is not exposed. All the processing is done locally, therefore there is no information leakage. In the case of on-line systems, usually a third party has to be consulted. The privacy characteristics depend highly on the information given to this third party, and therefore on the scheme used.

For instance, for Digicash[24], the server knows the amount of the transaction, and identifies the payee, but cannot learn anything on the payer. In a check-based system, the central authority learns the amount, the ID of the payer and the ID of the payee, in order to check that there is enough money on the payer's account and to authorize the payment. Thus, each implementation of the primitive Authorization will have different privacy attributes.

[24] DigiCash will be examined in detail in chapter 5.

Transfer

Definition

The transfer primitive takes care of any transfer of information between two parties that represents an actual transaction. It is formalized by a transaction object, described in section 4.3.1.

Transfer of value is the transfer of something that has a value by itself, like a coin, a stream of bits representing an electronic token, a paper check, etc. It can be notational or token-based money.

A transfer of data can be a transfer of information required to conduct the transaction. This can be credentials, personal information such as account number, etc. Transfers can be done over phone lines, Internet, hand-to-hand, etc. An important characteristic of transfers is whether they are encrypted or not.

A transfer can be modeled as a "copy" or a "move" function. The difference between the transfer of value and the transfer of data is that a transfer of value is necessarily a "move" operation, and the transfer of data is a "copy" operation. In effect, when transferring something that has a value, the sender cannot keep it; otherwise the value of it would be doubled. Therefore, a transfer of value can be seen as a "move" operation of this object. On the other hand, when transferring data, the sender usually keeps the information and this operation is close to a "copy" operation.

As shown in Figure 4-3, there are two types of transfer primitives. The first one is transfer-out; the second is transfer-in. Transfer-out is responsible for sending the transaction object to the other party. An external operation builds a transaction object (see section 4.3.1), and gives it to the transfer-out primitive, which sends it to the other party. Transfer-in receives the information, formats it into a transaction object, and triggers another operation by passing the transaction object.

Functional diagram

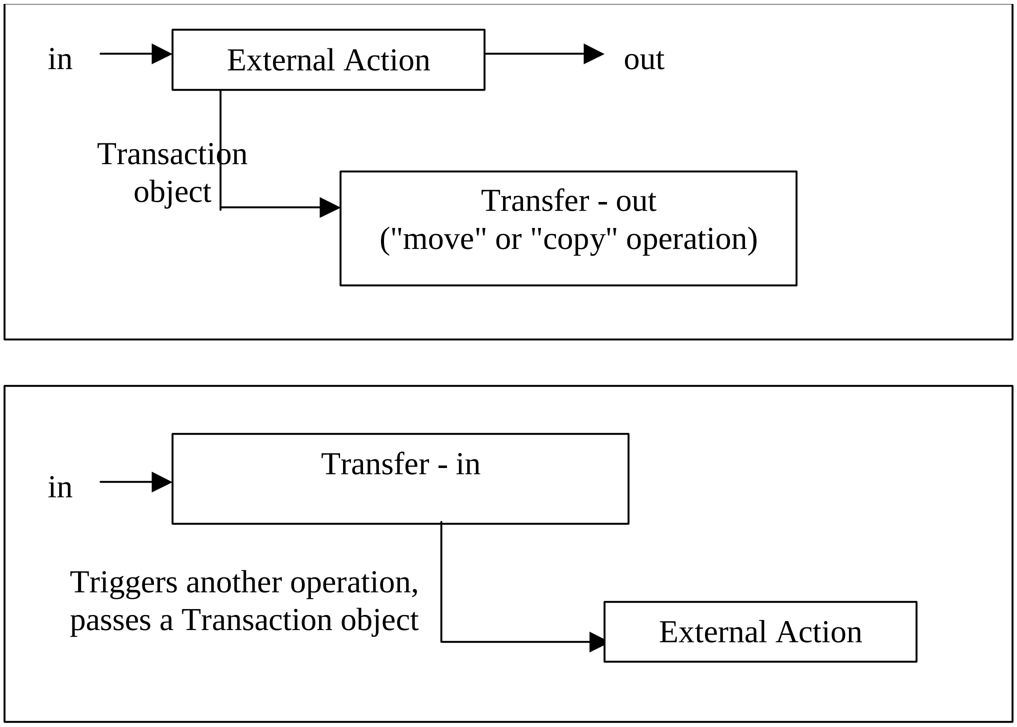
in
External Action
out
Transaction object
Transfer - out
("move" or "copy" operation)
in
Transfer - in
Triggers another operation, passes a Transaction object
External Action

Figure 4-3: Diagram of primitive Transfer, differentiated as "Transfer-out" and "Transfer-in"

Figure 4-3 shows the diagram of primitive *Transfer*. The transaction object is defined in section 4.3.1 on primitive objects.

Classes

The attributes of the following classes are summarized in Table 4-4.

Physical transfers: hand to hand

This occurs for example when paying with real cash or when giving a check to someone. This method is not used on-line and cannot be implemented. It may be used for initialization for instance.

Physical transfers: mail

This happens when sending something by mail, or for instance when a bank needs to send a check that was deposited to the issuing bank for authentication.
This method incurs high communication costs. Fraud occurs if the mail is intercepted.

As in the case of hand-to-hand transfer, this method cannot be used on-line and cannot be implemented.

One-way electronic transfers

In this case, the transfer is made over a network, which can be the Internet for instance. This is the only method used to conduct on-line transactions. There are many ways to implement this; the main characteristic is whether it is encrypted or not. Fraud occurs if the bit stream is intercepted or modified.

Transfers can be one-way or two-way.

One-way transfers are transfers where the receiver does not have to respond to the sender. This is the case of emails for instance. This does not require a direct network connection, and this system is often used asynchronously, meaning that the receiver and the sender do not have to interact synchronously. PayPal works on this mode: when the payer initiates a payment, the payee receives an email and can claim the money on PayPal server later.

Two-way transfers are made on a two-way direct network connection to interact. The sender and the receiver are in a synchronous transfer mode. The receiver will take immediate action to respond to the sender’s message.

Attributes

Table 4-4: Attributes of primitive Transfer

	Hand to hand transfer	Mail transfer	Electronic transfer
Computation	None	None	Low
User interface	N/A	N/A	N/A
Communication	N/A	High	Depends on the context (e.g. mobile vs. DSL)
Time	Medium	Slow (a few days)	Quick, depends on the link
Fixed costs	None	None	High (subscription to the provider)
Transaction cost	N/A	High	Depends on the context (flat rate vs. per bit rate)
Fraud risk	None	High (interception)	Low if encrypted
Transaction Time	Medium	Slow	Quick
Non refutable	N/A	No, except if certified with receipt	Depends on the implementation
Security	N/A	Low (interception)	High if encrypted
Unobtrusive	N/A	Send a letter via Post Office	Yes

The privacy attribute of *Transfer* is highly dependent on the scheme and the implementation; it cannot be generalized. In some cases, transfer could certainly worsen the traceability characteristics of the transaction: for instance, a transfer could leak an IP address. In some others, transfer would not have any consequence on privacy.

Record keeping

Definition

Different parties in the payment system may have the will or the obligation to keep records of transactions. This may come from and also be limited by legal requirements. This is the part where privacy and security interact the most.

The most interesting fact about a record keeping operation is what triggered the operation. Record keeping is highly dependant on other operations. In fact, it is either triggered by an action of wrapped around other actions. The different characteristics, which depend on the payment scheme, are whether it is triggered before or after another operation, whether it records the input, the output, or both of other operations, whether the record is public and whether it keeps track of identities (privacy issues).

The record keeping operation can also have a result, and can trigger another operation in result. For instance, a record keeping operation could be in charge of detecting fraud patterns. This role can be modeled by a record keeping operation running permanently, and take advantage of the offline or idle time of the machine to perform different analyses of the record.

Another important distinction is the difference between what has to be recorded for the good conduct of the transaction, what has to be recorded for law enforcement reasons, and what parties wish to record for their own use.

Functional diagram

Figure 4-4 shows the diagram of primitive *Record Keeping*.

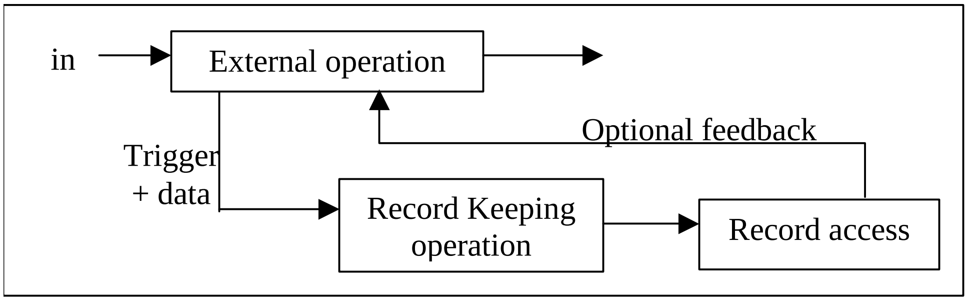

Figure 4-4: Diagram of primitive *Record keeping*

Classes

The attributes of the following classes are summarized in Table 4-5.

Local computer database

Records are stored locally in a database. This requires some computation (database queries, maintain the database), Fraud occurs if entries are added, deleted or modified, and this should not be possible, as the database is stored locally.

Distant computer database

Records are stored on a distant server. In this case, if the user wants to access records on the server, she has to trust that she will always have access to it, and that records will not be modified or deleted. The advantage of this method is that the tasks of maintaining and querying the database are left to the distant server. Whenever a user needs information, she has to access the server and make a query. This requires communications.

Attributes

Table 4-5: Attributes of primitive Record Keeping

	Local database	Distant database
Computation	Low	Low
User interface	N/A	N/A
Communication	N/A	Depends on the context
Time	Quick	Medium
Fixed costs	None	High (subscription to the provider)
Transaction cost	None	Depends on the context (flat rate vs. per bit rate)
Fraud risk	None	Low if encrypted
Transaction Time	Quick	Slow
Non refutable	N/A	Depends on the implementation
Security	N/A	High if encrypted
Unobtrusive	Yes	Yes

Storing data locally do not modify the privacy attributes of a scheme. However, using a distant server do. The information leakage comes from two sources. The first one is during the actual transaction, when data about the current transaction is stored. The party maintaining this database has access to the information. The second one is when parties make subsequent queries on the database. The party maintaining the database can learn information about users connecting to the database, how often they connect, what information they are looking for, maybe from where they access the database.

The information leakage is dependant on the scheme, because different schemes leak different types of information.

Aggregation

Definition

This is the action of grouping several operations together. It is usually done to make a single big transfer instead of many small value transfers. It is typically temporally decoupled from the transaction itself, and mainly occurs between two financial institutions. For instance, in the credit card system, transactions are aggregated and settled regularly between banks. The time between settlements depends on parameters fixed by the parties. It could be for instance once a day, once a month, as soon as a fixed amount is exceeded.

Aggregation is also used by micropayments schemes to reduce costs.

In many cases, aggregation can be modeled with a record keeping operation.

There are different types of aggregators:

- Forward aggregators: in the case of payments, the actual transfers of money are made before the paying operation (e.g. pre-paid cards)
- Backward aggregators: in the case of payments, the actual transfers of money are made after the paying operation (e.g. phone monthly bill)
- Discount aggregators: the more you buy, the cheaper it is
- Averaging aggregators: average a large number of operation and keep only the mean
- Flat rate aggregator: pay a monthly fee for unlimited access

Aggregation gets transaction objects as an input. When it is triggered (typically by another primitive such as timing), it gives an output that can trigger other actions.

Functional diagram

Figure 4-5 shows the diagram of primitive *Aggregation*.

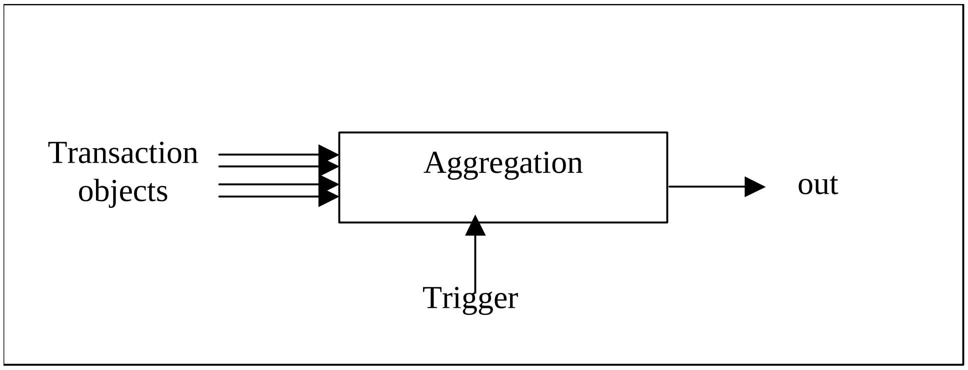

Figure 4-5: Diagram of primitive *Aggregation*

Attributes

The attributes of this primitive are highly dependant on the scheme and on the implementation.

The privacy attributes of aggregation are nonetheless very interesting, as aggregation can actually improve privacy performances of the scheme.

In fact, aggregation allows hiding specific information in the aggregated data. For instance, instead of recording when a communication was made, from where, and what sites were visited, it is much better, from a privacy perspective, to add the price of the communication to a monthly bill.

Timing

Definition

Transactions may take a certain delay to process, or may accept a delay. For instance, a check will usually be voided if deposited more than one year after it was written. For cash transactions, withdrawing or depositing cash is temporally decoupled from the transaction itself. Check and account transfer transactions may take a certain time to settle, sometimes up to a few days.

This time component of a transaction can be separated in two: timing and delay.

The delay comes from the time required by primitives to complete their tasks. They can be included in the primitives themselves.

However, timing primitive is necessary when something has to be used or made before a certain time (e.g. "do by…" or "use by..."). For instance, a check has to be deposited before a certain date before being void, and an e-cash coin must be spent or redeemed before a certain date before being void.

The delay primitive is easy to implement in other primitives. A timing operation is more difficult to simulate in the case where it triggers another operation.

Functional diagram

Figure 4-6 shows the diagram of primitive *Timing*.

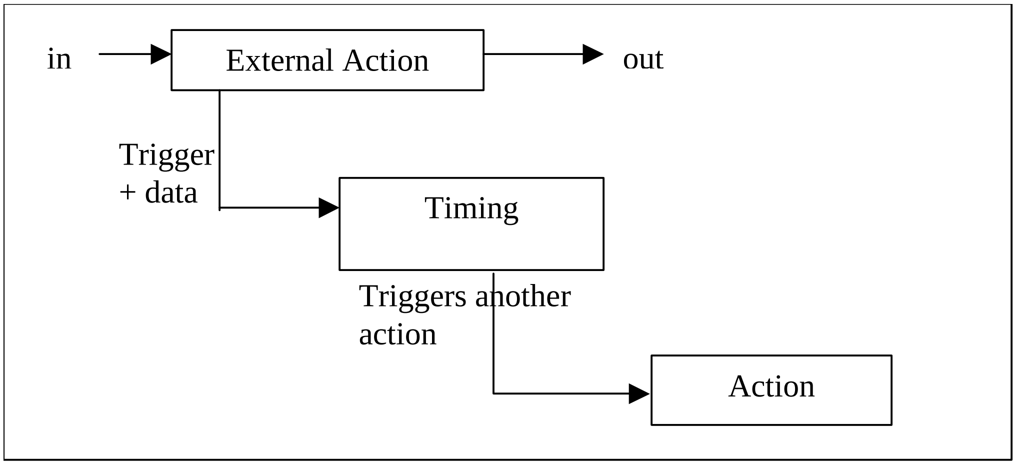

Figure 4-6: Diagram of primitive *Timing*

Electronic cash transaction

The delay allowed between the e-cash withdrawal and the actual payment has a cost. As all token numbers must be recorded to prevent double spending, the longer the delay, the bigger the database, and the more expensive it is. This cost can be limited by having an expiry date on transactions.

4.4 Selection process

The selection process would be very similar to the one proposed in (MacKie-Mason and White, 1997). It should handle first the binary constraints, and eliminate all potential solutions that do not satisfy these constraints. Then the process should deal with fuzzier constraints and try to optimize the solution to find the best implementation given all the constraints.

The process would be able to select the payment scheme satisfying the greater number of the most important preferences ranked by the user.

4.5 Summary

In this chapter, we have first described different transaction models in the context of the Personal Router. These models help understand the spectrum of possible exchange models that the Personal Router will need to implement.

To help manage the requirements of users and match them with the necessity of a payment system, a decomposition of payment schemes in primitive operations is proposed. Each of these primitive represents a particular operation of a payment system, and the combination of these operations acts as the real payment system. There are different ways to implement those primitives, while keeping the same results. With this framework, it is therefore possible to modify the properties of a payment system by changing an implementation of a primitive.

Properties of these primitives are defined and called attributes, and users' preferences are identified through these attributes.

Matching users' preferences with a specific payment system means selecting the payment system that is the closer to the set of preferences previously defined.

In the next chapter, we will give several examples of payment systems decompositions into primitives, and some possible modifications of these schemes.

5 RESULTS / EXAMPLES

To demonstrate possible applications of the framework of primitives defined in chapter 4, this chapter shows how three real payment systems can be constructed from them. These three payment systems have been selected because they cover very different areas of the rich environment of payment systems, and they are likely to be used in the Personal Router. Each of these is first decomposed and several suggestions of implementations are given, with the modifications induced.

The three payment systems are DigiCash, credit cards and PayPal. DigiCash and PayPal are symmetric, in the sense that a payer can also be a payee and vice versa. However, credit card systems require payees to be registered as merchant before being able to accept credit card payments.

DigiCash is an electronic cash system, providing total anonymity to the payer. It is interesting to consider because of it is fully anonymous. The Personal Router environment is likely to value this property.

The second system chosen is the credit card system. This is one of the most used on the Internet and in the “real” world. Its pervasive presence in payments makes it very important. It is likely to be used a lot in the Personal Router system.

The last payment system examined in this chapter is PayPal, which is one of the few successful payment systems specifically designed for the Internet. It has more than 15 million subscribers and represents a growing part of peer-to-peer Internet transactions.

These characteristics also make PayPal a potentially very successful payment system in the Personal Router environment.
From these examples, it will be possible to conclude about the weaknesses and the advantages of the construction of primitives.

5.1 DigiCash

DigiCash[25] was one of the first companies to launch an electronic cash payment scheme. This company was founded by David Chaum. The main characteristic of this scheme is that it is fully anonymous. The client withdraws coins from the bank in such a way that the bank is unable to find out the serial numbers of those coins. The coins can then be spent anonymously with a merchant, who cannot learn anything from them except that they are valid. Double-spending is prevented by having payee redeem coins to the bank and getting the bank's authorization before completing the transaction. Even collusion between the bank and a merchant cannot reveal any information about the user.

5.1.1 Quick technical overview of DigiCash

A precise description of the DigiCash scheme and of its details can be found in (O'Mahony et al., 2001), in (Camp, 2000), and in (Schneier, 1996). DigiCash uses blinding techniques to hide the coin's serial number from the bank when the bank signs a new coin. To withdraw a coin, users first choose a random number, which will be the serial numbers of the coin. This number is large enough so that it is extremely unlikely that two users choose the same serial number. The serial number is cryptographically signed by the bank, using blinding techniques that prevent the bank to see the serial number (see (Schneier, 1996) for more details on blinding). The bank uses a different signature key for each coin denomination. Coin denominations are chosen so that any amount can be paid with these coins (for instance, denominations can be chosen as the successive powers of two). This allows payments of any dollar amount with a combination of coins. The client then unblinds the coins, and stores them into a "wallet." When a coin is spent, the merchant redeems the coins to the bank, and the bank checks by looking for the coins' serial numbers in a central database that they have not been already

[25] http://www.digicash.com

spent. If not, the bank sends back an acknowledgement and credits the merchant's account. Hash values of a secret created by the client is cryptographically associated with the coins to provide the client with the possibility to later prove that she initiated the payment. Figure 5-1 represents a complete transaction with the DigiCash scheme.

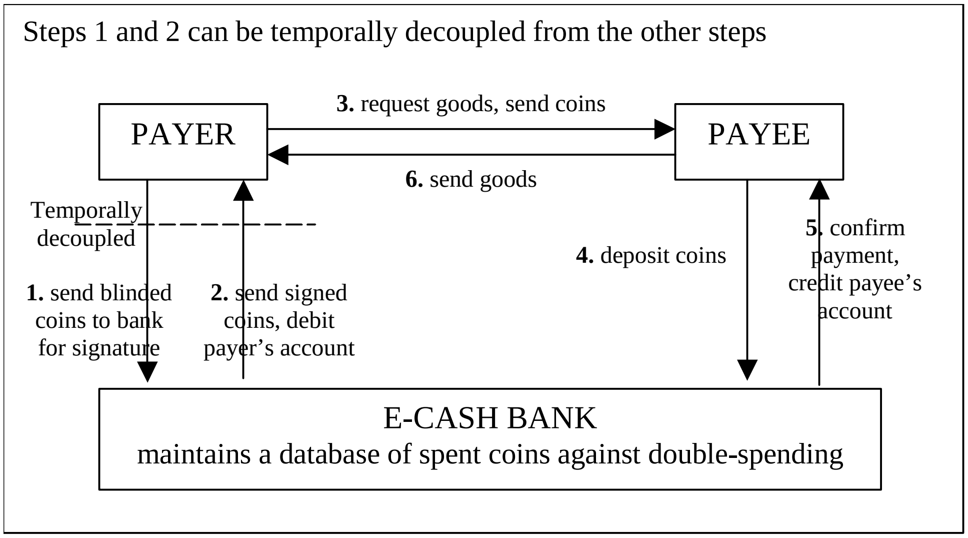
Steps 1 and 2 can be temporally decoupled from the other steps
PAYER
PAYEE
3. request goods, send coins
6. send goods
Temporally decoupled
1. send blinded coins to bank for signature
2. send signed coins, debit payer's account
4. deposit coins
5. confirm payment, credit payee's account
E-CASH BANK
maintains a database of spent coins against double-spending

Figure 5-1: The DigiCash scheme

The bank's database must store all the serial numbers and secrets of spent coins; thus, it grows very quickly. To prevent this database from being too large and unmanageable, an expiry date is associated to every signature of a denomination. Before a signature expires, clients must redeem all coins with the corresponding signature to the bank and withdraw new ones.

A coin is therefore valid if it was signed with the signature associated with the correct denomination, if the current date is before the expiry date of the signature key, and if the serial number of the coin does not appear in the database.

5.1.2 Primitives of a DigiCash payment system

In this section, we will go over the different parties in a DigiCash system and describe the primitives and their attributes. The system is supposed to be already initialized; this means that the payer and the payee both have a DigiCash account and that the payer has already withdrawn valid coins from the bank.

Figure 5-2 shows how DigiCash is broken into primitive operations during a transaction between a payer and a payee. Figure 5-2 does not include the initial withdrawal of money.

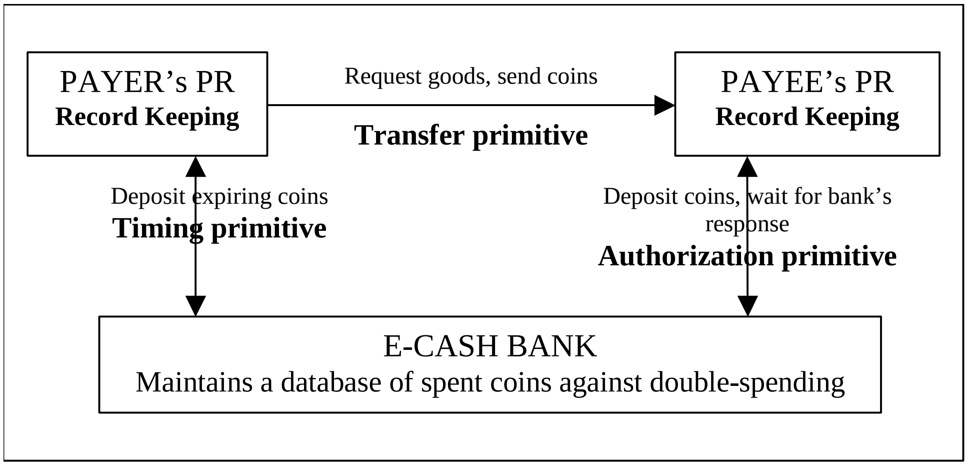

Figure 5-2: The DigiCash primitives

The following sections will describe primitives for the payer and for the payee. Section 5.1.3 will show several possible modifications of these primitives.

On the payer's side

We assume here the function of the payer. The initialization phase consists first in having an account set up with the bank minting coins, and second in withdrawing coins from the bank. The first phase will often involve human interactions, while the second phase can be done automatically. The second part of initialization could be designed with the primitives, as shown in Figure 5-3. We will first go over the withdrawal of coins, and then examine the transaction between the payer and the payee.

Initialization phase: withdrawal of coins

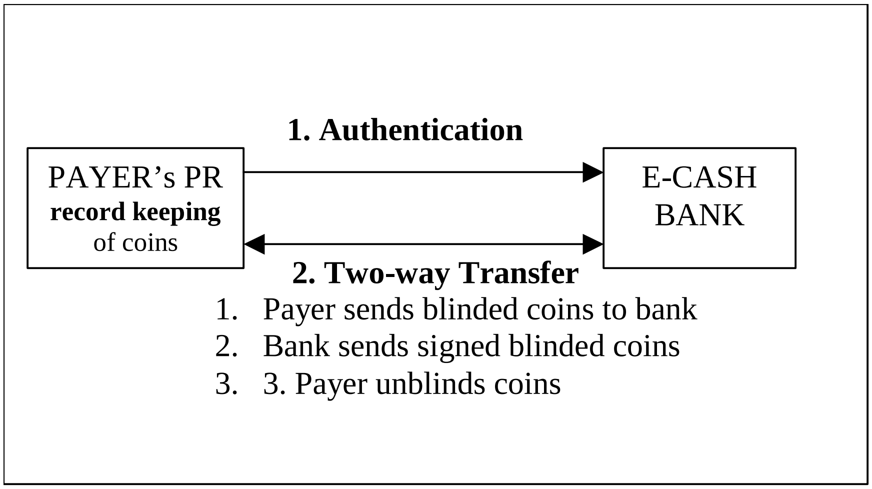
1. Authentication
PAYER's PR
record keeping
of coins
E-CASH
BANK
2. Two-way Transfer
1. Payer sends blinded coins to bank
2. Bank sends signed blinded coins
3. 3. Payer unblinds coins

Figure 5-3: Withdrawal of coins in DigiCash with primitives

Figure 5-3 shows how the withdrawal of coins could be done with primitive. First, users would need to authenticate themselves to the bank. Then they would prepare blinded coins and send them to the bank for signature. The bank would return them signed, and payers would unblind the coins to obtain valid coins that they can spend later.

It uses three primitives: *authentication*, *transfer* and *record keeping*. *Authentication* could be any kind of authentication supported by both parties. *Transfer* is a two-way transfer, and needs to perform some cryptographic operations leading to higher computational needs. *Record keeping* involves keeping track of created coins and spent coins. It is done locally by the payer, because the bank does not know the serial numbers of issued coins. Other primitives are not involved.

Description of the primitives

In this section, we will only consider the actual transaction between a payer and a payee where value is transferred. The payer is assumed to have already withdrawn coins from the bank, and is ready to pay the payee.

There is no *authentication* by the payer. However, when the payer needs to withdraw coins from the bank, authentication is required. There is no *authorization* either: in effect, the burden of authorizing the payment is on the payee, who must check coins against double spending.

The *transfer* primitive is in charge of sending the coins to the payee. It is a one way electronic transfer. Coins are encrypted with the bank’s public key to prevent them being stolen in transit and to prevent the merchant to tamper with them. Additional secret information is added to provide the user with the ability to later prove that she did the payment in the case of a dispute (see (O’Mahony et al., 2001, pp. 172-186) for more information). Therefore, the transfer primitive is fairly complex and requires several computations of signatures and hash functions.

The *record keeping* primitive keeps a record of the coins sent (their serial number and their secret information) and of the payees they were given to. This allows the payer to prove later that she spent the coins and what she spent them for in case of a dispute.

There is no *aggregation*.

The *timing* primitive is in charge of redeeming coins to the bank before they expire and withdrawing new coins. It is not used during a particular transaction, but it is needed as a part of the client implementation.

Analysis of the attributes of the primitives

Table 5-1 lists the primitives involved in the client implementation.

Table 5-1: Primitives’ attributes of a DigiCash payer

	Transfer: one way electronic transfer	Record keeping: local database	Timing	**Result of combination**
Computation	High	Low	High	High
User interface	N/A	N/A	N/A	N/A
Communication	Low	None	Low	Low
Time	Quick, depends on the link	Quick	Quick	Quick
Fixed costs	None	None	None	None
Transaction cost	Low	Low	Low	Low
Fraud risk	Low	None	Low	Low
Transaction Time	Quick	Quick	Quick	Quick
Non refutable	Yes	N/A	N/A	Yes
Security	High	N/A	High	High
Unobtrusive	Yes	Yes	Yes	Yes

Privacy analysis

On the client's side, due to the perfect anonymity provided by electronic cash systems, all primitive are anonymous and there is no information leakage.

On the payee's side

We assume here the function of the payee. The payee is supposed to have already an account at the bank, and is ready to receive a payment from the payer. The payee must have completed a phase of registration that is not described here and that could require physical interaction as in the case of the payer.

Description of the primitives

There is no *authentication*. DigiCash does not require the payer to be authenticated; the payer remains anonymous. However, the payee may have to authenticate herself when depositing the coins at the bank. This is done by signing the payment made by the payer and encrypting the message with the bank's public key. This can be considered as part of the *Authorization* primitive.

Authorization: The payee has to authorize the payment by forwarding the coins to the bank. If the coins are valid, the bank returns the result to the payee, credit the payee's account, and the payment is authorized.

The *transfer* primitive is in charge of receiving coins from the payer. It is a one way electronic transfer. Coins are prepared by the payer and do not require additional processing by the payee before being passed to the *authorization* primitive.

The *record keeping* primitive keeps a record of the coins received and authorized.

There is no *aggregation*.

There is no *timing* primitive.

Analysis of the attributes of the primitives

Table 5-2 lists the primitives involved in the payee implementation.

Table 5-2: Primitives' attributes of a DigiCash payee

	Authorization	Transfer: one way electronic transfer	Record keeping: local database	**Result of combination**
Computation	High	Low	Low	High
User interface	N/A	N/A	N/A	N/A
Communication	High	Low	None	High
Time	Quick, depends on the response from the bank	Quick, depends on the link	Quick	Quick
Fixed costs	None	None	None	None
Transaction cost	Low (fee from the bank?)	Low	Low	Low
Fraud risk	Low	Low	None	Low
Transaction Time	Quick	Quick	Quick	Quick
Non refutable	Yes	Yes	N/A	Yes
Security	High	High	N/A	High
Unobtrusive	Yes	Yes	Yes	Yes

Privacy analysis

On the payer's side, the only piece of information released is the merchant ID and a description of the order. This information cannot be accessed by the bank, because it is hashed before being released to the bank. The aim of this point of detail is to provide dispute resolution: in case of a dispute, the payer can prove to the bank what was ordered, because the hash value of the order's description will match what was earlier sent to the bank.

The privacy properties are summarized in Table 5-3. Information released because of the *transfer* primitive is in italics, information released because of the *authorization* primitive is not. The *record keeping* primitive does not need to release any information to the outside world.

Table 5-3: Privacy characteristics of a DigiCash transaction

			WHAT THEY KNOW				
			Identity of buyer	Identity of seller	Amount of transaction	Time and place of transaction	Nature of transaction
WHO KNOWS	Buyer	Identifiable	N/A	*Merchant ID*	Everything	Everything	Everything
		Traceable	N/A	Nothing			
	Seller	Identifiable	Nothing	N/A	Everything	Everything	Everything
		Traceable	Nothing	N/A			
	Bank	Issuer	Nothing	Nothing	Nothing	Nothing	Nothing
		Acquirer	Payer's bank	Merchant ID	Amount	Time if on-line	Nothing
	Observer		Nothing	Nothing	Nothing	Time	Partial

5.1.3 Examples of modifications of these primitives

In this section, modifications of the primitives described at the previous section are proposed. These modifications will slightly modify the behavior of the primitives, thus modify the behavior of the local implementation of the scheme on the payer's side or on the payee's side.

Privacy of the payer

DigiCash provides a perfect anonymity to the payer. However, the payer may prefer for diverse reasons to be not anonymous. For instance, users could get cheaper prices or additional services if they release the name. Another case could be that the merchant wishes to ensure the user's identity for a specific reason, such as verifying the right to access some information. Although anonymity is preferred in most cases, it can be that it is an undesirable property. In this case, the transfer primitive could be modified so that it sends some identity along with the payment. This identification can be partial, like a pseudonym, a birth date, an address, last name, or any other information. The DigiCash system would then be selected only for other reasons because it is preferred by one or more of the parties, and its privacy properties can be changed to fit the wishes of users.

Authorization by the payee

Depositing coins to the bank has a cost. The bank may charge a fee per coin, and the communication and computation involved have a cost. If the payee wants to reduce marginal costs relative to a payment (in particular in the case of a payment of small value), she could decide to deposit coins later when for instance the communication is cheaper, and trust the payer to give valid coins. Another strategy would be to use a probabilistic approach, and to check randomly whether the coins are valid or not randomly. The risk associated with these strategies could be cheaper than the cost of checking the coins right away.

The *authorization* primitive could implement those different strategies and use one or the other depending on the context. The choice of an implementation could modify the attributes of the payment system for the payee.

5.1.4 Summary

Attributes of DigiCash

Table 5-4: Primitives' attributes of a DigiCash payer

	Transfer: one way electronic transfer	Record keeping: local database	Timing	**Result of combination**
Computation	High	Low	High	High
User interface	N/A	N/A	N/A	N/A
Communication	Low	None	Low	Low
Time	Quick, depends on the link	Quick	Quick	Quick
Fixed costs	None	None	None	None
Transaction cost	Low	Low	Low	Low
Fraud risk	Low	None	Low	Low
Transaction Time	Quick	Quick	Quick	Quick
Non refutable	Yes	N/A	N/A	Yes
Security	High	N/A	High	High
Unobtrusive	Yes	Yes	Yes	Yes

The analysis is summarized in Table 5-4. Therefore, a DigiCash transaction is quick, requires a fair amount of computation and provides good security features. The privacy characteristics are very good, as this scheme provides full anonymity to the payer. DigiCash is an important system in the world of electronic payments, because it is one of the first fully anonymous payment systems. The primitive approach enables modifications of the behavior of the system. Two examples of modification have been studied: privacy and authorization. Privacy modifications are done when total anonymity is not desired, and authorization permits off-line transactions and cheaper fees in exchange for higher risks.

This decomposition is working with DigiCash, but do not provide a lot of customization. This is due to the fact that interfaces between players cannot be changed. Therefore, the system built with primitive allows only a small number of modifications. Nevertheless, this framework manages to offer several options to the payer and the payee, and these options are easy to implement and to modify.

5.2 Credit Cards

Detailed information about credit cards systems can be found in (O'Mahony et al., 2001), in (Camp, 1999).

5.2.1 Description of the scheme

Credit cards have been introduced a few decades ago.

There are two main distinct types of credit cards, which tend to overlap a lot:

- Credit cards: payments are set against a special-purpose bank account of the user, with an interest rate on unpaid balances.
- Debit cards: they are directly linked to a checking account, and work almost exactly as checks. In particular, payments can be made only if there is enough money to meet the amount of the transactions.

Both systems have a spending limit set by the user, the bank or the card issuer.

One major drawback of the credit card system is that the information needed to conduct an online transaction is the credit card number, the expiration date and the name of the cardholder. This information can be easily stolen, especially online, and this increases fraud costs. When doing an online transaction, the payer does not have to be present and to present physically the credit card, thus making fraudulent action easier. Fraud costs represent a large part of the transaction costs, which is fairly high compared to other payment systems. In this section, we will only consider online credit card transactions. This is why there have been many approaches aimed at reducing fraud costs, which would result in a reduced transaction costs and in an increased profitability of the system.

- Secure Socket Layer: this general-purpose cryptographic protocol is implemented on most browsers, and provides a way to securely exchange information between a client and a server. Using SSL is an easy way to protect sensitive communications (such as credit card numbers) on the Internet.
- SET[26] (Secure Electronic Transactions) is a major effort led by MasterCard and Visa, the two major credit card companies, along with other important players in the field of payment systems such as Microsoft and IBM. It provides a secure framework for paying online with a credit card. Due to its complexity, SET has not been deployed beyond trials, and simplified versions of this scheme are under research.
- In Europe, all credit cards[27] are equipped since 1992 with a chip performing different cryptographic operations. These smart cards are much more difficult to counterfeit than common magnetic cards. In addition, it would be very easy to authenticate the credit card owner with a smart card reader connected to the computer. However, although it greatly improves the security features of real-world transactions, it does not change anything for on-line payments until card readers are widely deployed.
- Another example of improvement of credit card security is the one time credit card, developed in particular by American Express. When an Internet purchase is to be made, the payer's application requests a credit card number from the card issuer. The generated number has a fixed validity period and can be used only once: after the transaction, the card issuer will mark the card number as invalid.

[26] http://www.setco.org/

[27] http://www.cartes-bancaires.com/GB/Pages/Accueil2.htm

Merchants (and potential observers) cannot see the difference between one of these one-time numbers and regular credit card numbers. The rest of the transaction is normal.

Credit card transaction

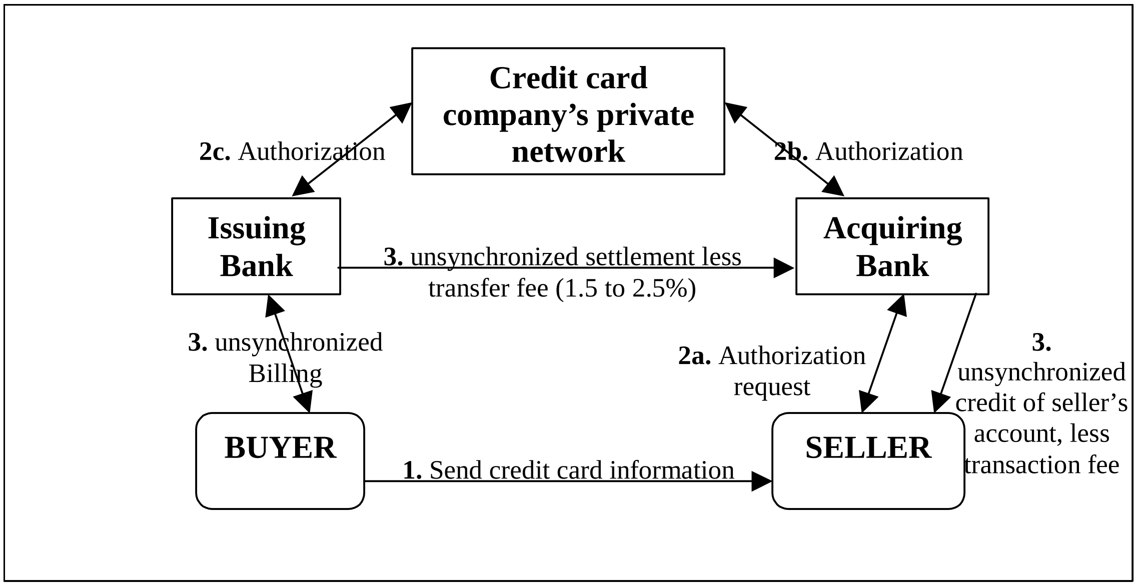
Credit card company's private network
2c. Authorization
2b. Authorization
Issuing Bank
Acquiring Bank
3. unsynchronized settlement less transfer fee (1.5 to 2.5%)
3. unsynchronized Billing
2a. Authorization request
3. unsynchronized credit of seller's account, less transaction fee
BUYER
SELLER
1. Send credit card information

Figure 5-4: The credit card scheme

Figure 5-4 shows the complexity of a credit card transaction and the multiple parties involved. Banks that belong to the credit card association may act as card issuers to their clients who wish to have a credit card, and as acquirers to their clients who wish to accept credit cards. When the buyer pays the seller, she transmits the card's details. The seller forward this information along with the amount of the transaction to the acquiring bank, which deals with the credit card association's network and credits the merchant's account if the transaction is authorized. A fee is deducted from the amount paid to the payee.

If the issuing bank and the acquiring bank are the same, the credit card network does not charge anything for the transaction.

Credit card transactions are very close to check transactions. However, there are two important differences. The first is that all transactions go through the private network of the credit card company, which links all banks together. The second difference is that the payer does not order a fund transfer or write a check; the payer actually gives all information that the seller needs to conduct the transaction, and the seller makes the payment himself.

The latter difference tends to fade. For example, SET (Secure Electronic Transactions), which is an evolution of the credit card system proposed by Visa and MasterCard, requires certificates and electronic signatures, and may be closer to on-line check-based systems than to credit card systems.

5.2.2 Primitives of an online credit card transaction

In this section, the different primitives used on the payer's side and on the merchant's side are described. Only the payer and the merchant are considered here. We assume that the payer has a valid credit card and is ready to pay the merchant. The merchant has a valid merchant account and can accept credit card payments. The initialization phase is not considered here, as it is a one-time operation requiring complex interactions between users and banks.

The same kinds of primitives are used on both sides, therefore they will be examined together and the difference will be pointed out in each case. In addition, for each primitive studied, several examples of implementations are shown, and their attributes are presented.

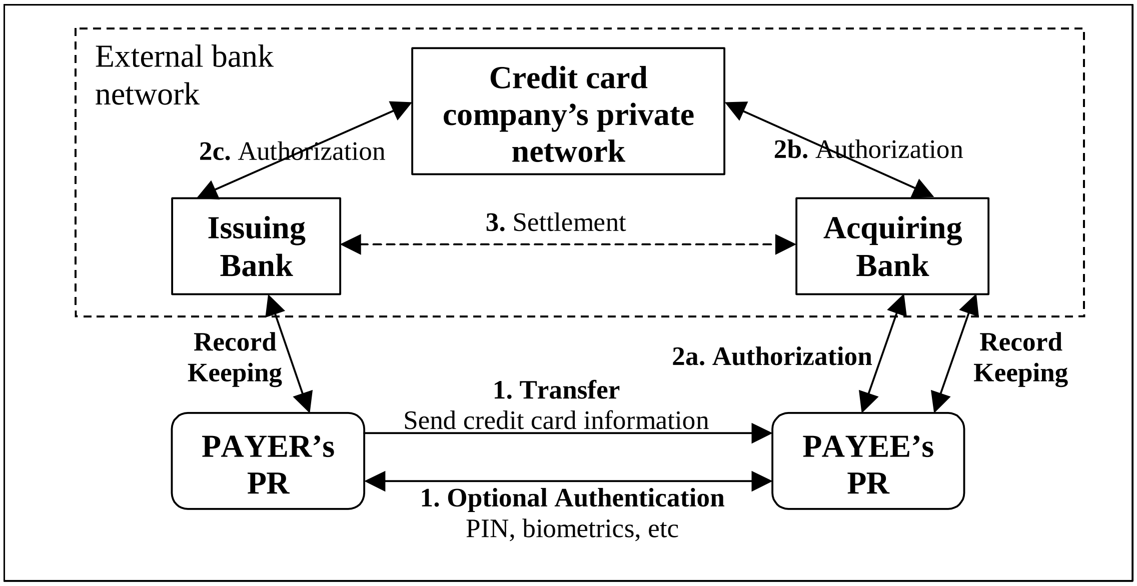

Figure 5-5: Primitives in the credit card scheme

Figure 5-5 shows the place of primitives in a credit card transaction between a payer and a merchant. The dotted box represents the network of banks external to the payer and the payee.

Authentication

Depending on the context, *authentication* may be performed via a Personal Identification Number, a handwritten signature, or may not exist. We will consider online payments only. Three ways of conducting this authentication are presented:

- Authentication with a PIN: before using the card, the user has to enter a PIN code. The user could also have the PIN code memorized in the Personal Router, or use authenticating devices like a signet ring for example.
- Authentication with biometrics: users need a biometric authenticating device, and authenticate themselves when needed.
- No authentication; this is the current situation on the Internet. No authentication is necessary before the card details are sent to the merchant.

Table 5-5 compares the different possible implementations of primitive *Authentication*.

Table 5-5: Attributes of primitive *Authentication*

	Authentication with PIN code	Authentication with biometrics	Authentication by credit card number
Computation	Low	High	None
User interface	Keyboard or any device used to enter a PIN	Biometrics reader	None
Communication	None	None	None
Time	Quick	Quick	Quick
Fixed costs	None / low	High	None
Transaction cost	N/A	N/A	N/A
Fraud risk	Low	Low	High
Transaction Time	Medium	Medium	Quick
Non refutable	N/A	N/A	N/A
Security	Medium	High	Low
Unobtrusive	No (enter PIN)	No (biometrics)	No (enter name and card number)

In terms of privacy, those three implementations are taking place exclusively locally on the payer's computer. Only the output of the operation may be sent out to the merchant and the bank. Therefore, no information is released to any other party.
On the merchant side, *Authentication* is used only when the merchant logs on the credit card network.

Authorization

Authorization is done exclusively by the merchant. This corresponds to the operation 2 on Figure 5-4. The client does not need to perform this operation. *Authorization* occurs when the merchant receives the details of the payer and submits them to the credit card network. The merchant receives then a response, indicating whether the payment is authorized.
Authorization can be performed in different ways. The merchant can trust the user, store the credit cards details and conduct the authorization procedure later, when the merchant sends batched transactions to the acquiring bank. The *authorization* is off-line in this case. The merchant could also check that the transaction is valid on-line before completing the transaction. Off-line authorization may be cheaper than on-line

authorization, and may be preferred in the case of small value payments. In this case, a floor limit may be set, where any transaction exceeding this limit requires an on-line authorization.

Table 5-6: Attributes of primitive *Authorization*

	On-line authorization	Off-line authorization
Computation	Low	Low
User interface	N/A	N/A
Communication	Low	None (occurs later)
Time	Medium	Quick
Fixed costs	High	Low
Transaction cost	Medium	Low
Fraud risk	Low	High
Transaction Time	Medium	Quick
Non refutable	Yes	No
Security	High	Low
Unobtrusive	Yes	Yes

This primitive does not influence privacy, except that the time of the transaction cannot be known precisely by other parties than the payer and the merchant if the authorization is off-line.

Transfer

Transfer transfers from the payer to the payee the data necessary to conduct the credit card transaction; only a one-way transfer is necessary. This information is usually the card number, the expiration date, and the name of the cardholder. This data is critical, and must be protected as much as possible against fraudulent action. It is a one-way transfer. Two types of *transfer* could be used: encrypted transfer or not encrypted transfer. Today, transfers of credit card numbers are almost always encrypted, mostly using SSL. This is a simple protection and the additional costs of encryption are much lower than the risks of transferring a credit card number in the clear.

Another possible implementation of the *transfer* primitive would be to use a one-time credit card number system (such as the system proposed by American Express[28]) if this service is available. It requires only a modification of the payer's client, but it is invisible for the merchant.

Table 5-7 lists the primitives involved in the client implementation.

Table 5-7: Attributes of primitive *Transfer*

	Non-encrypted Transfer	Encrypted Transfer	One time credit card number
Computation	Low	Medium	High
User interface	N/A	N/A	N/A
Communication	Low	Medium	High
Time	Quick	Quick	Medium, depends on server response's time
Fixed costs	None	None	None
Transaction cost	Low	Low	Low
Fraud risk	Very high	Low	Low
Transaction Time	Quick	Quick	Quick
Non refutable	N/A	N/A	N/A
Security	Low	Medium	High
Unobtrusive	Yes	Yes	Yes

Privacy

Table 5-8 analyzes the privacy characteristics of the *transfer* primitive in the first two cases.

[28] http://www.americanexpress.com/

Table 5-8: Privacy characteristics if *transfer* is encrypted or not encrypted

			WHAT THEY KNOW				
			Identity of buyer	Identity of seller	Amount of transaction	Time and place of transaction	Nature of transaction
WHO KNOWS	Buyer	Identifiable	N/A	Merchant ID	Everything	Everything	Everything
		Traceable	N/A	Merchant ID			
	Seller	Identifiable	Name/credit card number	N/A	Everything	Everything	Everything
		Traceable	Name/credit card number	N/A			
	Bank	Issuer	Everything	Everything	Amount	Time	Nothing
		Acquirer	Name/credit card number	Everything	Amount	Time	Nothing
	Central authority		Name/credit card number	Everything	Amount	Time	Nothing
	Observer		Nothing if encrypted, Name/credit card number if not	Merchant ID	Nothing if encrypted, Amount if not	Time	Partial

If the *transfer* primitive is using the one-time credit card system, then the privacy analysis looks the same, except that the credit card number that is revealed loses any interest after the transaction. In effect, as this number can only be used once, they lose all value after the first use.

Record Keeping

Record keeping is done for every transaction the user makes. This primitive is in charge of storing the details of the transaction, such as amount, date, merchant's ID. It is particularly important in credit card systems due to the relatively high fraud risks and of the generally high payment values.

Record keeping can be done either by the credit card holder locally, or the credit card company can be trusted to do it for the user. It is also possible to use both systems and compare the results to detect an error or a fraud.

Table 5-9: Attributes of *Record Keeping*

	Local database	Distant database
Computation	Low	Low
User interface	N/A	N/A
Communication	None	Medium
Time	Quick	Medium
Fixed costs	None	High (subscription to the provider)
Transaction cost	None	Low
Fraud risk	None	Low if encrypted
Transaction Time	Quick	Quick
Non refutable	N/A	N/A
Security	N/A	High if encrypted
Unobtrusive	Yes	Yes

This primitive has little impact on privacy, because the credit card network keeps record of transaction systematically. Therefore, the only additional information released when the user is using a distant database is what information is she interested in and how often she checks this information, which is of little interest. Nevertheless, a distant database could be hacked and personal information could then be in wrong hands. The database must be very protected in order to prevent anybody to access the private data held.

Aggregation

There is no *aggregation* in a credit card system at the level of the payer. If the *authorization* is off-line, the merchant will submit a lot of transactions at the same time, but each transaction will be separated from the other and will not be really aggregated.

Timing

Timing is not involved in credit card systems, except when the expiration date is reached.

5.2.3 Summary

Attributes of credit card systems

Table 5-10 summarizes the attributes of the most common implementations of each primitive. Table 5-11 presents the attributes of the merchant. Table 5-12 lists the privacy characteristics of the credit card system.

Table 5-10: Attributes of a credit card transaction for the payer

	Authentication by credit card number	Encrypted Transfer	Distant database	**Result of combination**
Computation	None	Medium	Low	Medium
User interface	None	N/A	N/A	None
Communication	None	Medium	Low	Medium
Time	Quick	Quick	Quick	Quick
Fixed costs	None	None	High (subscription to the provider)	High
Transaction cost	N/A	Low	Low	Low
Fraud risk	High	Low	Low if encrypted	High
Transaction Time	Quick	Quick	N/A	Quick
Non refutable	No	No	No	No
Security	Low	High	High if encrypted	Low
Unobtrusive	No (enter name and card number)	Yes	Yes	No (enter name and card number)

Table 5-11: Attributes of a credit card transaction for the merchant

	On-line authorization	Encrypted Transfer	Distant database	**Result of combination**
Computation	Low	Medium	Low	Medium
User interface	N/A	N/A	N/A	None
Communication	Low	Medium	Low	Medium
Time	Medium	Quick	Quick	Medium
Fixed costs	High	None	High (subscription to the provider)	High
Transaction cost	Medium	Low	Low	Medium
Fraud risk	Low	Low	Low if encrypted	Low
Transaction Time	Medium	Quick	N/A	Medium
Non refutable	Yes	No	No	No
Security	High	High	High if encrypted	High
Unobtrusive	Yes	Yes	Yes	Yes

Table 5-12: Privacy characteristics a the credit card system

			WHAT THEY KNOW				
			Identity of buyer	Identity of seller	Amount of transaction	Time and place of transaction	Nature of transaction
WHO KNOWS	Buyer	Identifiable	N/A	Merchant ID	Everything	Everything	Everything
		Traceable	N/A	Merchant ID			
	Seller	Identifiable	Name/credit card number	N/A	Everything	Everything	Everything
		Traceable	Name/credit card number	N/A			
	Bank	Issuer	Everything	Everything	Amount	Time	Nothing
		Acquirer	Name/credit card number	Everything	Amount	Time	Nothing
	Central authority		Name/credit card number	Everything	Amount	Time	Nothing
	Observer		Nothing if encrypted, Name/credit card number if not	Merchant ID	Nothing if encrypted, Amount if not	Time	Partial

Credit card systems are used commonly on the Internet and for many types of electronic transactions. This system offers many different ways to personalize the transaction, resulting in different attributes. The main modifications can be done on *Authentication*, *Authorization* and *Transfer*. This can change the speed of the transaction and the security / fraud attributes of the system.

Privacy is almost not affected by these modifications, due to the fact that the interactions between the merchant and the credit card network involve exchanging a lot of information about the payer. To deeply modify privacy attributes, the system needs to be redesigned, which is a much more difficult problem. This is due to the fact that interfaces and messages exchanged cannot be changed a lot, which limit customizations. SET tries to address this issue by redesigning the interfaces between the payer and the merchant, and between the merchant and the network. This again is a limitation of the primitive. However, designing the payer's and the payee's software with primitives makes it much easier to modify their behavior.

5.3 PayPal

PayPal[29] is a company that was created in 1998. It serves about 15 million registered users. It is an account-transfer system.

The idea is that users have a PayPal account, maintained by PayPal, on which they can add money by paying off-line with their credit card. Each account earns a rate of return on the account balance. This helps money to stay within the system.

Anyone can send money to anybody by just giving to PayPal the recipient's email address and the amount to be transferred. PayPal then sends the recipient an email in which there is an identification code. If the recipient is not yet registered on PayPal, he has to create a new PayPal account to retrieve the money.

It is possible to recover "real" money from the PayPal server through checks or direct deposit transactions. A personal account is free; nevertheless, users have to choose a premium service if they wish to send more than $1000 per six months to their bank account.

[29] http://www.paypal.com

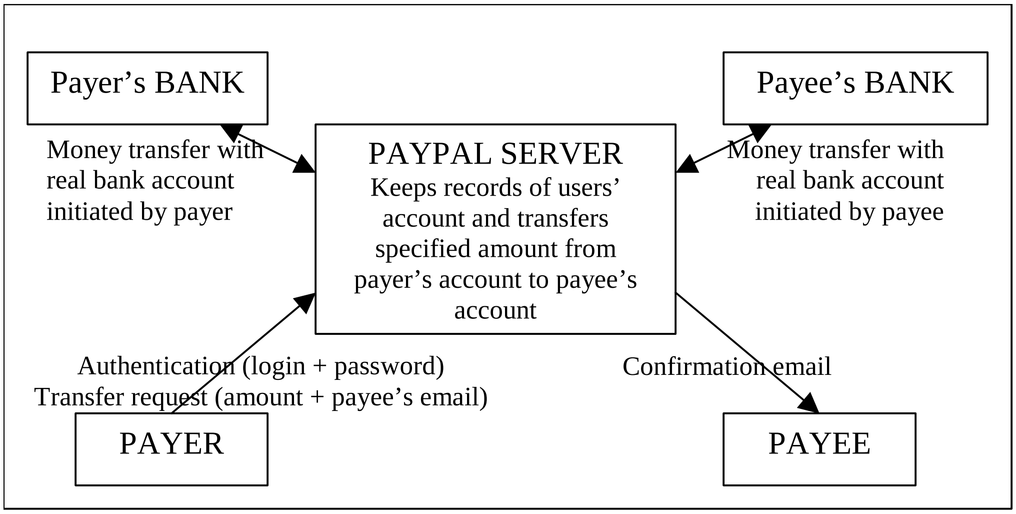
Payer's BANK
Payee's BANK
Money transfer with real bank account initiated by payer
PAYPAL SERVER
Keeps records of users' account and transfers specified amount from payer's account to payee's account
Money transfer with real bank account initiated by payee
Authentication (login + password)
Transfer request (amount + payee's email)
Confirmation email
PAYER
PAYEE

Figure 5-6: The PayPal scheme

Figure 5-6 shows how the PayPal schemes works. In the PayPal system, every transaction is made in PayPal money. Users can transfer this money to or from their bank account. However, PayPal encourages users to keep as much money as possible on their account. Because this creates strong positive externalities, PayPal can get interest from the money deposited and avoids making repetitive money transfer to users' real bank accounts. PayPal is fairly secure because every payment is made through the PayPal's server. When a payer pays a payee, PayPal directly credits the payee's account and sends an email to the payee to acknowledge the payment. The payee does not have to do anything to accept a payment. Therefore, the money transfer is done only on PayPal servers, and the payee cannot access personal information of the payer. Moreover, only PayPal knows financial details of the payer.

The link to the "real world" is done via users' real bank account. For security, PayPal can make a money transfer only to the user's bank account.

PayPal is a very simple way of sending money between two individuals. It is symmetric (payer and payee's roles are interchangeable) and relatively easy to use. It is one of the few electronic transaction systems that has had a success compared to other transaction schemes designed for the Internet. Most of PayPal transactions come from eBay.com.

PayPal can handle small payments down to one penny. Although PayPal allows transfers of any amount, this system would not be efficient if too many payments were of small value. In effect, it is not designed for handling small value payments. If PayPal were to be used as a micropayment scheme, it would have to deal with too many requests, and this would be inefficient. Therefore, it is not fitted to be a micropayment scheme.

5.3.1 Primitives of a PayPal transaction

A PayPal transaction is very asymmetric: the payer has to log in and to make the payment, while the payee has only to receive an email. This email does not convey any critical information. It is only an acknowledgement of the payment sent to the payer. The loss of the email does not jeopardize the payment.
In this section, we will go over the primitives involved in a PayPal transaction for the payer and the payee.

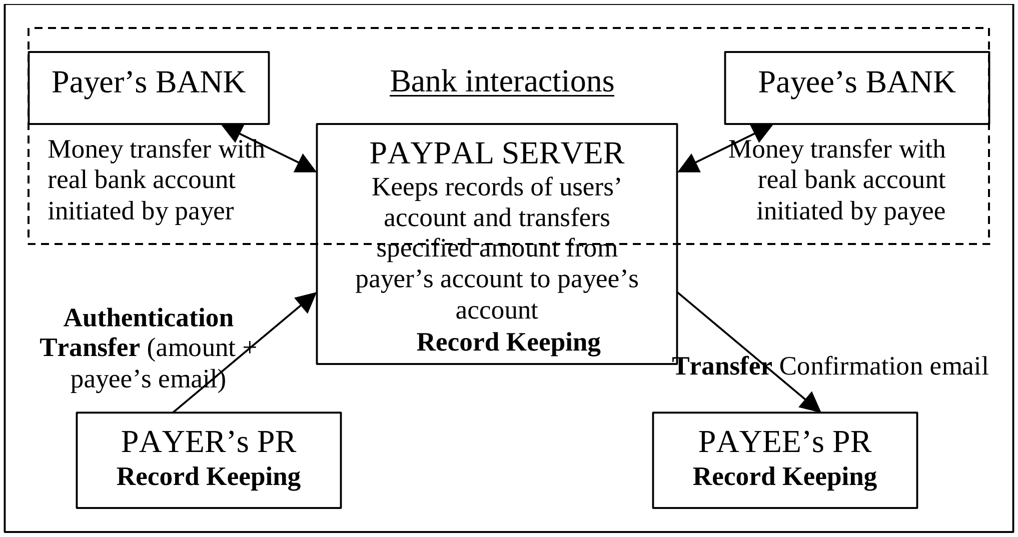
Payer's BANK
Bank interactions
Payee's BANK
Money transfer with real bank account initiated by payer
PAYPAL SERVER
Keeps records of users' account and transfers specified amount from payer's account to payee's account
Record Keeping
Money transfer with real bank account initiated by payee
Authentication
Transfer (amount + payee's email)
Transfer Confirmation email
PAYER's PR
Record Keeping
PAYEE's PR
Record Keeping

Figure 5-7: The PayPal primitives

Figure 5-7 shows the roles of primitives in a PayPal transaction. In the following sections, we will go over the primitives involved in a PayPal transaction between a payer and a payee. Only the payer and the payee will be considered here. The payer is assumed to have already a PayPal account and to have enough money on the account to complete the transaction. The initialization of the system involves human interaction and is performed only once. It is not examined here.

Authentication

Authentication is done via a login and a password on a SSL[30] connection. It could be possible to modify this authentication to login automatically (the client implementation knows the login / password information and can login on the user's behalf automatically). The user may choose this solution if the client is going to conduct a lot of transactions on the user's behalf, and the user does not want to be prompted for a login and a password repeatedly. There is of course a trade-off with security, because if the device is stolen, the thief can send himself money from the user's account.

It would be also possible to strengthen the *authentication* by requiring a stronger authentication from the user, like biometrics. In this case, the device would ask the user for the authenticating action and would login with the user's login and password if the authentication was successful. Other types of devices, like signet rings, could also be used.

Table 5-13: Attributes of primitive *Authentication*

	Authentication with PIN code	Automatic Authentication	Strengthened authentication
Computation	Low	High	High
User interface	Device to enter login and password	None	Depends, e.g. fingerprint reader
Communication	Low	Low	Low
Time	Medium	Quick	Medium
Fixed costs	None	None	Device's cost
Transaction cost	None	None	None
Fraud risk	Low	Medium	Low
Transaction Time	Medium	Quick	Medium
Non refutable	No	No	No
Security	Medium	Low	High
Unobtrusive	No (enter login and password)	Yes	No

[30] SSL: Secure Socket Layer, used to secure communications on the Internet

On a privacy standpoint, the payer cannot learn anything at this point. However, PayPal knows very well who was authenticated and can trace the user to personal information that is already stored on PayPal servers.

Authorization

On the payer’s side, there is no real *authorization*. On the payee’s side, the payment is authorized as soon as PayPal sends a confirmation email. This confirmation emails includes the email address of the payer, which is the only information accessible by the payee.

Transfer

On the payer’s side, *transfer* represents the transfer of the amount and of the email address of the payee.

On the payee’s side, the *transfer* represents the reception of the confirmation email.

As the central authority, PayPal needs to know all the payment details such as payer, payee, amount, and time. However, PayPal cannot learn anything about the nature of the transaction.

Table 5-14: Privacy characteristics of the *transfer* primitive of PayPal

			WHAT THEY KNOW				
			Identity of buyer	Identity of seller	Amount of transaction	Time and place of transaction	Nature of transaction
WHO KNOWS	Buyer	Identifiable	N/A	Payee’s email	Everything	Everything	Everything
		Traceable	N/A	Payee’s email			
	Seller	Identifiable	Payer’s email	N/A	Everything	Everything	Everything
		Traceable	Payer’s email	N/A			
	Bank	Issuer	Aggregated flow of money	Nothing	Amount	Time	Nothing
		Acquirer	Nothing	Aggregated flow of money	Amount	Time	Nothing
	Central authority (PayPal)		Everything	Everything	Amount	Time	Nothing
	Observer		Payer’s email if email not encrypted	Payee’s email if email not encrypted	Nothing	Time	Nothing

Record keeping

As in other cases, *record keeping* can be made online or offline, or both. PayPal keeps a record of all transactions. Users can choose to keep a local record for each transaction, or to trust PayPal to do it for them.

Table 5-15: Attributes of *Record Keeping*

	Local database	Distant database
Computation	Low	Low
User interface	N/A	N/A
Communication	None	Low
Time	Quick	Slow
Fixed costs	None	None
Transaction cost	None	Low
Fraud risk	None	Low
Transaction Time	Quick	Slow
Non refutable	N/A	N/A
Security	N/A	High
Unobtrusive	Yes	Yes

As in the credit card system, privacy is not much affected by this primitive. PayPal learns a lot about actions of users. PayPal knows about who users transact with, when, what the amounts are. Nevertheless, PayPal cannot learn anything about the nature of the transaction.

The more users transact with PayPal, the more precise are users' profiles accessible to PayPal. Moreover, if the database is compromised, the personal information will not be private anymore. However, there is no way to prevent PayPal to build this database. PayPal even need this database to detect and fight fraudulent actions.

Aggregation

There is no *aggregation* made by users in the PayPal system. The PayPal system could be considered aggregating payments between the deposit of money on PayPal accounts and the transfer of money to real bank accounts.

Timing

There is no *timing* made by users in the PayPal system.

5.3.2 Summary

Table 5-16 summarizes the attributes of the most common implementations of each primitive. Table 5-17 lists the privacy characteristics of the PayPal system.

Table 5-16: Attributes of PayPal

	Authentication with PIN code	Distant database	**Result of combination**
Computation	Low	Low	Low
User interface	Device to enter login and password	N/A	Keyboard
Communication	Low	Low	Low
Time	Medium	Medium	Medium
Fixed costs	None	None	None
Transaction cost	None	None	None
Fraud risk	Low	Low	Low
Transaction Time	Medium	Medium	Medium
Non refutable	No	N/A	No
Security	Medium	High	Medium
Unobtrusive	No (enter login and password)	Yes	No (enter login and password)

Table 5-17: Privacy characteristics of PayPal

			WHAT THEY KNOW				
			Identity of buyer	Identity of seller	Amount of transaction	Time and place of transaction	Nature of transaction
WHO KNOWS	Buyer	Identifiable	N/A	Payee's email	Everything	Everything	Everything
		Traceable	N/A	Payee's email			
	Seller	Identifiable	Payer's email	N/A	Everything	Everything	Everything
		Traceable	Payer's email	N/A			
	Bank	Issuer	Aggregated flow of money	Nothing	Amount	Time	Nothing
		Acquirer	Nothing	Aggregated flow of money	Amount	Time	Nothing
	Central authority (PayPal)		Everything	Everything	Amount	Time	Nothing
	Observer		Payer's email if email not encrypted	Payee's email if email not encrypted	Nothing	Time	Nothing

PayPal is a system based on a single central authority, and is therefore very difficult to customize without modifying the protocols set by the central authority. Nevertheless, it is possible to modify the *authentication* and the *record keeping* primitives so that they act differently and have different attributes. The success of PayPal is due to its simplicity. The complexity of PayPal is in its sophisticated payment server, but not in the interactions with users. Therefore, the interfaces with this server are well defined, and this simplicity is also the reason why only few modifications can be done.

5.4 Conclusions

In this chapter, three important types of electronic payments have been decomposed into the primitives defined in chapter 4. This decomposition allows modifications of some of these primitive and of their attributes. DigiCash supports some modifications of its *transfer* and *authorization* primitives. Credit card systems supports a lot more modifications, in particular in the primitives *authentication, authorization, transfer* and *record keeping*. PayPal, a successful payment system on the Internet, can support modifications of its *authorization* and *record-keeping* primitives.

Table 5-18 compares the attributes of the three payment systems examined in this chapter: DigiCash, credit cards and PayPal.

Table 5-18: Comparisons of attributes of DigiCash, Credit Cards and PayPal

	DigiCash	Credit Cards	PayPal
Computation	High	Medium	Low
User interface	N/A	None	Keyboard
Communication	Low	Medium	Low
Time	Quick	Quick	Slow
Fixed costs	None	High	None
Transaction cost	Low	Low	None
Fraud risk	Low	High	Low
Transaction Time	Quick	Quick	Slow
Non refutable	Yes	No	No
Security	High	Low	Medium
Unobtrusive	Yes	No (enter name and card number)	No (enter login and password)

DigiCash requires more computation power because of the cryptographic features of this scheme. This provides also greater security. Credit Cards and PayPal requires input from

the user. This can be modified by changing the implementation of the primitives according to the previous sections.
When comparing Table 5-3, Table 5-12, Table 5-17, which are respectively listing the privacy properties of DigiCash, Credit cards an PayPal, we can balance the type of information released to some of the parties. DigiCash has of course the better privacy record, because of its design allowing fully anonymous transactions. PayPal users release information mainly to the PayPal server, thus providing fairly good privacy properties, if users trust the central server. Credit cards are not doing so well because of the information that the payer has to release to the merchant before a transaction (name and credit card number), letting the merchant learn a lot of personal details about the user.

The framework of Primitives provides a good way to analyze payment systems and their characteristics. This decomposition seems to work well in very different systems. Complex systems such as credit cards support many ways to conduct a payment, and therefore are much easier to modify thanks to the primitives. However, simpler systems, or very rigid systems that do not allow much personalization are much more difficult to modify. This is due to the fact that the framework presented here does not allow modification of the scheme on a high level. It is limited to the devices of individual players of a transaction. Interfaces have to remain unchanged, and this limits greatly the scope of modifications. Deeply modifying attributes of these schemes would require a modification of the complete system and not only of the primitives of the payer or the payee. The interfaces would need to provide more flexibility in order to allow more possible implementations and more flexibility for users. The strength of primitives lies in the possibility to dynamically modify the characteristics of a system depending on user preferences. There would be two ways to extend this framework: first, it is possible to design payment systems in such a way that multiple interfaces between parties are possible. In this case, the parties would have to first negotiate to decide which interface they will use during the transaction, and then decide which implementation corresponding to the chosen interface they will use internally. Second, it may be possible to improve the framework of primitive to allow automatic negotiation of the interfaces.

In conclusion, the framework of primitives seems to be a good way to analyze and to compare payment systems, and allow users to easily conduct small changes to their system, thanks to the modularity and the description of differences it provides. Nevertheless, it lacks some power to greatly modify payments attributes, as it cannot act on the interfaces regulating exchanges between parties.

6 CONCLUSION

This thesis has addressed the problem of supporting a wide range of payment schemes in a single wireless communications device, the Personal Router. The approach taken has been the development of a set of building blocks, or primitives. To conclude this thesis, this chapter first summarizes the major findings of this work and proposes several areas of future work that would extend and improve the analysis presented here.

6.1 Findings

This section gives a summary of key results of this thesis.

Primitives can support three of the payment systems

The framework of primitives, presented in chapter 4, is a response to the issues raised in chapters 2 and 3. Through a modularization of elemental operations of payment systems, it is possible to provide a better way of defining and comparing payment systems.
There are two major advantages to this approach. First, comparing two payment systems is difficult, and would be even more difficult to do automatically. This framework provides a way to compare sub-elements of payment systems, thus making comparisons much easier to perform. Second, it becomes possible to interchange a sub-part of a payment system for another, which gives a way to easily modify the characteristics of a payment scheme. This could even lead to a way to build payment schemes as results of negotiations between payers and payees.

The primitives defined in section 4.3 are Authentication, Authorization, Transfer, Record keeping, Aggregation, and Timing.

In chapter 5, it is demonstrated that this framework can successfully decompose 3 very different payment systems, which are DigiCash, the credit card system, and PayPal.

The framework of primitives provides a way to modify payment systems on local user's device…

In chapter 5, three payment systems are reviewed and broken down into their primitive elements as defined in chapter 4. These three payment systems are DigiCash, the credit card system, and PayPal.

DigiCash is an electronic cash system, providing full anonymity to users. After the study of its primitive both on the payer's and the payee's side, two main modifications of the primitives have been pointed out: first, a modification of the transfer primitive, allowing to change the privacy characteristics of the scheme, following the user's preferences when full anonymity is not a desirable property; second, a modification of the authorization primitive of the payee providing a way to receive an off-line payment in exchange for a higher fraud risk.

Credit card systems are one of the most common payment systems on the Internet. As there are many different ways to use credit cards, there are also more possibilities to modify the scheme. Several implementations are presented. Several modifications concerning the primitives authentication, authorization and transfer are proposed. Privacy is not much affected by these modifications, because the scheme requires that users present their credit card number and name to merchants, who then transmit this information to the credit card network.

PayPal is a very successful payment system on the Internet. Its size and its number of subscribers makes it one a the most important players in the market of electronic payments over the Internet. Its scheme makes it very easy to transfer money from any user to any other user; contrary to the credit card system, there is no need to be registered as a merchant to be able to receive a payment.

As PayPal is very standardized and simple, it is not easy to modify. Possible modifications are proposed for the primitives authentication and record keeping.

The framework of primitives has been successful in breaking down those three payment systems into sub-elements. The true test of primitives will be their adaptability to new payment schemes that could be invented in the future.

… But lacks strength to change the whole system, because of fixed interfaces

The framework of primitives facilitates the analysis of a scheme and the comparisons. It also allows users to select between different implementations of these primitives to slightly modify the behavior and the characteristics of payment systems.

However, as this framework is only used independently of the other parties, it lacks the possibility to deeply modify a payment scheme by involving the entire set of players in the modification process. Therefore, modifications are limited to a reduced set of possibilities. This is due to the fact that interfaces regulating the exchanges between parties and the sequence of these exchanges cannot be modified. Therefore, the behavior of the actors cannot be changed in depth.

It could be possible to modify payment systems to allow important modifications. There could be two ways to do that: first, by designing payment systems directly with the framework of primitives and pre-designed interfaces that users could choose from out of a negotiation; another method, more difficult, would be to create a framework that would let users negotiate on interfaces and interactions necessary to conduct a transaction.

Privacy is a major characteristic of payment systems; in this context, it can be more easily analyzed with the use of privacy tables

Privacy must be analyzed by the information considered and the parties accessing the information. What kind of information is released? Is it a name, a pseudonym, a user ID? Is this information obtained by the bank, by the merchant, or by an observer? Privacy issues are multidimensional. In effect, it is much too simplistic to describe a system as private or not. In fact, only a few payment schemes, such as DigiCash, can claim to be fully anonymous, thus being private.

When evaluating the privacy characteristics of a system, different properties must be defined. First, anonymity and traceability must be differentiated. Pseudonymity must also be defined. Some schemes provide a way to revoke anonymity in specific cases such as law enforcement. These terms are discussed in section 2.2.13.

Section 2.2.13 also proposes to use a privacy table, used to analyze privacy characteristics of payment systems. This table proposed has two axes. The first one lists the type of information being released, and the other one lists the party it is released to. This table is an efficient way of summarizing clearly the privacy requirements and preferences of users, and the privacy characteristics of payment systems. By comparing the user preferences with the privacy properties of schemes, it is possible to make machines reason about privacy.

However, a lot of work needs to be done in the field of smart agents and user interface to support the complexity of the information presented in this privacy table.

6.2 Future Research and Open Issues

Privacy

Chapter 3 has given an overview of the potentially difficult questions raised by the Personal Router. It also lists three areas of threats, and states some solutions that have been proposed. But the place taken by the Personal Router is much more complex than the mere addition of these three worlds. The combination of these threats and the risks of collusion between players will require a deeper analysis of the probable risks and of the solutions that can be applied. It is true that other mobile communication means such as third generation mobile networks with data capabilities will raise a number of similar questions. But the Personal Router adds the perspective of a large number of providers, which will change the threat model.

User Interface

The issue of user interface has not been approached at all in this thesis and is absolutely essential for this framework to work. In effect, in order to use this framework, users will need to define fairly precisely what their preferences are. It would be wrong to believe that users will go through the attributes defined in this thesis and precisely specify what they want. It is more likely that they will not understand the meanings of the chosen attributes. In particular, privacy, as a multidimensional attribute very sensitive to context, will be difficult to analyze.

Therefore, these attributes can only be defined through a smart user interface able to understand and learn the will of users and the context in which they are acting. The most promising approach seems to be the observation of the user's behavior. The decision maker algorithm used to decide which payment system to use and how it should be used according to users' guidelines is also an open issue that will have to be researched. This decision maker algorithm should be able to learn from the user and should have a good user interface allowing users to teach it their preferences and to control the decisions of this algorithm.

Negotiation

More generally, the Personal Router will need a negotiation interface, capable of negotiating not only for the services, in terms of bandwidth, quality of service and delay, but also for the terms of the contract (in terms of payment scheme chosen, price, information revealed, modification made to the payment scheme, etc). Once again, this negotiation agent will also need a very good user interface to let users express what they want and to simplify the complexity of the users' requirements. It is particularly important given the fact that it is likely that the Personal Router will need to negotiate and pay for connectivity without the active participation of the user.

The Framework of Primitives

Finally, this framework could be extended to include deeper changes and maybe the possibility to create new payment systems. In this thesis, it is limited to partial modifications on parts of the system. Therefore, the proposed framework cannot modify fundamental characteristics of the system. It has to be supported by a fixed structure of the payment system. This greatly limits the modularity of the framework.

It may be possible to extend the approach presented in this thesis to the whole payment system and all the players. In this case, the whole system would be able to evolve and to modify itself depending on inputs and requirements of parties engaged in the transaction. Ultimately, it could even be possible to build for each transaction a new payment scheme that would perfectly fit the wishes of the parties involved.

Composing primitives into payment systems

In the presented framework, primitives are only applied to existing payment systems. The aim is here to compare payment system, and to apply small modifications. However, it could also be possible to build payment system from this set of primitives. In this case, a composition system should be created to first implement a way to create new payment systems, and then to verify that the created payment scheme is valid and works properly.

Legal requirements for applying the framework to a payment system as a whole

Payment systems deal with money, and thus provide a highly sensitive service. Legal requirements on payment systems are already complicated. But what would happen if those payment systems could be partially modified as result of negotiation between users? Another issue is the choice of a payment system as a contract. Parties would enter in a contract, and this raises issues, among others, of digital signatures, digital legal bindings, and of choice of laws.

Such a system could not be widely accepted without a good understanding of its legal implications.

7 BIBLIOGRAPHY

7.1 Articles

Agre, P., and Rotenberg, M. (1997), *Technology and Privacy: The new Landscape*, MIT Press, Cambridge, MA, USA.

Asokan N., Janson, P., Steiner, M., and Wagner, M. (1996), *Electronic Payment systems,* IBM research report RZ2890.

Bellare, M., Garay, J. A., Hauser, R., Herzberg, A., Krawczyk, H., Steiner, M., and Tsudik, G. (2000), *Design, Implementation, and deployment of the* i*KP secure electronic payment system*, IEEE Journal on selected areas in communications, vol. 18, no. 4, April 2000.

Boucher, P., Shostack, A., Goldberg, I. (2000) *Freedom System 2.0 Architecture*, white paper, Zero-Knowledge Systems, Inc.

Brands, S. (2000), *Rethinking Public Key Infrastructures and Digital Certificates,* MIT Press, Cambridge, MA, USA.

Camp, L. J. (2000), *Trust and Risks in Internet Commerce*, MIT Press, Cambridge, MA, USA.

Camp, L. J., Harkavy, M., Yee, B., and Tygar, J. D. (1996), *Anonymous Atomic Transactions,* Proceedings of the 2nd Usenix Workshop on Electronic Commerce, November 1996, pp. 123 – 133.

Camp, L. J., Tygar, J. D., and Sirbu, M. (1995), *Token and notational money in electronic commerce*, in Proceedings of the 1st Usenix Workshop on Electronic Commerce**,** July 1995, pages 1 - 12.

de Carvalho Ferreira, L., Dahab, R. (1998), *A scheme for analyzing electronic payment systems***,** Computer Security Applications Conference, 1998. Proceedings. 14th Annual, pp. 137 - 146

Chaum, D. (1992), “Achieving Electronic Privacy,” *Scientific American* 267, no. 2 (February 1992), pp. 96 - 101.

Chaum, D., Fiat, A., and Naor M., (1988) *Untraceable Electronic Cash*, Advances in Cryptology CRYPTO '88, S. Goldwasser (Ed.), Springer-Verlag, pp. 319-327.

Clark, D. D. and Wroclawski, J. T. (2001), *The Personal Router Whitepaper*, white paper, MIT Laboratory for Computer Science.

Cox, B., Tygar, J. D., Sirbu, M. (1995), *Netbill security and transaction protocol*, In Proceedings for the 1st USENIX Workshop on Electronic Commerce, July 1995.

Cranor, L. F. (1999), *Internet privacy*, Communications of the ACM 42, no. 2 (Feb. 1999), pp. 29 - 31.

Cranor L. F. and Reagle, J. (1999), *The platform for privacy preferences*, Communications of the ACM 42, no. 2 (Feb. 1999), pp. 48 – 55.

Croker, S. (1999), *The siren songs of Internet micropayments*, iMP Magazine, issue of April 1999, http://www.cisp.org/imp/april_99/04_99crocker.htm

Etzioni O., Lau, T., and Weld, D. S. (1999), *Privacy interfaces for information management*, Communications of the ACM vol. 42, no. 10 (Oct. 1999), pp. 88 – 94.

Falkner, M., Devetsikiotis, M., and Lambadaris, I. (2000), *An overview of pricing concepts for broadband IP networks*, IEEE Communications Surveys & Tutorials, Second Quarter 2000.

Furche, A., and Wrightson, G. (1996), *Computer Money*, Dpunkt, Heidelberg, Germany.

Gabber E., Gibbons, P. B., Kristol, D. M., Matias, Y., and Mayer, A. (1999) *Consistent, yet anonymous, Web access with LPWA*, Comm. ACM 42, no. 2 (Feb. 1999), pp. 42 – 47.

Glassman, S., Manasse, M., Abadi, M., Gauthier, P., and Sobalvarro, P., (1995), *The Millicent Protocol for Inexpensive Electronic Commerce*, In World Wide Web Journal, Fourth International World Wide Web Conference Proceedings, pages 603-618. O'Reilly, December 1995.

Jakobsson, M., MRaihi, D., Tsiounis, Y., and Yung, M. (1999), *Electronic Payments: Where Do We Go from Here?*, CQRE ‘99, Lecture Notes in Computer Science 1740, pp 43 - 63.

Kindberg, T., Mowbray, M. (2000) *Privacy within CoolTown*, white paper, HP labs.

Lee, Z.-Y., Yu, H.-C., and Kuo, P.-J. (2001), *An analysis and comparison of different types of electronic payment systems*, Conference on Management of Engineering and Technology, 2001. PICMET '01, Portland. pp. 38 –45.

MacKie-Mason, J., and White, K. (1997), *Evaluating and selecting digital payment mechanisms*, in *Interconnection and the Internet*, G. Rosston and D. Waterman, eds. Lawrence Erlbaum, pp. 113 - 134.

Meeks, B. N. (1999), *The privacy hoax*, Communication of the ACM, Vol. 42, No. 2, pp. 17 - 19.

O’Mahony, D., Peirce, M., and Tewari, H. (2001), *Electronic Payment Systems for E-Commerce*, 2nd Edition, Artech House, Norwood, MA, USA.

Peirce, M., and O’Mahony, D. (1999), *Flexible real-time payment methods for mobile communications*, IEEE Personal Communications, December 1999, pp. 44 - 55.

Pfitzmann, B., Waidner, M. (1996), *Properties of payment systems: general definition sketch and classification*, IBM Research Report, RZ2823.

Rivest, R. L., and Shamir, A. (1996), *PayWord and MicroMint--Two Simple Micropayment Schemes*, CryptoBytes, volume 2, number 1 (RSA Laboratories), pp. 7 - 11.

Rotenberg, M. (2000), *The Privacy Law Sourcebook 2000, United States Law, International Law, and Developments*, Electronic Privacy Information Center, Washington, DC, USA.

Schneier, B. (1996) *Applied Cryptography, 2nd Edition: protocols, algorithms, and source code in C*, John Wiley & Sons, Inc.

Sherif, M. H., Serhrouchni, A., Gaid, A. Y., and Farazmandnia, F. (1998), *SET and SSL: Electronic payments on the Internet*, Proceedings of 3rd IEEE Symposium on Computers and Communications, ISCC '98, pp. 353 –358.

Zaba, S. (1999), *Selecting the right e-payment scheme*, Protecting Your Intellectual Property: Security, Encryption and Anti-Copy Technologies (Ref. No. 1999/083), IEE Seminar, pp 0 - 11 -10/7

United States Privacy Act of 1974 (Public Law 93-5795)
http://www.usdoj.gov/04foia/privstat.htm

United States Senate Judiciary Committee (2000), *Privacy in the Digital Age: A Resource for Internet Users*

Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, Official Journal L 281, 23/11/1995 p. 0031 – 0050:
http://europa.eu.int/eur-lex/en/lif/dat/1995/en_395L0046.html

7.2 Web sites

http://www.fcc.org

The World Wide Web Consortium http://www.w3.org
The Platform for Privacy Preferences (P3P) Project http://www.w3.org/P3P/

Electronic Privacy Information Center (EPIC): http://www.epic.org

US site on Safe Harbor (department of Commerce): http://www.export.gov/safeharbor/
List of companies adhering to the Safe Harbor framework:
http://web.ita.doc.gov/safeharbor/shlist.nsf/webPages/safe+harbor+list

http://europa.eu.int/eur-lex/en/lif/dat/1995/en_395L0046.html

http://passport.com

http://www.bell-labs.com/project/lpwa

http://www.research.att.com/projects/crowds/

http://www.anonymizer.com

http://www.setco.org/

http://www.cartes-bancaires.com/GB/Pages/Accueil2.htm

http://www.americanexpress.com/

http://www.visa.com

http://www.mastercard.com

http://www.digicash.com

http://www.epic.org

https://www.paypal.com

http://www.umts-forum.org

http://www.doubleclick.com

http://www.setco.org/

Zero Knowledge technologies: http://www.zeroknowledge.com and http://www.freedom.net

7.3 General Background

Alexandris, N., Burmester M., Chrissikopoulos V. and Desmedt, Y. (2000), *Secure linking of customers, merchants and banks in electronic commerce*, Future Generation Computer Systems, Volume 16, Issue 4, February 2000, pp. 393-401

Buttyan, L. (2000), *Removing the financial incentive to cheat in micropayment schemes*, Electronics Letters , Volume: 36, Issue: 2 , 20 Jan. 2000, pp. 132 –133

Khanh Quoc Nguyen; Yi Mu; Varadharajan, V. (1997), *Micro-digital money for electronic commerce*, Computer Security Applications Conference, 1997. Proceedings., 13th Annual , pp. 2 –8

Min-Shiang Hwang, Iuon-Chang Lin and Li-Hua Li (2001), *A simple micro-payment scheme*, Journal of Systems and Software, Volume 55, Issue 3, 15 January 2001, pp. 221-229

Romao, A.; Da Silva, M. M.; Silva, A. (2000), *Secure electronic payments based on mobile agents*, Distributed and Parallel Databases, Volume 8, Issue 4, October 2000, pp. 447-470

Seong Oun Hwang (1998), *Electronic exchange check system on the Internet*, Parallel and Distributed Systems, 1998. Proceedings. 1998 International Conference on , pp. 454 – 460

Sirbu, M.; Tygar, J.D. (1995), *NetBill: An Internet commerce system optimized for network delivered services*, Compcon '95.'Technologies for the Information Superhighway', Digest of Papers, pp. 20 –25

Turban, E.; McElroy, D. (1998), *Using smart cards in electronic commerce,* System Sciences, Proceedings of the Thirty-First Hawaii International Conference on, Volume: 4, 1998, pp. 62 -69

Zaba, S. (1999), *Selecting the right e-payment scheme*, Protecting Your Intellectual Property: Security, Encryption and Anti-Copy Technlogies (Ref. No. 1999/083), IEEE Seminar on