Methods to Improve Privacy and Loyalty in Online Commerce

Vom Fachbereich Informatik der
Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades
Doctor rerum naturalium (Dr. rer. nat.)

von

Dipl.-Inform.
Matthias Enzmann

aus Langen (Hessen)

Referenten:	Prof. Dr. Claudia Eckert Prof. Dr. Johannes Buchmann
Tag der Einreichung:	15. November 2007
Tag der mündlichen Prüfung:	19. Dezember 2007

Darmstadt, 2007
Hochschulkennziffer: D17

Wissenschaftlicher Werdegang[1]

1993 – 1999	Studium der Informatik an der Technischen Universität Darmstadt
seit 1999	Wissenschaftlicher Mitarbeiter am Fraunhofer-Institut für Sichere Informationstechnologie (SIT)

Erklärung[2]

Hiermit erkläre ich, dass ich die vorliegende Arbeit —mit Ausnahme der in ihr ausdrücklich genannten Hilfen— selbstständig verfasst habe.

[1] gemäß §20 Abs. 3 der Promotionsordnung der TU Darmstadt
[2] gemäß §9 Abs. 1 der Promotionsordnung der TU Darmstadt

Danksagung

Zuerst möchte ich Prof. Dr. Claudia Eckert für die Betreuung der vorliegenden Arbeit danken. Ihre kritischen und konstruktiven Anmerkungen und Hinweise in all den Jahren halfen mir stets noch einmal selbstkritisch die Inhalte meiner Arbeit zu reflektieren und dadurch auch den inhaltlichen Schwerpunkt dieser Arbeit immer im Auge zu behalten. Prof. Dr. Johannes Buchmann möchte ich für sein Interesse an dieser Arbeit und die Übernahme des Koreferats danken.

Mein Dank gilt außerdem meinen Kolleginnen und Kollegen am Fraunhofer-Institut SIT, vor allem aber jenen aus dem Bereich TAD, mit denen ich neben all der Arbeit auch eine Menge Spaß hatte. Ich danke im Besonderen meinen beiden Langzeitzimmergenossen Omid Tafreschi und Ruben Wolf für unzählige Hinweise, Anregungen und Hilfestellungen sowie auch dafür, dass sie mit mir den "langen Leidensweg" geteilt haben.

Zu großem Dank bin ich meinen beiden früheren Koautoren und Korrekturlesern Marc Fischlin und Markus Schneider verpflichtet, die sich durch weniger reife Versionen dieser Arbeit gekämpft und zahlreiche Verbesserungsvorschläge beigesteuert haben. Darüber hinaus bin ich Marc für die Einblicke dankbar, die er mir in alle kryptografischen Belange geben konnte, denn ohne sie hätte ich die vorliegende Arbeit in dieser Form nicht schreiben können. Den größten Einfluss auf meine bisherige wissenschaftliche Laufbahn, und damit auch auf diese Arbeit, hatte aber Markus Schneider. Seiner Unterstützung und Beharrlichkeit (nicht nur) in wissenschaftlichen Belangen, die mir als Kollege, Koautor und Freund zu Teil wurde, verdanke ich vor allem anderen, dass ich die vorliegende Arbeit zu Ende geschrieben habe.

Der letzte Dank an dieser Stelle gilt meinen Eltern, die mich stets unterstützt und bestärkt haben, meinen begonnen Weg auch zu Ende zu gehen.

Zusammenfassung

Die Einhaltung der Datenschutzerklärungen von Online-Händlern, bspw. im Hinblick auf die gesammelten Daten und deren Verwendung, kann von Kunden in der Praxis kaum überprüft werden. Die Sammlung von Nutzungsdaten an sich ist hierbei aber nicht das größte Problem, sondern dass diese Daten mit der realen Identität eines Nutzers verknüpft werden können und es Händlern somit möglich ist, Profile ihrer Kunden ohne deren Kenntnis zu erstellen. Ziel dieser Arbeit war es, Methoden und Verfahren zu entwickeln, die entweder ganz ohne personenbezogene Daten auskommen oder dazu beitragen, dass anfallende Daten nicht gegen den Wunsch der Nutzer mit deren realer Identität verknüpft werden können.

Im Rahmen dieser Arbeit wurde hierfür ein abstraktes Modell entwickelt, das es erlaubt, die Transaktionen einer Kunde-Händler-Beziehung aus Datenschutzsicht zu analysieren, um bspw. festzustellen, inwieweit die in einer Transaktion anfallenden Daten von der Identität des betroffenen Kunden entkoppelt werden können. Das Modell bezieht Phasen ein als Mittel zur Unterteilung einer Transaktion in kleinere semantische Einheiten. Bisherige Ansätze in Mehrparteienmodellen betrachteten lediglich die Nichtverknüpfbarkeit einzelner Transaktionen der beteiligten Parteien, während der hier vorgestellte Ansatz darüber hinaus geht und zusätzlich die Nichtverknüpfbarkeit von Phasen mit einbezieht, was den Datenschutz weiter stärkt.

Die weiteren Beiträge dieser Arbeit stellen Anwendungsfälle des Modells dar und befassen sich näher mit Phasen von Transaktionen, die in besonderem Maße dazu angelegt sind, einen Einblick in das Privatleben von Kunden zu erlangen und damit wesentlich zur Erstellung bzw. Anreicherung eines Kundenprofils beitragen können. Diese Phasen sind zum einen die Suchphase, in der Kunden zunächst nach Produkten Ausschau halten, dabei aber u.U. bereits Vorlieben und Abneigungen Preis geben, ohne dass sie dies beabsichtigt hätten — hierfür wurde eine Architektur vorgeschlagen, die auf die Abkopplung der Suchphase von darauf folgenden Phasen abzielt. Zum anderen wurden Phasen betrachtet, die mit digitalen Anreizsystemen und Kundenbindungsprogrammen in Zusammenhang stehen. Letztere sind aus der realen Welt in Form von Bonus- und Kundenkarten als Mittel zur Kundenbindung, aber auch zur Ausforschung des Kundenverhaltens, hinlänglich bekannt. Als

Alternative zu herkömmlichen Bonussystemen wurde ein datenschutzfreundliches System entwickelt, welches den Belohnungsaspekt von Bonuskarten berücksichtigt, aber durch die Möglichkeit einer anonymen Nutzung bewusst auf die Möglichkeit zur Profilbildung verzichtet. Das Bonussystem wurde speziell für den Online-Handel entwickelt und ist das erste, das auf technischer statt auf organisatorischer Ebene die Privatsphäre von Nutzern schützt und auch die Sicherheitsbelange von Händlern berücksichtigt. Eine andere Art von Kundenbindungssystem sind so genannte Multi-Coupon-Systeme. Das hier vorgestellte Protokoll zum Einlösen eines Teil-Coupons erlaubt es dem Kunden, gegenüber dem Anbieter anonym nachzuweisen, dass sein Multi-Coupon mindestens noch ein gültiges Coupon aufweist, wobei im Rahmen dieses Nachweises ein Coupon entwertet wird, sodass dessen erneute Einlösung nicht möglich ist. Das hier vorgeschlagene Multi-Coupon-System stellt nach Kenntnis des Autors weiterhin das erste System dar, das eine Nutzungsbegrenzung eines Berechtigungsnachweises erlaubt (im Englischen als *limited-show credential* bezeichnet) und dabei dennoch die Anonymität des Besitzers sicherstellt.

Contents

Chapter 1

Introduction

In recent years, the World Wide Web has evolved into a business platform with worldwide reach and 24h/7 service for selling various kinds of goods. Presently, more than 1 milliard people have access to this business platform and thus, are potential customers for online vendors. Many people welcome the liberation from early closings of brick-and-mortar shops and enjoy shopping around the clock. In addition, they can do their shopping from home and do not have to drive several kilometres to visit their favourite store. Internet vendors' prices also often undercut prices of brick-and-mortar shops and hence, are even more attractive to customers out for a bargain. However, this freedom of shopping and seemingly lower prices are often traded in for an important aspect of personal freedom, privacy.

Almost all Internet vendors demand their customers' names and addresses, either for payment or for shipping, and hence, purchases become personalised. By inspecting the customer's shopping basket, vendors get —at least— a glimpse of a customer's life that they would not have gotten if they were brick-and-mortar vendors. Of course, customers also pay with credit cards in brick-and-mortar shops and hence, even there vendors learn something of their customers' lives. The difference to Internet vendors, however, is that customers may pay cash instead of using their credit cards and therefore, they have the *freedom of choice* to stay anonymous and still get the desired goods. But even if they choose to use their credit cards, a vendor normally learns only what is in his customer's shopping basket. He does not learn, for instance, what other goods the customer eyed while visiting his shop. Conversely, an Internet vendor can effortlessly observe his customers' behaviour and shopping habits. He learns what goods the customer viewed, he learns how long they have been viewed, how often they have been viewed, if they have been viewed in conjunction with certain other goods, etc. In short, the online vendor practically learns everything that can be observed.

One may argue that the observation argument is also true for brick-and-mortar shops, as they often have cameras installed, e.g., to catch shoplifters. Thus, customers are being recorded by cameras even in normal supermarkets. And still, the brick-and-mortar and the Internet scenario can hardly be compared as the depth of detail of the observation and the order of magnitude of observable customers widely differ. Although this may change in the (near) future, cameras in brick-and-mortar shops today can hardly track *every move* of a single customer that walks around in the shop, as cameras are installed at fixed locations and it would require either human intervention or some kind of hand-over between cameras to automatically track a specific customer's shopping tour. But even if this is made possible, e.g., by attaching RFID tags to products and constantly monitoring them, it may still take a lot of additional human effort, or sophisticated recognition software, to determine what goods from the shelves had been eyed by the customer. In the Internet, such information is practically for free. It is effortlessly possible to record every move of *every customer* every time the customer visits the Internet store — imagine the effort for this in a brick-and-mortar shop. By collecting all this data, a vendor creates a history that allows him to recall everything that a customer has seen and done, even things which the customer had already forgotten. Such histories are usually called *profiles* and in this work, we are mainly concerned about them.

Another difference between a shopping tour in a brick-and-mortar shop and in an Internet shop is that a customer's trip can be unnoticeably and accurately observed by third parties. For instance, the customer's Internet service provider (ISP) may learn almost the same information as the Internet vendor. In fact, the ISP —or everyone else able to observe the customer's communication— may learn even more than the vendor, as the ISP may observe every Internet communication of every one of its own customers, and hence, potentially is able to create an even richer profile of the customer.[1]

So it seems the price Internet customers have to pay for flexibility (shopping hours), efficiency and convenience (no driving and home delivery), and possibly lower prices, is that they have to give up their privacy to some degree. And indeed, many are willing or at least have no choice but to pay this price, while others completely reject Internet shopping or are at least holding back. One may argue at length whether the price for this loss of privacy seems reasonable or too high, and whatever side one eventually takes in this argument, the choice will most likely be determined by several subjective factors, including cultural issues and how one perceives privacy in general. Even though this discussion is not the subject of this work, we will briefly touch this issue in Section 2.2 and we will also illustrate a few

[1] This threat could be alleviated a bit, if the *complete* communication between customers, vendors, and others would be encrypted. However, this is rarely done.

scenarios in Section 2.4 where a lack of privacy protection could have, or already had, seriously affected the lives of individuals.

In this work, we are not going to discuss the value of privacy *per se*, even though the author personally believes that privacy indeed has a value *per se*. Instead, we focus on the development of mechanisms which are primarily intended to give users more control over the disclosure of personal data related to Internet shopping. Thus, every customer shall be enabled to *decide for himself/herself* what information he/she is willing to disclose, given that this information can be withheld, i.e., the information is —strictly speaking— unnecessary to progress his/her transaction at hand.

The kind of control we aim at, however, cannot be exercised without the cooperation of the Internet vendor. That is, the approaches put forth in this work assume that the vendor also has an interest in giving his customers a guarantee of privacy that goes beyond lip service and sole trust. From the point of enforcement this cooperative approach is clearly a disadvantage compared to other privacy enhancing technologies (PETs), as a single customer can hardly force a vendor to offer more privacy-friendly shopping. However, if the group of privacy-demanding customers —actual or potential— is large enough, vendors may eventually start thinking of privacy as a competitive advantage or even as a unique selling proposition. We will get back to this point in Section 2.6. Before that, we will give an overview of some widely deployed PETs in Section 2.5 and also discuss some of their shortcomings when it comes to Internet shopping.

The main contributions of this work, however, are more technical in nature. Specifically, we have developed a model that formalises the notion of a profile and allows to analyse the creation and expansion of customer profiles, given customers' communication with a vendor. Basically, the model allows us to examine the interrelationships of some customer's transactions which may eventually be used as the basis for a consumer profile. The model itself is introduced in Chapter 3 and will be used it in Chapters 4–6 to discuss the privacy properties of the solutions presented therein.

Using our model, we can also take a deeper look into a transaction and find the finer grained sub-transactions of which the whole transaction is comprised of. We will generally refer to these sub-transactions as *phases*. Examples for such phases are searching a vendor's catalogue, ordering items, paying a bill, returning an item, etc. The main idea of this work, found in all subsequent chapters, is to prevent linking of phases either within the same transaction or between the phases of distinct transactions. Clearly, preventing linkage is not always desirable, as it is sometimes necessary, e.g., to determine if the bill for a particular purchase had been paid. However, phases and transactions are often linked, even if it is not necessary. For instance, having a link from a customer's current transaction to previous ones

will often be unnecessary and will only allow the vendor to build a profile of the customer.

The basic idea of preventing links between transactions had been employed before. For instance, electronic payment systems had frequently been designed such that payments (in different transactions) of the same customer could not be linked by a vendor. These works, however, focused on a single phase, payment, and the other phases of a transaction had not been considered, though they have the same potential for providing links as the payment phase, and consequently, may allow the building of profiles, too. Another difference between the concept of many payment systems and our works is that we are faced with two-party protocols, involving only a customer and a vendor, while many payment systems add additional online players, including but not limited to trusted third parties, in order to establish a privacy-protecting context. In this work, we aim at the larger context, i.e., we also keep an eye on the whole transaction when we introduce privacy-enhancing solutions for a single phase. By considering the whole transaction within our model, we acknowledge that even partial solutions, i.e., solutions for a single phase, can help to improve overall privacy, e.g., by providing *less* personal information to the vendor, though the vendor may still receive *some* personal information. The latter aspect is sometimes called "collection limitation" and is a principle that can be found in many privacy laws.

Collection limitation is also at the heart of the architecture presented in Chapter 4. It basically allows a customer to shop an Internet vendor's site free from the vendor's watch, as the vendor is taken the ability to identify a specific customer's shopping tour, i.e., her search phase. This is realised by preventing links from the customer's search phase to subsequent phases, which prevents the vendor from learning what things the customer is interested in, other than the ones the customer chose to buy.

Preventing the identification of a specific customer's interaction with a vendor is also the theme of Chapter 5 where we are concerned with loyalty systems. Although loyalty systems have traditionally been used to retain and reward loyal customers, nowadays they are primarily used to collect data for the creation of customer profiles. Normally, loyalty systems create links from one purchase, i.e., transaction, to another one by adding an additional issue phase for some sort of collectible, e.g., loyalty points. Clearly, the transactions linked in this way can be used to create a profile of the customer. We developed a loyalty system for the Internet that in a way goes back to the roots, meaning that it acts as a rewarding/retention mechanism but does not permit the creation of profiles. The latter is realised through a system that does not employ any identifying information to keep track of the collectibles gathered by the customer. Hence, the additional phase, introduced by the system to issue the collectibles, does not negatively effect the privacy of customers because

it does not produce any additional information that can be used as a link to other phases of the same transaction or to any phase of other transactions, including these transactions' issue phases for collectibles. The system is also the first of its kind, since —to the author's knowledge— no technical proposal for privacy-protecting loyalty systems had been made before. The system also does not sacrifice security for privacy, or vice versa, but realises both goals simultaneously. We provide two variants of the loyalty system, each based on different cryptographic assumptions giving rise to different properties. For the construction of both systems, we introduce *anonymous counters* that keep track of the number of collectibles gathered, however, cannot be manipulated by unauthorised parties.

Another type of customer retention mechanism encountered in the real world are multiple-use admission tickets, which we are concerned with in Chapter 6. This kind of tickets can be used for a preset number of times, such as an admission ticket for a cinema that allows its owner to see 10 movies, say, for the price of 9. Having similar tickets available for Internet offerings may be advantageous. Since the proliferation of broadband Internet access grows, it might become possible, say, for brick-and-mortar cinemas to complement their range of products with Internet home cinemas that offer Internet users the same kind of incentive that is offered to their customers in the real world by issuing multiple-use admission tickets. Such a scenario might be an application for the multiple-use tickets presented in Chapter 6. Similar to the loyalty system, these tickets had been designed with security and privacy in mind, i.e., they cannot be easily forged and allow customers to stay anonymous, just as with a real-world multiple-use admission ticket. More generally, these tickets can be seen as certificates that provide a means to anonymously and gradually disclose any kind of information encoded in the certificate.

Apart from their use in the application scenarios mentioned above, the cryptographic schemes developed for the loyalty system and the multiple-use admission tickets from Chapters 5 and 6, respectively, may be of independent interest, as they can be used in a wider range of applications.

Chapter 2

On Privacy

Privacy is an extremely complex matter as it touches many areas of our daily life — even though we are not always aware of that. For instance, when someone uses a cell phone in a crowded street, the person gives up privacy to some degree, as every casual bystander can overhear the phone call. But usually people do not care to listen or try not do so because, by social norms, eavesdropping is considered impolite. So even if there is potential for a privacy invasion in this situation, it is rarely realised. However, this may not be true for people living in non-democratic countries where a bystander might be a government agent taking notes or recording the call. But even people living in a free society should be wary as their privacy may be challenged too, e.g., in the name of fighting terrorism. For instance, as of this writing, it is but a year that the European Parliament and The Council introduced a directive that demands the retention of data from electronic communication services, such as phone calls, or data from communication networks, such as the Internet [EU06]. This action is merely one example where legal, technical, and also ethical issues pose a challenge to the privacy of individuals. Although all three issues would demand proper treatment, it is beyond the scope of this work to even come close to this, due to each topic's inherent complexity. Instead, we provide a crash course on privacy issues where we highlight some ethical and technical aspects relevant for this work and introduce terminology used in subsequent chapters. We completely leave out legal issues, as national privacy laws differ from country to country, and the goal of this work is to provide privacy for Internet users on a technical level, irrespective of any privacy law in place.

2.1 Demand for Privacy

Often it is assumed —rather than argued— that Internet users all over the world have similar concerns for privacy when they go online. In the following, we will

backup this presumption by presenting the results of surveys, conducted in the last couple of years in several (western) countries, which show that users have had concerns about their privacy since the beginning of the commercialisation of the World Wide Web in the late 90's. Although initial concerns may have been due to the new media "Internet" and the uncertainty that came along with it, the surveys show that these concerns persist and are even felt stronger by experienced Internet users.

The first survey we present was conducted in 1999 by Harris Interactive in the U.S.A., the U.K., and in Germany. In this survey, only 21% (US)/ 13% (UK)/ 10% (DE) of the respondents said that they were confident or very confident when asked about their confidence in the proper handling of their personal data by Internet vendors [Int99]. Consequently, 58%/45%/61% of the respondents said that it is very important "that those [vendors] adopt and follow strong privacy protection policies". A total of 59%/48%/69% among the Internet users said that they at least once refused to provide information that they thought was too personal or unnecessary and 57%/41%/56% said that they at least once decided not to purchase from a retail site because they were not sure about the handling of their personal information.

In a 2001 survey conducted in the U.S. [CM01], sensitivity for privacy was even higher than in the 1999 survey by Harris Interactive, as 82% of the respondents said that they had at least once refused to provide personal information and 64% responded that at least once they chose not to purchase because of privacy concerns. In the same survey, 70% of the respondents agreed or strongly agreed that they are usually bothered "when companies ask [...] for personal information".

A more recent survey (2005) conducted in Canada, Germany, the U.S., and the U.K. [Res05] shows that concerns for privacy have not dropped over the years. In this survey, 72% of all respondents said that they are somewhat, very, or extremely concerned about their Internet security, where the highest concerns in this regard were selling of personal information to third parties and identity theft.

The series of surveys conducted by the German television networks ARD and ZDF among Internet users in Germany, also supports previously mentioned findings. In the 2004 survey [vGF04], 86% of the respondents said that they are concerned about the transfer of personal information to third parties which, compared to the 2003 study, was another 4% higher (cited in [vGF04]). These numbers are relatively stable, as the 2006 survey [FG06] showed that 85% of the respondents still worry about a transfer of personal information.

Although almost all surveys confirm that customers feel that their privacy concerns are not properly addressed, the online business is flourishing. Thus, there seems to be a dichotomy or ambiguity between reported attitude and actual behaviour. An explanation for this might be that privacy itself is an ambiguous con-

cept, i.e., views on privacy widely differ. Acquisti and Grossklags [AG04, AG05a] argue that the behaviour of customers is in part due to their incomplete information about a vendor's intentions and due to their "bounded rationality", i.e., their inability to properly process all information and reach a rational decision.

Conversely, Syverson [Syv03] claims that choosing immediate gratification in exchange for personal data may well be the result of a rational decision making. He conjectures that customers are perfectly rational when they disclose certain information in exchange for some benefit, if they have a low expectation that something bad will happen from the disclosure. However, Acquisti [Acq02] found that this expectation, or rather behaviour, is myopic, as customers are usually overestimating short-term benefits while underestimating negative consequences in the future, and in addition often have an "optimism bias".

In the ARD/ZDF survey [vGF04], another explanation is offered for online users' seemingly ambiguous behaviour, namely that customers will usually not have a choice but to provide personal information, if they want to use a certain service.

Irrespective of why customers, despite uttered privacy concerns, eventually decide to buy online, the presented results from the surveys show that privacy seems to be on the customers' mind. Therefore, also from a marketing perspective, it might be beneficial for a vendor to invest in privacy safeguards and use the privacy argument to gain a competitive advantage over other vendors.

2.2 Privacy in General

The previous section suggests that some kind of common understanding of the term "privacy" exists between Internet users, though the details are still fuzzy and seemingly ambiguous. The latter might be one reason why privacy theorists have been working for decades on answering the question what privacy really means and how it can be defined in terms of a theory which should —ideally— cover all aspects of privacy in any given situation. In the following, we give a brief overview of some widely respected theories in this field in order to illustrate the diverse possibilities for the meaning of privacy.

Given these different meanings, it comes as no surprise that the notion of privacy is perceived and interpreted differently by different people in the world [Tav00] — and, to make matters worse, the situational context also influences the perception of privacy [Gav95, Nis04]. The reasons for these diverse views on privacy are manifold, ranging from interests of certain parties, such as privacy advocates and marketeers, to cultural issues, e.g., freedom of speech vs. personal rights. Next, we will briefly review some privacy theories and also mention some of their shortcomings.

Non-intrusion. In the seminal work of Warren and Brandeis [WB90], privacy is defined as "the right to be let alone". This is often interpreted as being free from intrusion [Tav99b], however not free from surveillance. In that sense, an Internet café owner or an ISP spying on its customers would *not* violate their privacy. □

Seclusion. Another theory, put forth by Westin [Wes67], describes privacy as "the voluntary and temporary withdrawal of a person from the general society through physical [means] in a state of solitude". In other words, Westin equates privacy with solitude. This means that to enjoy a state of privacy, one has to be alone. Critics of this theory further point out that it is possible to have 'privacy' (in a non-Westin sense) but no solitude and to have no 'privacy' while being alone [Tav99b]. Indeed, the latter situation arises if one accesses the Internet from one's home computer —while being alone— and yet, one's complete communication might be observed by third parties. □

Control. The control theory, stated, e.g., in [Fri90], argues that one can have privacy if and only if one has control over information about oneself. This implies that one has the freedom or, in other words, *choice to deny or grant* access to information about oneself. In practice, however, one will never have complete control over one's information as soon as it is released. And in addition, this kind of control implies that one is *aware* of the fact that personal information is disclosed. Clearly, the latter cannot be assumed in general, let alone for Internet applications. □

Limitation. The ability to limit or restrict access to information about oneself *in certain contexts* is at the heart of the limitation theory, c.f. [Gav95]. This suggests that the more restricted the access to one's information is, the more privacy one enjoys [Tav00]. However, it is often not only the number of persons having access to some information that is important but also the context in which this information is released. For instance, telling a doctor in a private room about an illness and later on telling another doctor in a hospital the same will not necessarily lead to less privacy despite the fact that another person learned about the illness, i.e., gained access to the private information. And also, voluntarily telling a group of close friends personal things is usually not perceived as giving up privacy, though clearly a number of persons learned about these matters. Nissenbaum [Nis04] makes a similar point in saying that "personal information revealed in a particular context is always tagged with that context". In that sense, she also speaks of *contextual integrity* meaning that privacy is maintained, if some information that was disclosed in an appropriate context is not being distributed or used outside this context. □

Control/restricted access. The theory of control/restricted access developed by Moor [Moo97] combines the ideas of the limitation and control theory. In Moor's theory, the notion of privacy is always attached to a context and not to the information itself, i.e., some information is not private *per se* but may be considered private in certain contexts. Unlike the limitation theory, one may choose to disclose some information in a certain context, i.e., one grants someone access to some information, and still maintains the *same level* of privacy. This means that, say, a vendor may rightfully learn and process information obtained from some customer at some specific point in time, e.g., during the customer's purchase, but may violate the customer's privacy if he would use it for a different purpose or even at another time, i.e., in a different context. Note that this is also consistent with Nissenbaum's definition of a *violation of contextual integrity* [Nis98, Nis04]. □

We do not extend this discussion any further, as it seems clear that the notion of privacy in general is not easily defined in any 'formal sense'. Still, the theory of control/restricted access comes close to the kind of control over information that we have in mind for this work, as it recognises that —to maintain privacy— information from one context, e.g., a purchase at an online shop, must not be combined with information obtained in another context, such as downloading personal information from community portals, e.g., Facebook[1], StudiVZ[2], Xing[3], or from a private homepage. The latter is particularly interesting, as such sites are publicly accessible, which leads to the problem of privacy in public [Nis98] and to the question whether there can be a reasonable expectation of privacy in public areas. If such an expectation is unreasonable then, perhaps, Internet users should not expect any privacy while being online.

All of the theories above seem to implicitly assume that disclosed information can always be related to a specific person. However, as we will see in the next section, there is a whole range of possibilities between the ongoing identification of an individual and anonymity, i.e., the inability to even recognise a specific person.

2.3 Shades of Identities

Identities are usually associated with *natural persons*, and we refer to the identity of a person when we speak of combined information that relates to her, like her name, address, date of birth, etc. Often, we also relate a person's identity to a device's identification, e.g., when we refer to the number of her cell phone or to the

[1] `http://www.facebook.com`
[2] `http://www.studivz.net`
[3] `http://www.xing.com`

IP address of her computer. In such cases, we make the presumption that only a specific person has access to the device, i.e., takes/makes the phone call or uses the computer. In the Internet, *automated services* like Web-servers can also be thought of as having an identity, as they can be addressed, e.g., by their domain name. However, in this work, we will only consider identities which are related to natural persons.

An identity need not necessarily be determined by an identification (ID) such as a name or a device's ID. For instance, we often refer to people whose names we do not know, e.g., "the newsdealer around the corner", and yet, the identity of this person is determined — at least for a certain audience. If it is unclear where "around the corner" is or if there are several newsdealers "around the corner" then the identity might be undetermined.

A person might be known to others by different *traits* [Wal99], e.g., the newsdealer's friends will know him by his full name and not just as "the newsdealer around the corner". This example also shows that for any number of a person's traits, there might be a group which is able to relate them and others who are not, e.g., the newsdealer's friends and the customers who do not know his name, respectively. Note that the newsdealer's friends and his customers also make up two different *contexts*. Thus, one could say that an identity is the sum of the traits known about some person in a given context.

Traits can also be *temporary*, such as, "the man with the red cap who crossed the Broadway". Without wearing the red cap, the same man will possibly not be recognised when crossing the Broadway two days later, as opposed to the newsdealer who always sells newspapers at the same corner.

In the following, we will introduce the terminology used throughout this work to distinguish various shades of identities. As this work is settled in the field of electronic commerce, we usually have Internet users in the back of our minds, when we speak of "some person".

Personally Identifiable. If the real-world identity of an Internet user, e.g., her name and address, is known by her communication partner, or can be inferred or learned from it through other information, e.g., her credit card number, we say that the user is *personally identified*. The information which contributed to the identification of the user we call *personally identifiable information* (PII).

A user might have several, context-dependent identities. However, if they contain PII, they can be related and traced back to the same natural person and thus, they can be regarded as a single identity. If an identity is comprised of or contains PII, hence allows matching with a natural person, we call such an identity a *person identity*. □

Pseudonymity. If a user is not personally identifiable, she may still be *recognisable* by some distinguishing mark, though not necessarily something that would intrinsically reveal her real-world identity. We will therefore call a person *identifiable* (without qualification), if she can be distinguished from others through some mark, while her real-world identity remains unknown and cannot be inferred from this mark. A mark can be virtually anything: a (fictitious) name, a unique number, a cryptic string, etc. We call such a distinguishing information a *pseudonym*. As such, a pseudonym can be regarded as an identity.

An important difference between a person identity and a pseudonym is that a pseudonym can be arbitrarily selected. In our context, this is usually done by the vendor's shopping software, e.g., by setting a session cookie [KM97]. In other applications or contexts, users may choose pseudonyms by themselves, e.g., in the IRC nicknames are chosen by users. The person using a certain pseudonym, we will call the pseudonym's *holder*. By using a pseudonym more than once, its holder allows others to recognise her/him and, possibly, also allows them to relate other information to the pseudonym. Another difference to a person identity is that a pseudonym, in general, can be dropped without effort, e.g., by stopping its use.

If one is using a pseudonym over a considerable period of time, a communication partner might be able to piece together enough information from their common communication to finally allow the personal identification of the pseudonym's holder. So even if, at first, no PII is conveyed to a communication partner, care must be taken regarding the disclosure of new information and the pseudonym's duration of use.

Pfitzmann and Köhntopp [PK01] distinguish pseudonyms regarding their duration and their context of use. For instance, a *person pseudonym* is used as a substitute for a user's real-world identity and may be used over a long, possibly lifelong, period in *many* contexts. Conversely, a *relationship pseudonym* is used only in a *single* context, possibly also for some time. Hence, a relationship pseudonym differs from context to context. In this work, however, we are primarily interested in a third type of pseudonym, a so called *transaction pseudonym*. Such a pseudonym is used only once in any single context/transaction. Thus, even if the user enters the same context again, her pseudonym will be different from previous ones. In summary, using either type of pseudonym mentioned before, the holder's communication partner may recognise her in a given (temporal) context. □

Anonymity. The traditional understanding of anonymity, i.e., concealing or withholding one's name, is no longer adequate (for the Internet), as other information exist, e.g., social security numbers (SSNs), credit card and passport numbers, IP and email addresses, etc., which allow identification of a natural person [Wal99]. Even more, these traits allow the creation of dossiers by linking, matching, and

mining data that was once disclosed about some identity [Nis99]. Such a dossier, containing not only facts but also inferred data, is commonly referred to as a *profile*.

Information that feeds a profile is often created by the user's own interactions, e.g., by sending and receiving information over open networks, like the Internet. In order to link a message to a profile, the message must contain some identifying or recognisable information that can be associated with the profile. Conversely, *anonymity* refers to the absence of such identifying or recognisable information.

In the (technical) privacy literature, it is understood that anonymity still holds, even if a message's sender, receiver, or both have a transaction pseudonym. On this account, we can have anonymity with respect to the sender and/or with respect to the receiver of a message [PW86]. By anonymity we mean that a user, i.e., a sender or a receiver, is non-identifiable, much like in the definition of pseudonymity, and *in addition*, he cannot even be recognised by his communication partner once the communication has ended. Note that when we speak of anonymity, we normally focus on a particular layer, e.g., the application layer. However, it is clear that true anonymity is only possible, if all layers, possibly including low-level communication layers, are anonymised — at least to the point where an identity cannot be determined or resolved by the communication partner.

In this work, we are mostly concerned with sender anonymity, since in electronic commerce scenarios the receiver of a message is usually some vendor which is known to the sender/customer. Sender anonymity implies that an anonymous user, returning to some site, cannot be distinguished from a first-time visitor by the site's operator. Still, the vendor is free to make a guess whether the user under watch is a returning or first-time user. If the vendor bets on the former then the user's anonymity depends in part on the number of former visitors of the vendor's site and so depends the vendor's chance for making a correct guess with respect to the user's identity. The group of former visitors is called the *anonymity set* or *anonymity group*, as the size of this group gives a lower bound on the vendor's chance of correctly guessing the user's identity (person identity or pseudonym).

The anonymity set for a message is defined as the group of other network participants who *could have* sent the message. Intuitively, anonymity is the stronger the larger the anonymity set. This intuition, however, is not always correct. Assume we got an anonymity set consisting of n users. Now, if for a given message m, a group of l members from the set can be ruled out as potential senders with a high probability, the size of the whole anonymity set, n, would not be the decisive factor. Because in this case, the anonymity of a particular sender does not depend on the size n but on l (and its probability), which determines the *effective size* of the anonymity set (for message m), i.e., $n - l$. Therefore, besides the absolute size of the anonymity set, other factors such as side information conveyed by messages of a specific group

of participants may influence overall anonymity (see [SD02, Ser04, DSCP02] for a discussion). □

Unlinkability. At first sight, unlinkability seems to be somewhat misplaced here, as it has nothing to do with identity classes, such as the ones introduced above. It mainly has to do with profiling. However, as a profile leads to an identity and every identity belongs to one of the classes above, introducing the term "unlinkability" in this section seems to be appropriate.

The notion of *unlinkability of sender and receiver* first appeared in [PW86] and was described as a measure that "hides the relation between sender and recipient of a message from everybody but [...] the sender of the message". Later, Reiter and Rubin [RR98] described it as a state where a sender and a receiver can be recognised as participating in some communication while they cannot be identified by an attacker as communicating with each other.

Although achieving this kind of unlinkability is desirable, we do not aim that high. In a typical e-commerce scenario, the sender of a message will usually be some customer and the receiver will be a vendor who, with respect to privacy, plays the role of an attacker. Therefore, we are particularly interested in preventing that actions of the same customer can be related, i.e., *linked* by a vendor. Thus, what we look at in this work is *message unlinkability*, meaning that two or more pieces of information, e.g., messages sent by the same user, cannot be related by the recipient of the information. In other words, message unlinkability ensures that the recipient's *a priori* probability for relating two or more messages of the same user does not increase, if he in fact (unwittingly) receives another message of this user. That is, the *a priori* probability for linking two messages of the same user is the same as the *a posteriori* probability.

2.4 Privacy Concerns

The widely known "know your customer" mantra of marketeers can be taken to new heights given the data made readily available by customers in electronic commerce processes. Online customers provide online vendors with all kinds of data that marketeers could only have dreamt of before the Internet era. In a typical web shop, customers provide their names, addresses, payment information —such as credit card numbers—, and purchased goods. This is not a new threat to the privacy of individuals as this had been possible before the Internet era of shopping. However, in the "offline world", e.g., in brick-and-mortar shops, one has the *choice* to stay anonymous and still get the desired goods without having to accept a (financial) penalty. In the Internet, choosing to stay anonymous usually means choosing not to buy at all. In a 2004 survey [vGF04], this kind of 'take it or leave it' choice is also

offered as an explanation for the phenomenon of users reporting concerns about the treatment of personal data provided to online vendors and the users' actual behaviour to provide data nonetheless. However, the survey also shows that 86% of the respondents are concerned about their privacy and that a total of 58% try to do something about it, e.g., by deleting cookies at least once a month. As the primary means of cookies is to store data about a customer and also to track the customer's behaviour, one suggestion of this finding is that users are not too fond of being re-identified and/or tracked, despite the alleged advantages of being more quickly informed about new potential items of interest.

Before we outline specific concerns / threats, we should perhaps argue why collecting information from individuals may be seen as a privacy infringement. In the privacy literature, autonomy, liberty, democracy, etc. have been identified as being core values [Joh85, Moo97, Nis98, Tav99b], i.e., values which are good in themselves. And many agree that privacy has instrumental value, i.e., leads to something good, and is —at least— the "expression of a core value" [Moo97] or even an "essential aspect of [the core value] autonomy" [Joh85]. Privacy leads to something good as it can protect us from harm, e.g., discrimination, harassment, stigmatisation, and so forth [Ved99]. It also supports autonomy, liberty, and eventually democracy, as fear from being watched all the time will influence where we go, how we interact with others, and how we act in the public, in general.

In some countries, the former notion, though seemingly a philosophical one, also led to the adoption of concrete privacy laws. For instance, in 1983 the German Federal Constitutional Court ruled on similar grounds that people have the right of "informational self-determination" which it saw as an expression of personal freedom [BVe83]. The right of informational self-determination is often paraphrased as "having the right to know who learns which information on which occasion about oneself". The Court's decision eventually led to Germany's first Federal Data Protection Act, first enacted 1990, which in later years had been amended several times [BDS03]. Such a right of self-determination, however, will be of little use, if individuals cannot keep track —or do not know— of all the database systems where some piece of information about them is stored, as they will be unable to seek access to these systems in order to review, correct, or even demand deletion of their data [Lau96].

As mentioned before, many threats to privacy are not new or genuine threats of Internet practice. However, in the real world, we would normally object to certain practices whereas in the online world we take them for granted. In the following, we will present some scenarios which make up privacy issues in the real world but can also be encountered in the Internet in a similar form and are even exacerbated by the speed and scale of modern information technology.

Data Links. Often, it is not the data by itself which is private but the *linkage* of that data to a specific person — the data might just be publicly available. To illustrate such a case, Fulda gives the following example [Ful99]. Suppose Alice visits her friend Bob. While waiting for him in his living room, she sees a stack of magazines. Obviously every magazine from the stack had been published and hence, can be considered public information. Also, the fact that Bob possess a stack of magazines is publicly available on sight. Now, suppose that in the stack a 'men's magazine' is buried. Of course, the magazine by itself is also some public information — anyone can buy one at a newsstand. Now the question Fulda asks is, if Alice would go through the stack, noticing this magazine, wouldn't she violate Bob's privacy? Probably yes, even though all information by itself is publicly available. It is, however, the linkage of the magazine to Bob as a reader which was private.

Another example in this regard can be found in [Ful98] where the case of a pregnant woman is made who uses an ATM and is watched by a camera. Again, the author asks: Is the woman's privacy infringed, if being caught on camera triggers targeted advertisements, spam-mails, and the likes promoting products for newborns?

In the Internet, it is even easier to match and combine observed public information from different sources, once the linkage to a specific person is known. For instance, by personalising purchases, online vendors *a posteriori* create a link to the customer's search phase which enables them to see what the customer is interested in, similar to Alice who *a priori* knew the link from Bob to the stack of magazines. Essentially, online vendors create a link from a customer's digital/online identity to her real-world/offline identity [Acq02]. This enables vendors to tap information sources, online and offline, that link to the same offline identity, i.e., the same natural person, and therefore can be used to further expand the vendor's knowledge on a particular customer. Nissenbaum [Nis98] adds that the sum of this information can be further aggregated, e.g., to find similarities between certain groups of people, and this in turn may allow to "incorporate a richer portrait of the individual" into profiles. Consequently, an information asymmetry will arise that clearly favours vendors, since the vendor will know much more about the customer than vice versa, and the customer will not even know about it. The problem with this is that customers who are not aware of what others know about them may be easier targeted, discriminated, and manipulated than others [Nis98]. □

Data Mining. Data mining is basically about finding useful patterns and relationships in data [FU96, Tav99b]. "Useful" means that additional, new knowledge can be inferred from data and subsequently put to use, e.g., for targeted marketing campaigns [FPSS96]. The whole process for extracting non-obvious, previously unknown, and potentially useful information from data comprises several steps,

including data preparation, data selection, incorporation of prior knowledge, etc. This process is usually called knowledge discovery in databases (KDD) [FPSS96], however, we will use it interchangeably with the more familiar term *data mining*.

Although, data mining can be applied to all kinds of data, we are especially interested in cases where new data is extracted, or rather inferred, by aggregating and generalising data collected from a group of individuals. Note that there is a big difference between fact data, i.e., things that actually happened, and inferences, e.g., predictions, as the latter are afflicted with uncertainty. Therefore, Gotterbarn uses the term *virtual information* to distinguish inferences from the fact data used to produce them [Got99]. Virtual information may consist of inferences about a person's future behaviour, personality traits, and so forth. Gotterbarn also remarks that virtual information is often afforded the same veracity as actual fact data, as soon as it becomes an electronic fact, i.e., is recorded in a database.

An illustration of virtual information is given in [Tav99a], where Tavani makes the following hypothetical case. Suppose Bob, an executive working for a marketing firm, applies for an automobile loan at his bank. To this end, he fills out the usual forms required by the bank. In these forms, Bob indicates that he annually earns $90,000 and currently repays a $15,000 loan for a family vacation to Europe taken during the previous year. He also consents to internal processing of this data by the bank, i.e., his data is not given to third parties. Now, assume that the bank uses KDD to mine data from its databases and the KDD process comes up with the following pattern: "Executives earning more than $70,000 but less than $120,000 annually, and who purchase luxury cars (such as BMWs), and who take expensive vacations, often go into business for themselves within five years of employment." Further assume that a separate data mining process finds that the "majority of marketing entrepreneurs who go into business for themselves declare bankruptcy within one year of starting their own businesses". Now this virtual information can be combined to a new virtual information, making Bob a member of "the group of marketeers likely to start a business and declare bankruptcy within a year". Consequently, Bob poses a long-term credit risk for the bank and should be denied his loan.

At first sight, the example does not seem to illustrate a privacy problem, as Bob gave his consent to the processing of his data. However, recall from the beginning of this section, that privacy had been identified as an expression of autonomy and, according to the control/restricted access theory (see Section 2.2), one should be able to limit the context or purpose for which the provided data will be used. Now, the problem is, if we cannot possibly know what *new data* will be unearthed by KDD processes, how are we supposed to limit access to data we have neither provided nor are ever made aware of? And, looking at the example again, this does not even take into account that one may never truly become a member of the group found

by a KDD process — Weichert called this a statistical prejudice [Wei06]. If we have to fear that harmless facts recorded in some database, like a trip to Europe, lead to unforeseeable consequences, like the denial of a loan, then this will surely have an impact on our autonomy.

Another problem in this regard is, especially for legislative approaches, that it is hard to give an 'informed consent' to the processing of one's data, if it is unclear which new data will be found by the KDD process and how it will be used [Tav99a]. In addition, there is also an ethical issue with generalisations produced in a data mining process, as generalisations give rise to *stigmatisation* and *discrimination*, e.g., due to a predisposition for a certain disease, a certain type of lifestyle, or a particular job [Ved99]. Thus, anonymity, in such a case, would also act as an equaliser [Joh97]. □

Profiles. A profile is usually the sum of information gathered from and about a particular person and may also contain information which had been derived from the available fact data, e.g., by data mining techniques. In certain business branches, data mining will be used to amend personal profiles, e.g., in order to reduce risks associated with certain groups found by the KDD process. In this case, the generalised data is effectively treated as if it were personal data [Ved99, Wei06], i.e., virtual information is treated as fact data even if it is by far less than 100% accurate.

Another problem is that a customer, who regularly purchases at some vendor's site, enables the vendor to constantly expand on her profile, allowing more accurate inferences on her household. In addition, hypotheses drawn from inferences can be easily tested and recorded by the vendor, the next time the customer visits the online shop. As an example for this, imagine a crafted list of items of interest which are presented to the customer. If the customer clicks a certain item from the list, she supports the hypothesis, if she ignores the list, she rejects the hypothesis. This way, the vendor may easily test the customer's likes and dislikes. Note that such lists may also come with special offers and thus, can be much more manipulative than ordinary advertisements which are directed at the masses. Also note that the negative event "item was not clicked" can be recorded just as easily as its positive counterpart. Therefore, it is not only possible to observe what a customer did but also what she did not. This, together with the fact that information is already collected in a digital format, makes it much easier to process, verify, and enrich electronic customer profiles than, say, using surveillance cameras or instructing a store detective to create dossiers of only a small group of customers which can be observed this way. Therefore, the potential for privacy invasion in the online world is much greater than in the offline world. Still, we may not even have seen the full potential of recorded Internet traffic yet, as, according to [SDP06], today's vendors

often underutilise their available clickstream data, i.e., they do not fully exploit the potential of Web analytic tools yet.

In addition, the collection of the data at a central point, the ease of access to the information, including sophisticated searches and queries, the speed of sorting, updating, and sifting through the data, as well as the practically endless amount of memory to store all the data makes up the privacy issue [Joh85]. □

Surveillance. Surveillance is one of the building blocks for the creation of profiles. Internet technology allows to put every single user under surveillance in ways that had not been possible before. To distinguish surveillance aided by information technology from classic surveillance technology, e.g., cameras and microphones, Clarke coined the term *dataveillance*, meaning "the systematic use of personal data systems in the investigation or monitoring of the actions or communications of one or more persons" [Cla88]. However, with concepts like "ubiquitous/ pervasive computing" and "ambient intelligence" the line between classical and IT aided surveillance diminishes more and more — see [DTG06] for a discussion on some more privacy aspects in pervasive computing environments. Indeed, such converging sources of information may become one of the major future problems with respect to privacy, as most likely it will not be possible to draw a line between online and offline data pools, and everything observed in the real world immediately becomes an electronic fact available in the online world. □

Merging of Information. Users often cannot foresee the consequences resulting from the disclosure of certain information, since they only have limited knowledge of actions taken by the receiver of the information [AG05b], such as passing on the data to other parties. For instance, a user may authorise the collection of different information relating to her on various occasions, e.g., purchase information at *different stores*. However, this in no way means that for the union of this information —such as a profile comprised of the purchase information from *all stores*— a similar authorisation can be assumed, since the union of this information may collectively release more information than was originally desired [Got99, Sch03]. Still, it can be assumed that the predominant way of dealing with information obtained from various sources —online or offline— is to simply merge them. This consequence is often overlooked by users when they give an 'informed' consent permitting the transfer of some of their data to third parties.

In [Got99], Gotterbarn cites a real case where two information sources had been merged, without the persons' knowledge or consent, which had serious consequences for the persons involved. In the particular case, a banker in the U.S., who was on his state's health commission, pulled a list of cancer patients and matched them with his bank's records of outstanding loans. Then he called in the loans of persons

on the list in expectation of their premature demise. This case also illustrates a violation of contextual integrity [Nis04] because the health and financial information were extracted from their initial contexts and combined in a new context. □

Trading of Personal Data. It is clear that profiling will benefit some groups while others will be discriminated. People belonging to the advantageous group can be expected to willingly release certain information in order to receive some sort of benefit [AG05b]. In principle, there is nothing wrong with this, provided that people gave their consent for the processing of their data and accept the potential consequences of the disclosure, i.e., that they forfeit any control over the secondary use of their data (c.f. the paragraph "Data Mining" in this section). As already outlined in an earlier paragraph, disclosing information may have unforeseeable consequences for the future and thus, even the advantageous group may realise that information provided in the past can turn against them, e.g., if the information provided, later associates them with some risk group. So the dilemma is, if one trades in personal data to achieve some benefit today, one may be denied benefits in the future because of information provided in the past. However, withholding information may also not be ideal, e.g., if one will be heavily penalised for not providing certain data, e.g., paying twice as much interest for a loan. The latter is also an ethical issue as, in such a case, voluntariness cannot be assumed anymore and then, profiling will give rise to an increased erosion of privacy.

It is, however, not only the trade between business and consumer but also the trade from business to business which raises concerns. Indeed, the market for consumer information is driven by companies that wish to offer personalised services and to make use of dynamic pricing in order to maximise their profits [AV05, Tay02]. Since a market exists, online vendors do not have to solely rely on their own profiles but may also buy (sell) additional information from (to) professional data brokers, such as DoubleClick[4], I-Behavior[5], or LocatePlus[6]. The said companies are particularly active in the U.S. market, however, this does not necessarily mean that, say, the privacy of Europeans would be better protected from information brokers because of European privacy regulations. As many U.S. based companies, such as Amazon or Google, process their collected data outside the European Union, the EU's privacy regulation is hardly effective.

Another concern —easily overlooked by customers— arises, if a company is taken over by another company and all of the collected customer information is transferred to the new owner. The new owner may or may not have made a privacy statement about the treatment of customer information and may consequently have a different

[4] `http://www.doubleclick.com`
[5] `http://www.i-behavior.com`
[6] `http://www.locateplus.com`

perspective on the processing of personal information than the company which had originally collected the data. This could have happened in the past to customers of the now bankrupt online retailer ToySmart.com, which offered its customer database for sale when it went into bankruptcy [EAASS06, TM01]. This, however, was prevented by an action of the U.S. Federal Trade Commission (FTC), but only because the company's privacy policy stated that personal information would *never* be shared with a third party and the FTC's position was that ToySmart.com had to abide to this promise, even in case of the company's failure [EAASS06, Fed00]. Meanwhile, companies have learned from this 'mistake', and nowadays even large companies, such as Amazon.com, have made it explicitly clear in their privacy policy that customer information is an asset that will also be transferred, should the company be acquired by another firm (see also [MD03]). □

Price discrimination. Although not strictly a privacy problem, price discrimination is presumably one of the consequences of profiling that will be immediately felt by customers. Price discrimination means that a vendor charges customers different prices for the same good. The posted price seen by a customer will usually be based on some extra information, such as a profile. In certain business branches in the real world, price discrimination is very common, e.g., at car dealers or at flea markets. However, there are at least two differences to online price discrimination. In the real world, it is done more openly, i.e., one usually knows that prices are subject to haggling and that the prices are a starting point for further negotiation. In the online world, posted prices are fixed and customers will assume that the prices are the same for all customers. If not, customers feel treated unfair and are outraged, as experienced by the online bookstore Amazon.com when its experiment in price discrimination was revealed [Ram05, Ros00]. For this reason, Odlyzko [Odl03] expects that *open* price discrimination is not going to happen in the near future because customers will be able to circumvent this. For instance, customers who are charged with a low price for a certain good will re-sell the good to customers who are charged with a higher price. However, this does not necessarily suggest that price discrimination will not happen, it rather suggests that if it is done, it is likely to be done in secret and hence its manipulative character will not be obvious to customers.

Vendors may also be interested to learn an individual customer's valuation for certain goods in order to maximise their profits. They could do this, e.g., by periodically raising prices until the customer rejects the offer and then post a price the customer is likely to accept, according to her recorded behaviour [Ken01]. This can also be regarded as a variant of the aforementioned hypothesis testing capability (see the paragraph on Profiles).

However, Taylor's analysis [Tay02] suggests that, if customers anticipate a transfer of their personal data and act strategically, customers with a high valuation for a good will even reduce their demand, i.e., reject initial price offers today, thereby misrepresenting themselves as low-valuation customers, in order to be offered a cheaper price in the future, i.e., the next period. Thus, if most customers were forward-looking, buying consumer profiles would not be attractive for vendors because, from the recorded behaviour, vendors would not be able to distinguish between strategically acting customers, who would also buy at a high price, and low-valuation customers, who would only buy at a low price. Thus, in some sense, customers can be said to be 'anonymous' as they are indistinguishable —from the vendor's point of view— with respect to their valuation for some good. However, if customers are myopic or do not act strategically, they choose not to misrepresent themselves and vendors will want to employ price discrimination, as they are better off with this [AV05]. In the myopic case, customers discount the potential losses from losing control of their personal information with the uncertain probability that such an outcome will take place [Acq02]. In the latter case, customers may value some service which saves them time and effort, e.g., longer checkouts due to entering data anew or having to install an anonymising software. However, as the authors admit, these results should be regarded with caution, as they depend on the fact that only two selling periods had been considered and by adding more periods several additional factors would have to be taken into account. □

ID Theft. Identity theft means to assume someone else's name or some other trait, usually in order to run up bills or commit crimes in someone else's name [Com04]. ID theft is, on the one hand, a security problem, as it should not have been possible to steal personal information in the first place, but on the other hand, it is also a privacy problem, as personal information had been learned by unauthorised third parties — the thieves. By stealing profiles, criminals gain a dossier of a person which makes it all the more easier for them to literally become that person.

A story on MSNBC [Sul03] shows to what extremes such thefts can be actually stretched. The article reports about Malcolm Byrd, an innocent man, who had been arrested several times for crimes committed by others in his name. He had been filed a criminal record, had his driver's license suspended for unpaid traffic fines, was temporarily denied unemployment benefits because of 'his' criminal record, and even ended up in jail because someone else used his identity every time he got caught by the police. Byrd's case also illustrates that once an event becomes an electronic fact (c.f. the paragraph on Data Mining), e.g., Malcolm Byrd's criminal record is added to his profile, the data is believed to be true and automated decisions are made based on this grounds, such as the denial of unemployment benefits. If the thief of Byrd's identity had been more careful with using it, Malcolm Byrd may

have lived years without noticing that someone is damaging his reputation. Had Byrd applied for a loan, he might have been turned down because of 'his' criminal record and the bank may not even have told him.

ID theft is not a singular incident. For instance, the FTC reported 247,000 complaints in 2004, which is a rise in online ID theft by 15% compared to the previous year [Com04]. Additionally, ID theft may not only happen because someone broke into a secured database but may also happen because companies act sloppy and willingly publish information that can be related to individuals. As a witness to that, the ISP AOL unsuspectingly published on its research web site about 19 million search requests of more than half a million of its customers [Hei06]. These requests had not been anonymised and contained, among other information, the users' screen names, i.e., the online IDs of AOL customers. Apart from learning the AOL users' screen names, it was possible to learn some of their interests, names of friends, etc., which can be helpful not only in assuming the person's identity but also in identifying the person. And indeed, it took only a few days until the New York Times could report that the first person behind one of the screen names had been identified [Tim06].

So, one could also say make the argument that the more information is known about a particular person, the easier it becomes to assume this person's identity, and this in turn means lower security. □

We have made the point above that one of the privacy problems in the Internet is that everything said and done will likely be recorded in some file, e.g., profile, audit log, server log file, and so forth. Even though this is already a greater problem than in the real world, where the default is not to record everything, the real problem is that all this recorded information, and conclusions drawn from it, will be linked to a specific person, possibly without the person being aware of that, and potentially being stored for over a lifetime. This also raises concerns that even minor incidents in one's life or errors in records will follow one through life [Joh85] and may even be passed on to one's children, e.g., if some kind of gene defect had been diagnosed and stored in the profile of one of the child's parents. In addition, the collection of the data at large central / inter-linked databases, the scale, speed, and ease of access to the collected information, including sophisticated searches and queries, the speed of sorting, updating, and sifting through the data, as well as the practically endless amount of memory to store all this information raise major concerns for the privacy of individuals.

The problems outlined in this section mainly arise because the affected persons are personally identified. By using the same identity in all or many contexts, it is easy to link, extract, and process information from these contexts, compile them into a profile, and eventually learn information that the data subject believed to

be private. Without persistent identities, however, these problems would be non-existent, or at least greatly alleviated, as the information disclosed by a person in one context could not be related to information from other contexts, if the person did not intend this, e.g., by deliberately using the same pseudonym in more than one context. Thus, by acting anonymously or pseudonymously people will reduce the odds for becoming subject to privacy infringements. Therefore, the solutions developed in this work focus on anonymity as this protects online customers in the first place and allows them to avoid privacy problems such as the ones outlined before.

2.5 Privacy Enhancing Technologies

The term "privacy enhancing technology", or PET for short, is collectively used to refer to schemes which aim to give users more control over the information, especially personally identifiable information (PII), that is disclosed by interacting with others in the Internet. Some of this information cannot be completely withheld, as it is inherent to the protocols used to communicate over the Internet, such as a computer's IP address. In the following, we provide a quick overview of PETs aimed at anonymising network traffic or content received by individual users and thus, can be said to be universally applicable. These PETs are complementary to the ideas developed in this work and will sometimes need to be employed in order to provide the level of privacy protection we aim for. The selection of PETs is not meant to be comprehensive and just includes those PETs which the author deemed relevant for this work and universally applicable, and which allow a kind of technical enforcement by the data subjects, e.g., the users. A more comprehensive list of PETs, including some of their histories, can be found in [GWB97, Gol02].

Proxies. The Anonymizer[7] is an HTTP proxy server that sits between a user's browser and every web server the browser connects to. This means that the user's complete HTTP traffic is routed through the Anonymizer and the addressed server will see the Anonymizer's IP address instead of the user's. Therefore, a site operator only learns that someone browses via the Anonymizer but not the user's IP address.

Another example of a proxy solution is the Lucent Personalized Web Assistant (LPWA) [KGG+98, GGK+99]. It is similar to the Anonymizer in the sense that it also acts as a proxy between the user's browser and a web server. However, it offers some additional functionality that allows users to establish a pseudonymous relationship with web sites. □

[7] `http://www.anonymizer.com`

Proxy approaches, however, suffer from at least two major problems. First, the proxy must be ultimately trusted by its users since it sees all of its users' traffic, which may include sensitive information. Second, it is a single point of failure/attack. That is, attackers do not have to target a large number of servers in order to learn private information, they can get all the information by monitoring or hijacking the proxy.

Anonymity Networks. The problem of single proxies can be alleviated by introducing a *network of independent proxies* where a user's message travels along a path that consists of a subset of these proxies. This way, if one proxy becomes corrupted, the user's privacy will not necessarily be compromised, as the corrupted proxy may not know who sent the information and who is to receive it. Two popular schemes exist for anonymity networks, Crowds [RR98] and *Mix-based* systems [Cha81], such as Onion Routing [GRS99] (and its successor Tor [DMS04]), Freedom [BSG00], and Web Mixes [BFK00].

Mix-based systems can be used to hide the communication relationship between any two parties. In a nutshell, a Mix-based anonymity network is comprised of servers —the Mixes— and clients, i.e., senders and receivers of messages. Each Mix collects encrypted messages from a number of sources —senders of messages or other Mixes— and forwards them in random order to other Mixes or to the messages' final receivers. Thereby, Mixes make use of a technique called *layered encryption*. As the name suggests, a message is encrypted in layers and each Mix *en route* peels off a layer of encryption and sends the resulting message either to the next Mix able do decrypt the next layer or to the final receiver, if the removed layer was the last one. An observer of the Mix network may only notice that some users send messages while others receive them but not who communicates with whom — this is sometimes referred to as *unobservability*. A variant of classic Mixes are Peer-to-Peer based Mix systems (P2P mixes) [RP02, RP04, FM02], where each peer participating in the Mix network is a Mix by itself. □

Content Filters. Filters can prevent Web sites from stealing information from the user, e.g., her browsing history, or conveying (identifying) information to third party sites. Filter mechanisms usually include, but are not limited to, the blocking of cookies in various forms (totally, per site, third party only), filtering of ads, blocking of certain images, and disabling the execution of code, e.g., JavaScript, Flash. In this respect, today's browsers offer a lot of filtering capabilities, either off the shelf or through extension mechanisms. Alternatively, centralised proxies can provide similar filtering capabilities for user groups. Filtering out content, however, may come at the price of "page quality" [KMW07], i.e., the page looks 'ugly' because some elements had been filtered out, resulting in the displacement

of other elements. Krishnamurthy *et al.* [KMW07] provide a survey on the impact of filtering mechanisms on page quality and the effectiveness of these mechanisms with respect to privacy.

Apart from the PETs above, other mechanisms, such as privacy policy languages, are sometimes mentioned in the context of PETs. These mechanisms, however, are just an alternative means to communicate a certain privacy policy to users. Thus, in general, they provide notice but no enforcement. However, even the aspect of notifying users might fail, as human-readable privacy policies in the Web tend to be incomplete [Pol07], e.g., are silent on issues like data sharing, third-party data collection, etc. Thus, there is little reason to believe that machine-readable policies will give users more complete information than their human-readable counterparts. However, if the machine-readable policy mandates answers to certain questions then honest vendors who thought that, say, *not* sharing data is not worth mentioning may indeed provide more information. Still, we do not consider policies or any of the following items as PETs or even adequate means to protect the privacy of users because their enforcement is completely up to the vendor and merely telling users what is being done with their data does not improve their privacy. Nevertheless, we provide a brief discussion in this section in order to point out some of their other shortcomings.

Privacy Seals. Privacy seals are not so much driven by technology but more by organisational means. Its providers, e.g., TRUSTe[8] [Ben99] or BBBonline[9], are paid by vendors to audit their sites with respect to privacy rules established by the seal's provider. If the site in question passes the audit, the vendor is granted the seal and may subsequently display it on his Web pages. This suggests to customers that the shop is privacy compliant, since it got a certification from the seal's provider.

However, seal providers usually perform a relatively superficial review of a web site, looking for, e.g., privacy notices, opt-out functions, procedures for complaint resolution, etc., but offer no privacy [FKH00] — insofar the term "privacy seal" is also misleading, as it usually does not attest any privacy safeguards in place but rather the mere existence of certain operational procedures. In addition, even if a vendor collects all data he can get from his customers, some providers may still grant a seal, provided that the vendor declares in his privacy policy all the data he collects. Friedman *et al.* once compared this to "a hotel garnering a five-star rating simply by promising not to guarantee its customers good service and then faithfully keeping its promise" [FKH00].

[8] `http://www.truste.com`
[9] `http://www.bbbonline.com`

In a study, Moores [Moo05] found that on the one hand users associate seals with privacy but on the other hand generally do not know what the seal guarantees or how even a genuine seal looks like. Consequently, about 14.7% of the respondents in this study also 'recognised' a privacy seal that was completely made up. In addition, even genuine seals can be easily copied and incorporated into Web sites that have not undergone an audit [MD03]. Such sites can, of course, be sued by the seal provider but this requires that the provider receives notice of such an act in the first place. Thus, sites that pop-up out of nowhere (and which may disappear just as quickly as they showed up) have a good chance of tricking customers by illegally displaying a privacy seal and getting away with it. □

Privacy Policy Languages. Privacy policy languages are a means to encode a human-readable privacy policy in a machine-readable format that is ready for processing, e.g., P3P [CLM+02], EPAL [AHG03], and CPExchange [BH00] — see [KCLC07] for more examples. Such policies usually include statements of data processors saying what they intend to do with the data obtained from some entity, e.g., a user or a company, or what actions the entity consented to.

However, even if data processors internally use some kind of enforcement system to adhere to some policy, e.g. the E-P3P system introduced in [KSW02] or the system developed by Casassa Mont *et al.* [MPT06], users have no way of knowing that the system is actually in place, works and is operated as promised. Furthermore, if users revoke their consents, they cannot verify that all processing of the data covered by the consent is stopped and that previously inferred data will not be used by the data processor henceforth.

2.6 Competitive Advantage

Privacy protection is not for free and there are costs associated with it. Switching to privacy enhancing technologies (PETs) will incur costs for adopting vendors and also to their customers. However, the customers' willingness to pay for privacy is not always consistent with the degree of their concern (c.f. the survey results of Section 2.1). There are many subjective factors that may lead customers to decline extra costs for PETs [AG04], including ideological beliefs, e.g., that privacy is a human right that nobody should pay for. But once adopted, the actual usage costs for PETs will be low [Acq02].

However, if, on the one hand, the initial switching costs for PETs are greater than the expected net present value then vendors are unlikely to invest in PETs [FFSS04]. On the other hand, if such investments are not made, the expected loss in online sales in the U.S. due to consumers' security and privacy concerns are expected to be $2.4 billions, according to a 2002 study by Jupiter Research [Lea02, Jup02]. And

this just takes into account lost opportunities, i.e., customers choosing to buy less or not at all online. The surveys [CM01, Int99] already discussed in Section 2.1 allow similar conclusions, as some of their results indicate that vendors are already losing money from customers choosing not to purchase online because of privacy concerns. Another point is that the numbers from Jupiter Research do not include direct costs associated with damages caused, e.g., by ID theft, which additionally may incur indirect costs, such as damage to a company's reputation due to the inability of the company to protect its customers' information. Therefore, Internet vendors may want to protect themselves from such damages by adopting PETs and, by doing so, may additionally gain a competitive advantage over competitors which do not offer a similar protection.

Privacy issues may also hinder competition. In the Culnan-Milne survey [CM01], most respondents said that they are bothered when vendors ask for personal data. This implies that they are, in general, reluctant to provide such data. However, if they once did, they may not want to have other vendors to have their data and consequently, may stick with the first vendor they got in touch with. Such a behaviour would effectively create a kind of privacy lock-in, that would be governed by mere coincidence rather than by an attractive selling proposition.

However, even if vendors start competing in terms of privacy, Feigenbaum *et al.* warn that if the "amount of privacy" advertised by vendors cannot be accurately measured (by customers), vendors may be tempted to get sloppy with the entrusted information [FFSS04]. Measuring may indeed prove to be difficult, if not impossible, if privacy is to be enforced by vendors alone. This is more or less the situation today where vendors 'promise' their customers a certain amount of privacy. No matter if vendors actually live up to their 'promises', customers will usually be unable to verify such claims and again have to trust that the claims are indeed true. This is where the results of this work come into play.

In the approaches put forth in this work, a considerable amount of personal trust in the vendor is shifted to 'impartial' privacy enhancing components instead. These software components are ran by both customers *and* vendors. This way, a customer's agent communicating with some vendor's agent may better control the amount of personal data that is released and may also *verify* privacy-related outcomes of transactions, i.e., vendors' privacy guarantees can actually be verified and thus, a competition in terms of privacy is conceivable. This approach is different than those of general PETs (c.f. Section 2.5), since it also requires to change/augment the server side, i.e., the vendor's software. From the point of enforcement, this is clearly a disadvantage, since we need the vendors' cooperation. On the other hand, probably all client-side-only PETs are not enough to protect customers' privacy in an e-commerce transaction. This is because they only protect the 'lower' communication layer whereas transactions are normally taking place at

a 'higher' application layer. Consequently, our approach can be more effective, as it takes into account information exchanged at the application layer. The approach is also attractive to privacy-committed vendors, as it allows them to back their privacy promises by actual proof. However, we may still fail in our attempt to improve customers' privacy, if identifying information is conveyed at the communication layer, e.g., if a customer's computer has a static IP address. Thus, our approach cannot, and was never meant, to replace general PETs under all circumstances. Instead, we pursue a complementary approach that allows to put in place *additional safeguards* in order to minimise privacy invasions, in case one of the employed schemes is being circumvented or even broken. Therefore, our results can be used as stand-alone solutions as well as in combination with one another, and also with other PETs, making the resulting framework even more resistant to privacy infringements.

2.7 Conclusion

In this chapter, we addressed several issues surrounding privacy. First, we approached privacy from an ethical side and pointed out that the meaning of privacy has many facets and that its meaning is neither clear *per se* nor universally agreed upon.

This work's goal of enhancing privacy in e-commerce is best described in terms of the control/restricted access theory, as its goals seem to come closest to the ones pursued in this work. That is to say, customers should be able to control if they want to disclose certain information —including their own identity— and if they do, they should know what information they disclose and made aware of potential consequences of their choice. However, refusing to provide certain information may as well result in denial of service, as vendors seem to have embraced the idea that security is increased, if privacy is decreased, which clearly is just a presumption [Syv03]. For customers, the situation becomes increasingly unintelligible, as the more information is disclosed to different parties, the more difficult it is to understand who knows —or thinks to know— what about oneself. This raises privacy concerns as more and more data is being collected and processed in large databases in order to create detailed profiles of persons, which in turn form the basis for automated decisions made about them.

Persons subject to such processing are often unaware of it and as the results of the employed KDD process are unpredictable, they cannot exercise any rights with respect to their personal data, i.e., they can neither object to the processing nor restrict the use of the discovered data for a specific purpose. But even if the latter would be possible, the question would still be how customers should be able to verify that personal data is only used in a way permitted by the customer. Today, this comes down to sole trust in a vendor, who does not necessarily act in favour of

the customer or may not have implemented appropriate procedures to protect the personal data of his customers, e.g., from ID theft.

To limit the impact of privacy infringements or to prevent them right from the start, it is possible to modify procedures on the vendor's side such that less or no personal information is being collected. Privacy enhancing technologies (PETs) are means designed to do just that, prevent or limit the collection of personal data. Several approaches have been put forth in this regard, mainly focusing on anonymising the communication layer. Although this had been an important step forward, it is often not enough to protect users' privacy, as users are asked much of their private data at an application level that cannot be protected by PETs focusing only on the communication layer. So, in addition to such PETs, we need others that deal with the application layer. Clearly, customers would be better protected from privacy invasions, if by using PETs at an application layer less personal information would be released, as even vendors who sensibly handle personal information — without employing PETs— may be subject to data theft or may be acquired by another company which has a more liberal approach to privacy, as seen in the ToySmart.com example (see the paragraph on "Trading of Personal Data" in Section 2.4).

PETs for the application layer will not be as generally applicable as PETs for the communication layer because the latter can be easier and transparently incorporated into existing applications, as they can rely on certain standards, such as the Internet Protocol (IP) or the HyperText Transport Protocol (HTTP), and the former will mostly be faced with proprietary software and processes that build on it. However, even among processes realised with proprietary means, one can find common grounds. For instance, from an abstract point of view, buying at the online bookstore Amazon is not much different from buying at Barnes & Noble, another online bookshop. Both allow their customers to browse and search their shop, preview certain book content, order books, and of course, arrange payment for the books ordered. Hence, from an abstract point of view, processes are comprised of similar building blocks.

This work is driven by the idea that each building block can be redesigned to allow for more privacy protection of customers. Tackling every block separately is also advantageous as it allows to do a kind of local optimisation with respect to privacy and blend out issues of other building blocks. However, as the blocks together eventually form a process, we still need to pay attention to privacy issues when it comes to the process as a whole. For instance, an online customer does not gain much from using an anonymous payment system when she gave her address before to have the order delivered to her door. The case would be different, however, if no physical delivery is needed to complete the process, e.g., in case of online

services, software, music or video downloads. In this case, the whole process can be carried out anonymously.

Anonymity of customers may pose a risk to vendors, if the employed scheme deprives them of information that is needed, e.g., to guarantee payment. However, no one expects vendors to adopt such economically inviable schemes. Most privacy enhancing tools are therefore designed to only suppress/filter information that — in a strict sense— is unnecessary, say, to complete an order. Other tools may employ cryptography to balance the interests of both customers and vendors, and in particular do not sacrifice vendors' security for customers' privacy, or vice versa.

In the next chapter, we are going to formalise the idea of factoring out the abstract building blocks of processes. We will argue, how control over the linkage of information, treated only informally in this chapter here, can be used to improve the privacy of individuals. For instance, by preventing linkage in the first place, information can be withheld and by making links temporary, information cannot be permanently attached to a person's profile.

Chapter 3

Abstract Model

In our model, we usually consider two communicating parties, a customer and a vendor. As a convention throughout this work, we will speak of the former in the female and of the latter in the male form.

We assume that the customer is always trying to disclose as little information as possible or if certain information is required, she will at least try to distribute her data among non-cooperating parties, such that it becomes harder for any single entity to relate all of her data. Thus, it is her goal to prevent that someone is able to link data that belongs to her.

For the vendor, we assume that he is always eager to learn as much from his customers as possible. He could do this by making the customer provide the information directly (with or without her knowledge) or by accessing data provided by third parties, such as information brokers. In other words, the vendor is assumed to be always curious and he will try to learn which pieces of information relate to his customers or, more specifically, to any single customer. Hence, it is the vendor's interest to link as much data as possible to a specific customer, e.g., to allow for more accurate targeted promotions.

In the following, we are going to formally define what is meant by linking data and what is needed to do so. Since our main interest lies in technically preventing unsolicited profiles of customers, we first need to define what is technically meant by a profile and what its precursors are. Only then will we be able to develop methods which eliminate or minimise these precursors, such that profiles can be prevented in the first place or at least contain less information. However, for a formal definition of a profile, we need some more terminology and definitions which are introduced in the next sections.

$d_0 := (\ell_A,$ `GET /somepath/302-6125076-8274400 HTTP/1.1` $)$
$d_1 := (\ell_B,$ `Host: www.somesite.com` $)$
$d_2 := (\ell_C,$ `User-Agent: Mozilla/5.0` $)$
$d_3 := (\ell_D,$ `Accept: text/xml,application/xhtml+xml,text/html;q=0.9` $)$
$d_4 := (\ell_E,$ `Accept-Language: en-us,en;q=0.5` $)$
$d_5 := (\ell_F,$ `Accept-Encoding: gzip,deflate` $)$
$d_6 := (\ell_G,$ `Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7` $)$
$d_7 := (\ell_H,$ `Keep-Alive: 300` $)$
$d_8 := (\ell_I,$ `Connection: keep-alive` $)$
$d_9 := (\ell_J,$ `Cookie: ubid-acbde=028-1156239-9525019` $)$

$$d_0, d_1, \ldots, d_9 \in \mathcal{D}_{\text{HTTP}}$$
$$D := \{d_0, d_1, \ldots, d_9\} \in \mathfrak{D}_{\text{HTTP}}$$

Figure 3.1: *Example for a set of data elements $\mathcal{D}_{\text{HTTP}}$ and a data space $\mathfrak{D}_{\text{HTTP}}$*

3.1 Links

Let $\mathcal{D}$ denote a *set of data elements* and let data elements $d \in \mathcal{D}$ be pairs of the form (ℓ, x), where ℓ is a *label* and x is some *content data*, usually from $\{0,1\}^*$. Furthermore, let $\mathfrak{D} := 2^{\mathcal{D}}$ denote the powerset of $\mathcal{D}$, i.e., the set of all subsets of $\mathcal{D}$. We will call $\mathfrak{D}$ the *data space* (of $\mathcal{D}$). A particular subset of $\mathfrak{D}$ known by an entity V, e.g., a vendor, will be denoted by $\mathfrak{D}_V$ — we will sometimes refer to this set as V's *view*. Although $\mathfrak{D}$ consists of all subsets of $\mathcal{D}$, usually only a particular subset of $\mathfrak{D}$ will be considered. In the following, when we speak of a data set X it is understood that X is *finite*.

Example. As an example for $\mathfrak{D}$ and $\mathcal{D}$, consider the context of a communication protocol, such as HTTP. We can think of $\mathfrak{D}$ as a set containing all possible messages from the protocol and of $\mathcal{D}$ as the set of all possible records of a message (with an attached label). In Figure 3.1, it is shown how an HTTP message is represented in our model. The boxed elements of rows d_0 – d_9 are records of an HTTP message, i.e., content data, and together they make up a valid HTTP request. As mentioned before, content data is labelled in our model. In the example, the data element d_1, say, is labelled with ℓ_B and also every other "`Host: ...`" data element from $\mathcal{D}_{\text{HTTP}}$ will be labelled ℓ_B. Every "`User-Agent: ...`" element will be labelled ℓ_C, and so forth.

The data space $\mathcal{D}_{\text{HTTP}}$ itself contains all possible instances of valid HTTP records. Consequently, the shown HTTP message $D := \{d_0, d_1, \ldots, d_9\}$ is an element of the data space $\mathfrak{D}_{\text{HTTP}}$ that is associated with $\mathcal{D}_{\text{HTTP}}$. Clearly, all valid HTTP messages are included in $\mathfrak{D}_{\text{HTTP}}$. From the definition of $\mathfrak{D}$ above, it is clear that in this particular context we can also find invalid HTTP messages in $\mathfrak{D}_{\text{HTTP}}$. For instance, the set $D' := \{d_1, d_2, \ldots, d_9\}$ surely is an element of $\mathfrak{D}_{\text{HTTP}}$ but not a valid HTTP message as it is missing an HTTP request / status line. However, this is of no concern here. Usually, we will not deal with the whole data space anyhow and consider only a certain subset, e.g., $\mathfrak{D}_V \subset \mathfrak{D}_{\text{HTTP}}$ containing the HTTP messages exchanged between some vendor V and its customers (which we can expect to be valid because otherwise they would have been rejected by either the vendor's Web server or the customers' browsers). Therefore, when we speak of concrete instances of $\mathcal{D}$ and $\mathfrak{D}$, we silently ignore elements of $\mathfrak{D}$ which are not in accordance with the considered context, if such elements exist at all.

Remark 1. Strictly speaking, for HTTP we would not need labels as the content data itself is already labelled ("`GET ...`", "`Host: ...`", "`User-Agent: ...`", ...), due to the protocol specification. However, if this would not be the case, e.g., if the position of some content data in a message determines its meaning, then the elements may need to be labelled, e.g., with their position. Without labels, content data of the same type that appears at different positions in a message, signifying different meanings, cannot be distinguished. For instance, if a message contains two timestamps, one for the message's creation time at position i and another one for its sent time at position j, then it would not be clear which one is which if we just consider the content data, i.e., the timestamp, and ignore its label, i.e., the position information. In other words, labels help to disambiguate content data, if necessary.

In later chapters, we will largely omit the labels and just give the content data, as ambiguity is usually not an issue when we deal with abstract data, i.e., it will be clear from the context / notation what kind of content data we are dealing with. □

Remark 2. Note that it is not our intention to capture the meaning or property of any data element, i.e., we do not care about semantics in our definitions. For instance, neither are we concerned with the semantics of element d_2 from Figure 3.1, e.g., that the HTTP request might have been sent by a browser of the Mozilla family, nor is d_2's label ℓ_C meant to imply such an interpretation. Here, we are only interested in the fact that some data is collected by some entity. In our view, making sense of all the collected data is subject to a later processing step that is *not* part of our model. □

Next, we let $\mathcal{I}$ denote an *ID space* that contains certain elements which we call *identifiers*. Identifiers allow us to distinguish or relate different data sets, i.e., elements from $\mathfrak{D}$. Examples for identifiers are nonces in a communication protocol, SSNs, credit card numbers, cookies in HTTP, and so forth. The set $\mathcal{I}$ will be induced by an *extractor function* $\xi : \mathcal{D} \to \mathcal{I}$ that extracts identifiers from ordinary data elements. For instance, the cookie element d_9 from Figure 3.1 is comprised of tag information "`Cookie: ubid-acbde=`" and identifying information "`028-1156239-9525019`". The function ξ extracts the latter, e.g.,

$$\xi((\ell_J, \text{“}\texttt{Cookie: ubid-acbde=028-1156239-9525019}\text{”})) = \text{“}\texttt{028-1156239-9525019}\text{”}.$$

If an element $x \in \mathcal{D}$ does not include an identifier then ξ returns the *empty identifier* ε, e.g., taking data element d_8 from the HTTP example, $\xi(d_8) = \varepsilon$. Furthermore, for some data set $X \in \mathfrak{D}$, we let $ID(X) := \{\xi(x) \mid x \in X\} \subseteq \mathcal{I}$ refer to the set of identifiers contained in X. Note that different contexts give rise to different extractor functions and hence, different ID spaces. In the following, we will not explicitly mention ξ again and instead speak of the ID space $\mathcal{I}$ induced by it.

Identifiers are basically shortcuts for referring to certain sets of data, e.g., a specific transaction or a person's profile. A person's name, for instance, can be regarded as a shortcut for referring to the information known about her and at the same time it can be regarded as a data element. An identifier may not necessarily be globally unique, though it will often be unique in a certain context. For instance, the bar code of a product is unique in the context of other products. Conversely, the bar code is no more unique in the context of receipts, as the same bar code may appear on many receipts. Hence, when we refer to elements of an ID space $\mathcal{I}$, its elements may possibly instantiate the uniqueness property only within the context at hand. Note that this means that only the context (or rather the extractor ξ that is determined by the context), and not the element itself, implies some form of semantics for identifiers.

Using the terminology introduced above, we are now ready to formally define what it means that data can be linked.

Definition 1 (Linkable) *We say that two data sets, X and Y, are linkable (can be linked) if a set of identifiers L is known that relates the two. More formally, for a data space $\mathfrak{D}$ and an ID space $\mathcal{I}$, we have a relation $\mathcal{L} : \mathfrak{D} \times \mathfrak{D}$ which we define as*

$$\forall\ (X, Y) \in \mathcal{L}\ :\ \exists L \subseteq \mathcal{I}\ .\ (L \cap ID(X) \neq \emptyset \wedge L \cap ID(Y) \neq \emptyset) \quad .$$

We will write $\mathcal{L}(X, Y)$ as a shorthand for $(X, Y) \in \mathcal{L}$ and we call L the link data for X and Y.

At first sight, the definition above seems to be unnecessarily complicated. However, in the following we will give some examples which illustrate that somewhat

'easier' definitions can easily lead to false inferences. After these non-examples, we will give an example using the correct Definition 1. So, for the sake of illustration, we now expose two *unsatisfactory 'definitions'* which may come to mind.

(a) *Discounting Identifiers.* If X and Y are data sets, then we say that they are linkable if $X \cap Y \neq \emptyset$ is true.

(b) *Ignoring auxiliary information.* If X and Y are data sets, then we say that they are linkable if $ID(X) \cap ID(Y) \neq \emptyset$ holds, i.e., X and Y have common identifiers.

The first non-example serves to illustrate why (a) is unsatisfactory and why we introduced identifiers. Without identifiers, we would most of the time relate things which are really unrelated and this would most likely produce meaningless or wrong data. For instance, to learn how many different persons crossed a certain street between 2:00 pm and 3:00 pm, we let $\mathcal{D}$ be clothing, e.g. red shirt, blue socks, black boots, etc., and hence, $\mathfrak{D}$ would be any combination of pieces of clothing. In this case, the link relation could indicate that a person wearing combination X and crossing the street at 2:05 pm, is the same person as the one who wears combination Y and crosses the street at 2:30 pm, if they both wear, say, brown sneakers. Usually this alone hardly suffices to tell whether the two are the same and hence, the results we are interested in would only be correct by chance. □

Regarding (b), we will show that this definition is actually subsumed by Definition 1 and also that it is not universal enough, as we will explain in a moment. But first, let us see why (b) is indeed a special case of Definition 1. Suppose we have link data $L := ID(X) \cap ID(Y)$ and $L \neq \emptyset$. Using this and applying Definition 1, we get

$$L \cap ID(X) = [ID(X) \cap ID(Y)] \cap ID(X) = ID(X) \cap ID(Y) \tag{3.1}$$
$$L \cap ID(Y) = [ID(X) \cap ID(Y)] \cap ID(Y) = ID(X) \cap ID(Y) \tag{3.2}$$

Plugging Equations (3.1) and (3.2) into Definition 1, we get $ID(X) \cap ID(Y) \neq \emptyset \wedge ID(X) \cap ID(Y) \neq \emptyset$, which simplifies to $ID(X) \cap ID(Y) \neq \emptyset$, as in (b). Hence, (b) is a special case of the more universal Definition 1.

An example for such a case would be a vendor's charging data. If, say, $X, Y \in \mathfrak{D}$ are receipts of some vendor's customers and the receipts include information about purchased goods as well as payment information then $\mathcal{L}(X, Y)$ holds, e.g., if some customer used her credit card twice to pay for the goods found in X and Y. In this case linking is easy because the credit card number appears in both receipts. Note that, in contrast to the former example, we have an identifier, the customer's credit

card number, which is uniquely associated with a specific customer. However, we will not always have such a nice case where the link is apparent. □

We may encounter situations where two sets of data can be related, even if they do not have an identifier in common. This is the most general case which is also the motivation for Definition 1. As an example for the general case, where the link is not apparent, consider two data sets, X and Y, which belong to the same person. Assume that X contains the person's phone number (an identifier) and Y the person's name and address (another identifier). In absence of any other identifying data, the two sets cannot be linked. However, looking into an ordinary phone book (auxiliary information), one may find an entry E which is comprised of the person's phone number, her name and address, and possibly other data. By matching elements from $L := ID(E)$ with X's and Y's identifiers, a link from X to Y (and vice versa) can be found. Although, more than one link might be found for two data sets, for our purpose it is usually enough that at least one link is known.

For the last example, we will illustrate that not only the definition of "linkable" itself is important but also how it is applied. In particular, we stress that the ID space $\mathcal{I}$ also needs to be carefully chosen for the definition to be of use. One necessary requirement for $\mathcal{I}$, though not the only one as we shall see, is that its elements can be used to uniquely relate elements $X \in \mathfrak{D}$. However, uniqueness does not necessarily guarantee that links are always meaningful. For instance, let $\mathfrak{D}$'s elements be any data sets of purchase information, e.g., a receipt $R :=$ { "1 yogurt", "2 bananas", "1 pkg. of toast", ..., "grand total 9,59 €", "checkout 17", "2006/05/16/17:23:47"}[1]. Today, each good is usually assigned a bar code, which is a globally unique identifier for a certain good. However, choosing bar codes for $\mathcal{I}$ would be a bad choice, if the goal is to distinguish purchases of different customers. The reason for this is that two *unrelated* customers, say, Alice and Bob, may buy the same product and hence, the product's bar code will be printed on each of their receipts, R_A and $R_B \in \mathfrak{D}$, respectively. This, however, would indicate a link from Alice's purchase R_A to Bob's purchase R_B, which is clearly not what was intended. Hence, the ID space must also be chosen such that the same identifier will only be used if the entity assigned to the identifier is the same (with a high probability). □

The link relation $\mathcal{L}$, apart from its broad applicability, has some other nice properties which make $\mathcal{L}$ an *equivalence relation*. To show this, we check the required properties of an equivalence relation, i.e., reflexivity, symmetry, and transitivity, with respect to $\mathcal{L}$.

[1] Labels have been omitted.

Reflexivity. Given $X \in \mathfrak{D}$, it can easily be seen from the right-hand side of the definition, that, for $Y := X$ and $L := ID(X)$, $\mathcal{L}(X, X)$ holds.

Symmetry. Given $\mathcal{L}(X, Y)$, we have $L \cap ID(X) \neq \emptyset \wedge L \cap ID(Y) \neq \emptyset$, for some L, which is equal to $L \cap ID(Y) \neq \emptyset \wedge L \cap ID(X) \neq \emptyset$, which in turn is the definition of $\mathcal{L}(Y, X)$.

Transitivity. Let $\mathcal{L}(X, Y)$ and $\mathcal{L}(Y, Z)$ be given and let L_{xy} and L_{yz} be one of their sets of link data, respectively. Setting $Y := Z$, $L := L_{xy} \cup L_{yz}$ and plugging this into the right-hand side of Definition 1, we immediately get $\mathcal{L}(X, Z)$. ■

3.2 Transactions and Profiles

In this work, the core problem we are dealing with is the creation of profiles against the users' will, and possibly even without their knowledge. In the following, we will use the introduced notion of links to give a formal definition of a profile and model how profiles are created.

In practice, we usually do not encounter isolated data sets, as introduced in the previous section. Instead, we often have groups of data sets which, in a certain context, logically or semantically belong together. For instance, the process of finding an information via the Internet typically involves accessing an Internet search engine, entering certain search terms, and sifting through the returned hits, which may result in access to the hyperlinked pages. Every previously mentioned action produces data, e.g., access to the engine typically reveals the user's current IP address, possibly her operating system, and browser type, and the search engine's response to a query contains the search terms and hyperlinks to pages which may be related to the search terms. So we can say that the data set Q containing the user's query is related to the search engine's response data set R, i.e., $\mathcal{L}(Q, R)$. And also, the data set A_i produced by the user's access to any one web site hyperlinked in the response can be said to be related to R, i.e., $\mathcal{L}(R, A_i)$, and naturally also to the query Q, which already follows from the transitivity of $\mathcal{L}$. Note that the same query sent at a different time will usually be regarded as a different query and so are their associated data sets. In order to separate related sets of data from other sets of related data we introduce the following definition for a distinguisher.

Definition 2 (EID) *An ephemeral identifier (EID) is an element from the powerset $2^{\mathcal{I}}$ which relates a given number of data sets in a given period of time τ. That is, if for data sets $X_1, X_2, \ldots, X_m$ produced from time t until time $t + \tau$ it holds that $\mathcal{L}(X_i, X_j)$ for $i, j \in I := \{1, 2, \ldots, m\}$ then we will call the set of link data $\epsilon := \bigcup_{i,j \in I} L_{ij}$ an ephemeral identifier.*

In practice, however, we often encounter EIDs which only contain a single identifier, such as a session cookie or a dynamic URL. For instance, the element d_0 of Figure 3.1 is an example for an EID, as the dynamic URL of the shown request is a means to identify a certain session in the context of HTTP. In other words, it serves to relate all data that is exchanged between a client and a server as long as the session is active. Note, however, that we do not have a formal understanding of a "session", yet. □

We also want to be able to talk about the set of related data sets because they usually do not appear randomly but are means to serve some 'higher level purpose', e.g., "finding a paper" or "buying a book in an Internet shop". Although we do not intend to formalise the semantics of related data sets, we still want to look at them as a kind of high level object that is comprised of certain data. We call such a high level object a *transaction* and define it as follows.

Definition 3 (Transaction) *Let $\mathfrak{D}_V \subseteq \mathfrak{D}$ be the data space observed by V through communicating with others. Let $\mathcal{E} \subseteq 2^{\mathcal{I}}$ be a set of EIDs —possibly chosen by V— and let the data sets $X_1, X_2, \ldots \in \mathfrak{D}_V$ be given. We call a set T_i of data sets a* transaction *if they are related by the same EID $\epsilon_i \in \mathcal{E}$. More formally we have*

$$T_i := \{X_j \mid \mathcal{L}(X_j, X_l);\ L_{jl} \cap \epsilon_i \neq \emptyset;\ X_j, X_l \in \mathfrak{D}_V\},$$

where L_{jl} denotes the link data for the relation $\mathcal{L}(X_j, X_l)$. If we do not care about a particular EID, we simply drop the index i from the notation.

Informally, the definition says that a transaction is the sum of all related data sets, i.e., those which can be linked by the same EID. Using this definition, we can view a "session" as an instance of a transaction. In our example from Figure 3.1, all message exchanges between a certain client and a server, i.e., requests and responses, will use the same *session identifier* found in d_0.

Note that, by the definition above, $\mathfrak{D}_V$ is partitioned into transactions. A closer look reveals that these partitions are a consequence of $\mathcal{L}$ being an equivalence relation. Hence, in our model, "transaction" is just another name for "equivalence class under $\mathcal{L}$". Also note that an EID can be regarded as a *transaction pseudonym* and hence, it is the best we can hope for with respect to privacy (c.f. Section 2.3). □

In practice, vendors often use identifiers whose scope goes beyond a single transaction. We have briefly mentioned relationship and person pseudonyms in Section 2.3, which are identifiers that last for the duration of a business relationship or even for a lifetime, respectively. Since such identifiers outlive EIDs, we call them

persistent identifiers (PIDs). In contrast to an EID, a PID will typically be used more than once, e.g., to identify an already registered customer who proceeds to the checkout. As a consequence, PIDs can be used to link different transactions of the same customer and therefore, they are relevant with respect to the customer's privacy. With the next definition, we incorporate PIDs in our model.

Definition 4 (PID) *A persistent identifier π (PID) is an element from the powerset $2^{\mathcal{I}}$ which relates* distinct *transactions. More formally, let $\mathfrak{T} := \{T_1, T_2, \ldots, T_m\}$ be a set of transactions. If it holds that for any two distinct transactions $T_i, T_j \in \mathfrak{T}$, where $i, j \in I := \{1, 2, \ldots, m\}$, we have $\mathcal{L}(X_i, Y_j)$ for some data set $X_i \in T_i$ and some data set $Y_j \in T_j$ then we call the unison π of each related transaction T_k's EID ϵ_k a persistent identifier. Thus, we have $\pi := \bigcup_{k \in \hat{I}} \epsilon_k$ for a persistent identifier, where $\hat{I} \subseteq I$ is the index subset of the related transactions T_k. If two transactions T and T' are related by a PID we write $\mathcal{L}(T, T')$.*[2]

Note that a PID, similar to an EID, may also be comprised of only a single element, e.g., if the identifier is issued only once and always included in the transactions belonging to the same customer. The difference between a PID and an EID is that the latter might be reused (after some time) for another person while the former, in general, will not.

Examples for PIDs are static IP addresses, login information, name and address, credit card numbers, etc. PIDs may also allow to link transactions to data obtained from other sources, i.e., data obtained by other means than direct communication with the data subject. Although we will usually not call such data a transaction, it is technically not different from a transaction and hence, also fits into the model.

Finally, we have arrived at the point where we can give the term "profile" a formal meaning. Similar to transactions, we define a profile by using the concept of a PID.

Definition 5 (Profile) *Let a set of PIDs $\mathcal{P} \subseteq 2^{\mathcal{I}}$ —possibly chosen by V— and a set of transactions $\mathfrak{T}_V := \{T_1, T_2, \ldots\}$ accessible by V be given. We call a set $Prof_i$ of transactions a* profile *if they are related by the same PID $\pi_i \in \mathcal{P}$. More formally we have*

$$Prof_i := \{T_j \mid \mathcal{L}(T_j, T_l);\ E_{jl} \cap \pi_i \neq \emptyset;\ T_j, T_l \in \mathfrak{T}_V\},$$

where E_{jl} denotes the link data for the relation $\mathcal{L}(T_j, T_l)$. If we do not care about a particular PID, we simply drop the index i from the notation.

[2] This notation somewhat abuses the relation $\mathcal{L}$ because $\mathcal{L}$ is defined over pairs of data sets and not over pairs of sets of data sets. It is, however, straightforward to define the link relation for transactions too, as the definition does not change much, except for some 'adjustments' with respect to the underlying mathematical structure of the paired elements.

Informally, the definition above says that as soon as a user is assigned one or more persistent identifiers, a profile of the user can be build by linking all transactions that include these identifiers. For instance in Figure 3.1, a PID is shown in the form of a persistent HTTP cookie that is found in element d_9 of the HTTP request. In the context of HTTP, a user's browser will always include this element in messages exchanged with, e.g., a vendor's web server. Consequently, every time the user contacts the vendor, the vendor will recognise her and this allows him to relate information obtained from past sessions —in other words transactions— with her current session. The knowledge obtained from previous transactions of the user, i.e., the profile, may then be used by the vendor, e.g., to recommend goods to the user which she might be interested in according to her profile.

Note that a profile comes down to a transaction, if V can neither link distinct transactions nor relate transactions with other data — having said this, a transaction can be seen as a special case of a profile. Hence, if a customer's goal is to prevent some vendor V from learning anything beyond what is conveyed in a single transaction, the most favourable situation is the case where we only have *trivial profiles*, i.e., $Prof_i = T_i$, for all of V's profiles $Prof_i$ and transactions T_i.

3.3 Applications of the Model

We have modelled a transaction $T \subseteq \mathfrak{D}$ as a set containing abstract data sets (messages), which are normally exchanged by a customer C and a vendor V at some occasion. However, in economics, for instance, more fine-grained models are often used to characterise a transaction. An example for such a model is the three-phases-model consisting of an *information phase*, an *agreement phase*, and a *settlement phase* [SL98, PRW03, Reb00]. Hence, phases can be seen as a means to further group certain data within a transaction and give them additional semantics. Concrete models, such as the three-phases-model, can be generalised to a model where a transaction T is viewed as a finite sequence of phases P_i, i.e., $T := (P_1, P_2, \ldots, P_n)$. From the point of analysis with respect to privacy, however, phases P_i still constitute data sets, i.e., $P_i \in \mathfrak{D}$, and hence, also fit in our abstract model.

In Figure 3.2, the relations between EIDs, PIDs, transactions, profiles, and phases are illustrated. Each row in the figure represents a transaction T. The transactions consist of several phases $P_{i,j}$. For each phase, the data seen by the vendor is given in curly braces $\{\cdot\}$ below the phase ID $P_{i,j}$. This data is used to link the phases of a transaction with each other. The link data is an instance of an EID and can be viewed as auxiliary information available to the vendor. It is given in angled braces $\langle\cdot\rangle$. Since the links between the phases represent an equivalence relation, and hence include transitive links, we could have drawn links/edges between any two phases of the same transaction. However, for the sake of read-

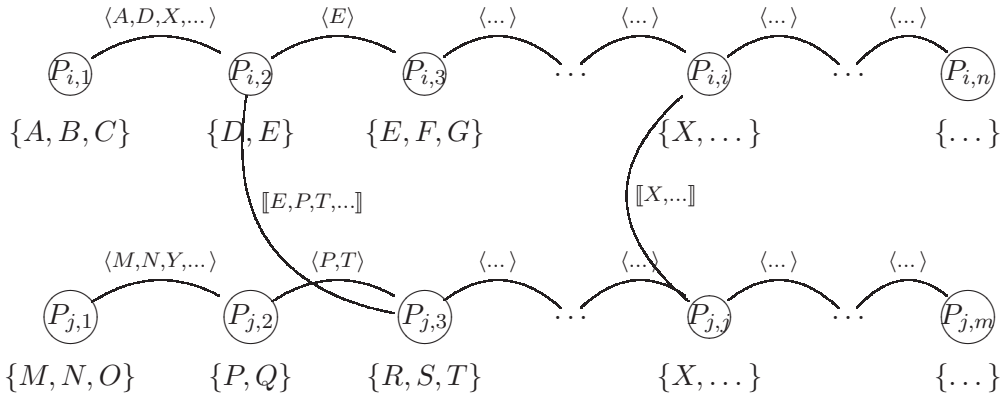


Figure 3.2: *Links within and between Transactions*

ability, we have left them out. Note that link data need not be minimal, i.e., link data may contain more information than necessary to establish a link between two phases. For instance, the link between $P_{i,1}$ and $P_{i,2}$ consists of $\langle A, D, X, \dots \rangle$ where the minimal link data would be $\langle A, D \rangle$, as these are the elements included in $P_{i,1}$ and $P_{i,2}$, respectively. Also note that for a link it is not necessary that some common data element exists in the linked phases: For the link $\mathcal{L}(P_{i,2}, P_{i,3})$ a common element exists while there is none for the link $\mathcal{L}(P_{i,1}, P_{i,2})$. The same is true for links between phases of different transactions, i.e., PIDs. Links between phases of different transactions are given in square brackets $[\![\cdot]\!]$. As we have links between two transactions in Figure 3.2, they together make up a non-trivial profile.

So far, we only talked about phases as abstract sets of data. In practice, of course, more concrete instances of phases are given for any business process. One such instance is the phase model frequently used in economics. This model, as possibly other phase-based models as well, can be used to give phases a meaning and therefore, add semantics to our abstract model. As an example of a model that uses four phases to specify a business transaction, we consider the following subdivision of a transaction into:

- Search
- Order
- Payment
- Delivery

The subdivision of an e-commerce transaction into these four phases is commonplace in economics [SL98, PRW03], though they are not always made explicit.

In cases where fewer phases are used, the left out phases are usually included as actions in the remaining phases [Reb00]. Sometimes even an additional fifth phase, *after sales*, is included [PRW03, Reb00] which refers to services such as customer support or marketing efforts for customer retention, which may take place after or within the transaction.

The abstract model introduced before, allows us to reason about the privacy of instances of such a phase-based model. Specifically, by using this formalism, we can determine the type and extend of data gathered in a given phase and whether the data from one phase is propagated to any subsequent phase or even to a subsequent transaction, the latter ultimately leading to a profile.

3.4 Privacy Protection by Decoupling

The model presented in the previous sections is an effort to systematically and formally approach the anatomy of Internet users' data trails, in order to better understand what can be done to improve the privacy of online users. Data trails are often generated by the users themselves —or with their assistance— while they interact with other online entities. If data trails can be linked to a personally identified user, the user's privacy may be at risk, as it is often unclear how this information will be used, i.e., if it is being used in the user's favour or to her disadvantage.

Our model, however, is completely oblivious to the intentions of the parties involved in the generation and potential exploitation of data trails. It does not allow to associate or calculate a specific risk whenever certain data is disclosed to a certain party. Instead, our approach is more qualitative than quantitative in nature. That is, we take on the conservative position of most privacy advocates that the more data is collected about an identified person, the higher the *potential* for privacy invasion.

Data trails are not all alike and thus, we intend to capture their anatomy/characteristics within an abstract model. Trails can be relatively short-lived, making them less of a threat, or long-lived causing them to potentially follow one through life (c.f. Section 2.4). In order to accommodate both, we introduced the notion of ephemeral identifiers (EIDs) and persistent identifiers (PIDs), giving rise to either short-lived transactions or possibly longer lived profiles, respectively.

The relation of transactions and profiles, i.e., the former being collected in the latter, brings up the natural idea of simply preventing such relations, i.e., preventing that transactions can be linked with each other and subsequently being stored in a profile. In the following, we will refer to the *design concept* of eliminating links as *decoupling*. Decoupling is not quite a methodology, as it does not govern the methods to achieve decoupling, other than to prevent links in the first place. No

matter how decoupling is eventually realised, the abstract model can always serve as a starting point to analyse the generation process of the data trail at hand and possibly provide guidance to design a more privacy-friendly process, yielding less or no personal data.

The idea of decoupling transactions had already been implicitly employed in the past when electronic payment systems emerged in the 80's and 90's. Many of these systems offered anonymity in the payment process with respect to the vendor and to the bank who were to accept and to clear the electronic money, respectively. In other words, the systems prevented links between the customer's withdrawal and her payment transaction. However, from our point of view, the latter was merely one phase in a larger transaction that usually includes more than just payment.

Still, much work with respect to privacy went into the payment phase, mostly from academia (see for example [BGH+95, BGK95, CFT98, Cha89, CFN90, Fer93, JY96, OO90]) but also industry made an effort to adopt some of the proposed schemes, e.g., [BGH+95, Cha89, MV97a, MV97b, MV97c] — a comprehensive list of payment systems and their respective properties can also be found in [SS03]. However, little efforts have been made for the other phases. Noteworthy exceptions are the European project SEMPER [LPSW00] and the German project DASIT [EE02, ER02, Roß02] which both pursued holistic approaches, i.e., all phases of a business transaction were taken into account for security and privacy. However, only in DASIT complete pseudonymity with respect to the vendor was possible when tangible goods needed to be shipped.

A problem with holistic approaches is, however, that they tend to produce frameworks which are complex, difficult to integrate in existing environments and therefore, hard to deploy. Another problem is that they often lack flexibility as one cannot easily replace some phase's component by another one without effecting components of other phases because of existing dependencies.

Our approach is more fine-grained and more general, as it also allows to deal with information at the phase level. Furthermore, it can be employed in any kind of transaction, as we are only dealing with data itself and not with its semantics — except for identifiers, of course. Specifically, our goal is to prevent unnecessary information of some phase to propagate to subsequent phases —from the same or from a different transaction— if that information is not required to progress the transaction as a whole. As an example for what this means, consider the information created during a customer's information phase. In this phase, customers often search through the vendor's catalogue, click at this promotion and that promotion, view this item and that item, before they eventually select some of them and proceed to the checkout, or simply leave the site. Now, assuming the customer proceeds to the checkout, i.e., moves on from the information phase to the agreement phase, all information required from the previous phase is which items she wants to buy and

how many of them. Everything else is not required, though it is normally available to the vendor and may be used for profiling the customer, e.g., to cross-sell products or to promote viewed items in future visits (c.f. Section 2.4).

Following this idea, we have employed our model to systematically go through the phases of typical online transactions and found that, by eliminating certain unnecessary links, data relating to a specific person can be reduced at several occasions which had not been considered before. However, these occasions, or rather phases, can provide great insights into a customer's behaviour, as well as her likes and dislikes, even if she is not aware of that. On this account, we developed privacy-enhanced components for phases which have received little or no attention so far. To be specific, we have developed components improving privacy in the customer's search phase and components dealing with privacy in the after sales phase.

The component-based approach which we pursue is also flexible as it allows us to replace a component of one phase while leaving in place others, e.g., payment components. By encapsulating components, we also support re-usability and extensibility. For instance, if an additional phase is to be introduced, previously developed components can be used for the old phases and only for the new phase will it be necessary to develop a new solution. This requires that components for the different phases do not functionally depend on each other, i.e., no component should require that any other component is present in order to achieve its privacy goals (with respect to a given phase). This isolation of components should, however, not preclude using the components in combination. Moreover it is even desirable to have several components in place to allow for more privacy safeguards.

Indeed, one component will generally not be enough to defend the whole transaction against privacy infringements, since the protection of one's privacy throughout a whole transaction will normally depend on all phases of the transaction —though some phases will potentially be more privacy intrusive than others— and possibly even on subsequent transactions. However, starting to improve privacy from the phase level allows us to employ a kind of divide and conquer strategy because by starting to solve small problems at first, i.e., improving privacy within phases, we will eventually have a solution for the big problem, i.e., enhanced privacy for the whole transaction. Therefore, even if we do not have solutions for every phase of a given type of transaction right now, this should not stop us from employing the available privacy-enhancing components as these may still improve privacy compared to the situation today.

Chapter 4

Enhancing Privacy by Decoupling Searches and Orders

In this chapter, we are going to formally model what it means, with respect to privacy, that search and order phases can be linked. For this, we will use the abstract model from Chapter 3 to derive a model for reasoning about the relationship of search and order phases. Furthermore, we will introduce a conceptual solution that can be employed to 'unlink', or decouple, phases from each other. We will see that this general concept can be implemented in a number of ways, each with different strengths and weaknesses, as well as different requirements for the technical environment.

Major portions of the following text have been presented before in [EKS02d, EKS02b, EKS02a, EKS02c, EKS03]. These works, however, did not include the model part and the conceptual solution introduced here.

4.1 Introduction

In today's Web practice, we can identify a lack of privacy enhancing technologies [Cla99]. It can be assumed that this lack of adequate privacy enhancing technologies is an additional barrier for the diffusion of e-commerce applications [HNP99, WLW98]. Thus, there is a need to change the present situation by the introduction of new technical solutions that allow to avoid or to reduce the invasion of privacy.

In academic work done so far, solutions for protecting the privacy of customers were mostly proposed for cases where the trade objects are restricted to intangible goods, e.g., see [BD01, BDF01, SSG99]. Intangible goods, such as electronic documents, images, music, or video files, can be delivered via communication net-

works, in contrast to tangible goods which require shipping. This means with intangible goods, all phases of a typical business transaction consisting of *search*, *order*, *payment*, and *delivery* can be handled electronically and the schemes developed for communication networks can be used to protect a customer's privacy, e.g., anonymity networks [RSG96, RR98, SRG97] and anonymous payment systems [Cha83, Cha89].

When dealing with tangible goods, e.g., books or CDs, these techniques can also be used. In the search phase, when the customer browses through a product catalogue, anonymity networks can be used to prevent re-identification, which also helps to protect against some other threats, such as price discrimination. Such techniques may help to prevent linking of transactions, however, they cannot be used to prevent linkability between phases, as phases are a concept of the application layer and general anonymisation techniques apply only to the lower communication layers.

In practice, in order to receive a tangible good, a customer has to reveal her identity and address to the vendor to allow for delivery. Alternatively, she could use a shipping address that does not reveal her real identity to the vendor, such as a P.O. box, or in a more advanced scenario, an additional third party may receive the package on behalf of the customer and re-sends it to the customer's real address. In practice, only a few people have a P.O. box and we are lacking an infrastructure like the one mentioned before. Thus, the vendor normally learns at least who is buying what. However, the vendor can learn much more today. He can link the data from a customer's order phase to her activities from the search phase and thus get a much deeper insight into the customer's interests than necessary for subsequent phases of the business transaction. This situation can be compared to real-world scenarios, where one is being completely observed while flipping through a catalogue before filling out a mail order form, or if one is being traced while walking through a brick and mortar shop before proceeding to the checkout.

The vendor is able to link these two phases, e.g., by using IP addresses, cookies, and dynamic URLs which allow to introduce the concept of sessions in HTTP communication. In general, countermeasures are available to customers which allow them to avoid linking by themselves. Unfortunately, such countermeasures are not very convenient because they require several additional steps.

In the following, we will introduce a conceptual solution which reflects the main idea of decoupling phases of a transaction started by a particular customer. Decoupling prevents the vendor from obtaining unnecessary information that only serves to create customer profiles. As we will see, decoupling phases can be difficult, given today's Web technology. An abstract solution, as in our concept of a "decoupling component" (DC), will convey the general idea of decoupling phases, however, it will not help in a concrete scenario. Therefore, we will explore two practical instances

of DCs which, among other things, will have different impacts on a customer's usual buying behaviour, on the necessary technical prerequisites for customers and vendors, and on the technical environment in general.

One of these instances is based on the concept of mobile agents. In this approach, a central mobile agent base station is required, which is assumed to be permanently online. At the base station, users can initiate their agents' shopping trips and they can also pick them up at this base station after the agent has finished all its tasks and returned to its station.

4.2 Tracking Users

The vendor's ability for linking customer activities can be based on several technical possibilities which allow user tracking in HTTP communication. Here, "tracking" means the re-identification of a user subsequently sending requests to a server. We can distinguish between the re-identification of users in distinct sessions and the re-identification of users within one session, possibly in different phases. In HTTP communication, the term "session" has the same meaning as transaction in the formal model introduced in Section 3.2. Hence, a session can be understood as a concrete instance of an abstract transaction. Likewise, the elements discussed in sections 4.2.2 and 4.2.3 are concrete instances of identifiers, EIDs or PIDs, which allow linking in the first place (see Sections 3.1 and 3.2).

4.2.1 Means to Collect Profiles

As long as customers do not prevent the vendor from tracking them, the vendor is able to record profiles. Such profiles can be understood as a sequence of requests for resources —like Web pages— that can be associated with a specific session, and hence with a specific customer, using the methods presented below.

When a customer C clicks on products $p_1, \ldots, p_\nu$ while browsing through an online catalogue, the vendor will be able to collect these events in a corresponding search phase. In the following, we will discuss several possibilities for user tracking which are suitable for creating profiles and associating profiles with customer identities. In general, user tracking can be achieved by exploiting characteristics of the underlying transport protocol(s) or by specially crafted content. In both cases, certain protocol and/or content elements can be used to introduce a link, which in the following is sometimes called a *hidden channel* because users are often not aware of such a link.

4.2.2 Tracking by Protocol

IP Addresses. The first means to link a transaction's phases is given by IP addresses. From the customer's perspective, one possibility to cope with this problem is to dis- an re-connect in order to conduct the order phase with a new IP address obtained from her ISP after re-connecting — a solution which is not very convenient. Another option to solve the problem could be the use of an anonymity network based on mixes [RSG96, SRG97] which hide her computer's IP address from the vendor.[1]

But if all requests are routed via the same sequence of mixes the probability for correctly linking information on the vendor's side can be very high, since they can be correlated with the IP address of the last mix in the chain. Another possibility for the customer to hide her own IP address could be the use of the *crowds* approach where routes of subsequent messages are different with high probability [RR98]. But all these anonymisation countermeasures can be circumvented if a vendor uses cookies or dynamically generates user-specific URLs, which are unique within a certain time frame (c.f. Figure 3.1, element d_0 and Section 3.2). □

Cookies. Another means to track the customer and to link the phases is given by cookies [KM97]. Cookies allow the vendor to create a stateful context, the session, which normally would not be possible because the HTTP protocol is stateless. To prevent a cookie-based session, a customer may refuse cookies. However, for shopping applications, they are often required. In a more laborious way, the customer could first browse through the product catalogue, then delete her cookies, and afterwards come back to the desired products and fill them in the virtual shopping cart without any further detours. Beside the inconvenience of this, the vendor can still track the customer with the method presented next. □

Dynamic URLs. The concept of a session has also been introduced to HTTP by way of so called dynamic URLs. In a dynamic URL, a dynamic part of the URL which is returned to a customer is unique in the sense that distinct requestors of the same Web resource can be distinguished via this part. This allows a Web server to track a customer. A countermeasure for this would be to shut down the browser after the search phase and restart it in order to go directly to the order phase, similar to the cookie case's solution. □

[1] Mixes sometimes also serve the purpose of preventing third parties from observing communication between any two communicating parties. However, this property, called *unobservability*, is of less concern here. We are just interested in *sender anonymity*, which is another property that can be realised with Mixes (c.f. Sections 2.3 and 2.5).

The previous considerations show, that presently, there are only inconvenient solutions for the customer to prevent a vendor from undesired linking.

4.2.3 Tracking by Content

A less obvious and more subtle method for tracking users is using the content information that a user requests. Using content tracking, a vendor serves all of his customers slightly different content in one phase in order to use this content in a subsequent phase for re-identification of the same customer. This is possible if the content, or a part of it, is re-sent to the vendor in a later stage. For instance, it is possible to associate a profile created in the search phase with a specific customer from the order phase, if such tagged content is presented in the order, no matter how carefully the customer avoided all protocol tracking. Next, we present some possibilities to realise content tracking which, of course, can also be used in combination.

Product identifiers. One way of identifying a customer's search phase *a posteriori* is using specially encoded product identifiers (*ProdID*s). Such an identifier could contain a static part that refers to the product and a dynamic part that identifies a certain search phase. Since customers are used to numeric *ProdID*s, they will not notice that they are given information leaking *ProdID*s. In a way *ProdID*s use the same concept as dynamic URLs, but on a content level. □

Prices. By giving different customers different prices, prices can also be used for the purpose of linking phases when contained in an order. In contrast to *ProdID*s, prices only allow unique re-identification in theory. In practice, a vendor only has a finite range of prices which customers are willing to pay for a given product. Thus, if this range is depleted, it is necessary to re-use some of the prices and hence, uniqueness is forfeited. However, if the customer eventually purchases more than one product than the combination of two or more tagged prices may establish uniqueness, again. □

Product ordering. Another method for content tracking is correlating the sequence in which products were clicked in the search phase with the product sequence submitted in a customer's order. This is possible, since the goods in the order normally appear in exactly the same sequence as in the customer's search phase. Although in the search phase, they may be interleaved with other products that had been viewed but not ordered by the customer. As with prices, using product ordering for mapping profiles to customers is probabilistic, since several customers may have produced the same click order of products.

4.3 Model for the Relationship of Phases

In this section, we put aside the practical considerations from above for the moment. We do this for the benefit of a more general and more abstract view on the subject. We will return to practical considerations after the introduction of the abstract model. In the following, we will stick with Internet shopping as a use case to illustrate the general options for the prevention of links between search and order phases. This serves to illustrate the general ideas developed for limiting the disclosure of data which is unnecessary for the purpose at hand.

We already outlined that rich information is provided by customers searching for their items of interest, such as, customers looking for books in an online store will often search for more than one book, potentially revealing many interests. Although we use the term "search" to denote a particular phase, we do not presuppose that search engines or the likes are actually used in this phase. Our perspective on search is fairly broad, i.e., we also view things like casual browsing of a Web site, clicking banner ads, etc. as potential actions for a search phase. For us, the important point is that the data from the search phase becomes personalised data in a later stage.

The data of a search phase can be roughly subdivided into two sets of data:

(a) *required data* which is information needed for the completion of any subsequent phase, e.g., the name of the book which a customer finally decided to buy, and

(b) *extra data* which is whatever other information the vendor may learn from the search phase, e.g., additional items of interest which the customer searched for.

In order to proceed with a transaction, it is clearly irrelevant to process the extra data. However, the extra data may contain information which is useful for expanding the customer's profile. Building or expanding a profile with this extra data is easily possible as soon as any phase exists in the same transaction that identifies the searching user. For instance, in the 4-phase model (see Chapter 3) the search phase is followed by an order phase where customers usually have to provide their names and addresses to get their purchases shipped.

Since our goal is to minimise personal data in general, in the following, we will introduce an approach which prevents that the extra information from a user's search phase can be exploited in a privacy-intrusive manner, e.g., being added to a profile in order to get a more detailed picture of the customer's personality.

Using our model from Chapter 3, we will next abstractly model the relationship between search and order phases in order to devise two general ideas that prevent this relationship, and which in turn help to eliminate extra data. For this, let $\mathfrak{P}$ denote an instance of a *set of phases* seen by a particular vendor V, i.e., $\mathfrak{P}$ is what

we denoted $\mathfrak{D}_V \subseteq \mathfrak{D}$ in earlier chapters. Note that the link data for relating the phases can be easily obtained by the methods presented in Sections 4.2.2 and 4.2.3.

Now, consider a transaction $T \subseteq \mathfrak{P}$ consisting of phases $P_1, P_2, \ldots, P_\nu$. Let $\mathcal{S} := P_i$ and $\mathcal{O} := P_j$, $i < j$, be search and order phases, respectively. Although we use the term "order phase" in the following to refer to a phase where the user is identified and some of her information from the search phase is required, we do not mean to imply that this is the only type of phase where our scheme can be employed. It is just more convenient to have a name for 'the phase where the user is identified and some of her information from the search phase is required'.[2] Hence, "order" could be substituted with, say, "bid" if the scenario at hand is an online auction. Also, we will simply call the customer's items of interest *goods*.

We let $\mathfrak{G}$ denote the *set of goods* offered by the vendor and we define $goods : \mathfrak{P} \rightarrow \mathfrak{G}$ as

$$goods(P) := \{g \mid g \in (P \cap \mathfrak{G})\} .$$

In particular, any search phase $\mathcal{S}$ will contain a number of goods $g_i \in \mathfrak{G}$. As mentioned before, some of the goods will be selected for purchase by the customer and become order data. Clearly, every $g_i \in goods(\mathcal{O})$ must also be in $goods(\mathcal{S})$.

Given two search requests $S_i, S_j \subseteq \mathcal{S}$, we necessarily have $\mathcal{L}(S_i, S_j)$ [3] — otherwise we would not be able to group them in $\mathcal{S}$. Hence, the customer's search requests will be linkable by some sort of session identifier, i.e., an EID or possibly even a PID.

If we have $\mathcal{L}(S_i, S_j)$, for $S_i, S_j \subseteq \mathcal{S}$, such that $goods(S_i) \subseteq \mathcal{O}$ and $goods(S_j) \not\subseteq \mathcal{O}$, i.e., the goods from search request S_i are purchased while S_j's are not, then pure search data is mixed up with order data in the customer's search phase $\mathcal{S}$, resulting in unnecessary extra data. Since we assumed that $\mathcal{S}$ and $\mathcal{O}$ are part of the same transaction, i.e., $\mathcal{L}(\mathcal{S}, \mathcal{O})$, the extra data becomes personalised data and further feeds the customer's profile. However, if the two phases can be separated, it is possible to minimise the information learned by the vendor by withholding the extra data. In other words, the vendor would only learn of goods g_i for which $g_i \in goods(\mathcal{S} \cap \mathcal{O})$ holds, i.e., goods searched for in $\mathcal{S}$ which become ordered goods.[4] To achieve this, the link data between the search and order phase needs to be eliminated. Assume for the moment that this can be done (we will later present an approach which achieves this). In this case, while the vendor will still know that for a given order phase $\mathcal{O}_j$ there must be a matching search phase $\mathcal{S}_i$, he may not know which particular search phase relates to $\mathcal{O}_j$.

[2] Note that "identified" does not necessarily mean personally identified.

[3] Since the relation $\mathcal{L}$ is defined over phases, one should think of S_i and S_j as phases consisting of only a single request.

[4] Note that this is the minimal information needed by the vendor to calculate the grand total of the purchase, assuming differently priced non-digital goods.

Let $\mathfrak{S} \subseteq \mathfrak{P}$ be the *set of search phases* recorded by some vendor. Given an order phase $\mathcal{O}_j$, we define $\mathfrak{S}_j \subseteq \mathfrak{S}$ as

$$\mathfrak{S}_j := \{\mathcal{S} \mid \mathcal{S} \in \mathfrak{S} \;.\; \mathit{goods}(\mathcal{S}) \supseteq \mathit{goods}(\mathcal{O}_j)\} \;.$$

In other words, the elements of $\mathfrak{S}_j$ are the search phases known by the vendor which at least contain all ordered goods of the order phase $\mathcal{O}_j$. In absence of any other data that makes $\mathcal{S}_i \in \mathfrak{S}_j$ more likely to be the corresponding search phase of $\mathcal{O}_j$, the vendor's chance for choosing the correct search phase is $1/|\mathfrak{S}_j|$, where $|\cdot|$ denotes set cardinality. Hence, if the number of candidates for the search phase is small then it becomes more likely that the vendor finds the correct link $\mathcal{L}(\mathcal{S}_j, \mathcal{O}_j)$ allowing him to expand the user's profile. $\mathfrak{S}_j$ may be small, e.g., if the order $\mathcal{O}_j$ contains some good g_r which is rarely bought or if it contains many goods g_i because the more goods are purchased in the same transaction, the bigger the chance that a particular set of goods is unique or at least rarely chosen. In contrast, $\mathfrak{S}_j$ may be large if $\mathcal{O}_j$ only contains a commonly chosen single good g_c or a common set of goods $g_{c_1}, g_{c_2}, \ldots, g_{c_l}$. Based on this observations, our goal must be

(a) to have a large set of candidates $\mathfrak{S}_j$ or

(b) to have search phases that do not contain any extra data in the first place.

4.3.1 Increasing the Size of the Anonymity Set

The size of $\mathfrak{S}_j$ depends on a user's exact choice of ordered goods and also on the number of other users searching the vendor's catalogue of goods. So the question is whether it is possible for a user to increase $|\mathfrak{S}_j|$. For the user, one option would be to engage in a new fake transaction in order to do additional, but related, searches by herself. In such a fake transaction, she would search the goods she really wants to buy and some other random goods. This, however, would be cumbersome and put a lot of burden on the user. Another, less burdensome option would be to simply wait until $|\mathfrak{S}_j|$ has grown to a suitable size, i.e., the *customer delays her order*. The problem is, the customer will usually not be able to tell the actual size of $\mathfrak{S}_j$ and hence, will not know when the ideal moment has arrived to send the order. It is clear, however, that the longer she waits, the better the chances that $|\mathfrak{S}_j|$ is large enough. In other words, the more time elapses the more difficult it becomes to use time as a means to correlate a customer's order with her search phase, assuming that in between these two events other customers also searched the goods from the customer's order and all other link data between search and order phase had been eliminated. Before we return to the question of how long to delay an order in order to allow $\mathfrak{S}_j$ to become large, we first discuss the factors that influence the size and growth of $\mathfrak{S}_j$.

For this, let $n := |\mathfrak{G}|$ be the number of different goods the vendor has to offer and let k be the number of ordered goods from a user U's order phase $\mathcal{O}_j$, i.e., $k := |\mathit{goods}(\mathcal{O}_j)|$. The number of possible subsets of $\mathfrak{G}$ is 2^n. Note that we can safely ignore that two or more instances of the same good are finally purchased, since this is not disclosed during the search.[5] Some of the subsets of $\mathfrak{G}$ contain each good from $\mathcal{O}_j$ and hence, are candidates for $\mathcal{S}_j$, which is the actual search phase corresponding to the user's order $\mathcal{O}_j$. We call these candidates *favourable sets*. The number of candidates, $c := |\mathfrak{S}_j|$, can be calculated as follows. Obviously, there is only one set G which has k elements and contains $\mathit{goods}(\mathcal{O}_j)$. Now let a set of goods $G \supseteq \mathit{goods}(\mathcal{S}_j)$ have size $k+1$. Then there is one 'free cell' in G for which $n-k$ goods are possible. Hence, the number of these sets is $n-k = \binom{n-k}{1}$. If $|G| = k+2$ then we have two cells for which there are $\binom{n-k}{2}$ potential choices. If $|G| = k+3$ then we have $\binom{n-k}{3}$ choices and so forth. The total number c of candidate sets from the power set $2^{\mathfrak{G}}$ which contain $\mathit{goods}(\mathcal{O}_j)$ is therefore given by Equation (4.1).

$$c := \sum_{i=0}^{n-k} \binom{n-k}{i} \tag{4.1}$$

Furthermore, since c is the sum of the $(n-k)$-th row of Pascal's triangle, we can simplify c's computation by using the well-known fact from Equation (4.2).[6]

$$\sum_{i=0}^{n-k} \binom{n-k}{i} = 2^{n-k} \tag{4.2}$$

As we have seen in previous chapters, the size of the anonymity group plays an important role for an individual person's chance to stay anonymous. Now, assume that a user U, who has searched V's shop, deems α to be the minimum size of the anonymity group with respect to her search phase $\mathcal{S}_j$. In other words, U expects $|\mathfrak{S}_j| \geq \alpha$. This leads us back to the initial question: When will $|\mathfrak{S}_j|$ be large enough? Unfortunately, there is no universal answer to this question, as the answer depends on the choices made by the customers of a specific shop. This means, we need the probability distribution of the goods purchased by the customers of some shop (or at least an estimate of it), in order to estimate how long it takes until α has become large enough.

Since we do not have such a probability distribution readily available, in the following, we will operate with an *idealised model* in order to illustrate the idea of

[5] In fact, we silently neglected this possibility already by defining orders as sets instead of multi-sets.

[6] This fact can be easily derived from the Binomial Theorem, which says that $(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^{n-i} y^i$, if we set $x = 1$ and $y = 1$.

delaying an order to increase the size of $\mathfrak{S}_j$. If we can assume that the customers' choices of goods are *uniformly distributed*, i.e., each subset of goods $G \in 2^{\mathfrak{G}}$ is equally likely to be picked by any customer, then we can make the following observations. The probability of some customer picking a favourable set with respect to U's order $\mathcal{O}_j$ is $p := \frac{c}{2^n}$ and the probability of the complementary event is given by $q := 1-p$. An interesting observation from Equations (4.1) and (4.2) is that the fraction p of favourable sets only depends on k and is *independent of* n under a uniform distribution, as

$$p = \frac{c}{2^n} = \frac{2^{n-k}}{2^n} = 2^{-k} \ . \tag{4.3}$$

Now, given m users *independently* searching V's goods, the probability that the anonymity set reaches size α after m trials is given by

$$Pr_m(|\mathfrak{S}_j| = \alpha) = \binom{m}{\alpha} p^\alpha q^{m-\alpha}$$

and for its Poisson approximation we have

$$Pr'_m(|\mathfrak{S}_j| = \alpha) = e^{-\lambda}\frac{\lambda^\alpha}{\alpha!} \ ,$$

where $\lambda = mp$ [Fel67]. Using the Poisson approximation, the probability for $|\mathfrak{S}_j| \geq \alpha$ in m trials is

$$Pr'(|\mathfrak{S}_j| \geq \alpha) = e^{-\lambda}\sum_{v=\alpha}^{\infty}\frac{\lambda^v}{v!} = 1 - e^{-\lambda}\sum_{v=0}^{\alpha-1}\frac{\lambda^v}{v!} \ .^7 \tag{4.4}$$

Equation 4.4 gives us the probability for the event that after m search transactions with respect to the vendor's catalogue the number of favourable sets of searches $\mathfrak{S}_j$, i.e., those which include the goods ordered by the customer U, has grown at least to the size α which is U's desired size for the anonymity group that will hide the relation between her search phase and her order phase.

For the following examples, we use the model above to finally answer a user U's question how long to delay her order to guarantee that the number of favourable sets, $|\mathfrak{S}_U|$, is large. In other words, we will estimate the time τ to wait until the anonymity group $\mathfrak{S}_U$ with respect to the customer's search phase $\mathcal{S}_U$ will have grown to size α with a high probability, e.g., $Pr'(|\mathfrak{S}_U| \geq \alpha) \geq 0.99$. In this case, submitting an order $\mathcal{O}_U$ after τ will leave the vendor a chance of $1/\alpha$ to link $\mathcal{O}_U$ with $\mathcal{S}_U$.

[7] Equation 4.4 can be derived by using the series representation of the exponential function $e(x) = e^x$, i.e., $e^x = \sum_{v=0}^{\infty}\frac{x^v}{v!}$, which yields $e^{-\lambda}\sum_{v=\alpha}^{\infty}\frac{\lambda^v}{v!} = e^{-\lambda}(e^\lambda - \sum_{v=0}^{\alpha-1}\frac{\lambda^v}{v!})$.

ordered products (k)	α	$Pr'(\lvert\mathfrak{S}_U\rvert \geq \alpha)$	trials (m)	waiting time
2	10	0.999	91	$\approx$ 3 hours
3	10	0.699	91	$\approx$ 3 hours
3	10	0.999	182	$\approx$ 6 hours
3	20	0.012	91	$\approx$ 3 hours
2	20	0.746	91	$\approx$ 3 hours
2	20	0.999	147	$\approx$ 5 hours
3	20	0.383	147	$\approx$ 5 hours
3	20	0.746	182	$\approx$ 6 hours
3	20	0.999	294	$\approx$ 10 hours

Table 4.1: *Waiting times for uniformly distributed sets of goods $\mathfrak{G}$*

Examples. Let V's catalogue consist of, say, $n = 5$ products (a concrete n is merely given for illustration, since according to Equation (4.3) it is irrelevant) and let U's order be comprised of $k = 2$ goods. Using Equation (4.1)/(4.2), we get $c = 8$ favourable sets and thus, assuming uniform distribution over the sets of searched products, the probability for a subsequent customer to pick one element from the set of favourable sets is $p = \frac{c}{2^n} = \frac{8}{32} = \frac{1}{4} = \frac{1}{2^k}$. Now, if U wants to have probability at least 0.999 that her anonymity set consists of, say, at least $\alpha = 10$ users, i.e., $Pr'(\lvert\mathfrak{S}_U\rvert \geq 10) \geq 0.999$, she would have to wait until approximately 91 users searched V's catalogue (see Table 4.1). Would she have bought $k = 3$ products, the probability for $\lvert\mathfrak{S}_U\rvert \geq 10$ after 91 users would merely be 0.699 — to obtain $Pr'(\lvert\mathfrak{S}_U\rvert \geq 10) \geq 0.999$ she would have to wait for twice as many customers as before, i.e., 182 users in total, which is due to the halved rate of success for $k = 3$ as compared to $k = 2$, i.e., $p = \frac{1}{8}$ for $k = 3$. Further assuming that a search phase is completed every 2 minutes, the user U would have to wait for about 3 hours if she selected 2 products and a bit more than 6 hours had she chosen 3 products. Table 4.1 gives some more examples. □

It must be stressed, however, that the results above are obtained in an idealised model and that, in general, customers will not be able to actually compute waiting times for a lack of statistical input data.

4.3.2 Preventing Extra Data

Although, several approaches are possible to prevent extra data, some of them will be more practical than others in a given scenario. For instance, if the number of goods $\lvert\mathfrak{G}\rvert$ offered by a vendor is small then it may be possible to retrieve $\mathfrak{G}$, privately

select the goods, and send the result to the vendor, e.g., by doing a local search in a downloaded catalogue and emailing the order list to the vendor afterwards. In this case, we have $\mathfrak{S} = \emptyset$ and hence no extra data. Consequently, the vendor's probability for obtaining extra data will be 0. However, if $\mathfrak{G}$ contains many elements then downloading it may be impractical. Fortunately, to prevent the collection of extra data, it is neither necessary to download $\mathfrak{G}$ nor to have an empty set of search phases.

In practice, it is sufficient that the vendor's set of collected search phases $\mathfrak{S}$ contains only search phases which are implied by the executed order phases. That is, given any order phase $\mathcal{O}_j$, if its corresponding search phase $\mathcal{S}_j$ would include only elements from the power set $2^{goods(\mathcal{O}_j)}$ then the vendor would not be able to extract any information from $\mathcal{S}_j$ that he could not have computed from $\mathcal{O}_j$ alone. In other words, if $\mathcal{S}_j$ contains only combinations of search request for goods which are contained in $\mathcal{O}_j$, the vendor gains no additional knowledge from the collected search phase.

However, often a user U's search phase $\mathcal{S}_U$ will not be as straight as this. Instead, it is likely to contain extra data, e.g., because U is indifferent what good to buy or because she might be tempted by the vendor to look for other goods she had not thought of before. Still, if we can make U's search phase *look* like she searched only for the goods she finally ordered, things would be good enough.

The problem here is that U's search requests $S_1, S_2, \ldots, S_m$ for the goods $g_1, g_2, \ldots, g_m$, respectively, are grouped in $\mathcal{S}_U$ by way of an ID. However, if we can *eliminate the link data* for any two search requests $S_j, S_k \subseteq \mathcal{S}_U$ then the search phase $\mathcal{S}_U$, comprised of many search requests, will be broken up into smaller phases $\mathcal{S}_{U,i}$ with less search requests. And even more, these search phases will be *unrelated* from the vendor's point of view. In the most favourable case with respect to privacy, the vendor will only see search phases consisting of no more than a single search request. In this case, he will only be able to find trivial links from search phases to order phases. That is, although the user's real search phase $\mathcal{S}_U$ is comprised of $m := |goods(\mathcal{S}_U)|$ search requests, the vendor will find at most $i = 1, \ldots, |goods(\mathcal{O}_U)|$ links $\mathcal{L}(\mathcal{S}_{U,i}, \mathcal{O}_U)$ for a given order phase $\mathcal{O}_U$. Moreover, since $goods(\mathcal{S}_{U,i}) \in 2^{goods(\mathcal{O}_U)}$, these links will not provide the vendor with any extra data. Therefore, the vendor's chance of learning additional information, which would expand the user's profile, will again be zero. □

In the next section, we will introduce a conceptual architecture which unites the two approaches from above. We will also present two possible instances of this architecture, each of which is an instance of one of these approaches.

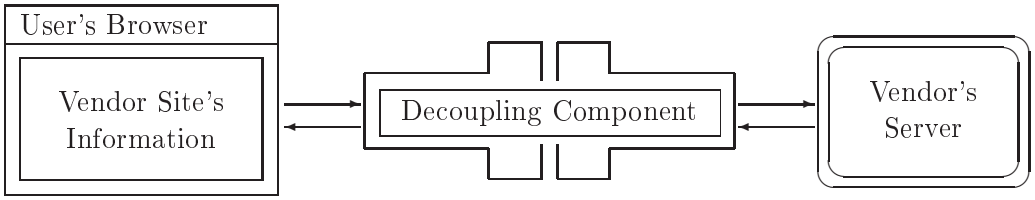


Figure 4.1: *Conceptual model of the Decoupling Component*

4.4 Conceptual Solution

We have seen that at least two levels of data exist which allow tracking of users — protocol and content data (see Section 4.2). The principal idea of removing the possibility of protocol tracking is to interrupt subsequent actions of the customer which are recordable by a vendor and thereby to eliminate existing links. For this, we introduce a decoupling component (DC) which is located between the customer and the vendor (see Figure 4.1).

In Section 4.3, we have discussed two methods for decoupling search and order phases — increasing the anonymity set with respect to a given order and preventing non-trivial transaction data in the first place. In the following, we are going to sketch two possible implementations for either method.

4.4.1 Preventing Extra Data

Using today's Web technology, it is possible to present each user with a customised view of Web sites. This, however, is often driven by the vendor, i.e., the user tells him what she likes and the vendor presents her with information deemed interesting for her. In this case, the user's view is more or less what the vendor wants the customer to see. On the other hand, Web technology would also permit the user to have her own view of the vendor's offered products. Using Web technologies, such as UDDI [BCvRE03] and WSDL [CCMW01], it would be possible to directly access information provided by the vendor and use a literally personalised representation of the information, instead of accessing a representation chosen by the vendor. If the vendor provides its products through Web services, the customer can execute, say, its searches by calling a specific service "search" and the results would be displayed according to the user's preferences. Examples of such services are Google's search API[8] or Amazon's Web services[9] which both are accessible via SOAP. In

[8] `http://code.google.com/apis/soapsearch`
[9] `http://www.amazon.com/webservices`

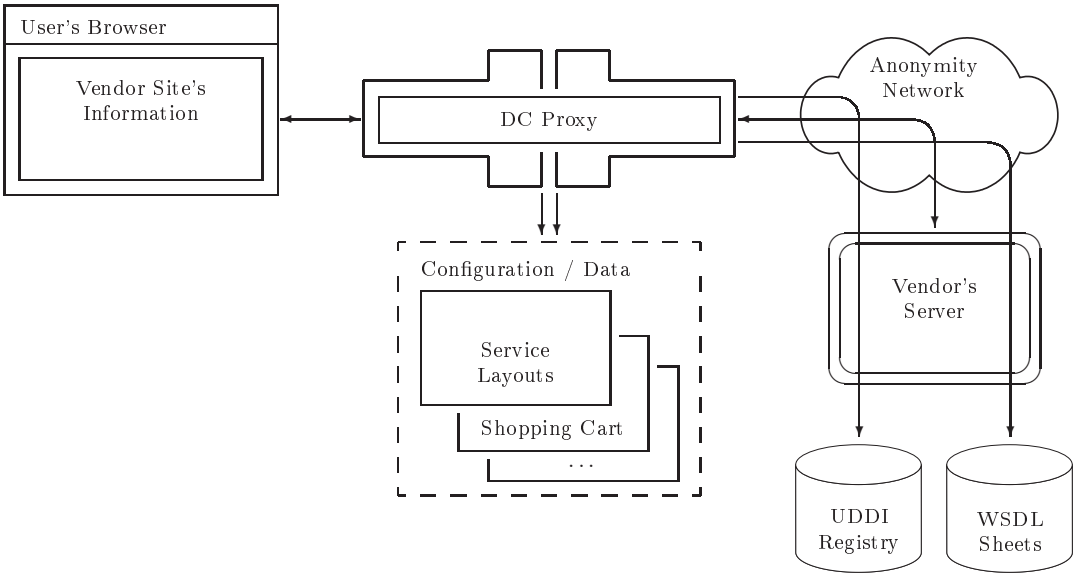


Figure 4.2: *Example of a DC instance*

such a case, session management in HTTP can be dropped, because it would be unnecessary, and consequently linking options given by the HTTP protocol would be eliminated. In addition, changing the IP address of the customer's computer for each service call would eliminate tracking by IP. This can be easily realised by sending each service request through an anonymity network.

The functionality described above could be combined in a decoupling component (DC). The DC would technically be a proxy which sits between the user's browser and the vendor's server (see Figure 4.2). This architecture guarantees that the DC sees all traffic exchanged between the customer C and the vendor V. In addition, the DC would provide client functionality for connecting to an anonymity network, such as TOR[10] or JAP[11], which allow to send and receive data anonymously.

Search. Now, if the user types the vendor's URL into her browser, the DC checks via UDDI whether V offers Web services, and if so, accesses their WSDL sheets, i.e., downloads the service descriptions. If V does not provide Web services to his customers, the DC simply connects to the vendor's Web server and the user is

[10] `http://tor.eff.org`
[11] `http://anon.inf.tu-dresden.de`

left with the 'normal', potentially more privacy-intrusive Web shop. Assuming the vendor provides Web services to his customers, the DC shows the list of services and other information according to the user's display preferences. In this case, the customer would use the DC's interface to access the vendor's Web service "search" or some other Web service to browse through the vendor's catalogue. This means that the customer could potentially have the same (graphical) user interface for all such vendors, which, as a side effect, may simplify the navigation. □

Order. The ordering procedure is likewise executed through Web services. In this case, the DC offers a virtual shopping cart to allow the customer to select goods for purchase, beforehand. In addition, it is conceivable that the DC stores commonly asked for customer information, such as name, address, and financial information, in a local database to relief the customer from entering her data for every new vendor. Should any of these information be required by the vendor, the DC simply notifies the user which information is asked for by the vendor and asks for her confirmation, before it finally sends the order data to the vendor's ordering service. □

Privacy. By sending every service request through an anonymity network, the user's IP address can be made different for each request. Hence, given two search requests, S_i and S_j, the vendor's only source of information for deciding whether $\mathcal{L}(S_i, S_j)$ holds is the content information provided in the requests. The requests, however, are just service calls which are comprised of information that potentially could have been sent by any customer because it only contains parameters for the respective Web service. Even if, say, some vendor-supplied session token t would be a required parameter for accessing some service, the DC could simply ask the vendor's server for a new token t' before accessing the service again. Therefore, the vendor would be prevented from relating requests of the same user by content information. In the end, the vendor would see access to his ordering service which would provide him with the customer's selected goods. Although the vendor might at best be able to find a single search request for each ordered good, he will not be able to relate it to the customer's other search requests for goods which did not became part of the order. This holds since every request is self-contained or in other words 'isolated' from the vendor's point of view. In this case 'isolated' means that a different sender identification, i.e., IP address in this case, is used each time and the content information provided in each service request is independent of any information provided in any other request. Therefore the vendor will be unable to correlate a specific user's actions with respect to information provided in all observed requests, except for trivial links from the set $2^{goods(\mathcal{O}_j)}$, where $\mathcal{O}_j$ is the user's order (see Section 4.3). □

By using a DC instance, as the one described above, the way the search phase is perceived by a user is totally different from the search phase seen by a vendor. While the user still perceives its searching activities as a 'semantic phase', since the act of searching is a sequence of related actions, the vendor has no clue of this semantics because he is unable to recognise the user during her search. The vendor can only technically observe calls to his Web services which are —from his perspective— independent of each other. Because of this, content-related tracking mechanisms will be meaningless, since they disclose almost nothing beyond what is conveyed in a single call to a Web service, as the call is an isolated event from the vendor's perspective. An information that may clearly be associated with a specific call will be the time when the user accessed the service. However, this does not disclose extra data, in the sense that no additional personal information is disclosed, such as, other interests.

The price to pay for this additional privacy is that the user's 'browsing experience' might be changed significantly. Since the DC practically builds the Web pages for the user, the visual results of this pages will be highly dependent on the DC's layout templates, its implemented logic, and, of course, on the Web services offered by the vendor. In addition, providing the user with navigational hints may become complex. On the other hand, users who are looking for specific items may not be affected by this loss. Essentially, they will use a search interface constructed by the DC, instead of one supplied by the vendor. The DC's interface will, in essence do the same as the vendor's, i.e., it will use the customer's search requests as input to the vendor's (search) Web service.

4.4.2 Increasing the Size of the Anonymity Set

In order to prevent linking of search and order phases, we proposed another approach that aims to increase the number of candidates which could have instantiated a given search phase $\mathcal{S}_i$. The basic idea exposed in Section 4.3 was to wait until a sufficiently large number of candidates exists. The main problem of this approach is that it is hard to determine the time to wait. For this, one can rely on heuristics or statistical information, as we did before. However, in theory, it would be possible to exactly time the release of an order, given that sufficiently many customers work together and anonymously share their access statistics for vendor sites. We will briefly give an idea of such an approach in Section 4.7.4. Before that, we introduce a concrete framework based on mobile agents, which can be seen as another possible instantiation of a decoupling component. This DC's aim is to establish a framework which makes it possible to send out arbitrarily delayed orders for the purpose of introducing uncertainty to the vendor when it comes to relating order data to a specific search phase.

The following sections on mobile agents as a means for privacy protection have been presented and elaborated in various forms in [EKS02d, EKS02b, EKS02a, EKS02c, EKS03].

4.5 Mobile Agents

In the following sections, we will look into mobile agents to introduce another possibility for implementing a decoupling component (DC). This particular DC instance will rely on large anonymity sets in order to allow customers to hide the relationship between their search and order phases, as discussed in Section 4.3.1.

The mobile agent DC should be considered only as a possibility for implementing a DC that decouples search and order phases. Although we have devoted a fair amount of space to mobile agents, the reader should not cling too hard to the idea of using mobile agents for implementing such a DC, as the idea of using anonymity sets for decoupling is not bound to a particular technology. Other implementations are certainly possible and may in fact be needed, if the technological environment is different or changes over time.

As we do not expect all readers to be familiar with the concept of mobile agents, we will briefly introduce some definitions needed in subsequent sections to develop the agent-based DC instance.

A mobile agent is an autonomous program which follows a given route by migrating through a network of hosts. At each host, it carries out certain tasks on behalf of its owner. After the tasks have been accomplished, the agent returns to its base station and delivers the results collected during its trip to its owner. One of the advantages of using mobile agents technology is that transaction costs for the agent owner are considerably reduced because after leaving its owner the agent migrates from one host to the next autonomously. During this period, the owner is not required to maintain his online connection to the network or in some other way stay in touch with the agent. In the past years, a lot of work has been done in the area of mobile agent systems. A variety of mobile agent systems is available today, e.g., Aglets [KLO98, LO98], Agent TCL [GCKR97]. However, we will not focus on a specific mobile agent system. Instead, we will consider mobile agents in a more abstract way. This means that we will only consider components of mobile agents which are important for the solution presented here. In our level of abstraction, a mobile agent a consists of the following components:

$$a = (bc, r, d, \delta).$$

The component bc denotes the *binary code* of the agent to be executed and r describes the mobile agent's *route* as an $(n+1)$-tuple (with $n \geq 1$) consisting of the

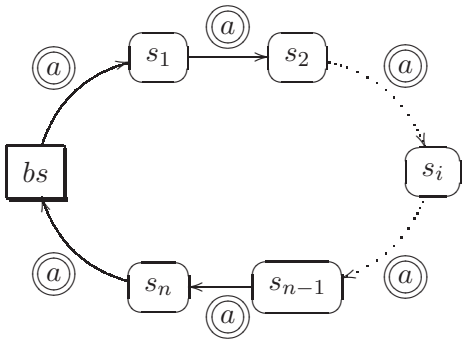


Figure 4.3: *An agent a's round trip*

host addresses $ad(s_i)$ of shops s_i that have to be visited on the agent's trip:

$$r = (ad(s_1), \ldots, ad(s_n), ad(bs)) \,.$$

The route is given by the agent's owner. The agent starts its trip at a base station bs where it returns to when it has visited the stations contained in the route (see Figure 4.3). Since the first migration step is $bs \rightarrow s_1$ the first route entry is given by $ad(s_1)$. The component d denotes the *data* given by the agent's owner. This data will be used as input for the computations at the hosts $s_1, \ldots, s_n$. Thus, we can think of it as $d = (d_1, \ldots, d_n)$ where d_i refers to the input data of s_i, where $1 \leq i \leq n$. The output data obtained from the computations are contained in δ. Similarly, we have $\delta = (\delta_1, \ldots, \delta_n)$.

If we set $bc = \emptyset$, then the 'agent' would not be able to carry out computations on its own. And still, the model, though less flexible, would make sense. In this case, the 'agent' merely provides the visited stations with data and the stations will compute the results on their own. Such a scenario is conceivable, if we think of such an agent as a kind of digital chain letter, which is forwarded by each station to the next one *en route*.

4.6 Adaption of Agent Components

In the following, we will adapt the previously introduced components of a mobile agent according to the requirements of our solution. Therefore, the main goal of protection we have in mind is *privacy*. However, we also consider *data origin authentication* (including *data integrity* of the mobile agent) and *non-repudiation*.

For this, we introduce some definitions. Let $E_{e_i}(d_i)$ denote a ciphertext obtained by using s_i's public key e_i of an asymmetric encryption algorithm E, e.g., *RSA*

[RSA78], on the data d_i. Furthermore, let $Sig_x(y_1, \ldots y_\nu)$ denote a digital signature of party x on some contents $y_1, \ldots y_\nu$. This allows us to introduce an element

$$\tilde{d} = (E_{e_1}(d_1), .., E_{e_n}(d_n), Sig_C(E_{e_1}(d_1), .., E_{e_n}(d_n), bc))$$

consisting of encrypted input data for $s_1, \ldots, s_n$ and a signature of the agent's owner, i.e., the customer C. Furthermore, we protect the agent's route by using layered encryption, similar to Westhoff *et al.* [WSUK00]. In each layer of the onion structure below, we have a host address $ad(s_i)$ and a signature of the agent's owner C on $ad(s_i)$ combined with other agent components.

$$\begin{aligned}
\tilde{r} \;=\; & ((ad(s_1), Sig_C(ad(s_1), bc, \tilde{d}), \\
& E_{e_1}(ad(s_2), Sig_C(ad(s_2), bc, \tilde{d}), \\
& \quad E_{e_2}(ad(s_3), Sig_C(ad(s_3), bc, \tilde{d}), \\
& \qquad \ddots \\
& \qquad\quad E_{e_{n-1}}(ad(s_n), Sig_C(ad(s_n), bc, \tilde{d}))\ldots)), \\
& ad(bs))
\end{aligned} \tag{4.5}$$

The encrypted route is processed as follows. The base station learns from the first entry of $\tilde{r}$ where to dispatch the agent. Before the agent is sent to s_1, bs deletes its own entry from $\tilde{r}$. When s_1 receives the agent with the new $\tilde{r}$, it decrypts the ciphertext from $\tilde{r}$ which is destined for it and obtains its successor's address $ad(s_2)$, a signature, and a 'new' ciphertext. Before sending the agent to s_2, s_1 deletes its own address and the signature from $\tilde{r}$. This procedure will be repeated until the agent arrives at s_{n-1}. Here the last decryption is necessary, i.e., s_{n-1} gets the last address $ad(s_n)$ and signature contained in the onion structure. After these parameters have been removed from $\tilde{r}$, the agent is sent to bs as specified by the last entry $ad(bs)$. The idea of the route protection is that a visited host does not learn which hosts the agent had visited before and which it is still going to visit, except for the host's respective predecessor and successor. If the latter is still to much information for the customer's taste, she may introduce dummy hosts in the route that just forward the agent, e.g., the base station.

The signatures contained in $\tilde{r}$ and $\tilde{d}$ are included to ensure data integrity and to allow easy detection of unwanted modifications. The signature also guarantees that components from $\tilde{r}$ cannot be replaced by components from earlier data $\tilde{r}'$ without detection.

After having produced the computation results δ_i at host s_i, they will be signed by s_i. Thus, we define $\tilde{\delta}_i = E_C(\delta_i, Sig_{s_i}(\delta_i, bc, \tilde{d}))$. At the end of the trip, the agent contains all computation results, i.e., $\tilde{\delta} = (\tilde{\delta_1}, \ldots, \tilde{\delta_n})$. Initially, portions $\tilde{\delta_i}$ are empty. For the upcoming sections, we assume that an agent $\tilde{a}$ consists of the

following components:

$$\tilde{a} = (bc, \tilde{r}, \tilde{d}, \tilde{\delta}).$$

Now that we are done with the building blocks, we are going to present how agents can be used to reduce profile data.

4.7 Achieving Privacy via Agents

In the following, we show how to prevent a vendor from linking information gathered in a customer's search phase to her real identity, which we assume is disclosed in the order phase. Furthermore, we give some more requirements that have to be fulfilled in order to avoid linking via product IDs.

In the following, we assume that the base station is maintained by a specialised mobile agent base provider. However, this provider does not play the role of a trusted third party in the usual sense. In fact, the customer's trust in the provider is quite minimal with respect to privacy, as the provider does not get any information about the search phase and therefore, he cannot exploit this information for himself nor pass it to others. The provider is trusted, however, not to give the customer's IP address to the vendor, not to delete agents after having received them, and not to release the agents before they are scheduled for departure. More details on the base station will be given in Section 4.8.2.

In Subsection 4.7.1, we show how to deliver an order with an agent, where the shopping tour contains only one vendor to be visited. In Subsections 4.7.2 and 4.7.3, we discuss how the tracking mechanisms described in Subsections 4.2.2 and 4.2.3, respectively, can be ruled out using the mobile agent approach. Afterwards, we point out how the linking probability for the vendor can be decreased. Routes containing several vendor addresses are finally discussed in 4.7.5.

4.7.1 Order Delivery via Agents

For simplicity, we assume that a customer C decides to buy at just one vendor, say s_1. While searching through s_1's catalogue, C has a look at various products, and finally decides to buy a subset $p_1, \ldots, p_k$ of these products. During this time, s_1 can track C's activities in a transaction log T_i. However, s_1 is unable to map them to C's identity *per se*. C forms the data d_1 by putting together the order information, including the identifiers of the products and her name and address. Furthermore, C should have the possibility to arrange the product identifiers in an arbitrary sequence, possibly one that does not match the order in which she selected the products herself. This is necessary since the sequence of product identifiers in the order d_1 must be protected against a potential correlation with the click sequence

contained in T_i. After C has created d_1, she creates $\tilde{d} = E_{e_1}(d_1, Sig_C(E_{e_1}(d_1), bc))$ and $\tilde{r} = (ad(s_1), ad(bs))$, and finally $\tilde{a} = (bc, \tilde{r}, \tilde{d}, \tilde{\delta})$, with $\tilde{\delta} = \emptyset$.

In the next step, C will transfer $\tilde{a}$ to bs and instruct bs to dispatch $\tilde{a}$. Now, $\tilde{a}$ can migrate to s_1 and deliver its data d_1. Since d_1 is asymmetrically encrypted, it can only be opened by s_1, i.e., bs does not learn which products were ordered by C. Furthermore, by verifying the signature, s_1 can check if the order was undeniably created by C and if it had been modified. When the data contained in the agent is valid, s_1 creates the output δ_1 and the encrypted result $\tilde{\delta}_1 = E_C(\delta_1, Sig_{s_1}(\delta_1, bc, \tilde{d}))$ which is given to the agent. This output could be, e.g., s_1's notification that he received the order or a confirmation that the order had been carried out immediately. To ensure integrity, all outputs should be signed by the vendor as well. After s_1 is done, the agent is sent to bs according to the last route entry $ad(bs)$. There, $\tilde{a}$ waits for C until she connects again to bs in order to receive the agent.

4.7.2 Tracking by Protocol

Using our mobile agent approach, the vendor cannot use IP addresses, cookies, or dynamic URLs to link the received order to search phase activities because he is no more directly receiving the order from the customer C. In order to get a deeper insight into C's interests, s_1 can try to link available transaction data $T_1, T_2, \ldots$ with the data obtained from the agent to find out C's real transaction T_C. But this means that s_1 has to carry out a random experiment, assuming that no other hidden information exists that makes linking easier. Since C has ordered $p_1, \ldots, p_k$, the vendor searches his database for profiles which include $p_1, \ldots, p_k$ and which have been recorded in a 'relevant time interval'. The relevant time interval needs to be chosen by s_1. Although, any time interval can be chosen for the search space, it can be assumed that transactions of a certain age can be ignored, e.g., recently submitted orders will most likely not belong to a transaction that has been recorded one year ago. If we assume that s_1 finds $\alpha \geq 1$ transactions $T_1, \ldots, T_\alpha$, each containing at least $p_1, \ldots, p_k$, then for s_1 the probability of correctly linking any stored transaction T_i to C's search phase $\mathcal{S}_j$ is given by $Pr(\mathcal{L}(T_i, \mathcal{S}_j)) = \frac{1}{\alpha}$. Hence, the probability depends on the number α of transaction candidates, which is what we called the size of the anonymity group in earlier chapters. Obviously, α decreases monotonically when k increases, i.e., including a higher number of products in the order may increase the linking probability.

4.7.3 Tracking by Content

Since content tracking is not easily detectable *per se*, rules and procedures need to be introduced to rule out content tracking, or at least make it detectable by a customer.

Product identifiers. In accordance with our "honest but curious vendor" assumption, a vendor could offer an agent-based privacy protecting service and may still try to link the search and order activities via leaking product IDs (*ProdID*s). In this case, the linking probability for the vendor would be $Pr(T_C = T_i) = 1$. In order to prevent such attacks, *ProdID*s must be verifiable by the customer such that hidden channels become obvious. This, for instance, could be achieved by some natural language encoding of *ProdID*s. In such a scheme, the set of potential candidates for *ProdID*s is quite small, e.g., for a music CD the *ProdID* could be *(artist's name, title)* instead of some ID *B0000262WI* and therefore, a customer will be able to detect with high probability if there is some hidden information. □

Prices. Another possibility to increase the linking probability could be achieved via variable pricing strategies, similar to price discrimination. However, in our scenario, the vendor's primary intention is to use different prices in order to identify individual customers and not to maximise his profit — remember that we are strictly talking about computer science and not economics. Thus, an unfair vendor could offer the same product at slightly different prices. When he later receives an order which contains the price presented in the offer, he can easily exclude all those transactions from the set of candidates in which he offered the specific product for a different price.

If the customer does not send the prices, the vendor might sell her the ordered goods for any price, which he had once charged for the given products. In order to prevent this, we propose to let the vendor give a temporary price guarantee for any product he sells, i.e., he commits himself to charge a fixed price for a given product for some period of time. This could be realised, e.g., by creating and publishing vendor-signed documents, each containing a product, its price, and the validity period, similar to price offers found in advertisements in the real world. These guarantees can be downloaded by the customer and kept for later reference.

If a vendor serves customers with distinct price guarantees in order to track them, then he does not know the exact prices at which he offered his products to a specific customer. Thus, if he is trying to identify a customer by the offered prices, the vendor can only guess the price a customer expects according to her downloaded guarantee. Should the vendor make a wrong guess to the customers disadvantage, she could present her price guarantee, proving that the vendor tried to deceive her. In this case, since the price guarantee is disclosed, the vendor would be able to link a transaction to the customer. However, disputes are unattractive for the vendor because if they occur frequently the vendor's reputation will degrade. In addition, if the vendor makes a wrong guess to the customer's advantage, i.e., billing her for a lower price than she expected, then she will surely accept and the vendor lowers his own profit and in addition, he would create a link to the wrong transaction. □

Product ordering. In order to counter the effects of correlating the click sequence from the search phase with the product ordering in the order, the customer should have the option to arrange the *ProdID*s in an arbitrary sequence. This sequence can be randomly created or obtained by sorting the *ProdID*s, e.g., in lexicographic order. This is sufficient as long as there are other transactions stored in s_1's database produced by other customers who also viewed at least $p_1, \ldots, p_k$ in one session and therefore help to decrease the vendor's linking probability.

4.7.4 Decreasing the Probability for Linking

In the following, we will show how a customer can decrease the linking probability. Since this probability depends on the number of candidates α, we would be better off, if the vendor's site is visited by many users. Therefore, we would be fine if we could somehow motivate a large group of users to access the vendor's catalogue because this will most likely increase α but this approach does not seem to be very realistic. However, from a broader perspective, the idea of motivating many users to increase α is just a means to shorten the time until α is sufficiently large. Therefore, we could likewise wait long enough until α is sufficiently large.

In order to achieve anonymity sets of size at least α, we propose to introduce a service at the base station that allows customers to define a delay for the start of the agent's shopping trip. That is, the customer tells the base station to send out the agent, say, in 10 hours. Since this delay is unknown to the vendor, he does not know whether to link the order with profiles stored in the past minutes, hours, or even days. The only thing the vendor knows is that an order is always preceded by a search. This causal interrelationship cannot be undone but it can be hidden in a group of similar events. By randomly extending the time between an order and its corresponding search, a time-based correlation of the two events becomes more difficult. On the one hand, the time interval considered by the vendor should be large, since this increases the probability that the correct profile is included in the set of candidates. On the other hand, the size of the anonymity set α can also be assumed to have increased. Of course, this depends on the access statistics of the vendor's Web site. The dispatch time needs to be specified by the customer. For instance, she may tell the base station not to release the agent before a delay $\Delta\tau$ or not before time *MM:dd:hh:mm*. When price guarantees are used, the delay must not exceed the validity period of the price guarantee.

This solution is quite simple, since it only requires the existence of the base station. But on the other hand, there is no guarantee that the number of profile candidates is large enough. From a theoretical perspective, even such a guarantee can be given, if a community effort is used to measure the access statistics of a shop. This could be realised, for instance, if customers anonymously upload the

transaction data of their search phases to a third party. A customer would then be able to query this party for access statistics of a specific vendor's products. This would enable her to determine the current size of the anonymity set with respect to her order. In [EKS02a] such a party has been described in greater detail.

4.7.5 Routes with Several Shops

So far, we have considered only cases in which a mobile agent only visits one vendor for delivering its order information. Shopping tours with several vendors can be advantageous, e.g., in cases when an order for vendor s_i should only be ordered if products from a previously visited shop s_j could have been ordered. Furthermore, this option improves overall efficiency of the network, since it reduces the number of mobile agents which migrate through the network. Visiting more than one shop introduces no privacy problems, since every order information is encrypted for a specific vendor.

One could argue that with longer routes vendors learn which other hosts have to be visited on the shopping tour which gives some further insight into the customers behaviour or interests. But this threat can be tackled with the route protection scheme shown in (4.5). With this solution, a vendor only learns about his predecessor and successor in the shopping tour. If a customer does not want the vendor to learn his predecessor or successor then she could create a route $\tilde{r}$ with bs as an intermediate hop which separates the vendors from each other. This increases the number of migration steps on the agent's trip but allows the customer to enhance her privacy.

In case of dependent agent computations, e.g., when order delivery at s_i depends on computation results from s_j, the computational results have to be communicated from one vendor to another. If s_i is immediately followed by s_j then the exchange of the required results can be done directly — provided that the customer is willing to reveal the shopping tour stations to the vendors. If the customer decides to hide vendor identities from each other by agent exchange via bs, the results can be encrypted for bs which then needs to decrypt and re-encrypt the data for the vendor that requires it. By applying this concept, a vendor will not learn what C is ordering at other vendors as long as the vendors are not colluding and exchange their trade data.

4.8 Architecture

In the following, we will sketch the architecture including the infrastructure components that are needed for our solution. Figure 4.4 depicts the basic components which are required in our system. The customer uses a simple Web browser and

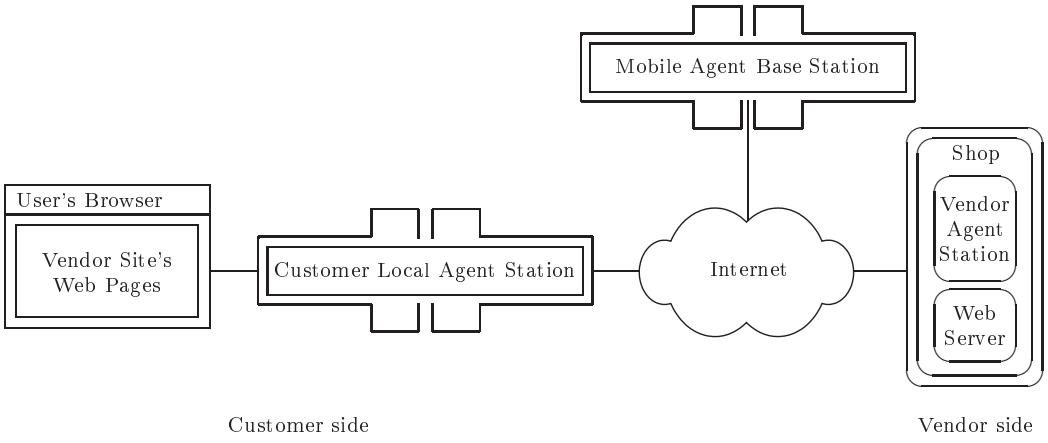


Figure 4.4: *Overview of the Agent-Based DC Instance*

the *Customer Local Agent Station* (CLAS) which is a component of our solution. The vendor runs a Web server and a *Vendor Agent Station* (VAS). In addition, a third party provides a *Mobile Agent Base Station* (MABS) which offers mobile agent services to customers.

4.8.1 Customer Components

The customer is running two applications, a browser and a CLAS. The CLAS component is independent of a specific vendor's VAS and provides the following services and functions:

- a client-side proxy,
- a *Product Selector* (PS) from which products can be put into an agent that becomes the shopping cart,
- a *Mobile Agent Generation* (MAG) component, e.g., an agent workbench, and
- a client for using the services of the MABS.

When the customer browses through the vendor catalogue all requests are first sent to the CLAS (Figure 4.5, message 1) and subsequently forwarded to the vendor system (Figure 4.5, message 2). The corresponding response of the vendor is received by the CLAS (Figure 4.5, message 3). Message 3 consists of data destined for the CLAS and data destined for the browser. After having extracted its own

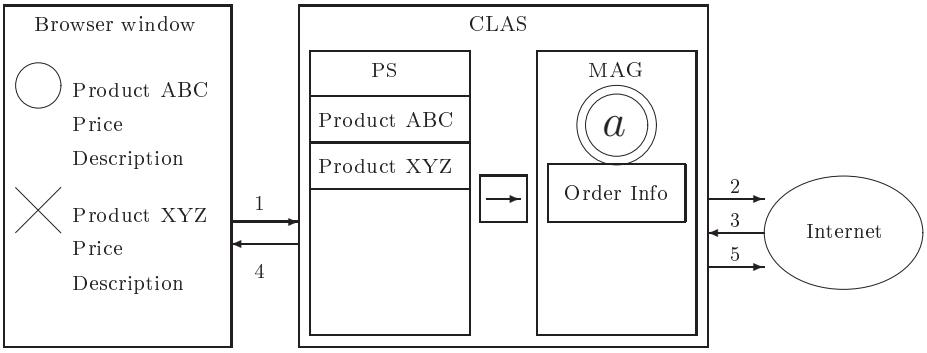


Figure 4.5: *CLAS and browser interaction*

data, the CLAS forwards the remaining message to the browser (Figure 4.5, message 4). The browser's message consists of the usual Web page content, e.g., product information. The CLAS's part of the message consists of the product identifiers of the products currently displayed by the browser. These identifiers will also shown by the PS sub-component. In contrast to common scenarios where the customer puts a product to be bought in a virtual shopping cart by clicking some icon or button in the shop's Web interface, in our scenario the Web interface is merely used for navigation and product information. To prepare some good for the order, the customer transfers the product ID from the PS to the MAG. Using the MAG, the customer composes the agent in the desired manner, i.e., she selects the appropriate agent from the set of available agents. Then, she has to create the input data and the constraints which may exist for the shopping tour, e.g., buy at shop B only if the order at shop A can be fulfilled. Such constraints influence the order of hosts to be visited and thus, have to be considered when the agent route is composed. After having selected an appropriate order of shops, the CLAS re-arranges the sequence of the selected products as described in Subsection 4.7.3 and the MAG automatically creates a route according to Equation (4.5).

After having finished the shopping tour, the customer transfers the agent to the MABS (Figure 4.5, message 5). In addition, she may enter some additional instructions, such as the agent's release delay or the threshold α.

4.8.2 Mobile Agent Base Station

The MABS provides a docking station for an agent, as well as an alarm clock and relaying/sending/receiving facilities. The alarm clock is used to wake up the agent after a pre-determined period of time or at a specific time and date.

When any condition for the agent's release is satisfied, the MABS lets the agent start its trip. When the agent returns to the MABS, it is stored in a personal inbox which is only accessible by the corresponding customer. After having received the agent again, the MABS will inform the customer of the agent's return, i.e., that the agent is ready for pickup. Then, the customer connects to the MABS and downloads the agent in order to review and verify the results of its trip. Therefore, the customer's MAG also must provide the functionality for decrypting data which was encrypted for the customer by the visited vendors.

4.8.3 Vendor Components

On the vendor side, the solution requires a *Vendor Agent Station* (VAS) beside the usual vendor infrastructure. This VAS provides the required functionality that is necessary for the interaction with the customers' agents, i.e., basic agent execution facilities and the required security mechanisms as mentioned before. Furthermore, the vendor has to provide messages comprised of CLAS data and browser data. Whenever a customer requests product information for goods, the vendor's Web server responds with corresponding messages for the CLAS and for the browser.

By using the CLAS for selecting the products to be ordered in the proposed way, we can be sure that no hidden channel to the vendor exists. Although, hidden channels could still be possible when a customer selects a product for purchase directly from the Web interface, i.e., she does not use the functionality provided by our DC instance. Such a hidden channel would increase the linking probability by reducing the set of candidates to those customers who really purchased the corresponding products. In our solution, the set of candidates is larger, since it includes not only the aforementioned group of customers but also the group of customers which *only viewed* the product(s), i.e., the size of the anonymity group is larger.

4.9 Related Work

Classic works in the area of anonymisation techniques, e.g., [RSG96, RR98, SRG97], aimed at anonymising IP addresses, e.g., to prevent the relation of entries in log files. Furthermore, these works wanted to achieve *unobservability* as much as possible, i.e., that a third party observing the whole network cannot determine who is communicating with whom. In other words, such a third party must not be able to link messages sent from a customer to a vendor and likewise answers sent back from the vendor to the customer. In our work, we have not been concerned with such third parties, as we can obtain this kind of unlinkability by employing the said anonymisation techniques. However, searches and orders take place in the higher

application layer and hence, the solutions developed for the lower communication layer can be circumvented by using link data in the layers above.

Privacy protection in electronic payment systems, e.g., [BGH+95, BGK95, CFT98, Cha89, CFN90, OO90], is more close to our ideas. In these systems, the goal is to achieve unlinkability of a customer's payment transactions with respect to a vendor. We want to achieve unlinkability of transactions, too. However, our approach is more general, as it recognises that a transaction may consist of more than just a payment phase and that other phases may disclose (personal) data as well that can be used for linking transactions. Our approach also allows for more fine-grained control with respect to the disclosure of personal data, as we do not only try to prevent links between transactions but also within transactions.

Most of the work that deals with online selling and privacy (e.g., [BD01, BDF01, SSG99]) exclusively focuses on the trade of intangible goods where anonymity networks and electronic payment systems can be used without problems, as no real-world identities and shipping addresses need to be revealed.

Other works dealing with privacy protection with respect to gathering information on customer activities in business processes was presented in [AJJ+00, Jue01, Kob07]. These works focused on privacy protection in customisation, personalisation, and targeted advertising.

The idea to protect the privacy of users in searches had later been picked up by Shen *et al.* [STZ07] and Xu *et al.* [XZCW07] as well. Their works aimed at reducing the amount of personal information that a server receives in personalised search queries. In addition, the client-side personalisation solution proposed in [STZ07] can be seen as an instance of a decoupling component, as introduced in Section 4.4.

In the area of agent-based intermediary infrastructures, further results were presented in [TM00]. This work, however, was focused on the agent-based implementation of typical commercial activities, such as product brokering, negotiation, and matching customers' and vendors' needs and bringing together both of them.

4.10 Conclusion

We have presented an approach to reduce personal information produced in a customer's interaction with a vendor in order to improve the customer's privacy. The approach focuses on decoupling the customer's search phase from subsequent phases by eliminating unnecessary links between them. This way, the information learned by the vendor in the customer's search phase cannot be related to data obtained in later phases, such as the customer's name and address in case of tangible goods that need to be shipped. Hence, even in cases where the vendor can relate transactions

of a customer by her name, our approach helps to reduce the amount of personal information that can be collected in a profile.

We discussed two variants of this approach, one focusing on preventing non-trivial search profiles entirely and the other one allowing search profiles in principal but leaving the vendor in doubt whether a given profile truly belongs to the customer at hand, as links from the search phase to the customer's identity are afflicted with uncertainty. In practice, the degree of uncertainty will depend on several factors, including, but not limited to, the usage statistics of the vendor's site, buying habits of his customers, the customer's choice of goods, the number of goods chosen by her, and also her time frame for executing an order. Finally, we have sketched two architectures, one for each of the two variants. The first one, being based on Web services, can prevent links from search phases to customers' identities entirely, and the other one, making use of an agent infrastructure, only hinders the association of search phases to customers' purchases but may be less obtrusive than the first variant, as the customers' browsing experience does not need to be changed significantly. The idea of decoupling a search phase from subsequent phases of a transaction is however not restricted to the technologies presented in this chapter. Therefore, the two architectures are just possible instances for decoupling components and others are certainly possible.

Chapter 5

Privacy-friendly Loyalty Systems

In this chapter, we present a privacy-friendly loyalty system allowing online vendors to issue loyalty points to their customers. The system protects customers' privacy as it cannot be used to create profiles by linking transactions in which points had been issued or redeemed. To this end, two variants of such a system had been developed, one based on the RSA problem and the other one being based on the Diffie-Hellman problem. For both systems, we introduced anonymous counters which are used to keep track of issued loyalty points.

Most parts of this chapter had been previously published in [EFS04, ES04, EEOS05]. This publications, however, did not include references to the link model of Chapter 3 and were missing detailed security proofs in part.

5.1 Introduction

Naturally, every online vendor's interest lies in attracting new customers and creating a large base of loyal customers. Since loyal customers create regular revenues, the goal of online vendors, as well as real-world vendors, is to turn occasional customers into loyal ones. Thus, in the past, online and real-world vendors have introduced loyalty programs, e.g., frequent flyer programs or online consumer reward systems, in order to retain customers.

Aside from customer retention, another incentive for vendors is to learn more about their customers to exploit this information for purposes such as customer profiling, data mining, or direct marketing. Thus, from the customer's perspective, loyalty programs have two sides. On the one hand, customers value the financial benefits, on the other hand, they may fear an infringement of their privacy. Hence, if privacy concerns outweigh the expected benefits from the loyalty program the vendor's strategy for attracting and retaining customers will fail. Thus, if privacy is a barrier for customers to participate in the program it may be worthwhile for

vendors to reconsider their strategy of collecting personal data. Indeed, according to [HNP99, Kob01], there are many customers that are concerned about their privacy in electronic commerce scenarios. Thus, privacy non-invasive loyalty systems might be of particular interest to vendors.

In this work, we deal with loyalty systems in which customers receive points from vendors for their purchases. Points can be redeemed at the vendor in exchange for a reward, where the value of the reward depends on the number of points. The system had been designed such that it does not permit the vendor to collect or relate personal information by means of the system. This technical approach guards against vendors who merely promise to respect their customers' privacy when in fact they do not. Thus, by using our system, a vendor committed to the privacy of his customers can actually prove that he lives up to his promises, which should foster trust in his business. Here, proof means that anyone looking at a protocol transcript of our loyalty system can verify that no identifying or personal information is conveyed by the protocol.

More specifically, to protect the customers' privacy, we require that by examining the loyalty system's protocol transcripts, the vendor must not be able to link customers' transactions, e.g., to create consumer profiles. Hence, it must be ensured that issued or redeemed loyalty points carry no information which would permit the vendor to link any two customer transactions. As a consequence, when points are handed in by the customer, it must not be possible for the vendor to determine the purchases in which the points were obtained. Of course, this is only meaningful if the vendor does not have access to linking information outside the loyalty system. In addition to unlinkability of points to transactions, there are security requirements with respect to unforgeability of points and preventing that the same points are redeemed more than once. Another aspect, which may be of interest to vendors, is to allow vendors to decide if they want to permit or prevent customers from pooling their loyalty points in order to receive a more valuable reward.

5.2 Loyalty Programs

A loyalty program is a structured marketing effort that rewards, and therefore encourages, loyal behaviour of customers, which is hopefully beneficial to the vendor [SS97]. We say that a customer is loyal if she prefers a certain vendor over its competitors. The motivation of vendors for adopting a loyalty program is, in general, twofold. First, vendors want to retain present customers and stimulate repeated purchase behaviour which would guarantee regular future earnings. Second, they want to learn more about their customers in order to refine their company's strategy.

In general, the basic condition for loyal customer behaviour in the real world are different from the electronic world [OO00]. Connecting to a vendor's site is as

easy as connecting to its competitor's site. This is in contrast to the real world where barriers exist, such as geographical distance or an existing inter-personal relationship between customer and shop personnel, that may prevent customers from instantly switching vendors. Thus, online vendors must be even more interested in loyalty programs than their real-world counterparts.

Different types of loyalty programs exist, such as rewarding systems and virtual communities. Rewarding systems are giving program members a financial incentive. They can be further classified according to the time the reward is given relative to the purchase. For instance, price promotions or rebates through membership cards are examples of immediate rewarding systems and programs instilling customers to collect points, such as frequent flyer miles or "buy 10 get one free", are examples of delayed rewarding systems. On the other hand, virtual communities focus on social and service aspects, e.g., online forum panels on product related problems, rather than on financial incentives.

In addition, point-based loyalty programs may appear in different forms. For instance, the number of points awarded to the customer may depend on the monetary value of a purchase, e.g., one point for each Euro spent, or it may depend on specific types of products, e.g., after having bought 10 MP3 files the customer may download one for free. Furthermore, we can categorise point-based programs according to the way points are collected. In a token-based approach, a token is issued to the customer for each point awarded, e.g., chips issued by a supermarket. By contrast, in a counter-based approach, the number of points awarded to the customer is added to her current balance of points, e.g., frequent flyer miles. A third aspect is whether vendors want to ensure that their customers individually achieve the redemption threshold or if they allow customers to pool their points. Pooling means that two or more customers are allowed to combine their collected points, e.g., to reach some redemption threshold earlier or to reach a higher threshold in order to receive a more valuable reward.

Loyalty programs are often beneficial to the vendor, as its members have a greater tendency to be loyal to the vendor and also have an increased interaction frequency compared to non-members [SS97]. Furthermore, it can be assumed that members are less willing to try offers of competing vendors, even when negative experiences with their preferred vendor occur since these effects are moderated by the loyalty program membership [BKB00]. According to [DU97], loyalty program members are also less price sensitive, spend more money at a shop and are more likely to pass on positive recommendations than non-members.

The customer information gathered in loyalty programs can be used by the vendor for direct marketing, data mining, and customer profiling in order to promote products, infer new customer data, and optimise their range of products. This means that vendors have a large database of consumer data where they record

every single transaction of their customers. Thus, common loyalty programs do not look so bright anymore from the customer's perspective since they may see this monitoring as an invasion to their privacy. In this context, customers may fear to lose control over their personal data, since vendors' may disclose their data to third parties. For these reasons, loyalty programs are also critically eyed by privacy advocates [Con06]. Clearly, customer loyalty strongly depends on the customers' trust in the vendor. Thus, if customers are convinced that they participate in a privacy-friendly loyalty program, their loyalty may even increase.

It is clear that vendors primarily interested in customer profiles will certainly not use the type of loyalty system we have in mind. Hence, our system is mostly of interest to vendors who do not work with customer profiles and who want to give their customers an incentive to buy again at their shop for both enhanced privacy and financial benefits.

5.3 Requirements

When designing an electronic loyalty system, both customers' and vendors' interests need to be taken into account. Therefore, requirements such as customers' privacy and of course, security regarding the unforgeability of loyalty points must be considered. In the following, we describe requirements that a loyalty system must fulfil.

5.3.1 Privacy

Customers have the fundamental requirement to protect their privacy. In our context, this means that it should not be possible for the vendor to create customer profiles from the awarding and redeeming processes of loyalty points. More precisely, it should not be possible for the vendor to link any two customer transactions by means of the loyalty system. This includes both awarding or redeeming transactions. In other words, given a redeeming transaction, the vendor should be prevented from linking it to

1) the corresponding awarding transactions and

2) other redeeming transactions of the same customer.

And likewise, given an awarding transaction, the vendor cannot link it to awarding and redeeming transactions of the same customer. Note that we focus only on the loyalty system's properties that are necessary to achieve unlinkability. Clearly, linkability may be possible outside the loyalty system. However, preventing this 'entirely' is out of scope of this work. In order to achieve unlinkability for electronic

purchases in general, additional technologies have to be used, e.g., unlinkability of search and order phases proposed in [EKS02b], payment systems that allow the customer to remain anonymous with respect to the vendor [Cha89, CPS96], anonymity networks as in [Cha81, RSG98], or privacy-friendly delivery in case of tangible goods, similar to the approach proposed in [EE02].

5.3.2 Security

The security requirements considered here can be summarised as *system integrity*. The property of system integrity in the context of a point-based loyalty system means that no other party beside the vendor should be able to create valid loyalty points. We have two aspects of system integrity —unforgeability and double-spending detection— that need to be considered. In addition, we consider a third property, pooling prevention, which is specifically related to loyalty systems.

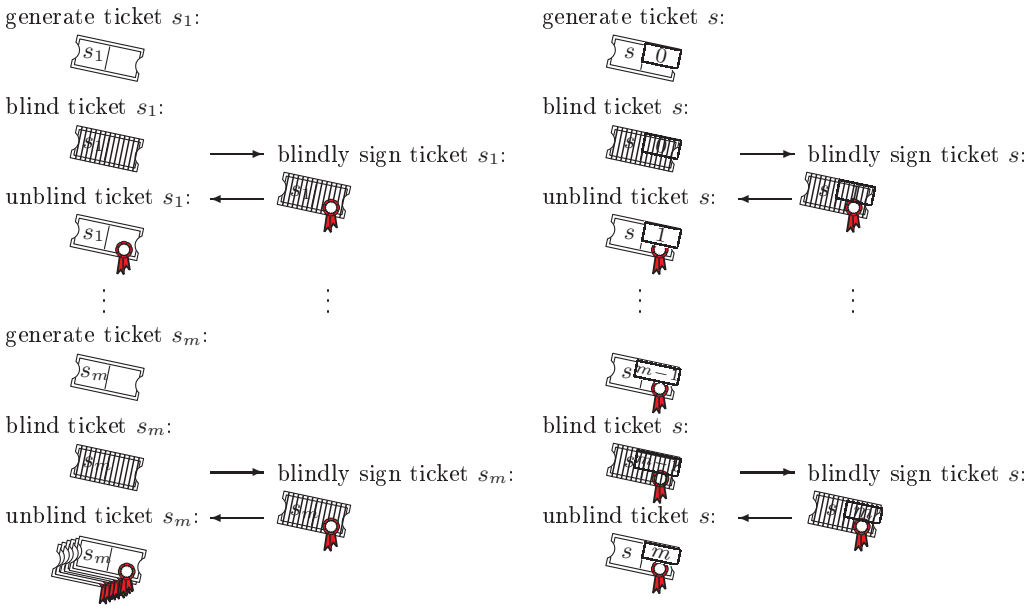
Unforgeability. Loyalty points may only be created by the vendor himself, i.e., customers should not be able to produce them. At the very least, the vendor should be able to tell false points from genuine ones.

Double-spending detection. In contrast to real-world loyalty points, their electronic counterparts can be easily copied and are indistinguishable. As a consequence, parties may try to hand-in copies of loyalty points to the vendor. Thus, we require that it must be *detectable* whether the same loyalty points have been spent before.

Pooling prevention. It should not be possible that, say, two customers, each having a counter of 5 loyalty points, can transform their individual counters into a joint one worth 10 points. Note that this is different from the problem of customers sharing a counter which, in general, cannot be prevented in systems with perfect privacy.

5.4 Electronic Loyalty Systems

In this section, we start by informally describing how the two basic variants of the electronic loyalty systems work. In order to point out the advantages of the counter-based systems over *ad hoc* solutions based on electronic coin systems, we will compare it to such systems. Henceforth, we will refer to coin systems as token-based systems, in contrast to the counter-based systems proposed here. As an example for a token-based system, we consider a scheme that is based on the idea of blind signatures proposed by Chaum [Cha83]. We will use this scheme to compare it to our proposal.



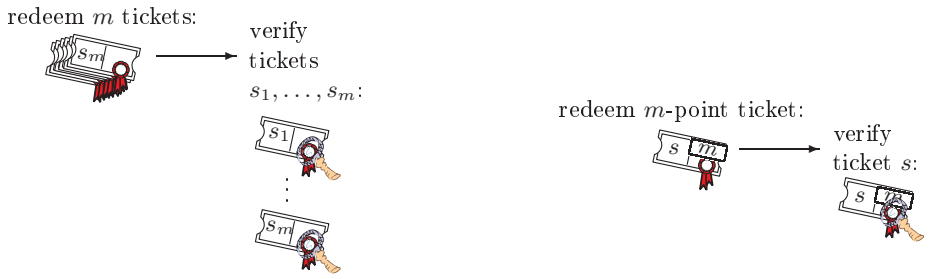
(a) *Token-based issue process* **(b)** *Counter-based issue/add process*

Figure 5.1: *Issuance of loyalty points*

The main difference between a token-based system and our counter-based system is that, in the former system, customers have to keep a token for every point issued to them, while in the latter system, they have to keep only one "token", i.e., an account balance, which represents a counter that is increased for every point issued to the customer. Figure 5.1 sketches the issue process for both types of loyalty systems, where the tickets represent the aforementioned tokens.

In the token-based scheme as well as in the counter-based scheme, random serial numbers s_i and s, respectively, are chosen by the customer which identifies the ticket holding the customer's point(s). This serial number is used by the vendor within the redeem phase in order to determine whether the ticket has been handed in before. In the counter-based scheme a single serial number s is sufficient for collecting, e.g., m loyalty points while in a simple token-based approach m different serial numbers $s_1, \ldots, s_m$ are needed.

In both schemes, the ticket's serial number must be hidden from the vendor to prevent him from linking the ticket to a particular issue step. This is necessary because a point, i.e., a ticket, is given to a customer as a reward for her purchases and therefore the ticket's serial number, if not hidden from the vendor, could be used afterwards to uniquely identify a certain purchase. Thus, in the token-based approach, the vendor would learn that m certain purchases, identified by the tickets'



(a) *Token-based redeem process* **(b)** *Counter-based redeem process*

Figure 5.2: *Redemption of loyalty points*

serial numbers, were made by the same person as soon as that person hands in her collected tickets. This would allow the vendor to *a posteriori* create a profile for this customer. In the counter-based approach, protecting the customer's privacy might be even more demanding, since the ticket's serial number is always the same for all counter values. Thus, the vendor could easily build a purchase-by-purchase customer profile, based on the serial number, whenever another loyalty point is added to the customer's ticket.

In order to prevent such building of customer profiles, the customer blinds her ticket before sending it to the vendor in order to hide the serial number from the vendor's prying eyes — in the counter-based approach, this blinding also hides the current balance of the ticket. In the next step, the vendor digitally signs the ticket given to him, without knowing its content — he will learn (and verify) the ticket's content only after it is handed in by the customer. In the counter-based scheme, the vendor 'automatically' increases the ticket's counter by signing it (for the details see Sections 5.6 and 5.7). The signing itself is actually the 'core' issue step of a loyalty point and also serves to protect the ticket from unnoticeable tampering. The ticket is afterwards returned to the customer who unblinds it and after that owns a ticket worth one (additional) loyalty point. Figure 5.1(a) shows that in the token-based approach m different tokens, i.e., tickets, must be generated and stored by the customer in order to collect m loyalty points. By contrast, Figure 5.1(b) shows that in the counter-based approach only one 'token' needs to be generated and stored in order to hold m loyalty points. If more than one point, e.g., k points, is to be issued at once, the counter's advantage becomes even more apparent. In this case, the vendor merely has to change the addend from 1 to k which means no additional effort for him. However, in the token-based approach, the customer would practically have to go through k issuing steps, considerably reducing efficiency for both the customer and the vendor.

One could think of improving the efficiency of the token-based scheme by issuing tickets of higher values, e.g., tickets worth $k, 2k, \ldots$ points. From the privacy perspective, however, this approach discloses more information to the vendor than the approach using uniformly valued tokens. This additional information can be exploited by the vendor to increase the probability for linking a redemption process to a particular purchase of some customer. For instance, if some customer is given a ticket worth k' points for some purchase and no one else is given a ticket of this value, the vendor will be able to link the corresponding purchase to the ticket, as soon as it is redeemed. This also implies that counter-based schemes must follow our natural understanding of counters, i.e., they have to represent the value of a sum. For instance, a counter-based scheme permitting recovery of the terms contributing to the sum would just be as bad as token-based schemes with non-uniform values, since one of the terms might just be some re-identifiable k' as argued before. However, if a counter worth k' points is represented as a single value, the vendor will never know if k' points have been issued at once or if they are the result of, e.g., k' issuing processes worth 1 point each. Hence, he cannot trace the counter back to specific purchases.

When the customer has collected a certain number of loyalty points, say m, she may redeem them at the vendor's for some reward. Figure 5.2 shows the redemption process for both schemes. The obvious disadvantages of the token-based scheme from Figure 5.2(a) are that m tokens must be transferred to the vendor and every single token must be verified by the vendor. By contrast, the customer needs to send only a single token, worth m loyalty points, in a counter-based approach (see Figure 5.2(b)) and consequently the vendor has to do only one verification.[1]

In summary, the counter-based approach uses, for $m > 1$ loyalty points, less storage space[2] on the customer side, less bandwidth in the redeem step, and less computation time in the vendor's verification step than the token-based approach. This improvement in efficiency can be achieved without sacrificing privacy or security as we will show in the detailed protocol descriptions within the next sections. We start with the description of a token-based system and afterwards introduce our counter-based schemes.

[1] Obviously, this argument is only reasonable if the workload for counter verification is smaller than the verification of a corresponding number of tokens.

[2] This assumes, of course, that the space required to store counters only grows less than linear in the number of loyalty points. As we will see in Sections 5.6 and 5.7, this property holds for our counter-based loyalty systems.

5.5 A Token-based Loyalty System

In this section, we describe an electronic token-based loyalty system which uses blind signatures proposed by Chaum [Cha89]. The following also provides a brief introduction to Chaum's scheme for readers not familiar with it.

The token-based loyalty scheme consists of two protocols, the *issue* and *redeem* protocol, which correspond to the withdrawal and deposit steps, respectively, of Chaum's blind signature protocol [Cha89]. Chaum's blind signatures are in turn based on RSA [RSA78].[3] In contrast to Chaum's original proposal, which involves three parties, vendor, bank, and customer, we use it as a two-party protocol involving only a vendor and a customer. The vendor in our protocol also plays the role of the bank in Chaum's protocol. The goal of Chaum's protocol was to establish unlinkability of withdrawal and deposit transactions with respect to the bank. Translated to our context, this means that Chaum's protocol can guarantee the unlinkability of issuing and redeeming transactions. According to Section 5.3, we also require the unlinkability of any two issue transactions and any two redeem transactions. This aspect is not considered in Chaum's payment protocol, e.g., withdrawals are linkable. However, as we will see, Chaum's protocol can be set up to achieve all unlinkability requirements of Section 5.3 by simply modifying Chaum's original scenario.

5.5.1 System Setup

In an initial step, the vendor runs a key generator $KeyGen^{\mathrm{RSA}}$ to produce the system parameters for the underlying RSA cryptosystem. On input k, where k is a security parameter, $KeyGen^{\mathrm{RSA}}$ outputs two k-bit primes p and q, as well as public and private exponents e and d, respectively, such that $ed = 1 \bmod (p-1)(q-1)$. The values p, q, d are kept secret by the vendor and $(e, n := pq)$ are published as his public key. The vendor also chooses an appropriate cryptographic hash function $h(\cdot)$ which maps integers to the group $\mathbb{Z}_n^*$. We denote the vendor's signature on some message x by $\sigma(h(x)) := (h(x))^d \bmod n$. Unless otherwise noted, computations in the following are $\bmod n$.

[3] For our illustration, we could have also used a blind signature protocol based on, say, the intractability of the Diffie-Hellman problems (see Figure 5.8). However, as we are not concerned about the nature of the underlying cryptographic assumption here, RSA was merely chosen for convenience.

Customer		Vendor
choose $b \in_R \mathbb{Z}_n^*$, $s \in_R \mathbb{Z}_n$;		
compute $t := b^e h(s)$;	$\xrightarrow{t}$	
	$\xleftarrow{\sigma(t)}$	(blindly) sign t: $\sigma(t) := t^d$;
compute unblinding factor b^{-1};		
unblind $\sigma(t)$:		
$\sigma(t)b^{-1} = t^d b^{-1} = (b^e h(s))^d b^{-1}$		
$= b\,h(s)^d b^{-1} = h(s)^d = \sigma(h(s))$;		
verify: $\sigma(h(s))^e \stackrel{?}{=} h(s)$;		
store point $\langle s, \sigma(h(s))\rangle$ if signature is valid;		

Figure 5.3: *Token-based issue protocol*

5.5.2 Protocols

Issue. First, the customer randomly chooses a serial number $s \in_R \mathbb{Z}_n$ which serves as an input for the hash function $h(\cdot)$.[4] Furthermore, she randomly chooses the blinding factor $b \in_R \mathbb{Z}_n^*$ for computing the blinded serial number t.

Figure 5.3 shows the protocol for issuing a loyalty point $\langle s, \sigma(h(s))\rangle$. If $m > 1$ loyalty points are awarded to the customer for some purchase, then the issue protocol is repeated m times and the customer receives m tokens $\langle s_1, \sigma(h(s_1))\rangle, \ldots, \langle s_m, \sigma(h(s_m))\rangle$.

Redeem. If the customer has reached the redemption threshold m, i.e., gathered enough points to hand them in for a reward, she may execute the redeem protocol shown in Figure 5.4. If the protocol was successfully completed, the vendor sends the reward to the customer. Finally, the vendor stores the serial numbers of all redeemed and valid loyalty points in a local database.

5.5.3 Attack Model

A successful attack on the security of the token-based loyalty system considered above means for an adversary to create one or more additional tokens that have

[4] Chaum proposed to select s from a special set $\mathbb{S}_n \subset \mathbb{Z}_n^*$ which is predetermined by the vendor. For instance, $\mathbb{S}_n$ can be a set of palindromes mod n, i.e., all numbers less than n whose binary representations are the same when read from left to right or from right to left. In [Cha89] other examples for $\mathbb{S}_n$ are considered.

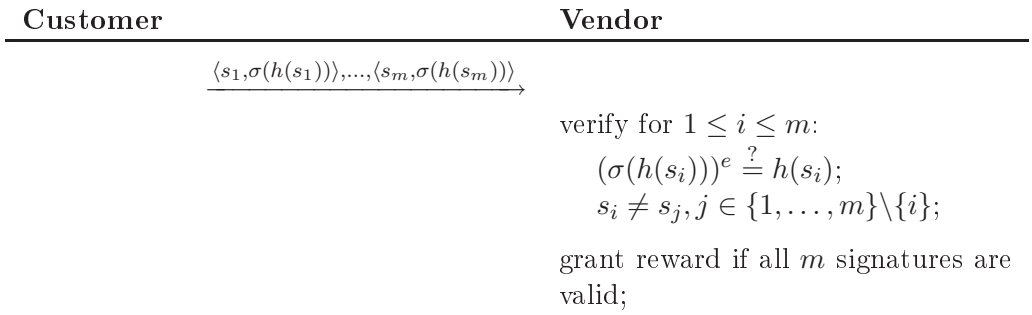


Figure 5.4: *Token-based redeem protocol*

not been issued by the vendor. Specifically, in l interactions with the vendor where l is polynomial in the security parameter k, the adversary must have managed to gain $m \geq l + 1$ points. This type of forgery of the RSA blind signature scheme is known as *one-more-forgery* and was introduced in [PS00]. We will come back to this point in Section 5.5.5 where we discuss the security of the token-based loyalty system. Before that, we argue the privacy aspects of the system.

5.5.4 Privacy

The privacy requirement states that it should not be possible for the vendor to link any two customers' transactions by means of the loyalty system. Using our formalism from Section 3.1, this means that any two issue protocol runs I and I' cannot be linked, i.e., $\mathcal{L}(I, I')$ does not hold, and likewise any two redeem protocol runs R and R' cannot be linked, i.e., $\mathcal{L}(R, R')$ does not hold. In addition, for any I and any R, $\mathcal{L}(I, R)$ must not hold, either. Note that the statement "$\mathcal{L}(X, Y)$ does not hold" means that *the vendor* possesses no link data that allows him to decide if $\mathcal{L}(X, Y)$ is true. Clearly, any customer may decide this for her own protocol runs because she possesses the necessary data, e.g., serial numbers, blinding factors, and blinded tokens.

More precisely, let $\mathfrak{D}$ be the set of protocol runs carried out between a vendor V and his customers, and let $\mathcal{L}$ be the link relation defined over $\mathfrak{D} \times \mathfrak{D}$. Furthermore, let $\mathfrak{I} \subseteq \mathfrak{D}$ be the set of issue transactions in which V was involved and let $\mathfrak{R} \subseteq \mathfrak{D}$ be the set of his redeem transactions — clearly, $\mathfrak{I} \cap \mathfrak{R} = \emptyset$. An element $I \in \mathfrak{I}$ constitutes the set of messages exchanged between V and his customers in an issue protocol run, e.g., the transcript $I := \{t, \sigma(t)\}$ (c.f. Figure 5.3), and analogously, an example for $R \in \mathfrak{R}$ would look like $R := \{s_1, s_2, \ldots, s_m, \sigma(h(s_1)), \ldots, \sigma(h(s_m))\}$ (c.f. Figure 5.4). Recall that for the analysis, as well as throughout this whole chapter, only data gathered by the vendor in his loyalty system related communication with his

customers is considered, i.e., we assume that no other link data is available to him that allows him to decide the link relation $\mathcal{L}$.

Issue transactions. Given $I, I' \in \mathfrak{I}$, the vendor's goal is to decide if $\mathcal{L}(I, I')$ holds, where I and I' are assumed to be distinct. We will now argue that this is not possible, if the customer follows the protocol.

For contradiction, assume that V is able to relate two issue transactions I and I' by analysing their transcripts $I := \{t, \sigma(t)\}$ and $I' := \{t', \sigma(t')\}$. By assumption, the customer selects a blinding factor's base b uniformly at random from $\mathbb{Z}_n^*$. Raising b to the e-th power results in a permutation on $\mathbb{Z}_n^*$ and hence, the blinding factor $b^e \bmod n$ is also uniformly distributed on $\mathbb{Z}_n^*$. Furthermore, blinding the hashed serial number $h(s)$, i.e., forming $t = b^e h(s)$, is also a permutation on $\mathbb{Z}_n^*$ and again, t is uniformly distributed on $\mathbb{Z}_n^*$. Now, suppose we randomly and uniformly choose $\hat{t} \in_R \mathbb{Z}_n^*$ and set $\hat{I} := \{\hat{t}^e \bmod n, \hat{t}\}$. Then $\hat{I}$ is indistinguishable from a real transcript as it has the same distribution. Since I and $\hat{I}$ are identically distributed, we can essentially replace I with $\hat{I}$. If V ruled that $\mathcal{L}(I, I')$ holds, he must also decide that $\mathcal{L}(\hat{I}, I')$ holds, as I and $\hat{I}$ have the same distribution. However, since $\hat{I}$ is a random transcript, it is unrelated to any real transcript and hence, $\mathcal{L}(\hat{I}, I')$ does not hold and V's verdict was not correct. This contradicts our assumption that V can decide the link relation $\mathcal{L}$ for issue transactions. Hence, we conclude that V cannot relate issue transactions, as long as the customer applies the blinding specified in the protocol.

Redeem transactions. Let us now turn to the unlinkability property of two distinct redeem transactions $R, R' \in \mathfrak{R}$. A redeem transaction differs from another one by its serial numbers and possibly in the number of points which are redeemed. Suppose we have two transcripts from real redeem transactions

$$\begin{aligned} R &:= \{s_1, \ldots, s_l, \sigma(h(s_1)), \ldots, \sigma(h(s_l))\} \quad \text{and} \\ R' &:= \{s'_1, \ldots, s'_m, \sigma(h(s'_1)), \ldots, \sigma(h(s'_m))\}. \end{aligned}$$

If the customer(s) followed the protocol, every serial number s_i and each s'_j was randomly and uniformly chosen from the set $\mathbb{Z}_n$. Similar to the case of the issue transactions, we assume for contradiction that V is able to decide if $\mathcal{L}(R, R')$ holds. In contrast to our approach in the previous paragraph, we cannot simply make up a random transcript which is identically distributed to a real transcript, as this would imply that we can either forge signatures or invert the hash function. Therefore, let $\tilde{R} := \{\, \tilde{s_1}, \ldots, \tilde{s_u}, \sigma(h(\tilde{s_1})), \ldots, \sigma(h(\tilde{s_u}))\}$ be a redeem transaction that is unrelated to R and R' if $\mathcal{L}(R, R')$ holds and otherwise, let $\tilde{R}$ be related to, say, R'.

Suppose, $\mathcal{L}(R, R')$ holds, then by assumption neither $\mathcal{L}(R, \tilde{R})$ nor $\mathcal{L}(\tilde{R}, R')$ holds. However, since the serial numbers from R, R', and $\tilde{R}$ are randomly and

uniformly chosen from $\mathbb{Z}_n$ they are identically distributed and, of course, independent of each other. For a fixed public key (n, e) and hash function $h(\cdot)$, the signatures $\sigma(\cdot)$ from the transactions can be ignored for this analysis, since they fully depend on the serial numbers. Now, as argued before in the paragraph of the issue transactions, since the distributions of R and $\tilde{R}$ are identical, we can replace R with $\tilde{R}$. Hence, R and $\tilde{R}$ are indistinguishable, which means that V must rule that $\mathcal{L}(\tilde{R}, R')$ also holds, which is not the case because $\tilde{R}$ is unrelated to R'. Hence, this contradicts the assumption that V can decide $\mathcal{L}$.

A similar contradiction occurs if $\mathcal{L}(R, R')$ does not hold and we replace R with $\tilde{R}$ — note that by assumption $\mathcal{L}(\tilde{R}, R')$ holds in this case. As V will not see any difference between the distributions of R and $\tilde{R}$, his verdict must be the same for $\mathcal{L}(\tilde{R}, R')$ and for $\mathcal{L}(R, R')$, i.e., that they are unrelated. Again, this is a contradiction. Consequently, V cannot decide if $\mathcal{L}(R, R')$ holds, given that customers select their serial numbers at random from $\mathbb{Z}_n$.

Issue and redeem transactions. Finally, we argue that the vendor V cannot decide if $\mathcal{L}(I, R)$ holds for any $I \in \mathfrak{I}$ and any $R \in \mathfrak{R}$. Suppose we have

$$\begin{aligned} I &:= \{t := b^e h(s) \bmod n, \sigma(t)\} \quad \text{and} \\ R &:= \{s_1, \ldots, s_l, \sigma(h(s_1)), \ldots, \sigma(h(s_m))\}. \end{aligned}$$

To decide $\mathcal{L}(I, R)$, the vendor needs to determine if $s \in \{s_1, \ldots, s_m\}$. Assume that V is able to do this for any $I \in \mathfrak{I}$ and any $R \in \mathfrak{R}$ — note that he does not even know s. As already argued in the issue paragraph, I can be replaced with a random transcript $\hat{I}$ which has a distribution identical to the one of I. Hence, we can replace I with $\hat{I}$. If V can decide $\mathcal{L}(I, R)$, he must be able to detect that $\mathcal{L}(\hat{I}, R)$ does not hold. This is a contradiction, since the distributions of the sets I and $\hat{I}$ are indistinguishable, i.e., $\mathcal{L}(\hat{I}, R)$ 'looks like' $\mathcal{L}(I, R)$, and hence, V cannot decide differently for $\mathcal{L}(I, R)$ and $\mathcal{L}(\hat{I}, R)$. Hence, we conclude that V cannot relate issue transactions with redeem transactions as long as the customer blinds her serial numbers according to the specification of the issue protocol.

5.5.5 Security

The goal of the security requirements given in Section 5.3 was to prevent a fraudster from creating valid loyalty points $\langle s, \sigma(h(s))\rangle$. The fulfilment of these requirements can be directly derived from the security of both RSA and Chaum's work. Nevertheless, in the following we mention a few points in this regard.

Unforgeability. We require that no one except for the vendor can create valid tokens $\langle s, \sigma(h(s))\rangle$ such that $\sigma(h(s))^e = h(s)$. Of course, this assumes that only the vendor

knows the signing key d and that the signature scheme is secure. In the presented scheme, tokens are produced using the blind RSA signature scheme. This scheme was proven to be secure against one-more-forgeries in the random oracle model, as long as the *known-target inversion problem* is hard [PS00]. The known-target inversion problem RSA-KTI is defined by the following experiment due to Bellare *et al.* [BNPS03].

Experiment $\mathbf{Exp}_{A,m}^{\text{RSA-KTI}}(k)$:

$(n, e, d) := KeyGen^{\text{RSA}}(k)$;
For $i := 1, \ldots, m$: $x_i \in_R \mathbb{Z}_n^*$;
$\langle y_1, \ldots, y_{m(k)+1} \rangle := A^{(\cdot)^d \bmod n}(n, e, x_1, \ldots, x_{m(k)+1})$;
If all of the following conditions hold, return *success* else return *failure*
- $\forall i \in \{1, \ldots, m(k)+1\} : y_i^e = x_i \bmod n$
- A made at most $m(k)$ queries to the inversion oracle $(\cdot)^d \bmod n$

Basically, what this definition says is that a forger A for RSA-KTI[m] succeeds, if it can produce, for (at least) one of the x_i given to it as input, an RSA signature on its own, i.e., without calling the RSA inversion oracle $(\cdot)^d \bmod n$. The inversion oracle serves to provide A with message-signature pairs $(\xi, \sigma(\xi))$ at its discretion. Of course, at most $m(k)$ such pairs are given to A because otherwise the experiment would be trivial. The relation of the experiment and our attack model from Section 5.5.3 is outlined next.

The inversion oracle $(\cdot)^d \bmod n$ plays the role of the vendor which issues points / tokens to his customers. A cheating customer, i.e., the adversary A, may engage in $m(k)$ interactions with the vendor to produce m tokens $(s_i, \sigma(h(s_i)))$ or whatever other useful signatures she may take advantage of. Note that the vendor/inversion oracle signs any value t provided to it. Also note that A must 'compensate' calls to $(\cdot)^d \bmod n$ which do not provide her with a direct signature on any x_i. That is, if A submits $y \notin \{x_1, \ldots, x_{m(k)+1)}\}$ to $(\cdot)^d \bmod n$, this call still counts for her total allowable number of oracle calls and she must still produce $m(k)+1$ signatures, one for every x_i given to her.

In conclusion, since solving RSA-KTI[m] in polynomial time is conjectured to be infeasible, a polynomial time algorithm for producing one-more-forgeries of the blind RSA signature scheme is assumed to be non-existent. Hence, the blind RSA signature scheme is assumed to be secure. An extensive discussion on the security of the blind RSA signature scheme and its relation to various other RSA related problems can be found in [BNPS03].

Double-spending detection. Since a customer can try to redeem copies of loyalty points, the vendor stores all serial numbers in a local database and compares each

s from a newly submitted point to the database's. If some s is already stored in the database then it has been double-spent and will not be accepted. This can be assumed because the probability that two customers pick the same $s \in \mathbb{Z}_n$ as a serial number is negligible.

Pooling prevention. Unfortunately, pooling of loyalty points cannot be prevented, since each point is represented by a serial number s_i which is uniformly chosen at random from the set $\mathbb{Z}_n$, and the vendor does not learn the number before redemption (assuming that the blind signature scheme is secure). Thus, the vendor cannot detect whether loyalty points to be redeemed had been issued to any one customer or to a group of customers who pooled their points.

5.5.6 Efficiency

Issuing m loyalty points always requires the creation of m tokens. Thus, costs in the issue protocol are linear in the number of issued loyalty points. This means that the size of data to be stored and transferred, the amount of required serial numbers, the customer's effort for blinding and signature verification, and the vendor's effort for signature generation grow linearly in the number of loyalty points. The redeem protocol shows similar properties. Also here, the size of data to be transferred and stored (database for serial numbers) grows linearly with the number of redeemed points. However, in contrast to the signature creation, the verification can be condensed to a single modular exponentiation and $2(m-1)$ modular multiplications by checking

$$\prod_{i=1}^{m} h(s_i) \stackrel{?}{=} \left(\prod_{i=1}^{m} \sigma(h(s_i)) \right)^e \bmod n. \tag{5.1}$$

We note that more efficient techniques for batch verification of RSA signatures have been developed [PMPS00] which are suitable even for super-polynomial m, however, with a small soundness error[5].

5.6 A Counter-based Loyalty System based on RSA

In this section we introduce a counter-based optimisation of the scheme from the previous section. As in the token-based system, we have two protocols, *issue* and *redeem*. However, instead of issuing a token for every loyalty point earned by the customer, as in the issue protocol of Section 5.5, the vendor now adds the loyalty points to be issued to a counter owned by the customer. This counter-based loyalty

[5] In these tests, a small probability exists —usually $2^{-\ell}$, where ℓ is the security parameter for the test— that a batch with at least one bad signature passes the test.

system also makes use of Chaumian blind signatures [Cha83] and introduces what we call *anonymous counters*.

In the following, we describe the details of the proposed loyalty system. The system itself mainly consists of two protocols, *issue* and *redeem*, which correspond to the processes shown in Figures 5.1b and 5.2b, respectively, from Section 5.4.

5.6.1 System setup

Let k be the vendor's security parameter. First, the vendor runs an RSA key generator algorithm $KeyGen^{\mathrm{RSA}}$ which on input k returns (n, e, d), where n denotes the public modulus, which is the product of two odd primes, p and q, and satisfies $2^{k-1} < p, q < 2^k$. By e and d we denote the public and secret exponent, respectively, where $ed = 1 \bmod (p-1)(q-1)$. Along with his public key, (n, e), the vendor publishes the description of a cryptographic hash function $h(\cdot)$. Unless otherwise noted, computations in the following are $\bmod n$. By $\sigma^m(x)$ we denote an *m-times signature* on some input x, i.e., $\sigma^m(x) := \sigma(\sigma(\ldots\sigma(\sigma(x))\ldots)) = x^{d^m}$. For simplicity, we assume that a vendor only uses one public/private key pair for his loyalty system and therefore, we omit the denotation of d for the signing function σ.

In the proposed system, a counter value representing m loyalty points contains an m-times signature on a hashed serial number s, i.e., $\sigma^m(h(s)) = h(s)^{d^m}$ represents a counter value worth m loyalty points. Thus, each counter value is associated with exactly one serial number s.

5.6.2 Protocols

Initialisation. Before the customer starts the issue protocol for the first time, she has to setup her initial counter. This means that she randomly chooses a serial number $s \in_R \mathbb{S}$, where $\mathbb{S}$ is a sufficiently large, but finite, set of integers, and sets the counter's start value m_0; usually $m_0 = 0$.

Issue. Assume that we have a counter value with $m_{i-1} \geq 0$ points that the customer gathered in $i-1$ purchases so far, where $i \geq 1$. Further assume that she will obtain $l_i > 0$ points in her i-th purchase, yielding $m_i := m_{i-1} + l_i$. In order to obtain l_i points, she hands in a blinded counter value t_i as shown in Figure 5.5. Then, the vendor blindly l_i-signs t_i and returns the result to the customer. Here, l_i-signing means that the vendor raises t_i to the d^{l_i}-th power, which can be done in a single step. Upon receiving the l_i-signed t_i, the customer unblinds it and obtains an m_i-times signature of $h(s)$. This scheme is then used for subsequent purchases in the same manner, i.e., the result of the issue protocol in purchase $i-1$

Customer		**Vendor**
choose $b_i \in_R \mathbb{Z}_n^*$;		
compute		
$t_i := b_i^{e^{l_i}} \sigma^{m_{i-1}}(h(s)) = b_i^{e^{l_i}} h(s)^{d^{m_{i-1}}}$;	$\xrightarrow{t_i}$	
		(blindly) l_i-sign t_i
	$\xleftarrow{\sigma^{l_i}(t_i)}$	$\sigma^{l_i}(t_i) := t_i^{d^{l_i}}$;
compute b_i^{-1};		
unblind $\sigma^{l_i}(t_i)$:		
$\sigma^{l_i}(t_i) b_i^{-1} = t_i^{d^{l_i}} b_i^{-1} = (b_i^{e^{l_i}} h(s)^{d^{m_{i-1}}})^{d^{l_i}} b_i^{-1}$		
$= b_i h(s)^{d^{(m_{i-1}+l_i)}} b_i^{-1} = h(s)^{d^{(m_{i-1}+l_i)}}$		
$= \sigma^{(m_{i-1}+l_i)}(h(s)) = \sigma^{m_i}(h(s))$;		
verify $(\sigma^{m_i}(h(s)))^{e^{m_i}} \stackrel{?}{=} h(s)$;		
store $\langle s, m_i, \sigma^{m_i}(h(s)) \rangle$ if valid;		

Figure 5.5: *Issue protocol for l_i loyalty points in the customer's i-th purchase* $(i \geq 1)$

becomes an input for the issue protocol in purchase i. After i purchases with iterated l_j-signing for $j = 1, \ldots, i$, the customer has a hashed serial number s which is raised to the $d^{m_0 + \sum_{j=1}^{i} l_j}$-th power. Then, the actual counter, which represents the number of loyalty points, is given by the number of signatures on $h(s)$, i.e., $m_i = m_0 + \sum_{j=1}^{i} l_j$. The customer keeps the current balance of loyalty points by storing $\langle s, m_i, \sigma^{m_i}(h(s)) \rangle$. For carrying out exponentiations $b_i^{e^{l_i}}$, $(\sigma^{m_i}(h(s)))^{e^{m_i}}$ on the customer side, and $t_i^{d^{l_i}}$ on the vendor side, efficient techniques can be applied, such as *square-and-multiply algorithms* or the *Montgomery exponentiation algorithm* [MOV97]. Furthermore, since the vendor knows the factorisation of n, he can also apply reductions $\bmod (p-1)(q-1)$.

Redeem. If the customer has collected enough points to gain a reward, say m points, she executes the redeem protocol as shown in Figure 5.6. The vendor grants the reward if $\langle s, m, \sigma^m(h(s)) \rangle$ is a valid triple and then stores the serial number of the triple in a local database. In order to reduce the computation costs for exponentiations $(\sigma^m(h(s)))^{e^m}$ in the verification, the vendor can use the methods suggested in the issue protocol.

Note that the customer always has to hand in the triple with the most current counter value in the redeem protocol. Handing in triples with an intermediate counter value is to the customer's disadvantage since each serial number is only accepted once by the vendor.

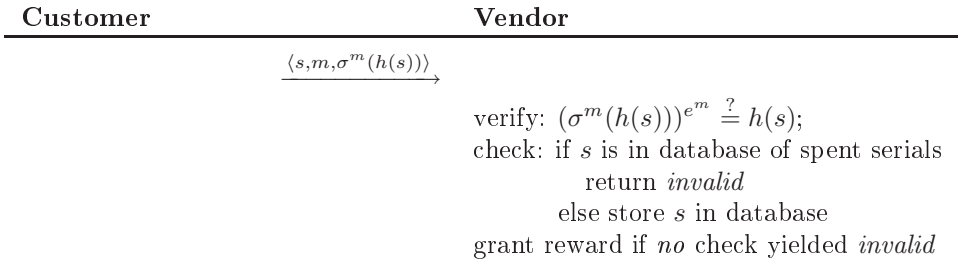


Figure 5.6: *Redeem protocol*

In case a customer hands in a counter representing m loyalty points but chooses a reward for $u < m$ points, the redeem protocol is carried out in combination with the issue protocol initialised with a new serial number s', $m_0 = 0$, and $l_1 = m - u$. This is similar to change given in payments.

5.6.3 Attack Model

In an attack, the adversary plays the role of several customers interacting with the vendor, i.e., he runs issue protocols and finally the redeem protocol to claim a reward. We call these feigned customers *controlled users*. Controlled users may deviate from the protocol in an arbitrary fashion. The adversary succeeds in his attack if he can claim more points than the total sum of points issued to all customers, similar to the setting of one-more-forgeries in the token-based attack model (c.f. Section 5.5.3). This definition captures forgery but not pooling, because if, say, two users, each having a counter value worth $m/2$ points, manage to combine their counter values into a joint one, worth m points, no more points than issued are claimed. Yet, m points have been wrongfully pooled.

In order to incorporate this event into our attack model, we define another group of customers, which we call *scheduled users*. A scheduled user basically starts as an autonomous customer honestly running the protocol. However, the adversary schedules the actions of these users, i.e., determines when these users run the initialisation or issue protocol. Whenever a scheduled user is done with its actions, it returns to an idle state and waits for the adversary to command it again. If the adversary creates a new scheduled user in an attack, this user immediately initialises itself and then becomes idle. Afterwards, the adversary may wake up a scheduled user and order it to step its counter. This means, the user runs the issue protocol with its current counter value, saves the updated counter value, and returns to an idle state, again. At his discretion, the adversary may corrupt a scheduled user which then becomes a controlled user. At this time, the scheduled user hands over

all protocol transcripts and intermediate data, such as intermediate counter values and blinding factors, generated during its interaction with the vendor.[6] Note that until a scheduled user is corrupted, the adversary is totally oblivious to the user's 'internal state', except for the user's current number of points which is identical to the number of steppings ordered by the adversary.

Thus, the adversary can interact with scheduled users in three ways: create a new user, step a user's counter (i.e., order it to run the issue protocol), corrupt a user.

System break. For the condition of a system break we count issued loyalty points as follows. For simplicity, we assume that the vendor adds only one loyalty point per run of the issue protocol, instead of l. Each time any controlled user runs the issue protocol, we increase the adversary's global counter of loyalty points, m_A, by 1. However, issue protocol runs by scheduled users are counted individually, i.e., for each scheduled user S a counter m_S is used. All counters are initially set to zero.

We say that an adversary A *breaks the loyalty system* if it claims *strictly more* than $m_A + \max\{m_S\}$ loyalty points, where the maximum is taken over the counters of all scheduled users S that were created during the attack.

Note that the bound $m_A + \max\{m_S\}$ can be trivially reached by an adversary that schedules a user $\max\{m_S\}$ times, corrupts it afterwards (and thereby taking over the user's counter values), and performs m_A subsequent runs of the issue protocol by itself, using the corrupted user's counter values as the starting counter value.

Claiming more than m_A points (with $\max\{m_S\} = 0$) captures cases of genuine forgery, while the second term of the sum reflects pure pooling attacks (with $m_A = 0$). The latter, perhaps, needs a little explanation. Recall the example from Section 5.3.2, where two customers each own a counter value worth 5 loyalty points. One can think of these two as scheduled users S_1 and S_2 which were corrupted by the adversary after they gathered 5 loyalty points. We have motivated pooling prevention by saying that it should not be possible for them to combine their counter values into a joint counter value worth 10 points. In fact, by specifying the condition for a system break as above, we even make it easier for poolers to satisfy this condition. Since $\max\{m_{S_1} = 5, m_{S_2} = 5\} = 5$, they only have to come up with 6 or more points, instead of 10, to break the system.

[6] Usually, honest users are supposed to delete such information. However, reliable erasure is in general hard to achieve and the adversary may later be able to recover the values from the user's hard disk. Thus, a conservative approach is to presume that the user in fact saves the values explicitly.

5.6.4 Privacy

For the analysis of counter-based loyalty system's privacy property, we use the same definitions as in the token-based approach (see Section 5.5.4). That is, we consider a link relation $\mathcal{L}$ defined over the set of loyalty system-related transactions $\mathfrak{D}$ seen by some vendor V and we denote the subset of issue and redeem transactions by $\mathfrak{I}$ and $\mathfrak{R}$, respectively.

Issues transactions. The blinding in the issue protocol of the counter-based approach slightly differs from the one of the token-based approach, if $l > 1$ point are to be issued. However, it can be easily seen that this blinding of $h(s)$ still ensures that the resulting value is still uniformly distributed on $\mathbb{Z}_n^*$. This means that we can setup a random transcript $\hat{I} := \{t^{(e^l)}, t\}$ which is identically distributed to real transcripts. Hence, using the same arguments as in Section 5.5.4, V will not be able to distinguish a random transcript from a real one. Consequently, he cannot distinguish between cases where he is given two real transcripts I and I' for which $\mathcal{L}(I, I')$ holds and cases where he is given $\hat{I}, I'$ such that $\mathcal{L}(\hat{I}, I')$ never holds, as $\hat{I}$ is independent of any issue transaction. Therefore, if V cannot distinguish these cases, he will not be able to decide if the link relation $\mathcal{L}$ holds.

Redeem transactions. Consider the following two distinct redeem transactions $R := \{s, l, \sigma^l(h(s))\}$ and $R' := \{s', m, \sigma^m(h(s'))\}$. As in the token-based variant, in order to determine if $\mathcal{L}(R, R')$ holds, i.e., if they had been carried out by the same customer, the vendor must be able to tell if s and s' had been selected by the same customer. As we have already seen in the analysis of the token-based scheme, this is infeasible if customers follow the protocol, i.e., select serial numbers uniformly at random from $\mathbb{S}$. Hence, V cannot decide $\mathcal{L}(R, R')$.

Issue and redeem transactions. In order to decide if $\mathcal{L}(I, R)$ holds for any issue transaction $I \in \mathfrak{I}$ and any $R \in \mathfrak{R}$, the vendor's goal is to determine if the serial number s from the redeem transaction R is encoded in I. Analogously to the $\mathcal{L}(I, I')$ case, assume that $\mathcal{L}(I, R)$ can be decided by V. Now, feed related I and R to the vendor. By assumption V will be able to confirm that $\mathcal{L}(I, R)$ holds. However, since I and a random transcript $\hat{I}$ formed as in the issue paragraph are indistinguishable, V will not be able to tell I from $\hat{I}$ and consequently, if he determined that $\mathcal{L}(I, R)$ holds, he will determine that $\mathcal{L}(\hat{I}, R)$ holds. The latter is clearly not the case and hence a contradiction to the assumption that V can decide $\mathcal{L}(I, R)$. In conclusion, the vendor V cannot decide $\mathcal{L}(I, R)$, either.

5.6.5 Security

The goal of the security requirements given in Section 5.3 was to prevent a forger from illicitly double-spending, increasing, or combining counter values of loyalty points. Double-spending can be easily prevented by storing serial numbers of previously redeemed counter values, checking presented serial numbers s' against the set of already spent numbers, and rejecting s' which had been stored before. Therefore, we move our attention to unforgeability and pooling prevention, for the rest of this subsection.

In the following, we show that the desired properties, unforgeability and pooling prevention, are preserved under the attack model presented in Section 5.6.3. The proof of security for this properties is based on the intractability of a new problem which we call *incremental RSA* (iRSA). Under the assumption that iRSA is hard, we prove the security of our loyalty system in the random oracle model [BR93].

The iRSA problem is closely related to the *RSA single-target inversion problem* (RSA-STI) treated in [BNPS03]. Still, we could not find a suitable reduction to our loyalty system from any well-known mathematical problem, such as factoring or a problem from the family of RSA-related problems [BNPS03, Poi99], e.g., RSA-STI. However, basing the security of our system on the intractability of a pure mathematical problem allows us to to investigate its security while hiding the details of the underlying protocol.

The incremental RSA problem. The task of solving the incremental RSA problem is as follows. Given a public RSA key (n, e) and some $x \in \mathbb{Z}_n^*$, where n denotes the modulus and e the encryption exponent, find $x^{d^m} \bmod n$ for unknown decryption exponent d and a given $m \geq 1$ polynomial in the security parameter k. For the attack, an adversary is given access to an RSA inversion oracle $(\cdot)^d \bmod n$, which on input y returns $y^d \bmod n$. The adversary's task is to come up with $x^{d^m} \bmod n$ while making at most $m - 1$ oracle queries. This means, for $m = 1$ the adversary must solve the problem without making any oracle queries. In this case, the iRSA problem is equivalent to the single-target inversion problem [BNPS03]. In the following, we use a notation similar to that of [BNPS03].

Definition 6 (incremental RSA problem: iRSA[m]) *Let $k \in \mathbb{N}$ be the security parameter and let $m : \mathbb{N} \to \mathbb{N}$ be a polynomial function of k. Let A be an adversary with access to an RSA inversion oracle $(\cdot)^d \bmod n$. The iRSA[m] problem is defined by the following experiment:*

Experiment $\mathbf{Exp}^{\mathrm{iRSA}}_{A,m}(k)$:

$(n, e, d) := KeyGen^{\mathrm{RSA}}(k)$;
$y := A^{(\cdot)^d \bmod n}(n, e, x, m(k))$;
If all of the following conditions hold, return *success* else return *failure*
- $y^{e^{m(k)}} = x \bmod n$
- A made *strictly less* than $m(k)$ oracle queries

The iRSA[m] problem is said to be hard if for any adversary A, whose time-complexity is polynomial in k, the success probability is negligible.

We now show that the iRSA problem reduces to the security of our loyalty system in the random oracle model [BR93], i.e., we show that breaking our loyalty system is at least as hard as solving iRSA. For this, we construct an adversary B that solves the iRSA[m] problem by using an arbitrary polynomial-time forger F for our loyalty system as a subroutine. The forger is given a public key (n, e) as an input and either outputs a triple (s, l, z) satisfying $z^{e^l} = h(s) \bmod n$ or *failure*. F has non-negligible probability of success ϵ, i.e., we have

$$Pr[F(n, e) = (s, l, z)] = \epsilon.$$

The following proof is by contradiction, i.e., under the assumption that the iRSA[m] problem is hard, we show how any successful forgery of F can be turned into a solution for the iRSA[m] problem. The forger's interactions with a vendor are simulated within a virtual environment, where the algorithm B uses an inversion oracle $(\cdot)^d \bmod n$ to simulate the vendor's responses in real runs of the issue protocol. Furthermore, the hash function $h(\cdot)$ used in the loyalty system is modelled as a random oracle, i.e., it maps inputs to uniformly and independently distributed group elements, repeating answers for previously queried inputs. The construction of the random oracle $h(\cdot)$ is such that in every output of $h(\cdot)$ the value x from the iRSA[m] problem is 'embedded' to ensure that any signature on a hash value can be turned into a signature on x, which we need to solve iRSA[m]. Of course, to the forger the output of $h(\cdot)$ is indistinguishable from a random value and hence, it will not notice the difference between a true random number and $h(\cdot)$'s output. Since this virtual environment looks like a real environment to the forger, it will succeed in its attack if it would do so in a real attack.

Now, the idea for the security reduction is roughly as follows. If the forger succeeds, it will come up with a counter value that has not been produced by the vendor, i.e., B. Since a counter value in the loyalty system must be an m-signature on some hash value $h(s)$ and because every hash value has x embedded, any such counter value can be turned into an m-signature for x. So if F succeeds, it must

have produced on its own at least one signature that is part of the m-signature on x, and hence, B will have used less than m calls to the inversion oracle $(\cdot)^d \bmod n$. Consequently, B will be able to produce an m-signature on x by having made strictly less than m calls to $(\cdot)^d \bmod n$, which is a solution to iRSA[m].

Before we start describing the algorithm B, we want to remark on some technical details. We assume that in the loyalty system, a polynomially bounded maximum number of loyalty points $m(k)$ can be redeemed at once. In the simulation, we assume for simplicity that the forger can only request a single point, instead of l points, in any run of the issue protocol. Furthermore, recall that the forger F succeeds in breaking our loyalty system, if it produced at least $m_A + \max\{m_S\} + 1$ loyalty points by interacting with the simulated vendor, i.e., B. Hence, B's goal is to answer all of F's requests by making at most $m_A + \max\{m_S\}$ oracle calls, in order to finally take advantage of F's capability to produce at least one additional (pooled) point on its own. Next, we give a detailed description of algorithm B, shown in Figure 5.7.

Description of iRSA[m] solver B. Lines 1–3 show the initialisation of three associative arrays, i.e., named arrays, and the simple indexed array X. If an array element is not present, e.g., a given name is not in the array or an index is out of bounds, the array yields the element $\emptyset$. Furthermore, we denote the size of any array A by $|A|$.

The associative array C is used to manage and store the loyalty points and auxiliary data of any scheduled user u created by F. The array S holds the serial numbers of any scheduled user u created in the simulation, i.e., $S[u]$ yields u's serial number s. For each serial number s created in the simulation, the associative array R holds a simple array of $m(k) + 1$ values used to randomise the output of the hash oracle $h(\cdot)$. Thus, R can be seen as a matrix with $m(k) + 1$ columns and an extendable number of rows. The simple indexed array X is used to store signatures on the value x.

Line 3 assigns x to the first empty index of X, i.e., x is appended to the array. Next, the simulation is started by running F with the input values (n, e) from the iRSA[m] problem. Lines 4–32 show B's overall behaviour during the simulation.

The construction of the hash oracle $h(\cdot)$ is given in lines 5–9. First, it is checked if the serial number s has been submitted to $h(\cdot)$ before. If not, i.e., $R[s] = \emptyset$, a new random value r_s is uniformly chosen and a new array is created which holds r_s raised to its e^i-th powers, where $0 \leq i \leq m(k)$. By $R[s][i]$ we refer to the i-th element of the array stored under name s in the associative array R, i.e., $R[s][i]$ yields $r_s^{e^i}$. Finally, $xr_s^{e^m}$ is returned to F as the result of $h(\cdot)$ — the reason for the construction of this result will become clear below. Note that, in the simulation, the forger F only sees a uniformly distributed random value returned to it as the result of querying $h(\cdot)$ with s and hence, it will not notice that x is embedded in the output.

Algorithm $B^{(\cdot)^d \bmod n}(n, e, x, m(k)+1)$:

[1] Initialise associative arrays C, R and S to empty;
[2] Initialise array X to empty;
[3] $X[0] := x$;

[4] Run F on input (n, e) and reply to its (oracle) queries as follows:

[5] — If F invokes $h(\cdot)$ with input s then
[6] If $R[s] = \emptyset$ then
[7] Choose $r_s \in_R \mathbb{Z}_n^*$;
[8] $R[s] := (r_s^{e^0}, r_s^{e^1}, \ldots, r_s^{e^{m(k)}})$;
[9] Return $X[0] \cdot R[s][m(k)]$ to F;

[10] — If F creates a scheduled user u then
[11] Choose $s \in_R \mathbb{S}$;
[12] $S[u] := s$;
[13] Invoke $h(\cdot)$ with s;
[14] Assign its response to H;
[15] $C[u][0] := (\emptyset, H)$;

[16] — If F tells scheduled user u to step its counter then
[17] $s := S[u]$;
[18] $l := |C[u]| - 1$;
[19] If $l = |X| - 1$ then
[20] Invoke oracle $(\cdot)^d \bmod n$ with input $X[l]$;
[21] Assign its response to $X[l+1]$;
[22] Choose $b \in_R \mathbb{Z}_n^*$;
[23] $t := b^e \cdot X[l] \cdot R[s][m(k) - l]$;
[24] $v := b \cdot X[l+1] \cdot R[s][m(k) - (l+1)]$;
[25] $C[u][l+1] := (b, v/b)$;
[26] Return (t, v) to F;

[27] — If F corrupts scheduled user u then
[28] Return $(S[u], C[u])$ to F;

[29] — If F runs the issue protocol with input t then
[30] Invoke oracle $(\cdot)^d \bmod n$ with input t;
[31] Return its response to F;

[32] Until F halts with some output (s, l, z) or *failure*;

[33] If F returned *failure* or $R[s] = \emptyset$ then
[34] Return *failure*;

[35] $y := z \cdot R[s][m(k) - l]^{-1}$;
[36] For $i = l, \ldots, m(k)$ do
[37] Invoke oracle $(\cdot)^d \bmod n$ with input y;
[38] Assign its response to y;
[39] Return y;

Figure 5.7: *Construction of algorithm B for solving iRSA[m]*

In lines 10–15 the forger creates a new scheduled user u. For u, a serial number is randomly and uniformly chosen (line 11) and stored in S using u as an identifier (line 12). Next, s is being 'hashed' and $h(\cdot)$'s output is assigned to H, shown in lines 13 and 14, respectively. At the end, a simple indexed array, $C[u]$, is created by initialising its first entry $C[u][0]$. Every entry $C[u][i]$ of the array $C[u]$ will hold a pair (b_i, c_i), where b_i is the blinding factor used in the issue protocol that resulted in the creation of counter value c_i. Since no blinding factor is needed to create the initial counter value c_0, b_0 is left empty.

If F tells scheduled user u to step its counter, lines 16–26 are executed. In this function, the simulator creates a new counter value for u. For convenience, in lines 17 and 18, values s and l are assigned u's serial number and current number of points, respectively. Then, it is checked if u's current number of points is the highest of all scheduled users.[7] If so, a new signature needs to be generated because no previously stored signature can be reused in order to compute a new counter value for u. Therefore, the inversion oracle $(\cdot)^d \bmod n$ is called with input x^{d^l} (line 20), which is the currently highest signature on x, and its output is assigned to the next empty entry of array X (line 21). Thus, the entry $X[i]$ holds the i-th signature of x, i.e., $X[i] = x^{d^i}$, for $i = 0, \ldots, |X| - 1$.

In line 22, a blinding factor b is randomly chosen, as in any real run of the issue protocol. Then, a blinded counter value is computed as $t := b^e x^{d^l} r_s^{e^{m(k)-l}}$ (line 23), which according to the construction of $h(\cdot)$ is equal to $b^e \sigma^l(h(s))$. Afterwards, a valid response v from the virtual vendor is computed, i.e., $bx^{d^{l+1}} r_s^{e^{m(k)-(l+1)}} = t^d =: v$ (line 24). Note that $v = b \cdot \sigma^{l+1}(h(s)) = b \cdot \sigma^{l+1}(x) \cdot \sigma^{l+1}(r_s^{e^{m(k)}})$. From this equation, the purpose of the array $R[s]$ becomes apparent. Namely, we can disguise/reuse signatures x^{d^i}, for $1 \leq i \leq |X|-1$, as i-signatures for any value $h(s)$ without the need to actually call $(\cdot)^d \bmod n$ with $h(s), \sigma(h(s)), \ldots, \sigma^{i-1}(h(s))$ because $\sigma^i(x) = X[i]$ (line 21) and $\sigma^i(r_s) = R[s][m(k) - i]$ (line 8). Also note that $|X| - 1 = \max\{m_S\}$ for all scheduled users S.

Eventually, in line 25 the blinding factor b and the new counter value $\sigma^{l+1}(h(s))$ of user u are stored to have them available for F in case u is later being corrupted by the forger. The two values t and $v = t^d$ that F would have observed in a real issue protocol run are returned to F in line 26.

In lines 27–28, the code is given that provides F with all of scheduled user u's data and intermediate data that u would have received in real issue protocol runs. Therefore, F is given u's serial number, stored in $S[u]$, every blinding factor b_i, and

[7] The value l will never exceed $|X| - 1$, since it is always increased after the check and so is the array X, in case $l = |X| - 1$.

all intermediate counter values of u, both of which are stored in $C[u]$. That is, F receives the following 'internal' data of u: s, $\langle(-, h(s)), (b_1, \sigma(h(s))), (b_2, \sigma^2(h(s))), \ldots, (b_{|C[u]-1|}, \sigma^{|C[u]-1|}(h(s)))\rangle$. The blinded counter values $t_i = b_i^e \cdot \sigma^{i-1}(h(s))$ and the vendor's responses t_i^d are omitted, since F received them as output of the "step" query and they can be re-constructed from the returned data, anyhow.

When the forger interacts with the vendor, via a controlled user, B must simulate the vendor's behaviour in runs of the issue protocol, i.e., it has to respond to F's signing queries. This is shown in lines 29–31. Since F may use arbitrary values t in an issue protocol run (c.f. Figure 5.5), B has no idea how to interpret t and thus, can only invoke the inversion oracle with t and return its result to F. Note that such calls to $(\cdot)^d \bmod n$ add to F's global counter m_A.

It may seem odd that we simulate two different issue transactions, one for scheduled users (lines 27–28) and one for controlled users (lines 29–31). The distinction, however, is vital for the simulation, since we do not have the luxury of submitting every t submitted by F or by a scheduled user u to the signing oracle $(\cdot)^d \bmod n$ and then just return its response to F or record the result for u, respectively. If we would do this, we would quickly run out of oracle queries in case the forger attempts a pooling attack. To see this, suppose that F created two scheduled users which it stepped for $l \leq \lfloor m(k)/2 \rfloor$ times, each. According to the breaking condition for the loyalty system, F also succeeds if it is able to successfully mount a pooling attack, i.e., it comes up with $l+1$ (or more) points. If it succeeds after having made $2l$ oracle queries, it leaves us with $m(k) - 2l$ queries to $(\cdot)^d \bmod n$ which may not be enough to produce the remaining $m(k) - (l+1)$ signatures needed to solve iRSA[m]. Thus, we must be cautious when making queries to $(\cdot)^d \bmod n$. To counter the problem of running out of oracle queries, B is reusing results from queries to $(\cdot)^d \bmod n$ in cases where the forger steps a scheduled user. Specifically, whenever a scheduled user u's counter is stepped whose current number of points l does not exceed $|X| - 2$, we will find the value $\sigma^{l+1}(x)$ in the array X and hence, by also looking up appropriate values in $R[s_u]$, we can construct intermediate counter values up to and including $\sigma^{l+1}(h(s_u))$ without making additional oracle calls. In other words, since counter values of scheduled users share the same signature at their core, the solver B can generate, say, two distinct counter values $\sigma^i(h(s))$ and $\sigma^i(h(s'))$ of scheduled users u and u', respectively, for which it only needs i calls to $(\cdot)^d \bmod n$ for *both* counter values, instead of i calls for *each* counter value.

After the simulation, B checks in line 33 if the forger failed, in which case B fails too, or if the forger came up with an appropriate triple (detailed explanations are omitted here). However, there is still a small chance that the forger F *succeeds* and B *fails*. This may happen, if F did not invoke $h(\cdot)$, i.e., no entry $R[s]$ exists (c.f. line 6), and still managed to come up with a valid signature z — the forger

may have simply guessed s and its guess is correct with probability $1/n$ [8]. If F had not used $h(\cdot)$ for its result, B cannot turn the resulting triple in a solution for iRSA[m] and hence, B fails.

Eventually, if F *succeeds* and used $h(\cdot)$ in its results, it comes up with a triple (s, l, z) representing a serial number s, a counter l, and an l-signature satisfying $z = \sigma^l(h(s))$, where $0 \leq max\{m_S\}+m_A < l \leq m(k)$ (see line 32).[9] Since $\sigma^l(h(s)) = h(s)^{d^l} = (xr_s^{e^{m(k)}})^{d^l} = x^{d^l} r_s^{e^{m(k)-l}} = z$ we get an l-times signature, y, for x by computing $y = z(r_s^{e^{m(k)-l}})^{-1} = x^{d^l}$. Clearly, if F succeeded, it used up less than l queries to $(\cdot)^d \bmod n$, thus, in lines 36–38, the remaining $m(k) - l + 1$ signatures for x are computed by iteratively querying the signing oracle $(\cdot)^d \bmod n$. Finally, B outputs $y^{d^{(m(k)-l+1)}} = (x^{d^l})^{d^{(m(k)-l+1)}} = x^{d^{m(k)+1}}$, as desired. Since B almost always succeeds whenever F succeeds, we have

$$Pr[(B^{(\cdot)^d \bmod n}(n, e, x, m(k) + 1))^{e^{m(k)+1}} = x \bmod n] = \epsilon - \frac{1}{n}$$

and hence a polynomial-time solution for the iRSA[m] problem, which contradicts our assumption that iRSA[m] is hard. ■

In addition to the proof of security in the random oracle model above, in the following, we provide some more arguments which indicate that breaking our loyalty system is indeed hard. Using these additional arguments one need not rely solely on the conjectured intractability of the iRSA[m] problem. Consider a forger who repeatedly applies the public exponent e to some $x \in \mathbb{Z}_n^*$ until he obtains $h(\tilde{s}) = x^{e^v} \in \mathbb{Z}_n^*$ for some $\tilde{s}$ and some v, if any, in order to forge $\langle \tilde{s}, v, \sigma^v(h(\tilde{s}))\rangle$. However, since $x^{e^{v-1}} = \sigma(h(\tilde{s}))$ this would provide a method for generating a valid pair $\langle \tilde{s}, \sigma(h(\tilde{s}))\rangle$ without knowing d, and thus, a method to break the security of Chaum's system. Since this attack is assumed to be infeasible, it is also infeasible to generate valid triples in this way.

Another aspect to be considered is related to the fact that exponents can be reduced $\bmod \lambda(n)$, where $\lambda(\cdot)$ denotes the *Carmichael* function which gives the smallest number $r > 0$ such that $a^r = 1 \bmod n$ for all $a \in \mathbb{Z}_n^*$ [Yan02]. This means that for $s \in \mathbb{Z}_n^*$ there exists a number $w > 0$ with $e^w = 1 \mod \lambda(n)$ such that $h(s)^{e^w} = h(s)$ [WS79], or $h(s)^{e^z} = h(s)^{d^{w-z}}$ for $z \geq 0$. In words, applying e iteratively z times to $h(s)$ yields the same result as applying d iteratively $w - z$ times to $h(s)$. It may be argued that this can be exploited by a forger. But for a successful attack, a forger has to find an appropriate w. If a forger would know

[8] For instance, F may have chosen $z \in_R \mathbb{Z}_n^*$, computed $y := z^{e^l} \bmod n$, and guessed s such that $y = h(s)$.

[9] The last inequality is due to the maximum number of redeemable points in any one redemption.

an efficient method to find w, then he would be able to carry out the so called *iterated-encryption attack* on the RSA cryptosystem which would eventually reveal d. In [Mau95], it is shown how system parameters p, q can be chosen such that the iterated-encryption attack is thwarted. Hence, using this construction, forging counter values by repeatedly applying e to $h(s)$ is infeasible.

Unforgeability and Pooling Prevention. From the security proof and the arguments above, we conclude that, under the assumption that iRSA is hard, it is infeasible to forge or pool loyalty points obtained by running the issue protocol. In case vendors want to allow pooling, they just have to remove the restriction to send a single counter in the redeem protocol and allow for l ones, i.e., customers may send

$$\langle s_1, m_1, \sigma^{m_1}(h(s_1), s_2, m_2, \sigma^{m_2}(h(s_2), \ldots, s_l, m_l, \sigma^{m_l}(h(s_l)\rangle.$$

Double-spending detection. Since a malicious customer may try to redeem copies of loyalty point counters, the vendor stores all serial numbers in a local database and compares each s from a newly submitted counter value to the database's. If s is already stored in the database then the counter value has been double-spent and will not be accepted. The probability that two customers pick the same serial number s is $1/n$ which can be neglected for a sufficiently large security parameter k.

5.6.6 Efficiency

In our scheme, loyalty points awarded to a customer are represented by a counter. In contrast to a token-based approach, the costs in the issue protocol are no longer linearly related to the number of loyalty points. Instead, the size of the data that has to be transferred and stored is constant for each purchase, regardless of the number of points awarded. This means, that the costs, for the vendor, regarding storage size only grow linearly with the number of redemptions. Furthermore, the number of signatures that need to be generated by the vendor in any one purchase is constant, i.e., one m-signature per purchase. In a token-based approach each token, i.e., loyalty point, requires its own serial number while in the counter-based approach the number of serial numbers is independent of the number of loyalty points. The redeem protocol of the counter-based approach is also more efficient than the redeem protocol in a token-based variant. In any one run of the redeem protocol, the size of the transferred data is constant in the number of redeemed points and at most two modular exponentiations are needed to verify any counter value. Storage costs for serial numbers are only linear in the number of redeem protocol runs in contrast to linear costs in loyalty points within a token-based approach.

Instance Generator $\mathbf{IGen}^{\mathrm{CDH}}(k)$:

$(g, G) := KeyGen^{\mathrm{DH}}(k)$
$a, b \in_R \mathbb{Z}^*_{ord(G)}$
$y_1 := g^a$ (in G)
$y_2 := g^b$ (in G)
Output (y_1, y_2, g, G).

Task: Given y_1, y_2, g, G,
compute g^{ab} (in G).

(a) *Computational Diffie-Hellman problem (CDH)*

Instance Generator $\mathbf{IGen}^{\mathrm{DDH}}(k)$:

$(g, G) := KeyGen^{\mathrm{DH}}(k)$
$a, b, c \in_R \mathbb{Z}^*_{ord(G)}$
$y_1 := g^a$ (in G)
$y_2 := g^b$ (in G)
$y_3 \in_R \{g^{ab}, g^c\}$
Output (y_1, y_2, y_3, g, G)

Task: Given y_1, y_2, y_3, g, G,
decide if $y_3 = g^{ab}$ (in G).

(b) *Decisional Diffie-Hellman problem (DDH)*

Figure 5.8: *Diffie-Hellman problems*

5.7 A Counter-based Loyalty System based on DH

Another variant of the loyalty system is based on a different cryptographic assumption, namely that the so called Diffie-Hellman problem is hard. Again, the loyalty scheme consists of two protocols, the *issue* and *redeem* protocol. Both protocols involve two parties, the vendor and the customer, as before, and make use of an anonymous counter. The goal of our construction is to achieve the unlinkability of issue and redeem and also the unlinkability of any two issue transactions and any two redeem transactions.

5.7.1 System Setup

The system is set up as follows. The vendor chooses an appropriate cyclic group G, along with a generator g for this group. The group G must be chosen such that the decisional Diffie-Hellman problem (DDH) can be decided efficiently but for which the computational Diffie-Hellman problem (CDH) is presumed to be hard — CDH and DDH are briefly stated in Figures 5.8a and 5.8b, respectively. In Section 5.7.6, we will provide suitable choices for G, but for the following sections the exact nature of G is of no concern.

The order of the group G should be a sufficiently large prime q for which we will later specify another condition, namely, that $q - 1$ does not have small prime factors (see Section 5.7.6). From now on, unless otherwise noted, it is understood that all computations are done in the group G.

Customer		Vendor
choose $s \in_R \mathbb{S}$		choose $v \in_R \mathbb{Z}_q^*$
compute $c_0 := h(s)$	$\xleftarrow{\langle g,V \rangle}$	publish $g, V := g^v$

Figure 5.9: *Initialisation*

The vendor randomly selects a value $v \in \mathbb{Z}_q^*$ and computes $V = g^v$. He publishes (g, V) (and a description of the group G) as his public key and keeps v private. The customer chooses a random serial number s from some finite set $\mathbb{S}$. This serial number will act as an identifier for her future loyalty points, and serial numbers should be chosen such that collisions do not occur. After that, the customer binds to s by computing her initial counter value $c_0 := h(s)$, where $h(\cdot)$ is some cryptographic hash function mapping to the group. This hash function should be specified and published by the vendor, too. The initialisation process is depicted in Figure 5.9.

5.7.2 Protocols

Issue. When the customer is to be credited with a loyalty point, she randomly chooses r_i from $\mathbb{Z}_q$. Then, she blinds her current counter value c_{i-1} by computing $b_i := c_{i-1} g^{r_i}$ and sends the blinded counter b_i to the vendor. The vendor raises b_i to the v-th power and returns the result. Next, the customer computes the unblinding factor V^{-r_i} and subsequently derives $b_i^v V^{-r_i} = c_{i-1}^v$. After that, the customer verifies that the vendor has sent a correct value. To do so, she checks whether (c_{i-1}, V, c_{i-1}^v) is a valid DH triple by running the efficient test for the group G.[10] Note that, in general, this validity test is intractable for groups like $\mathbb{Z}_p^*$. If the verification succeeds then the customer sets $c_i := c_{i-1}^v$ and stores (i, c_i). The issue protocol is shown in Figure 5.10.

Redeem. If the customer has reached some redeeming threshold, i.e., has gathered enough points to hand them in for a reward, she may execute the redeem protocol shown in Figure 5.11. There, the customer sends her serial number s, the number of collected loyalty points n, and the counter value c_n. The vendor validates this triple by checking that c_n is in fact $c_0^{v^n}$ for $c_0 = h(s)$.

In order to prevent customers from redeeming the same counter value more than once, the vendor checks if s is already stored in his database of redeemed serial numbers. If this is not the case the vendor stores the new serial number s

[10] To see why this is should be a DH triple, note that $V = g^v$ and c_{i-1} can be written as g^x for some $x \in \mathbb{Z}_q^*$, as g is a generator of G.

Customer		Vendor
choose $r_i \in_R \mathbb{Z}_q$;		
compute $b_i := c_{i-1} g^{r_i}$;	$\xrightarrow{b_i}$	
	$\xleftarrow{b_i^v}$	compute b_i^v;
compute unblinding factor V^{-r_i};		
unblind b_i^v		
$b_i^v V^{-r_i} = c_{i-1}^v g^{r_i v} V^{-r_i}$		
$= c_{i-1}^v g^{r_i v} g^{-r_i v}$		
$= c_{i-1}^v$;		
verify (c_{i-1}, V, c_{i-1}^v) DH triple?;		
set $c_i := c_{i-1}^v$;		

Figure 5.10: *Issue protocol in the customer's i-th purchase*

Customer		Vendor
	$\xrightarrow{\langle s, n, c_n \rangle}$	
		verify $c_n \stackrel{?}{=} h(s)^{v^n}$;
		s not yet stored in database?;
		grant reward if verification successful;

Figure 5.11: *Redeem protocol*

— alternatively, the serial number's hash value $h(s)$ may be stored and checked. Eventually, if all checks are completed successfully, the vendor sends the reward to the customer.

Note that if the serial numbers would be used directly, i.e., without applying the hash function or some similar measure, then the vendor might be easily tricked into accepting a forged counter value. Specifically, given two correct counter values $c_n = s^{v^n}$, $c'_n = (s')^{v^n}$ for some n it is easy to derive a third counter value $c_n c'_n = (ss')^{v^n}$ for serial number ss'.

5.7.3 Attack Model

The attack model for the DH-based loyalty system is the same as for the RSA-based system from Section 5.6. Although, the underlying cryptographic assumptions are different, the goal of an attacker is still to forge or pool loyalty points and hence, everything said in Section 5.6.3 also applies here. Consequently, in the following

we use the same terminology and similar ideas as in Sections 5.6.3 and 5.6.5 when discussing the security of the DH-based system.

5.7.4 Privacy

The DH-based loyalty system's privacy property follows from the same arguments as the RSA-based system's privacy property, as discussed in Section 5.6.4. In the DH-based protocol, the blinding and signing functions are necessarily different from the counter-based RSA protocol. However, their properties are essentially the same, and hence the same arguments apply with respect to unlinkability among issue transactions, among redeem transactions, and between issue and redeem transactions. We omit any further discussion.

5.7.5 Security

To claim security properties of our loyalty system we first have to specify the attack scenario and successful attacks. Afterwards, we show that our system achieves the desired properties.

We remark that the vendor in our system can easily thwart double spending by keeping track of used serial numbers s and by rejecting claims for previously submitted ones. As for the unforgeability and pooling prevention we prove security of our scheme based on the intractability of a new problem, called the incremental Diffie-Hellman (iDH) problem. This problem is related to the classical Diffie-Hellman problem as well as to the previously proposed one-more RSA and one-more Discrete Logarithm problems for proving Chaum's blind signature and its discrete-log variant to be secure [BNPS01, Bol03]. Although we were unable to reduce some standard cryptographic problem to this new problem, our reduction enables us to investigate the security of our system by considering a pure mathematical problem and hiding the details of the protocol. Indeed, we will also provide some discussion about the hardness of the iDH problem below.

The incremental Diffie-Hellman problem. The incremental Diffie-Hellman problem is to find $m \geq 1$ and $g^{v^{m+1}}$ for given group elements g and $V = g^v$ (where v is unknown). To facilitate the task one is allowed to query a special Diffie-Hellman oracle $\mathsf{DH}_{g,V}(\cdot)$ computing X^v for inputs X. Yet, the condition is that the oracle can only be queried at most $m-1$ times, e.g., to compute g^{v^3} from g, g^v one may make a single call to the oracle. Specifically:

Definition 7 (incremental Diffie-Hellman problem) *Let g be a generator of a group of prime order q and $V = g^v$ be a random element in this group. Given g, V and access*

to an oracle $\mathsf{DH}_{g,V}(X) = X^v$ *the incremental Diffie-Hellman (iDH) problem is to come up with an element* Z *and an integer* $1 \leq m < \mathrm{ord}_{\mathbb{Z}_q^*}(v) - 1$ *such that*

$$Z = g^{v^{m+1}}$$

and such that the oracle $\mathsf{DH}_{g,V}(\cdot)$ *has been queried at most* $m-1$ *times.*

The upper bound on the integer m rules out trivial solutions. Else, $Z := g$ would for example be a correct claim for $m = \mathrm{ord}_{\mathbb{Z}_q^*}(v) - 1$ because this would yield $g^{v^{m+1}} = g^{v^{-1+1}} = g = Z$ and another correct claim would be any multiple m of the order $\mathrm{ord}_{\mathbb{Z}_q^*}(v)$ of v in $\mathbb{Z}_q^*$ because $g^{v^{m+1}} = g^v = Z$. For our scheme, we therefore choose a sufficiently large order for v; see Section 5.7.6 for details.

Unforgeability and pooling prevention. The incremental Diffie-Hellman problem reduces to the security of our scheme in the random oracle model. To show this we present an iDH algorithm that uses a successful forger for our loyalty system as a subroutine. In order to use the forger in this way, the iDH algorithm will set up a "virtual" environment for the forger by impersonating the vendor and inserting the input for the iDH problem. As the experiment looks like a real interaction with the vendor from the forger's perspective, the forger will claim more points than issued in the experiment if she would do so in an actual attack. Any solution in the experiment will immediately give a solution for the iDH problem, and we conclude that each forger for our protocol must implicitly solve the iDH problem.

In the experiment, we assume the same kind of scheduled and controlled users as described in Section 5.6.3 and used before in Section 5.6.5. Also in analogy to Section 5.6.5, we model the hash function $h(\cdot)$, mapping serial numbers to group elements, as a random oracle [BR93]. That is, we assume that $h(\cdot)$ acts as a random function: it maps inputs to uniformly and independently distributed group elements, repeating answers for previously queried inputs. Note that the idealised random oracle model merely provides some heuristic evidence that the scheme is indeed secure; refer to [CGH98] for a discussion. Therefore, in Section 5.7.6 we also present a modification which completely forges random oracles but which essentially preserves the efficiency (with only a negligible loss in the initialisation protocol).

We next specify the construction of the iDH algorithm from an arbitrary forger. For this, the iDH algorithm first tries to guess the maximum $M_S := \max\{m_S\}$ of issued points for scheduled users in the upcoming experiment. This value is usually bounded by a parameter M representing the system's maximum of redeem points. Instructively, think of M as 10 or $1,000$.

To guess $M_S = \max\{m_S\}$ the iDH algorithm picks a uniformly distributed value between 0 and M. The forger's view in the following simulation is independent of this choice, and the iDH algorithm thus hits the right value with probability

$1/(M+1)$. If, on the other hand, the guess later turns out to be incorrect the iDH solver will stop with *failure* instead. The overall success probability of the iDH algorithm therefore decreases by a factor of $1/(M+1)$ compared to the forger. From now on, we condition on the event that the iDH algorithm selects the correct M_S.

Next, we describe the simulation of the forger. The iDH algorithm is given g and V, has access to the oracle, and has predicted M_S. It first computes $g^{v^2}, g^{v^3}, \ldots, g^{v^{M_S+1}}$ by iteratively querying the oracle, starting with $V = g^{v^1}$. This can be done with M_S queries. It next starts the simulation of the forger by providing g, V as the public key of the vendor. The emulation proceeds as follows:

- Whenever the forger queries the hash function $h(\cdot)$ about some serial number s, i.e., adds another *controlled user* to the system, then the iDH algorithm chooses $w_s \in \mathbb{Z}_q^*$ at random and returns V^{w_s} (or returns the previously given answer, if this serial number has been queried before).

- If the forger initiates the issue protocol for a controlled user and submits a value b to the virtual vendor then the iDH algorithm calls the DH oracle to derive b^v and answers on behalf of the vendor with this value. This actions increases the forger's counter of oracle calls, m_A, by one.

- If the forger adds another *scheduled user* to the system then the iDH algorithm chooses a number s and sets $h(s) := V^{w_s}$ for a random value $w_s \in_R \mathbb{Z}_q^*$ (or returns the previously given answer if this serial number has appeared before). The iDH algorithm from now on impersonates this scheduled user with values s and $c_0 = h(s) = V^{w_s} = g^{w_s v}$.

- If the forger asks a scheduled user to step the counter then the iDH solver fetches the current counter value $c_{i-1} = g^{w_s v^i}$ and runs a simulation of the issue protocol:

  - Take $g^{v^{i+1}}$ from the pre-computed list of powers. Note that, by assumption, i does not exceed the correct guess M_S and therefore $g^{v^{i+1}}$ must be in this list.
  - On behalf of the customer select $r_i \in_R \mathbb{Z}_q$ at random and compute $b_i := c_{i-1} g^{r_i}$.
  - On behalf of the vendor compute V^{r_i} and $(g^{v^{i+1}})^{w_s}$ and reply with

    $$b_i^v = V^{r_i} (g^{v^{i+1}})^{w_s} = V^{r_i} c_{i-1}^v$$

    Store $c_i = g^{w_s v^{i+1}}$ and r_i in the name of the customer. Note that all the values, including c_i and r_i, are distributed identically to an execution between a scheduled user and the vendor in an actual attack.

- If the forger corrupts a scheduled user the iDH algorithm hands over all the previously stored values on behalf of this customer and stops impersonating this user.

When the forger finally redeems a counter value Z and $m \geq 1$ for some serial number s then the iDH algorithm computes $w_s^{-1} \bmod q$ and outputs $Z^{w_s^{-1}}$ and m and stops.[11]

Note that the answers of the iDH algorithm are identical to those of the genuine vendor and the simulated hash function evaluation yields uniformly distributed values like the random oracle. This means that the view of any forger in the experiment is the same as in an actual attack, and if the forger is able to redeem more points in reality then it succeeds in the simulation with the same probability (under the condition that the iDH solver has guessed M_S in advance).

Finally, it remains to be shown that the construction above turns any forgery in the experiment into a solution to the iDH problem. For this note that, for a successful redemption,

$$Z^{w_s^{-1}} = \left(g^{w_s v^{m+1}}\right)^{w_s^{-1}} = g^{v^{m+1}}$$

Furthermore, $m > \max\{m_S\} + m_A$ which implies

$$m \geq \max\{m_S\} + m_A + 1$$

Since the iDH algorithm has queried its oracle exactly $\max\{m_S\} + m_A$ times this means that $Z^{w_s^{-1}}$ and m constitute a valid solution to the iDH problem. Therefore, we have presented an algorithm solving the iDH problem whenever the forger succeeds and the algorithm's initial guess is right. ■

As for the exact security of our loyalty system we note that, according to common practice, the running time of the attacker comprises its own steps and the ones of honest parties during the attack. But then the running time of the derived algorithm iDH differs only marginally from the one of the attacker, i.e., the iDH algorithm initially computes the powers g^{v^i} via the oracle and also performs some additional computations when simulating answers of the vendor. Our reduction hence shows that if the adversary breaks the loyalty system in t steps with probability ε, then there is an algorithm solving the iDH problem in time $t' \approx t$ and with probability $\frac{1}{M+1}(\varepsilon - \frac{1}{q-1})$.

[11] There is a very small probability that the forger successfully claims a counter value for a number s that has not been passed to the hash function before. However, this probability is equal to $1/(q-1)$ and we thus neglect it for the analysis.

On the hardness of the iDH problem. It remains to argue the intractability of the iDH problem. We are not aware of any reduction from well-established problems like the Discrete Logarithm problem or the canonical Diffie-Hellman problem. Still, we give a brief discussion about the intractability of the iDH problem and its relationship to similar problems.

The algorithm's task is to find some $m \geq 1$ and $g^{v^{m+1}}$ after having made at most $m-1$ calls to the oracle. *Under the condition that the algorithm never queries the oracle* the canonical Diffie-Hellman problem can be reduced to this problem and our problem is hence believed to be infeasible. Namely, without the help of the oracle the algorithm computes a variant of the Diffie-Hellman function, $g^v \mapsto g^{v^m}$ for unknown v and some $m > 1$. This function, however, has the same power as the classical DH function for m's of order $O(\sqrt{\log q})$, refer to [MW96, Kil01].

As for the power of the oracle queries, note that the iDH problem is related to another problem from computational complexity. Namely, it is believed that computation of powers V^{2^m} requires m *sequential* squarings and that there is no efficient improvement allowing a faster parallel computation. This problem has been applied in cryptography before to derive protocols with critical time release properties [BN00].

In our case the constant 2 in the computation of V^{2^m} is replaced by the unknown value v, even hampering the task. Hence, any successful iDH algorithm that, in addition to the oracle calls, only performs operations which are independent of the input would give rise to a new algorithm deriving powers V^{v^m} with less than m exponentiations (using some pre-processing).

In conclusion, we cannot prove that the iDH problem is as hard as, say, the computational Diffie-Hellman problem. However, the discussion above indicates that straightforward algorithms for the problem do not work and that more sophisticated algorithms would be required to solve the problem — if it can be solved efficiently at all.

5.7.6 Efficiency and Implementation Issues.

To implement the protocol one has to pick an appropriate group G for which the *computational* Diffie-Hellman problem (CDH) is conjectured to be hard, while the *decisional* Diffie-Hellman problem(DDH) is easy. Such a group can be defined over a certain class of elliptic curves.

Elliptic curves provide an alternative to well-known groups based on modular arithmetic over the integers. Compared to cryptographic operations like RSA over $\mathbb{Z}_n^*$ or Diffie-Hellman over $\mathbb{Z}_p^*$ elliptic curves usually offer smaller key sizes at a comparable security level. Nonetheless, our motivation for basing our protocol on elliptic curves stems from a special property of some of these curves. Namely,

we deploy special elliptic curves for which CDH (see Figure 5.8a) is believed to be intractable, whereas DDH (see Figure 5.8b) is known to be easy. Such elliptic curves had been suggested in [Jou00, JN01] and immediately gained a lot of attention because of their usefulness for the design of cryptographic protocols, e.g., [BF01, BLS01, Bol03, Dod03].

The decision procedure for elliptic curves separating the computational and the decisional Diffie-Hellman problem is usually based on the so-called Weil or Tate pairing. These pairings can be carried out efficiently and allow to decide whether a given tuple constitutes a correct DH triple or not. We omit further technical details as they are irrelevant for the conceptual design of our loyalty system here. Nonetheless, we remark that such curves have already been investigated quite well, in particular with respect to

- appropriate choices of such groups in light of efficiency and security (note that the computational DH problem must still be intractable for the group) [JN01, BLS01];
- fast computation of the pairing functions [BF01, BKLS02, GHS02], i.e., fast verification of putative DH triples (g^a, g^b, g^c);[12]
- hashing into the curve [BLS01]; that is, how to define a hash function $h(\cdot)$ mapping bit strings to the group.

Since we merely apply these properties we refer to these works for details. For an introduction to elliptic curves see [Men93].

To implement the protocol one has to pick an appropriate elliptic curve with a pairing function and define a hash function mapping strings to random group elements. We refer to [JN01, BF01, BLS01, BKLS02, GHS02] for such choices.

It is not hard to see that we can eliminate the hash function (and the random oracle model in the security proof), if we let the vendor choose a random value c_0 for the customer in an initialisation step. If this initialisation step is also carried out anonymously then the customer's privacy will not be affected by this and unforgeability now follows from the iDH problem alone. Of course, the same procedure can be used in the RSA variant of the loyalty system to eliminate the random oracle.

The variant with the vendor choosing the serial number can also avoid accidental collisions, which may happen when customers select the serial numbers, even if the collision probability is very small. Unfortunately, this variant has some drawbacks as well. First, it requires an additional interaction to get a new serial number for initialising a new counter. Second, requesting a serial number might be correlated with a purchase / issue transaction and this may allow the vendor to link the redeem

[12] We use the multiplicative notation for the elliptic curve generated by g.

transaction with the counter's first issue transaction. Furthermore, in this variant the vendor learns that no issue transaction prior to the creation of the serial number is related to the user. In summary, the creation of serial numbers by the vendor has some disadvantages regarding privacy. Another drawback is that a malicious customer could repeatedly request serial numbers from the vendor without really using them. Since each serial number can only be issued once, this may lead to an unnecessary waste of serial numbers.

Recall that we also require the order of the vendor's secret v in the multiplicative group $\mathbb{Z}_q^*$ to be quite large. This can be accomplished by letting $q-1$ have only large prime factors. Specifically, for $q \approx 2^{160}$ it suffices to let $q-1$ consist only of prime factors larger than 40 bits. Then any element $v \neq 1$ has order at least 2^{40} in $\mathbb{Z}_q^*$ which is sufficient for all practical purposes. Since $g^{v^m} = g^{v^{m+i\cdot \mathrm{ord}_{\mathbb{Z}_q^*}(v)}}$ for any $i \geq 0$, an adversary may claim higher counter values $m + i \cdot \mathrm{ord}_{\mathbb{Z}_q^*}(v)$ instead of m. But this can be tackled by defining a maximum counter value which is obviously smaller than $\mathrm{ord}_{\mathbb{Z}_q^*}(v)$, i.e., larger counter values will not be accepted in the redeem protocol. The vendor may publish this bound on the maximum number of points as part of the system parameters.

We address the vendor's effort for the verification in the redeem protocol. Note that the vendor first calculates $w := v^m \bmod q$ over $\mathbb{Z}_q^*$ and then $h(s)^w$ in the elliptic curve and finally compares it with the given c_m. Altogether these are only two exponentiations, and thus improves efficiency over the verification of m blind signatures in the token-based case. To decrease this effort further the vendor can also pre-compute and store powers of the universal value v, especially if all customers are likely to claim points for a fixed value, like $m = 10$. Verification of a claim then essentially boils down to a single exponentiation.

5.8 Comparison

In the previous sections, we have presented three schemes, all of which allow to implement a privacy-friendly loyalty system. The schemes differ with respect to their cryptographic assumptions and their time and space requirements. Table 5.1 shows a comparison chart where the protocols of each scheme are roughly compared along various dimensions.

For the comparison, we assume that a customer obtains m loyalty points in a single issue phase and redeems the same number of points in a single redeem phase. Furthermore, we assume that exponentiations roughly require the same effort in each of the underlying mathematical structures. To be brief, we do not consider other factors, such as multiplications or hashing operations, which also have an impact on the systems' overall performance.

Protocol	Party	Measure	Token-based	Counter-based (RSA)	Counter-based (DH)
Issue	*Customer/ Vendor*	#Protocol Runs	$m^\dagger$	1	m
	Customer	Blinding (#Exponentiations)	m	m	m
		Verification (#Exponentiations)	m	m	m^*
		Storage (Serial Number(s) + Points)	$2m$	2	2
	Vendor	Signing (#Exponentiations)	m	2	m
Redeem	*Customer/ Vendor*	#Protocol Runs	$m^\dagger$	1	1
	Vendor	Verification (#Exponentiations)	1	2	2
		Storage (Serial Numbers)	m	1	1
		Pooling prevention possible?	No	Yes	Yes

†Protocol runs can be executed in parallel.
*Disregarding potential costs for testing DDH.

Table 5.1: *Comparison chart for issuing/redeeming m loyalty points*

Issue. In the issue phase, m runs are required for the token-based and the counter-based DH variant of the loyalty system, respectively, as only one point can be issued in each run. However, in the token-based variant, it is possible to run multiple issue protocols in parallel, i.e., m blinded tokens can be sent (and retrieved) in one go. Conversely, the issue protocol of the DH version cannot be sped up that way because points have to be added to the previously issued counter value and hence, the protocol can only be carried out sequentially. The counter-based RSA variant requires only one run and hence, is more efficient than its DH counterpart.

In the row for the issue protocol, the numbers shown for the protocols are the total numbers, i.e., the number of operations/space required to obtain m loyalty points. In all schemes, the customer employs exponentiations linear in the number of loyalty points to be issued, in order to blind her token/counter value. All schemes *a priori* require the same number of exponentiations. However, the effort for carrying out these exponentiations may vary, as optimisation techniques might be employed that exploit the different characteristics of the computations. For instance, in the token-based variant, m different bases are independently raised to the

public exponent e and in the counter-based RSA variant, sequential exponentiations are computed. Irrespective of such considerations, the effort for verification is the same as for the blinding operation because essentially the same computations are made in order to verify the signature(s) of the loyalty points. Note that the table's entry for the DH variant may need to be amended with costs for deciding whether a given triple is a DDH triple or not, if such tests require exponentiations.

The storage required for the counter-based loyalty systems is constant, as only one storage unit is required for the serial number and another one for the m-signature — one may also store the number m but this is not strictly necessary. Conversely, the token-based variant requires 2 storage units per issued point. More precisely, it requires 2 units per *additional* point, in contrast to the counter-based variants whose storage needs to be allocated just once, i.e., when the first point is issued.

For the signing, the vendor needs m exponentiations in the token-based as well as in the counter-based DH variant, which is a consequence of the number of protocol runs required. In the counter-based RSA variant on the other hand, only a constant number of exponentiations is needed, i.e., 2 exponentiations. This is due to the vendor's knowledge of the group's order, i.e., he is able to compute $y := d^m \bmod \mathrm{ord}(\mathbb{Z}_n^*)$ and then $t^y \bmod n$. Hence, he does not need to employ sequential exponentiations. Note that a vendor employing the DH variant would have the knowledge to do the same, but he has to run the protocol m times, employing one signing operation in each run, because the customer cannot unblind an m-signature, for $m > 1$.

Redeem. In the redeem phase, the counter-based variants just need one run of the protocol, i.e., one roundtrip. The token-based variant's redeem protocol, however, can be executed in parallel, analogously to the issue protocol, and hence, can also be reduced to a single roundtrip. Still, the counter-based variants have an advantage over the token-based system, as the amount of data that needs to be send to the vendor is constant in the number of loyalty points, i.e., 2, as opposed to the linear-sized amount of data in the token-based system.

The number of exponentiations required to verify the loyalty systems' points is constant in all three cases, with the token-based variant needing just a single exponentiation (see Equation 5.1).

As vendors need to keep track of already spent serial numbers, the token-based variant uses up the most space, again, because every point uses up a serial number. The counter-based variants only have constant storage costs with respect to a run of the redeem protocol, as they just need one serial number.

Finally, vendors have the option to allow or disallow pooling of loyalty points in the counter-based variants, where there is no such choice in the token-based variant.

5.9 Prototype

The counter-based loyalty systems proposed here are especially suitable for Internet applications, like those in the World Wide Web, as they follow a simple request-response scheme, just as HTTP does. This gives developers a lot of freedom for choosing the technologies to implement the customer and the vendor part of the system, i.e., the client and server side, respectively. Examples of such choices are Flash, Active X, and Java Applets for the customer part and CGI, PHP, ASP, JSP, and Java Servlets for the vendor part.

For the customer side it is necessary to have access to local storage, as we need to store the customer's points. Apart from that, there is no other strong functional requirement on the client technology and there is also none for the server technology. For our own prototype implementation, we have chosen Signed Java Applets for the client side because Java Applets can be run in all major browsers, e.g., Internet Explorer and the Mozilla browser family, and in moderately recent versions of Java cryptographic primitives are provided. In addition, it is possible to determine the authenticity of the Applet's code. The server side was implemented using JSP and Java Servlets.

We have implemented the RSA variant of the counter-based scheme, as it is superior to the two other variants (see Section 5.8). The implementation itself is relatively straightforward and thus, we will not go into details here. Instead, we want to highlight different options to operate the loyalty system.

In our model so far, we have seen a customer talking only to a single vendor. However, it should, of course, be possible for the customer to collect points from different vendors and also to manage these different points. In Figure 5.12, this is shown for vendors B and C which independently issue and redeem loyalty points. That is, customers can collect points of *type* B which they can only redeem at *vendor* B, and likewise for C.

For vendors $A_1, \ldots, A_i, \ldots, A_n$, a different model is applied, which we have also implemented. In this model, customers may collect points of type A from any vendor A_i, $1 \leq i \leq n$. And likewise, they may spent their points of type A at any vendor A_i. That is, from the customers' point of view this group of vendors acts in concert. In practice, the vendors A_i need not know of each other. To allow this, we have introduced a third party, the operator, which is transparent to customers.

In the operator model, vendors do not actually issue or redeem points, they delegate these tasks to the operator. That is, any vendor A_i forwards a customer's issue or redeem message to the operator which increases counters and checks their validity. In short, the operator does everything that, e.g., vendors B and C do for their own points. This way, the private key for points of type A, used to increase customers' counters, need not be shared among the vendors A_i. However, in this

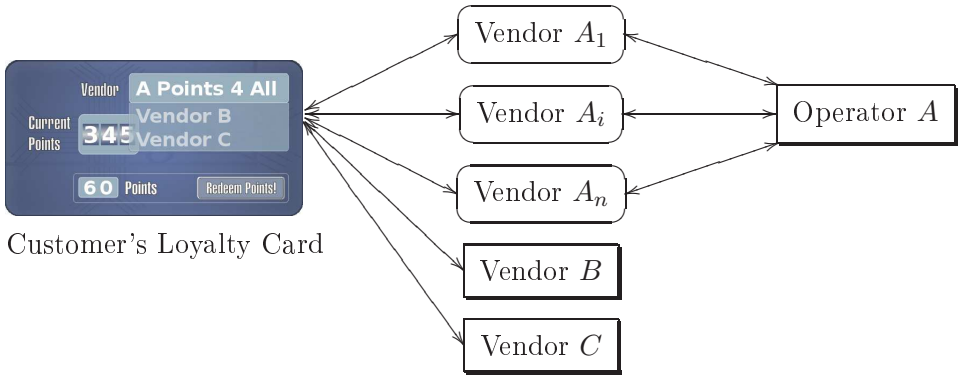


Figure 5.12: *A Customer's Relationships with different Vendors*

model some kind of bookkeeping is necessary, as some vendors may issue more points than they have to redeem, and vice versa, e.g., because the rewards of other vendors are more attractive than their own ones. Therefore, some kind of compensation mechanism will need to be established in order to balance the profits accompanying the issuance of points with the expenses accompanying the redemption of points. The exact nature of such a compensation mechanism is of no concern here, we just want to add that implementations for the vendor-to-operator interface may need to provide some kind of audit log, for both vendors and the operator, in order to have proof who asked for how many points to be issued or redeemed.

The card image of Figure 5.12 was actually taken from our prototype, however, the choice which type of points are to be received or to be spent is not left to the customer, as hinted by this illustration. Instead, the choice is made automatically by the client software depending on the public key advertised by the respective vendor. Note that vendors which have their own type of points, i.e., B and C, give their own names, "Vendor B" and "Vendor C", while a group of vendors uses a common name, "A Points 4 All", in order to make it clear to customers that the points of all vendors A_i go to the same counter.

5.10 Related Work

Much work has been done by economic and marketing experts in the field of loyalty systems, e.g., see [BKB00, SS97, DU97]. Furthermore, there has been lots of work stressing the importance of privacy for electronic commerce, e.g., see [HNP99]. A common goal of proposals for privacy enhancing systems in the area of electronic commerce is to prevent certain parties from linking activities of the same customer. In typical commercial relationships, there are many possibilities to link customer

transactions. For instance, in the area of payment systems, the unlinkability of withdrawal and deposit has been considered [Cha89, CPS96]. In [EKS02b], a solution to establish the unlinkability of the customer's search and order phases has been proposed (see also Chapter 4). In this context, we provide a solution to guarantee that unlinkability achieved by other techniques still holds when using a loyalty system. Other work regarding technical proposals for loyalty systems can be found in [Mah98]. In this work, an infrastructure based on smart cards is proposed which allows individuals to introduce their own currencies or loyalty systems. However, the authors do not deal with the problem of achieving privacy in loyalty systems. Another proposal for a loyalty system was presented in [WLT00]. In this work, the authors take the privacy aspect into account. However, the system was not designed to provide unlinkability of transactions, as it is based on pseudonyms and thus provides a weaker form of privacy.

5.11 Conclusion

We have presented privacy-friendly loyalty systems that do not allow vendors to link customers' transactions. Transactions of the loyalty systems are unlinkable, i.e., no persistent identifiers (PIDs) are employed, and an attempt of a malicious vendor to introduce such PIDs will be detectable by the customer. Hence, non-trivial profiles of customers cannot be build with these systems. Customers explicitly wishing to establish a profile with some vendor may do so by sending extra authentication information, which is not part of the loyalty system, but may easily be build around the loyalty system's protocols. However, if such a thing is done, it should be done openly and only after the customer has given her consent and has been informed beforehand about the consequences of sending authentication data, i.e., that it becomes possible to create a profile.

Loyalty systems can be part of a vendor's customer relationship management, as it is a means to retain customers and to increase the incentive for repeated purchases. The privacy aspect of the loyalty systems introduced in this chapter may also attract new customers that, under normal circumstances, reject loyalty programs for their lack of privacy protection.

In addition to the privacy aspect, the counter-based schemes proposed here are overall more efficient than the token-based approach in terms of processing time, used bandwidth, and storage space, if more than one loyalty point is to be issued and redeemed. Furthermore, these schemes provide vendors with the additional option to allow or disallow pooling of loyalty points, which is not possible in the token-based scheme where pooling cannot be prevented.

Chapter 6

An Anonymous Multi-Coupon System

In this chapter, we introduce the notion of multi-coupons. These multi-coupons are similar to vouchers, however, may be spent more than once. From a more abstract point of view, our multi-coupons can be seen as unlinkable m-showable credentials, which allow a gradual release of information encoded in the multi-coupons. An abstract of this chapter was previously presented in [CES+05]. This work, however, did not include detailed security proofs of the employed protocols and also no details of the protocols themselves.

6.1 Introduction

Today, coupons appear to be useful means for vendors to attract the attention of potential customers. Usually, coupons give the customer a financial incentive to purchase at a specific vendor. The purpose of coupons is many-fold. For instance, they can be used to draw the attention of customers to a newly opened shop or to prevent customers from buying at a competitor's shop [SZ95]. Of course, coupons can also be purchased by customers, e.g., gift certificates. Even drug prescriptions from a doctor can be seen as a kind of a coupon.

In general, a coupon is a representation of the right to claim some good or service, usually from the party that issued the coupon. The types of coupons mentioned before can, in general, be redeemed only once, i.e., the coupon is invalidated after the service or good has been claimed. However, there are also coupons which can be redeemed more than once, such as a coupon book of a movie theatre, where customers pay, say, for 9 movies and are entitled to see 10. We call such coupons *multi-coupons*. In this chapter, we are particularly interested in this type of coupons.

Typically, a real-world multi-coupon of value m is devalued by crossing out some field or by detaching a part of it. Offering such coupons can be beneficial for the issuing party, e.g., a movie theatre. First, customers pay in advance for services or goods they have not claimed yet. Second, they are locked-in by the issuer/vendor, i.e., they are unlikely to switch to another vendor to purchase the same or similar service or good as long as they have not redeemed all their coupons. Hence, multi-coupons can also be seen as a kind of loyalty program since they are specific to some vendor and induce loyalty, at least, as long as the customer has coupons left to spend.

Clearly, vendors are interested in creating loyalty and hence, it is likely that we are going to see such coupon systems in the Internet, too. For instance, with the proliferation of broadband Internet connections, we are beginning to see companies which stream movies directly to individual customers' devices. Apart from the palette of available movies, such a virtual movie theatre may attract and retain customers by offering financial incentives in the form of multi-coupons. In this case, multi-coupons may be used just like multi-admission tickets of brick-and-mortar theatres, which allow customers to see one movie per (single) coupon from the admission ticket.

6.1.1 Desirable Properties for Coupon Systems

At first, introducing a coupon system looks like a win-win situation, since both parties seem to benefit from such a coupon system. Vendors have a means to create a loyal customer base and customers value the financial benefit provided by coupons. However, since a customer normally redeems her coupons in different transactions, a multi-coupon can be used as a means to link transactions, and thus, to allow a vendor to create a record of the customer's past purchases. Such customer information might be exploited for data mining, to infer new customer data, customer profiling, promotion of new products, price discrimination, etc. [Odl03]. Thus, if customers expect that their data will be misused when they use the coupon system, e.g., by using it to create profiles for price discrimination [FKZ02], they are more likely to decline the coupon system. According to [HNP99, Kob01] privacy is a concern to Internet users, especially when it comes to electronic commerce scenarios — see also Section 2.1. Hence, a prudent vendor should take these concerns into account when planning to offer a coupon system.

In order to rule out privacy concerns of customers from the start, vendors might want to introduce a coupon system that does not infringe their customers' privacy. Thus, a coupon should disclose as little information as possible. For instance, a multi-coupon should only give vendors an indication that it is still valid, i.e., that at least one coupon is not spent, instead of disclosing the number of unspent coupons.

Such a property could be useful in sensitive areas, e.g., in health care scenarios, where a multi-coupon can be used as a prescription for a certain number of doses of some medicine. In this case, the pharmacist would deduct a single coupon from the multi-coupon and may only learn if the prescription has been used up or not. Also in welfare, paper-based cheques or food stamps could be replaced by electronic coupons. In fact, recently, the U.S. announced to replace their paper-based food stamp program with electronic benefits and debit cards [Pea04]. However, this electronic program does not protect the privacy of recipients, since the cards are processed similar to ordinary debit cards.

For vendors, in addition to common security requirements such as unforgeability, there are other requirements which are specific to a coupon system. As mentioned before, a vendor's driving reason for offering a coupon system is to establish a long-term relationship with customers. However, customers may be interested in sharing a multi-coupon, i.e., each customer obtains and redeems a fraction of the coupons in the multi-coupon. Moreover, this behaviour allows them, e.g., to sell coupons on an individual basis for a cheaper price[1], e.g., to one-time customers who otherwise would have purchased full-priced services or goods. Thus, ideally, vendors wish to prevent customers from *splitting* their coupons.

To illustrate splitting, we consider the following variants as examples of real-world multi-coupons. The first variant, being a coupon book with detachable coupons and the second one being a multi-coupon where spent coupons are crossed out, i.e., coupons cannot be detached. The coupon book can be easily shared by a group of customers, since each customer can detach its share of coupons from the book and each coupon may be independently redeemed by a different customer. In the second variant, the multi-coupon must be given to the vendor *as a whole* to allow him to devalue the multi-coupon by crossing out one of the coupons. Hence, in this variant, individual coupons cannot be split and redeemed separately and independently as in the first variant.

Nevertheless, even in the multi-coupon scenario with non-detachable coupons some kind of sharing is possible, if we transfer it to the digital world. Since digital coupons can be easily copied, colluding customers may jointly purchase a multi-coupon, distribute copies of it among each other, and agree to redeem only the share of the coupons for which each of them paid for. In this scenario, however, customers have to fully trust each other that none of them redeems more than its share of coupons. Since each of the colluders owns a copy of the multi-coupon, this means that every colluder has full control of *all* single coupons. Hence, each of them could redeem single coupons of other colluders without their knowledge. A colluder 'deceived' in such a way would only learn about it when he or she tries

[1] Recall that a multi-coupon for m goods is usually sold for the price of $m - k$ goods, $k \geq 1$

to redeem a single coupon and the vendor rejects it because it was already spent. Thus, it seems less likely that multi-coupons are traded between customers.

In this context, another scenario with multi-coupons is possible where trust is only one-way. If customer Alice buys a multi-coupon, uses up, say, half of the coupons and sells the remaining half of the coupons to customer Bob then Alice does not have to trust Bob. Only Bob has to trust Alice that he indeed received the purported half of the coupons from her. There is nothing that really can stop Alice from doing so, neither in a real-world scenario with paper coupons nor in the digital scenario, unless (a) the multi-coupon contains information that ties it to Alice's identity and which the vendor must be able to verify (b) Alice has a strong incentive to keep the multi-coupon, e.g., because some private and/or 'valuable' information is encoded into the multi-coupon.

We do not pursue any of these two approaches because, first, we do not want to identify customers because this may violate their privacy and, second, encoding valuable information seems to be unsatisfactory as well because encoding a 'valuable' secret implies that such a secret exists and that a customer is willing to encode it into a potentially much less valuable multi-coupon. Instead, we employ *all-or-nothing sharing* which has been used in other works before [Bra99, CL01] to *discourage* users from sharing/disclosing certain data, such as private credential information.

In case of a multi-coupon, all-or-nothing means that a customer cannot give away or sell any single coupon from its multi-coupon without giving away all other single coupons — this includes used and unused single coupons alike. Therefore, our scheme is comparable with the real-world multi-coupon example from above where used coupons are crossed out. The difference is that in the digital world one may effortlessly create identical copies of a multi-coupon while in the real world creating exact replicas of such a coupon may require some effort.

6.1.2 Overview of the Coupon System

The coupon system proposed here can be viewed as a digital counterpart to the real-world multi-coupon with non-detachable coupons, as mentioned before. In our coupon system, a multi-coupon M is a digital signature on a tuple X where $X = (x_1, \ldots, x_m)$. In the following outline of the system, we denote a set of coupons by M and a single coupon by $x \in \{x_1, \ldots, x_m\}$.

In the issue phase of the coupon, a user first convinces a vendor that she knows X without revealing the elements of X. Then, the verifier issues the coupon M by 'blindly' signing X, i.e., $M := \mathit{Sign}(X)$, and sending M to the user. Here we make use of the Camenisch-Lysyanskaya (CL) signature scheme [CL02].

When redeeming a single coupon x, the user reveals x to the vendor and proves that she is in possession of a valid multi-coupon $M = \mathit{Sign}(X)$ and $x \in$

$\{x_1, \ldots, x_m\}$. The vendor then checks if x is in a list of used coupons. If it is not, the vendor accepts x and puts it on the list. Beside satisfying common security requirements, the scheme has the following properties: The vendor is not able to trace x back to M or to link two redemptions because M is never given back to the vendor and the single coupons x are independent of each other. Furthermore, the vendor does not learn anything about the status of the multi-coupon, i.e., how many single coupons of the multi-coupon are still unspent. Regarding the vendor's requirement, the scheme does not allow users to split a multi-coupon without sharing all values $(x_1, \ldots, x_m)$.

A method used in the coupon system might be of independent interest. It proves knowledge of the CL signature M on a message tuple $X := (x_1, \ldots, x_m)$ and allows to reveal an arbitrary single message x_j, while the remaining elements of the tuple, the revealed message's index, j, and the signature M remain hidden. This method may be useful in a wider range of applications, as the multi-coupon can also be regarded as an m-showable credential that provides unlinkability between different showings of the credential and also allows gradual release of certified information, i.e., one x_j per showing.

6.2 Related Work

At first it may seem that the coupon system can be easily realised using an existing payment system or credential system which supports m-showable credentials or at least one-showable credentials of which m can be obtained. However, none of these systems satisfied all the requirements of the coupon system we had in mind, or could only satisfy them in an inefficient manner. In addition, some of the systems discussed below require the existence of a trusted third party to issue certificates of some sort. We do not have such a requirement.

The payment system of Chaum [Cha89] as well as the one of Brands [Bra93] use digital coins which can be anonymously spent. Withdrawal and spending of coins is roughly the same as issuance and redemption of single coupons. However, using m single-spendable digital coins as a multi-coupon easily allows splitting of the multi-coupon. Even if we would use multi-valued coins, such that one unit of an m-coin can be spent and an $m-1$ coin is returned, we would still not have the coupon system that we have in mind, since the number of remaining coins is disclosed to the vendor. In the coin system of Ferguson [Fer93] a multi-coin is introduced that can be spent m times. However, when paying with the same multi-coin, the vendor learns the remaining value of the coin and, in addition, transactions are linkable.

Okamoto and Ohta [OO90] proposed a scheme which resembles our coupon system in the sense that they use a multiple blind signature to issue a "large" coin which is comprised of "smaller" coins, or rather, can be subdivided into smaller

ones. However, subdividability in their system is considered a feature while in a coupon system this translates to splitting and, hence, is less desirable. In [Che96] and [Ver01], Chen and Verheul, respectively, proposed credential systems where the credentials are multi-showable, i.e., can be shown for an unlimited number of times. The credentials obtained through both systems are intended for pseudonymous usage, thus, our requirements for unlinkable redemptions and m-redeemability are not satisfied.

In the work of Brands [Bra99], attribute certificates were proposed that allow selective showing of individual attributes. These attributes are normally multi-showable but can be made m-showable, however, then different transactions become linkable. Persiano and Visconti [PV04] used some of the ideas of [Bra99] to build a credential system which is multi-showable and does not allow to link different showings of a credential. Still, showings of credentials cannot be limited.

An anonymous credential system where credentials can be made one-showable was proposed by Camenisch and Lysyanskaya [CL01]. Through this system, a user may obtain m one-showable credentials which can be regarded as single coupons. However, this approach is not very efficient when used in a coupon system, since a credential generation protocol must be run for each credential and the credentials can be shared by different users and independently shown[2]. This means, when applied as a coupon system, coupons can be independently spent and splitting is easily possible.

The aspect of technically supporting loyalty in commercial applications has also been explored before. Maher [Mah98] proposed a framework to introduce loyalty points, however, the privacy aspect was not an issue there. Enzmann *et al.* [EFS04, ES04] (see Chapter 5) proposed a counter for a privacy-friendly, point-based loyalty system, where users anonymously obtained points for their purchases. Finally, Wibowo *et al.* [WLT00] proposed a loyalty system, however, based on pseudonyms, thus, providing a weaker form of privacy.

The initial publication of this work [CES+05] inspired several other works, e.g., [Ngu06, CGH06a, CGH06b], that proposed new functionality for multi-coupon systems. Building on the ideas of this work, Canard *et al.* [CGH06a, CGH06b] proposed a more efficient approach which employed other but more efficient primitives [CHL05, DY05] than our system and consequently resulted in better overall efficiency. Nguyen [Ngu06] proposed a system that is even more efficient than the system of Canard *et al.* using roughly the same primitives as they did.

[2] In [CL01] a solution was proposed to deal with this kind of lending. However, this solution hurts performance because it adds complexity and additional protocol runs to the basic scheme.

6.3 Model

The coupon system considered here mainly involves two parties, a customer U (user) and a vendor V. The system itself is comprised of an *issue* protocol and a *redeem* protocol which are both carried out between U and V. The output of the issue protocol is a multi-coupon M for U and the result of the redeem protocol is a spent single coupon for V and a multi-coupon devalued by one single coupon for U. Next, we state the main security requirements for the involved parties.

6.3.1 Requirements

Unforgeability. It must be infeasible to create new multi-coupons, to increase the number of unspent coupons, or to reset the number of spent coupons.

Double-spending detection. A vendor must be able to detect attempts of redeeming 'old' coupons that have already been redeemed. This means, given two runs of the redeem protocol, where a single coupon x is deducted from multi-coupon M and y is deducted from N, the vendor must be able to decide if $x = y$.

Redemption limitation. An m-redeemable coupon M may not be accepted by the vendor more than m times.

Protection against splitting. A coalition of customers U_i should not be able to split an m-redeemable multi-coupon M into (disjoint) s_i-redeemable shares M_i with $\sum_i s_i \leq m$ such that M_i can only be redeemed by customer U_i and none of the other customers $U_j, j \neq i$, or a subset of them is able to redeem the share M_i or a part of it. We call this property *strong protection against splitting*.

A weaker form of this property is *all-or-nothing-sharing*. This means that splitting is possible, however, only if customers trust each other not to spent (part of) the other's share M_i. Another way of putting this is to say that sharing M means sharing all m single coupons. We call this *weak protection against splitting*. In other works [Bra99, CL01], a similar property, called all-or-nothing-disclosure, had been employed to discourage lending of credentials.

Unlinkability. It must be infeasible for vendors to link protocol runs of honest users. For this, we have to consider linking a run of an issue protocol to runs of corresponding redeem protocols and linking of any two redeem protocol runs.

(1) *issue vs. redeem*: Given a run I of the issue protocol with output a multi-coupon M and given a redeem protocol run R with output a devalued multi-coupon

N, the vendor must not be able to decide if $\mathcal{L}(I, R)$ holds. In other words, he can neither assert nor rule out that m is the 'initial version' of N and hence, he will not know if M and N are essentially the 'same' multi-coupon or 'different' ones.

(2) *redeem vs. redeem*: Given two runs of the redeem protocol, R and R', with output two multi-coupons, M and M', respectively, the vendor must neither be able to decide if $\mathcal{L}(R, R')$ holds, i.e., he cannot tell if M is a former 'version' of M', or vice versa, nor if they are unrelated altogether.

Minimum disclosure. As a result of a redeem protocol run, the vendor may only learn of the single coupon being redeemed but not the number of remaining coupons. This already follows from the unlinkability requirement but we make it explicit here, nevertheless.

6.4 Cryptographic Preliminaries

For the construction of our multi-coupon system, we first need some foundations. In the following, we will therefore introduce several cryptographic 'primitives' which we employ in subsequent sections to derive the building blocks for the final coupon system.

6.4.1 Notational Conventions and Definitions

Throughout this chapter, we will use some definitions and notational conventions which are introduced next. A prime p is called a *safe prime* if $p = 2p' + 1$, where p' is also prime. The order of an element g of a multiplicative group G is denoted by $\mathrm{ord}_G(g)$ and is defined as the smallest integer α such that $g^\alpha = 1$ (in G). If g is a generator for some group, we write $\langle g \rangle$ for the group generated by g.

The set of residues modulo n that are relatively prime to n are denoted by $\mathbb{Z}_n^*$ and the set of quadratic residues modulo n is denoted by QR_n. An integer a is a quadratic residue modulo n if $b \in \mathbb{Z}_n^*$ exists such that $a = b^2 \bmod n$.

Furthermore, we denote the binary length of an integer I by ℓ_I and we write "$a \in_R S$" if a is to be chosen uniformly and at random from the set of integers S.

We say that a function $\epsilon(k)$ is *negligible*, if $\epsilon(k) \leq 1/p(k)$ for all polynomials $p(\cdot)$ and sufficiently large k. The quantity $1 - \epsilon(k)$ is called *overwhelming*, if $\epsilon(k)$ is negligible. In addition, a function is *noticeable*, or simply *non-negligible*, if it is not negligible.

6.4.2 Commitment Scheme

A commitment scheme is a two-party protocol between a committer $\mathcal{C}$ and a receiver $\mathcal{R}$. In general, the scheme includes a *Commit* process and an *Open* process. In the first process, $\mathcal{C}$ computes a commitment C_x for a secret message x, such that x cannot be changed without changing C_x [BCC88]. $\mathcal{C}$ then gives C_x to $\mathcal{R}$ and keeps x secret. In the second process, $\mathcal{C}$ opens C_x by revealing x.

The commitment scheme we employ is due to Damgård and Fujisaki (DF) [DF02] which is a generalisation of the Fujisaki-Okamoto scheme [FO97]. We skip the introduction of the basic DF scheme for committing to a single value x and proceed to the scheme where the commitment is to a message tuple $(x_1, x_2, \ldots, x_m)$. Although, the DF scheme works on any finite Abelian group G, we are only interested in the case where G is the group $\mathbb{Z}_n^*$, where n is the product of two primes.

Let $h \in_R \mathbb{Z}_n^*$ be an element of large order and in addition, the order should be comprised of at least one large prime factor. Furthermore, let $g_1, g_2, \ldots, g_m \in \langle h \rangle$, i.e., $g_i = h^{\gamma_i} \bmod n$ for some γ_i, and let ℓ_x denote the maximum allowed binary length of any message x. To commit to a secret message tuple $X := (x_1, x_2, \ldots, x_m)$, where $x_i \in [0, 2^{\ell_x} - 1]$, the committing party $\mathcal{C}$ uses the public commitment key $PK := (g_1, \ldots, g_m, h, n)$ to form the commitment

$$C_X = \prod_{i=1}^{m} g_i^{x_i} h^{r_X} \bmod n, \tag{6.1}$$

where $r_X \in_R \mathbb{Z}_n$ is chosen at random. $\mathcal{C}$ then sends C_X to the receiving party $\mathcal{R}$. In the opening phase, $\mathcal{C}$ simply reveals X and r_X to $\mathcal{R}$ which allows $\mathcal{R}$ to verify that Equation (6.1) holds. Note that $\mathcal{C}$ in our case does not have to know the order of $\mathbb{Z}_n^*$.

6.4.3 Signature Scheme

The signature scheme stated in the following is a variant of the signature scheme developed by Camenisch and Lysyanskaya (CL) [CL02] for signing a block of messages. A similar variant was also used in [BCC04a, BCC04b]. The scheme had been proven secure under the strong RSA assumption which is briefly stated next:

Strong RSA assumption [BP97, FO97]. The strong RSA assumption conjectures that it is hard, on input an RSA modulus n and an element $x \in \mathbb{Z}_n^*$, to compute values $e > 1$ and y such that $y^e = x \bmod n$. □

For the variant of the CL signature scheme considered in the following, the message to be signed is a tuple denoted by $X := (x_1, x_2, \ldots, x_m)$ where $x_i \in [0, 2^{\ell_x} -$

1], $i = 1, \ldots, m$ and ℓ_x is a security parameter for the permissible length of a message.

Key Generation. Let k be a security parameter and let n be the product of two safe primes, i.e., $n = pq$, $p = 2p' + 1, q = 2q' + 1$, where p, q, p', q' are primes and $k = \ell_p = \ell_n/2$. Furthermore, let h be a generator of QR_n and $b \in_R \langle h \rangle$. For $i = 1, \ldots, m$, choose $\alpha_i, \gamma \in_R [1, p'q']$ and set $a_i := b^{\alpha_i} \bmod n$, $c := b^{\gamma} \bmod n$ — this particular choice for the parameters a_i, b, and c is a variation of the original CL scheme and is needed later on to statistically hide a signature in $\langle h \rangle$. The signer's public key then becomes $PK := (A, b, c, n)$ where $A := (a_1, a_2, \ldots, a_m)$ and the secret key is the factorisation of n, i.e., $SK := (p, q)$. We denote this algorithm by

$$(A, b, c, n, h, p, q) \leftarrow KeyGen_m^{CL}(1^k).$$

Signing. On input a message tuple $X := (x_1, x_2, \ldots, x_m)$, choose a random prime number $e \in_R [2^{\ell_e - 1} + 1, 2^{\ell_e} - 1]$, where ℓ_e is an upper bound for the binary length of e, and also choose a random number s of length ℓ_s. The quantities ℓ_e and ℓ_s are security parameters and should be chosen such that $\ell_e > \ell_x + 1$ and $\ell_s = \ell_n + \ell_x + \ell_{CL}$, where the parameter ℓ_{CL} is an additional security parameter that was necessary in the security proof of the CL signature scheme (see [CL02] for the details). Proposed values for these security parameters are $\ell_n = 1024, \ell_x = 160, \ell_e = 162, \ell_{CL} = 80$ — for the details of the exact choice of these parameters and the proofs of security, the reader is referred to [CL02]. Finally, the signature (v, e, s) on a message X is computed such that

$$c = v^e a_1^{x_1} \cdots a_m^{x_m} b^s \bmod n. \tag{6.2}$$

We denote this algorithm by

$$(v, e, s) \leftarrow Sign_{(A,b,c,n,p,q)}(X).$$

Verification. In order to verify that (v, e, s) is a signature on $X := (x_1, x_2, \ldots, x_m)$, Equation (6.2) must be verified and also that $x_i \in [0, 2^{\ell_x} - 1]$, $i = 1, \ldots, m$, and $2^{\ell_e - 1} < e < 2^{\ell_e}$. We denote this algorithm by

$$ind \leftarrow Verify_{(A,b,c,n)}(X, v, e, s)$$

where the indicator $ind \in \{accept, reject\}$.

Remark 1. The CL signature scheme is separable, i.e., the signature (v, e, s) on X is also the signature on any sub-tuple of X, if we change the public key accordingly. In the following, we use the notation $X \setminus (x_j)$ to denote the sub-tuple of

X which is comprised of all of X's components but its j-th one, i.e., $X \backslash (x_j) = (x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_m)$.

As for separability, the message x_j from the tuple X can be separated from the signature (v, e, s) on X with respect to the public key (A, b, c, n) if the public key is changed to $(A \backslash (a_j), b, c/a_j^{x_j}, n)$, i.e., given

$$v^e a_1^{x_1} \cdots a_m^{x_m} b^s = c$$

the signature for $X \backslash (x_j)$ is derived by dividing both sides by $a_j^{x_j}$, i.e.,

$$v^e a_1^{x_1} \cdots a_{j-1}^{x_{j-1}} a_{j+1}^{x_{j+1}} \cdots a_m^{x_m} b^s = c/a_j^{x_j}.$$

This holds for any sub-tuple Y of X. We will use this property in our coupon system to redeem a single coupon from a set of coupons, i.e., a multi-coupon.

Remark 2. As noted in [CG04], the CL signature scheme has the property of randomisation, i.e., the signature (v, e, s) can be randomised to

$$(v^* := vb^w, e, s^* := s - ew)$$

for an arbitrary integer w. For a verifier it makes no difference whether he verifies (v^*, e, s^*) or (v, e, s) because both are signatures on X with respect to the same public key (A, b, c, n), since

$$(v^*)^e \prod_{i=1}^{m} a_i^{x_i} b^{s^*} = v^e b^{ew} \prod_{i=1}^{m} a_i^{x_i} b^s b^{-ew} = v^e \prod_{i=1}^{m} a_i^{x_i} b^s \bmod n.$$

This property benefits our scheme because in subsequent sections we employ a proof of knowledge of a CL signature. Such a proof can be done more efficiently in an anonymous manner for a randomised signature (v^*, e, s^*) than for (v, e, s). The reason for this is that v^* can be made different in each run of the proof protocol by always choosing a new randomisation parameter w.

6.4.4 Proofs of Relations between Committed Numbers

For the construction of our coupon system, we implicitly employ several sub-protocols in order to prove certain statements and relations between committed numbers. These protocols are interactive proofs carried out between two parties, a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. We will give an informal introduction to interactive proofs and related notions. For a more formal treatment see the seminal paper on interactive proof systems by Goldwasser, Micali, and Rackoff [GMR89]. Camenisch [Cam98] provides a brief, yet formal, summary of the subject. We, however, continue with an informal approach to the subject.

Roughly speaking, in an interactive proof system, the prover tries to convince the verifier that she knows a certain secret or that a certain claimed statement holds. At the end of such a protocol, $\mathcal{V}$ should be convinced of $\mathcal{P}$'s knowledge or the validity of her statement. An important class of interactive proof systems are so called *zero-knowledge* proof systems. In these systems, the verifier receives no information except for the validity of the prover's claimed knowledge or statement. As a consequence, the verifier cannot use his gained knowledge to prove the statement to other parties.

Several models for interactive proof systems have been put forth (for an overview see [BC89]). Roughly speaking, these models differ in the power of the prover, e.g., if the prover is given an exponential amount of time to compute a witness (information to convince the verifier) or if he already knows a witness (or is able to compute one in polynomial time). In the latter case the prover's total time for computations is therefore bound by a polynomial. Interactive proof systems that prove knowledge of some witness are often called *proofs of knowledge* (PoK) and in the following, we will employ this kind of proofs. In a PoK, the prover's computation time is usually restricted to polynomial time because if he had unlimited computing power/time he could compute a witness whenever he wants, and hence, proofs of knowledge would be of lesser interest.

A proof of knowledge, as any other interactive proof system, must satisfy *completeness* and *validity*. The former means that if the prover $\mathcal{P}$ indeed possesses knowledge with respect to a certain statement and carries out a PoK with the verifier $\mathcal{V}$ then the probability that $\mathcal{V}$ will *not* be convinced by the PoK is negligible — in other words $\mathcal{V}$ will be convinced of $\mathcal{P}$'s knowledge with overwhelming probability. If $\mathcal{V}$ is convinced, we also say that he *accepts* the PoK. The validity property means that if $\mathcal{P}$ makes $\mathcal{V}$ accept, then $\mathcal{P}$ will indeed possess the claimed knowledge with overwhelming probability.

In the following, we briefly state a number of proofs of knowledge that can be found in the literature and which we implicitly employ in our scheme. The output of each of the protocols for the verifier is an indication $ind_{\mathcal{V}} \in \{accept, reject\}$. In the following proofs of knowledge, the commitments are formed using the DF scheme.

PoKRep. A prover $\mathcal{P}$ proves knowledge of a discrete logarithm representation (DLREP) modulo a composite to a verifier $\mathcal{V}$ [FO97]. Common inputs are a description of the group G, the public key $PK := (g_1, \ldots, g_m, h)$ with $h, g_i \in G$, and a commitment C. By this protocol, $\mathcal{P}$ convinces $\mathcal{V}$ of knowledge of $X := (x_1, \ldots, x_m)$, such that $C = \prod_{i=1}^{m} g_i^{x_i} h^r$ (in G).

PoKInt. A prover $\mathcal{P}$ proves to a verifier $\mathcal{V}$ knowledge of x and r such that $C = g^x h^r$ and $a \leq x \leq b$. Common inputs are parameters (g, h, n), the commitment C, and

the integers a, b. We use a straightforward extension to the basic protocol, such that the proved knowledge is two tuples, instead of two values, and the interval membership of each component from a tuple, instead of one value. Within this extension, we denote $G := (g_1, g_2, ..., g_m)$, $H := (h_1, h_2, ..., h_l)$, $X := (x_1, x_2, ..., x_m)$, $R := (r_1, r_2, ..., r_l)$, and $C := \prod_{i=1}^{m} g_i^{x_i} \prod_{j=1}^{l} h_j^{r_j}$. By running the protocol, $\mathcal{P}$ proves to $\mathcal{V}$ knowledge of X and R, and the interval membership, $a \leq x_i \leq b$. A number of protocols exist for proving interval membership, e.g., [FO97, CFT98, CM99b, CM99a, Bou00]. However, only Boudot's protocol [Bou00] ensures that X lies within the exact bounds of the interval, whereas the other protocols can only guarantee membership to an expanded interval, i.e., the bounds a, b are expanded by some factor δ.

PoKOr. A prover $\mathcal{P}$ proves to a verifier $\mathcal{V}$ an 'OR statement' of a commitment C, such that $C := (C_1, \ldots, C_m)$, where $C_i := \prod_{j \in I_i} g_j^{x_{ij}} h^{r_i}$ and $I_i \subseteq \{1, ..., m\}$, and $\mathcal{P}$ knows at least one tuple $\{x_{ij} \mid j \in I_i\}$ for some undisclosed i. We denote the OR statement as $\bigvee_{i=1}^{m} C_i = \prod_{j \in I_i} g_j^{x_{ij}} h^{r_i}$. Common inputs are C and parameters (B, n) where $B := (g_1, \ldots, g_m)$. By running the protocol, $\mathcal{P}$ proves to $\mathcal{V}$ knowledge of $\{x_{ij} \mid j \in I_i\}$ without revealing the values x_{ij} and i. A number of mechanisms for proving the OR statement have been introduced, e.g. [Cam98], [CDS94], and [dSdCPY94].

PoK. Sometimes, we need to carry out two or more of the above protocols simultaneously, e.g., when responses to challenges have to be used in more than one validity check of the verifier to prove intermingled relations among commitments. Instead of giving concrete constructions of these protocols each time, we just describe their aim, i.e., what the verifier wants to prove. For this we apply the notation used, e.g., in [CS97]. For instance, the following expression

$$ind_{\mathcal{V}} \leftarrow PoK\{(\alpha, \beta) : C = g^{\alpha} h^{\beta} \ \wedge \ D = \hat{g}^{\alpha} \hat{h}^{\beta} \ \wedge \ 0 \leq \alpha < 2^k\}$$

means that knowledge of α and β is proven such that $C = g^{\alpha} h^{\rho}$ and $D = \hat{g}^{\alpha} \hat{h}^{\beta}$ holds, and α lies in the integer interval $[0, 2^k)$. By convention, Greek letters denote values of which knowledge is proven of.

6.5 Blind Signature Scheme

The first building block for our coupon system is a so called blind signature scheme. We will employ it in the coupon system to anonymously issue multi-coupons to customers. In the following, we state a protocol for securely signing a blinded tuple of messages. The protocol is shown in Figure 6.1 and is based on [CL02,

BCC04a, BCC04b]. Figure 6.1 is meant as an overview and uses the abstract notation introduced above, which hides most of the technical details of the protocol. The purpose of the protocol is to provide a user U with a signature from the signer S on a tuple $X := (x_1, x_2, \ldots, x_m)$ without revealing X to S.

We assume that S has the public key $PK := (A, b, c, n)$, secret key $SK := (p, q)$, and public length parameters $\ell_n, \ell_x, \ell_e, \ell_s$, ℓ_ϕ, and $\ell_{\mathcal{H}}$. The last parameters do not belong to the signature scheme but are needed in the *PoK* which is carried out in the protocol. The parameter $\ell_{\mathcal{H}}$ determines the size of the signer's challenge (who also acts as verifier in this protocol) and ℓ_ϕ is a security parameter that controls the statistical zero-knowledge property of the employed *PoK*— we will elaborate this point in the analysis following the introduction of the protocol.

The user U's input to the protocol is the message tuple $X := (x_1, \ldots, x_m)$ for which U wants to obtain a signature. The values $x_1, \ldots, x_m$ are uniformly chosen at random from the interval $[0, 2^{\ell_x} - 1]$. Among the first steps in Figure 6.1, U computes the commitment C for the message tuple X according to Equation (6.3) and then sends C to S.

$$C := \prod_{i=1}^{m} a_i^{x_i} b^{s'} \bmod n \tag{6.3}$$

The next steps assure to S that U indeed knows the discrete logarithm representation (DLREP) of C with respect to the bases $(a_1, \ldots, a_m, b)$ and that the committed values in C have been chosen from the correct intervals. This assurance is provided by means of a proof of knowledge of a DLREP of C with respect to the bases $(a_1, a_2, \ldots, a_m, b)$.

If all proofs are accepted, S chooses a prime e and computes

$$v := (c/(Cb^{s''}))^{1/e} = (c/(\prod_{i=1}^{m} a_i^{x_i} b^{(s'+s'')}))^{1/e} \bmod n.$$

At the end, V sends the resulting tuple (v, e, s'') to U. Finally, U sets $s := (s' + s'')$ and obtains (v, e, s) which is the desired signature on X. We denote this protocol for blindly signing a tuple by

$$(v, e, s) \leftarrow \mathit{BlindSign}_{(PK)}(X).$$

Analysis. From the equations above it is clear that *BlindSign* yields a CL signature, if the user formed her commitments C and D according to the protocol's specification. In the following, we will argue two points. First, the signer will almost always sign only correctly formed commitments and second, if the commitments are correctly formed, he will almost always learn nothing of any committed value

User U		**Signer** S
Common Input:	Verification key PK :=	(A, b, c, n), $A := (a_1, \ldots, a_m)$
	Length parameters	$\ell_x, \ell_e, \ell_s, \ell_n, \ell_{\mathcal{H}}, \ell_\phi$
User's Input:	Message X :=	$(x_1, \ldots, x_m)$
Signer's Input:	Factorisation of n	(p, q)

choose $s' \in_R [0, 2^{\ell_n+\ell_\phi} - 1]$
$C := \prod_{i=1}^{m} a_i^{x_i} b^{s'} \bmod n$ $\xrightarrow{C}$

Run PoK { $(\xi_1, \ldots, \xi_m, \sigma) : C = a_1^{\xi_1} \cdots a_m^{\xi_m} b^{\sigma} \bmod n \ \wedge$
for $i = 1, \ldots, m : \xi_i \in \{0,1\}^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi+2} \ \wedge$
$\sigma \in \{0,1\}^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi+2}\} \rightarrow ind_S$

check $ind_S \stackrel{?}{=} accept$
choose $s'' \in_R [0, 2^{\ell_s-1}]$
choose $e \in_R [2^{\ell_e-1} + 1, 2^{\ell_e} - 1]$

$s := s' + s''$; $\xleftarrow{(v,e,s'')}$ $v := (c/(Cb^{s''}))^{1/e} \bmod n$

check $Verify_{(A,b,c,n)}(X, v, e, s) \stackrel{?}{=} accept$
[i.e., $c \stackrel{?}{=} v^e b^s \prod_{i=1}^{m} a_i^{x_i} \bmod n$]
output (v, e, s)

Figure 6.1: *Protocol for blindly signing a tuple:* $BlindSign$

x_i, i.e., *BlindSign* is indeed a blind signature protocol. Here, "almost always" means in an overwhelming number of protocol runs.

For the analysis of *BlindSign*, it is helpful to have the full protocol at hand because the arguments in the analysis are sometimes very technical in nature and therefore, hard to understand from the abstract notation used in Figure 6.1. Therefore, the full protocol is additionally shown in Figure 6.2 to allow for the necessary depth in detail. The analysis will show that *BlindSign* is indeed a blind signature scheme, i.e., the signer learns nothing of the message X that is to be signed.

In the first half of the protocol (see Figure 6.2), a zero-knowledge proof of knowledge of a DLREP (*PoKRep*) is given which shows that U knows a representation of the commitment C with respect to the signer's public key (A, b, c, n). Of course, the whole protocol *BlindSign* is not zero-knowledge because after the *PoKRep* the signer S sends a signature (v, e, s''), which 'proves' that U actually interacted with S. The analysis of the employed *PoKRep*, apart from showing that the signer does not learn X, also serves to introduce certain techniques with respect to zero-

knowledge proofs that we employ in subsequent protocols. We introduce the notion of zero-knowledge at this point because we will need it in said subsequent protocols, and it is more comprehensive to illustrate it with this straight-forward *PoK* protocol than with the more complex ones to follow.

Hence, for the analysis, we assume that *BlindSign stops* after the user U has sent her responses y_i and the signer S has verified them — later on we will also call this the *reduced BlindSign* protocol. In essence, what we do is to analyse the *PoKRep* carried out in the first half of *BlindSign*. We start the analysis by discussing the completeness and validity properties of *PoKRep* as shown in the first half of Figure 6.2.

6.5.1 Completeness

For completeness, we have to show that if the user U, acting as prover, knows a correct representation of the commitment C then the signer S must be able to verify this. Hence, we must show that the verification equation

$$C^t D = \prod_{i=1}^{m} a_i^{y_i} b^{y_{s'}} \bmod n \tag{6.4}$$

holds for correct responses $y_i = x_i t + r_i$, for $1 \leq i \leq m$, and $y_{s'} = s't + r_{s'}$ (see Figure 6.2). This is shown in the next equation.

$$\begin{aligned} C^t D &= (\prod_{i=1}^{m} a_i^{x_i} b^{s'})^t (\prod_{i=1}^{m} a_i^{r_i} b^{r_{s'}}) = \prod_{i=1}^{m} a_i^{x_i t} b^{s't} \prod_{i=1}^{m} a_i^{r_i} b^{r_{s'}} = \prod_{i=1}^{m} a_i^{x_i t + r_i} b^{s't + r_{s'}} \\ &= \prod_{i=1}^{m} a_i^{y_i} b^{y_{s'}} \bmod n \end{aligned} \tag{6.5}$$

6.5.2 Validity / Knowledge Error

In the protocol, if a cheating user U' does not know a representation of C, she may still be able to satisfy Equation (6.4), i.e., make S accept. She can do this, if she correctly predicts S's challenge t. That is, in the commitment phase, she chooses random values $C' \in_R QR_n$, $y'_i \in_R [0, 2^{\ell_x + \ell_{\mathcal{H}} + \ell_\phi} - 1]$, $y'_{s'} \in_R [0, 2^{\ell_n + \ell_{\mathcal{H}} + 2\ell_\phi} - 1]$, and prepares for a specific challenge t_0 by computing $D' := C'^{-t_0} \prod_{i=1}^{m} a_i^{y'_i} b^{y'_{s'}} \bmod n$. Then, the messages between U' and S are exchanged as usual. Obviously, the values $C', D', y'_1, y'_2, \ldots, y'_m, y'_{s'}$ sent by a cheating U' also satisfy Equation (6.4), if it

User U			**Signer** S
Common Input:	Verification key PK	$:=$	(A, b, c, n), $A := (a_1, \ldots, a_m)$
	Length parameters		$\ell_x, \ell_e, \ell_s, \ell_n, \ell_{\mathcal{H}}, \ell_\phi$
User's Input:	Message X	$:=$	$(x_1, \ldots, x_m)$, $x_1, \ldots, x_m \in [0, 2^{\ell_x} - 1]$
Signer's Input:	Factorisation of n		(p, q)

Choose:
$r_1, \ldots, r_m \in_R [0, 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi} - 1]$;
$s' \in_R [0, 2^{\ell_n+\ell_\phi} - 1]$;
$r_{s'} \in_R [0, 2^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi} - 1]$;
$C := \prod_{i=1}^m a_i^{x_i} b^{s'} \bmod n$;
$D := \prod_{i=1}^m a_i^{r_i} b^{r_{s'}} \bmod n$

$\xrightarrow{C,D}$

$\xleftarrow{t}$ Choose: $t \in_R [0, 2^{\ell_{\mathcal{H}}} - 1]$

For $i = 1, \ldots, m$:
$y_i := x_i t + r_i$;
$y_{s'} := s't + r_{s'}$

$\xrightarrow{y_1, \ldots, y_m, y_{s'}}$

Verify:
$C^t D \stackrel{?}{=} \prod_{i=1}^m a_i^{y_i} b^{y_s} \bmod n$;
For $i = 1, \ldots, m$:
$y_i \stackrel{?}{\in} \{0,1\}^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi+1}$;
$y_{s'} \stackrel{?}{\in} \{0,1\}^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi+1}$;
If not valid then *Abort*;
choose:
$s'' \in_R [0, 2^{\ell_s-1}]$;
$e \in_R [2^{\ell_e-1} + 1, 2^{\ell_e} - 1]$ such that e is *prime*;

$\xleftarrow{v,e,s''}$ $v := (c/(Cb^{s''}))^{1/e} \bmod n$

$s := s' + s''$;
Verify:
$c \stackrel{?}{=} v^e \prod_{i=1}^m a_i^{x_i} b^s \bmod n$;
$(e - 2^{\ell_e-1}) \stackrel{?}{\in} \{0,1\}^{\ell_e}$;
If not valid then *Abort*;
Store (v, e, s), $X := (x_1, \ldots, x_m)$;

Figure 6.2: *Detailed protocol* $BlindSign$

turns out that U''s guess t_0 was correct. The success rate of such an attack is determined by the security parameter $\ell_{\mathcal{H}}$ which gives the binary length of a challenge t and hence, determines the number of potential challenges. Since U' can, in any one run of the protocol, only prepare for one challenge, the success probability of such an attack is $1/2^{\ell_{\mathcal{H}}}$, which is a negligible quantity in the security parameter $\ell_{\mathcal{H}}$. In other words the protocol has an error probability of $2^{-\ell_{\mathcal{H}}}$, which is sometimes also referred to as a knowledge error.

6.5.3 Proof of Knowledge

We will argue next that the protocol so far is a proof of knowledge, i.e., it shows that if the signer S (acting as verifier $\mathcal{V}$) accepts, the user U (acting as prover $\mathcal{P}$) must indeed 'know' the DLREP $(x_1, x_2, \ldots, x_m, s')$ of the commitment C with respect to bases (A, b). To show this, we employ a technique that is commonly referred to as *rewinding*. Using this technique, we construct an expected polynomial-time algorithm $\mathcal{K}$, called a *knowledge extractor*, that computes the DLREP of C from two accepting protocol transcripts. If $\mathcal{K}$ succeeds, this proves that $\mathcal{P}$ indeed has the claimed knowledge. We require two conditions for the knowledge extractor $\mathcal{K}$ [BG93]:

1.) The success probability of $\mathcal{K}$ must be close to the success probability of the prover $\mathcal{P}^*$, i.e., it may only negligibly deviate from $\mathcal{P}^*$'s.[3]

2.) The knowledge extractor $\mathcal{K}$ must run in (expected) polynomial time.

In the construction of the knowledge extractor $\mathcal{K}$, we use a prover $\mathcal{P}^*$ as a resettable sub-routine. "Resettable" means that $\mathcal{P}^*$ can be run as often as $\mathcal{K}$ pleases *and* it can be reset to any one of its previously seen states of the current run of the protocol (*rewinding*) — this is similar to rewinding the tape of a VCR (and playing it again). If $\mathcal{P}^*$ has been rewound, it 'loses any knowledge' of the states after the reset starting point and may subsequently react differently, if it receives different inputs — recall that $\mathcal{P}^*$ is interactive. In addition, $\mathcal{P}^*$ may be a cheating prover, i.e., one that does not posses the claimed knowledge, but may still be successful in correctly responding to a certain subset of challenges asked by the (simulated) verifier $\mathcal{V}$. Therefore, instead of assuming that $\mathcal{P}^*$ always succeeds, we more generally let $\pi_{\mathcal{P}^*}$ denote the probability that $\mathcal{P}^*$ successfully completes the proof protocol — thus, potentially $\pi_{\mathcal{P}^*} < 1$. Next, we explain how the knowledge extractor works. This particular construction is due to [FF00].

[3] Of course, to be meaningful this condition implicitly assumes that $\mathcal{P}^*$'s success probability is greater than the protocol's knowledge error.

In step 1, the knowledge extractor $\mathcal{K}$, playing the role of the verifier $\mathcal{V}$, runs reduced *BlindSign* with $\mathcal{P}^*$ until the sub-protocol *PoKRep* has finished, and it received the transcript $T := [C, D, t, y_1, \ldots, y_m, y_{s'}]$. Next, $\mathcal{K}$ checks if Equation (6.4) holds — it does with probability $\pi_{\mathcal{P}^*}$. If not, $\mathcal{K}$ returns *failure*. Even if Equation (6.4) holds, i.e., T is an accepting transcript, with probability $2^{-\ell_{\mathcal{H}}}$ $\mathcal{K}$ will still return *failure* — we will explain this in a moment. If $\mathcal{K}$ completed its first step without returning *failure*, step 2 follows — otherwise $\mathcal{K}$ stops and the knowledge extraction has failed.

In step 2, $\mathcal{K}$ begins with rewinding $\mathcal{P}^*$ to the state where $\mathcal{P}^*$ had already sent its commitments C and D. Then $\mathcal{K}$ randomly chooses a new challenge t' and passes it to $\mathcal{P}^*$ who responds accordingly. The resulting transcript $T' := [C, D, t', y'_1, \ldots, y'_m, y'_{s'}]$ will be accepted by $\mathcal{V}$ with probability $\varpi_{\mathcal{P}^*}(C, D)$, where $\varpi_{\mathcal{P}^*}(C, D)$ is the conditional probability that, given that C and D had been chosen as commitments in step 1, $\mathcal{P}^*$ will succeed in convincing $\mathcal{V}$. If T' is rejected or $t = t'$, $\mathcal{K}$ repeats step 2 until it receives an accepting transcript with $t \neq t'$.

The success probability of $\mathcal{K}$ is $\pi_{\mathcal{P}^*} - 2^{-\ell_{\mathcal{H}}}$ because as soon as step 2 is started, $\mathcal{K}$ will (at any cost) extract a second transcript T' different from T so that the DLREP of C can be computed (see below). The overall success probability $\pi_{\mathcal{P}^*} - 2^{-\ell_{\mathcal{H}}}$ of $\mathcal{K}$ takes into account the event that in the loop of step 2 the challenge t is chosen (again) —which happens with probability $2^{-\ell_{\mathcal{H}}}$— in which case the transcripts are identical and knowledge extraction would fail. Since the knowledge extractor's probability of success is $\pi_{\mathcal{P}^*} - 2^{-\ell_{\mathcal{H}}}$, which only negligibly deviates from $\mathcal{P}^*$'s, we conclude that condition 1.) is satisfied.

The knowledge extractor will need an expected number of $1/(\varpi_{\mathcal{P}^*}(C, D) - 2^{-\ell_{\mathcal{H}}})$ many iterations of step 2 in order to receive a second accepting transcript T'. Hence, its expected running time is $\tau/(\varpi_{\mathcal{P}^*}(C, D) - 2^{-\ell_{\mathcal{H}}})$, where τ is essentially the running time of the (simulated) verifier $\mathcal{V}$ which is polynomial. Now, for any fixed pair of commitments C and D, the probability that $\mathcal{V}$ accepts is $\varpi_{\mathcal{P}^*}(C, D)$ (see above) and hence, the probability for entering step 2 can be expressed as $\varpi_{\mathcal{P}^*}(C, D) - 2^{-\ell_{\mathcal{H}}}$. Since step 2 is only entered with probability $\varpi_{\mathcal{P}^*}(C, D) - 2^{-\ell_{\mathcal{H}}}$, $\mathcal{K}$'s expected running time is $(\varpi_{\mathcal{P}^*}(C, D) - 2^{-\ell_{\mathcal{H}}}) \cdot \tau/(\varpi_{\mathcal{P}^*}(C, D) - 2^{-\ell_{\mathcal{H}}}) = \tau$ which, of course, is polynomial. Thus, $\mathcal{K}$ runs in expected polynomial time and condition 2.) is also satisfied.

From now on, we condition on the event that step 2 was entered. Therefore, $\mathcal{K}$ eventually receives a transcript $T' := [C, D, t', y'_1, \ldots, y'_m, y'_{s'}]$ which contains another accepting answer from $\mathcal{P}^*$, different from the one of T. We now show how $\mathcal{K}$ extracts the DL representation of C from the two transcripts, i.e., retrieves the values $(x_1, x_2, \ldots, x_m, s')$.

For the simulation, suppose $\mathcal{K}$ gave the public key $(a_1^*, a_2^*, \ldots, a_m^*, b, n)$ to $\mathcal{P}^*$ as common input, where, for $i = 1, \ldots, m$, $a_i^* = b^{\alpha_i^*} \bmod n$ and α_i^* is an integer with

no factor smaller than $2^{\ell_{\mathcal{H}}}$. Note that C and D are fixed in both transcripts, as $\mathcal{P}^*$ has only been rewound to the state *after* C and D have been sent. Since both transcripts have been accepted, Equation (6.4) holds and from (6.4) and the two transcripts we can derive the following equation.

$$\begin{aligned} C^{\Delta t} := C^{(t-t')} &= \prod_{i=1}^{m} (a_i^*)^{(y_i - y_i')} b^{(y_s - y_{s'})} =: \prod_{i=1}^{m} (a_i^*)^{\Delta y_i} b^{\Delta y_{s'}} \\ &= b^{\sum_{i=1}^{m} \alpha_i^* \Delta y_i + \Delta y_{s'}} \bmod n \end{aligned} \tag{6.6}$$

We will now show that either Δt divides all Δy_i and $\Delta y_{s'}$, which enables $\mathcal{K}$ to compute/extract the values $x_1, x_2, \ldots, x_m, s'$ (see Equations (6.7)), or otherwise, $\mathcal{K}$ would be able to compute non-trivial roots in $\mathbb{Z}_n^*$, which violates the strong RSA assumption. In the following, we assume that $C \neq 1$, as this would allow $\mathcal{K}$ to immediately output a trivial DLREP of C, namely $(0, 0, \ldots, 0)$.

Now, let $u := \sum_{i=1}^{m} \alpha_i^* \Delta y_i + \Delta y_{s'}$ and assume for contradiction that $\Delta t \nmid u$. Thus, $\gcd(\Delta t, u) = 1$ and, by the extended Euclidian algorithm, we get values ϑ, ν such that $\Delta t \vartheta + u\nu = 1$. Now, $\mathcal{K}$ computes (for any $b \in \mathbb{Z}_n^*$) the value $z \in \mathbb{Z}_n^*$ such that $b = b^{\Delta t\vartheta + u\nu} = b^{\Delta t\vartheta}(b^u)^\nu = b^{\Delta t\vartheta}(C^{\Delta t})^\nu = (b^\vartheta C^\nu)^{\Delta t} =: z^{\Delta t} \bmod n$, which is a Δt-th root of b. Clearly, since $\mathcal{K}$ runs in expected polynomial time, this contradicts the strong RSA assumption and hence, Δt must divide all Δy_i and $\Delta y_{s'}$ — note that by construction $\Delta t \nmid \alpha_i^*$. Therefore, from the two transcripts T and T', $\mathcal{K}$ can extract the DLREP of C by computing the quantities (for $i = 1, 2, \ldots, m$)

$$x_i = \frac{\Delta y_i}{\Delta t} = \frac{y_i - y_i'}{t - t'} \qquad \text{and} \qquad s' = \frac{\Delta y_{s'}}{\Delta t} = \frac{y_{s'} - y_{s'}'}{t - t'}. \tag{6.7}$$

6.5.4 Zero-Knowledge

Showing that a protocol is "zero-knowledge" means to show that anything that the verifier $\mathcal{V}$, i.e., in our case the signer, could have computed from the transcript generated in his interaction with a prover $\mathcal{P}$, he could have computed by himself, i.e., without interacting with $\mathcal{P}$. The idea of this notion is that if $\mathcal{V}$ can simulate the interaction with $\mathcal{P}$ without even talking to her, i.e., he is able to forge a transcript that looks like a transcript from a real interaction, then a real transcript cannot contain more knowledge than the verifier possesses, anyhow.

Now, if the protocol *BlindSign* would stop after the *PoK* of the DLREP of C, as we have assumed in the beginning, it would indeed be *honest-verifier statistical* zero-knowledge. The "statistical" in the qualification of zero-knowledge means that the zero-knowledge property is not violated if only a statistically small, i.e., negligible, chance exists that allows an adversary to distinguish a forged transcript from a real one. And finally, "honest-verifier" means that we assume that $\mathcal{V}$ does not deviate from the protocol, i.e., he chooses his challenge t at random.

The critical point in the previous paragraph is the notion of "*looks* like a transcript from a real interaction". In order to show that a transcript can be forged, i.e., a real interaction between $\mathcal{P}$ and $\mathcal{V}$ can be simulated, we first define an algorithm $\mathcal{S}$, called a simulator, that outputs transcripts $T' := [C, D', t', y'_1, \ldots, y'_m, y'_{s'}]$. All values, except for C which is taken from a real conversation and hence 'conveys' knowledge, are chosen/computed as follows:

$$\begin{gathered} t' \in_R \ [0, 2^{\ell_{\mathcal{H}}} - 1] \\ \text{for } i = 1, \ldots, m: \quad y'_i \in_R [0, 2^{\ell_x + \ell_{\mathcal{H}} + \ell_\phi} - 1] \\ y'_{s'} \in_R \ [0, 2^{\ell_n + \ell_{\mathcal{H}} + 2\ell_\phi} - 1] \\ D' := C^{-t'} \prod_{i=1}^{m} a_i^{y'_i} b^{y'_{s'}} \bmod n \end{gathered}$$

Note that the values chosen by the simulator for its transcript T' are just random values, i.e., they do not contain any knowledge, and that they have been uniformly chosen from their respective intervals. Our goal is to show that T', produced by the simulator $\mathcal{S}$, is *statistically indistinguishable* from a transcript T of a real interaction between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. This means, we have to show that the statistical difference between any T and any T' is negligible when only a *polynomially sized* sample set of transactions $\mathcal{T}$ can be examined. In other words, if $\mathcal{T}$ is fed to an algorithm that tries to distinguish sample sets from $\mathcal{S}$ and sample sets of real conversations from $\mathcal{P}$ and $\mathcal{V}$, the algorithm's chance of successfully deciding if $\mathcal{T}$ contains samples that stem either from the simulator or from real conversations, is at most $1/2 + \epsilon$, where ϵ is a negligible quantity. The rational behind this approach is that if one cannot even tell random values from 'real' values then one cannot gain any knowledge from them.

The following formal notion of statistical indistinguishability was developed by Goldwasser, Micali, and Rackoff [GMR89].

Definition 8 (Statistical indistinguishability [GMR89]) *Let $L \subset \{0,1\}^*$ be any language. Two families of random variables $\{U(x)\}$ and $\{V(x)\}$ are* statistically indistinguishable on L *if*

$$\sum_{y \in \{0,1\}^*} |Pr(U(x) = y) - Pr(V(x) = y)| < {\ell_x}^{-c}$$

for all constants $c > 0$ and all sufficiently long $x \in L$.

Using Definition 8 we are now ready to show that the distribution of T and T' are indeed statistically indistinguishable. First, we examine the distribution of the

y_i and of $y_{s'}$. We, however, drop the index i from the notation of the variables y_i and r_i for convenience. We start by examining the values y. Recall that,

$$\begin{aligned} y &= xt + r & r &\in_R [0, \ldots, 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi} - 1] \\ x &\in [0, \ldots, 2^{\ell_x} - 1] & t &\in [0, \ldots, 2^{\ell_{\mathcal{H}}} - 1] \end{aligned}$$

where $\in_R$ means *uniformly chosen at random*. In the following, let $u = xt$ and hence

$$u \in [0, \ldots, 2^{\ell_x+\ell_{\mathcal{H}}} - 1].$$

For the comparison with the distribution of the simulator's values y', we are seeking the probability distribution (distribution function) of the discrete random variable Y which takes on values y from real conversations between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$.

First, note that the distribution of Y is determined by the distributions of the two random variables $U = u$ and $R = r$, since $Y = U + R$. Now suppose $R = r$, then $Y = y$ if and only if $U = y - r$. Hence, Y is the simultaneous realisation of the two events

$$R = r \quad \text{and} \quad U = y - r. \tag{6.8}$$

Note that R and U are independent, as the verifier is assumed to be honest and hence, chooses t *independent* of any value previously sent by the prover. Therefore, we can compute Y's probability distribution as

$$\begin{aligned} Pr(Y = y) &= \sum_{r=0}^{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi-1}} Pr(U = y - r, R = r) \\ &= \sum_{r=0}^{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi-1}} Pr(U = y - r) \cdot Pr(R = r) \\ &= \frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \sum_{r=0}^{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi-1}} Pr(U = y - r). \end{aligned} \tag{6.9}$$

We are already given that the random variable R is uniformly distributed on the interval $[0, 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi} - 1]$, while U is clearly not distributed uniformly. However, the following two equations show that Y, the sum of U and R, is uniformly distributed over a certain interval in $[0, 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi} - 1]$. We use the lower and upper limit of the claimed interval to show its uniform distribution.

Let $y = 2^{\ell_x+\ell_{\mathcal{H}}} - 1$, which is the lower limit of the claimed interval. Plugging this value into Equation (6.9) leads to the following equation.

$$
\begin{aligned}
Pr(Y = 2^{\ell_x+\ell_{\mathcal{H}}} - 1) &= \frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \sum_{r=0}^{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1} Pr(U = 2^{\ell_x+\ell_{\mathcal{H}}} - 1 - r) \\
&\overset{(1)}{=} \frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \Bigg[Pr(U = 2^{\ell_x+\ell_{\mathcal{H}}} - 1) + Pr(U = 2^{\ell_x+\ell_{\mathcal{H}}} - 2) \\
&\qquad + \cdots + Pr(U = 0) + \underbrace{Pr(U = -1)}_{=0} + \underbrace{\cdots}_{=0} \\
&\qquad + \underbrace{Pr(U = 2^{\ell_x+\ell_{\mathcal{H}}} - 1 - (2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi} - 1))}_{=0} \Bigg] \\
&\overset{(2)}{=} \frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \sum_{r=0}^{2^{\ell_x+\ell_{\mathcal{H}}}-1} Pr(U = 2^{\ell_x+\ell_{\mathcal{H}}} - 1 - r) \qquad (6.10) \\
&\overset{(3)}{=} \frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \sum_{r=0}^{2^{\ell_x+\ell_{\mathcal{H}}}-1} Pr(U = r) \\
&= \frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \cdot 1
\end{aligned}
$$

In the equation, the sum includes a partial sum, starting from the sum's first through $2^{\ell_x+\ell_{\mathcal{H}}}$-th addend, where U takes on values $2^{\ell_x+\ell_{\mathcal{H}}} - 1$ through 0 for which the probability is greater or equal to zero (see Equation (6.10.(1))). The remaining probabilities are guaranteed to be zero because U does not take on values below zero, since $u = xt$ and $x \geq 0$ and $t \geq 0$. Thus, the sum's upper limit can be replaced with $2^{\ell_x+\ell_{\mathcal{H}}} - 1$ (6.10.(2)). Now the sum runs from highest to lowest sample point which can be rewritten to run from lowest to highest sample point (6.10.(3)). From this equation it can be seen that the sum of probabilities must add up to unity because U takes on every value from its sample space. Finally, the result shows that $Pr(Y = 2^{\ell_x+\ell_{\mathcal{H}}} - 1) = 1/2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}$ is indeed the expected value for a uniform distribution on $[0, 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi} - 1]$.

It can be easily seen that if y increases, the partial sum 'moves' from the beginning of the total sum to the end of the total sum. Hence, $y = 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi} - 1$ is the largest value for which the whole partial sum is part of the total sum. The latter is shown in Equation (6.11) where we plug in $y = 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi} - 1$ to show that the upper limit of the claimed interval is uniformly distributed on $[0, 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi} - 1]$, as well.

$$
\begin{aligned}
Pr(Y=y) &= \frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \sum_{r=0}^{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1} Pr(U=y-r) \\
&= \frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \Bigg[\underbrace{Pr(U=y)}_{=0} + \underbrace{Pr(U=y-1)}_{=0} + \underbrace{\ldots}_{=0} \\
&\qquad + Pr(U=(2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1)-(2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-2^{\ell_x+\ell_{\mathcal{H}}})) \\
&\qquad \ldots \\
&\qquad + Pr(U=(2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1)-(2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1)) \Bigg] \\
&= \frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \sum_{r=2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-2^{\ell_x+\ell_{\mathcal{H}}}}^{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1} Pr(U=y-r) \qquad (6.11) \\
&= \frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \sum_{r=0}^{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1-(2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-2^{\ell_x+\ell_{\mathcal{H}}})} Pr(U=y-(r+2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-2^{\ell_x+\ell_{\mathcal{H}}})) \\
&= \frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \sum_{r=0}^{2^{\ell_x+\ell_{\mathcal{H}}}-1} Pr(U=2^{\ell_x+\ell_{\mathcal{H}}}-1-r) \\
&= \frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \sum_{r=0}^{2^{\ell_x+\ell_{\mathcal{H}}}-1} Pr(U=r) \\
&= \frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \cdot 1
\end{aligned}
$$

Again, the result shows that the expected value is identical to a uniform distribution on $[0, 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1]$. It is clear that this distribution holds for the whole interval $[2^{\ell_x+\ell_{\mathcal{H}}}-1, 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1]$. However, if y increases beyond $2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1$, the partial sum 'leaves' the total sum and for $0 \leq y < 2^{\ell_x+\ell_{\mathcal{H}}}-1$ it 'enters' the total sum. The probability for $Pr(Y=y)$ will in these cases most of the time deviate from the expected value of a uniform distribution because only a subset of $(U \cap R)$'s sample space will be included in the total sum.

In summary, we have the following distribution function of Y for any value $y \in \mathbb{Z}$ and any distribution of x and t from $[0, \ldots, 2^{\ell_x}-1]$ and $[0, \ldots, 2^{\ell_{\mathcal{H}}}-1]$, respectively.

$$Pr(Y=y)=\begin{cases} =0 & \text{for} \quad y<0 & \text{(I)} \\ \leq 2^{-(\ell_x+\ell_{\mathcal{H}}+\ell_\phi)} & \text{for} \quad 0\leq y<2^{\ell_x+\ell_{\mathcal{H}}}-1 & \text{(II)} \\ =2^{-(\ell_x+\ell_{\mathcal{H}}+\ell_\phi)} & \text{for} \quad 2^{\ell_x+\ell_{\mathcal{H}}}-1\leq y\leq 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1 & \text{(III)} \\ \leq 2^{-(\ell_x+\ell_{\mathcal{H}}+\ell_\phi)} & \text{for} \quad 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}\leq y\leq 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}+2^{\ell_x+\ell_{\mathcal{H}}}-1 & \text{(IV)} \\ =0 & \text{for} \quad y\geq 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}+2^{\ell_x+\ell_{\mathcal{H}}} & \text{(V)} \end{cases}$$

Now, we want to compute the statistical difference between the distribution of any y from a real conversation's transcript and a uniform distribution on $[0, 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1]$. For this, let Y' be a random variable that is uniformly distributed on $[0, 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1]$ (like the simulator's choices for y'). Note that Y's deviations from a uniform distribution can be found in intervals (II) and (IV). In (I), (III), and (V) the distributions are identical to Y''s because (III) yields a uniform distribution, as shown above, and (I) and (V) are outside the range of the uniform distribution on $[0, 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1]$ and the distribution of responses y from a real conversation. Hence, the complement of y's probability of belonging to interval (III) yields the probability of y belonging to interval (II) or (IV).

Using this notion and Definition 8, we finally get the following equation for the statistical difference between the simulator's uniformly chosen values y' and the values y from a real conversation between a prover $\mathcal{P}$ and an honest verifier $\mathcal{V}$.

$$\begin{aligned} \sum_{y\in\mathbb{Z}} |Pr(Y=y)-Pr(Y'=y)| &= 1-\left(\frac{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}-1-(2^{\ell_x+\ell_{\mathcal{H}}}-1)+1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}}\right) \\ &= 1-\left(\frac{2^{\ell_x+\ell_{\mathcal{H}}}(2^{\ell_\phi}-1)+1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}}\right) \\ &= 1-\frac{2^{\ell_\phi}-1}{2^{\ell_\phi}}-\frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \\ &= \frac{1}{2^{\ell_\phi}}-\frac{1}{2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi}} \\ &< \frac{1}{2^{\ell_\phi}} \end{aligned} \tag{6.12}$$

This result shows that the statistical difference between a real conversation and the simulator's output is only about $2^{-\ell_\phi}$, which is a negligible quantity in the security parameter ℓ_ϕ, and also in ℓ_x, since $2^{-\ell_\phi} < {\ell_x}^{-c}$ for any constant c and sufficiently large ℓ_ϕ. Hence, the two distributions are statistically indistinguishable, as claimed.

Another interpretation of Equation (6.12) is that with probability $2^{-\ell_\phi}$ a verifier may gain some information on the committed values x_i. This happens if y is very small or very large, i.e., if it lies in the interval (II) or (IV), respectively. For

instance, if $y = 1$ and $\mathcal{V}$ sent $t = 1$ then either $x = 0$ and $r = 1$, or $x = 1$ and $r = 0$. But as y approaches the lower limit of interval (III), the more choices exist for x and r and the maximum number of choices is reached in interval (III). However, as y grows larger than the upper limit of interval (III), i.e., enters interval (IV), the number of potential choices for x and r become smaller again, up to the point where they are completely determined.

Next, we examine the distribution of $y_{s'}$ from the transcript T. Similar to y, we let $u' := s't$ and thus, we get

$$y_{s'} = u' + r_{s'}$$

$$\begin{aligned} r_{s'} &\in_R [0, \ldots, 2^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi} - 1] & \qquad u' &\in [0, \ldots, 2^{\ell_n+\ell_{\mathcal{H}}+\ell_\phi} - 1] \\ s' &\in [0, \ldots, 2^{\ell_n+\ell_\phi} - 1] & \qquad t &\in [0, \ldots, 2^{\ell_{\mathcal{H}}} - 1]. \end{aligned}$$

Again, we have to determine the distribution function for a random variable $Y = U + R$, where U and R are also random variables. Analogous to Equation (6.9), we receive the following equation.

$$\begin{aligned} Pr(Y = y_{s'}) &= \sum_{r_{s'}=0}^{2^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi-1}} Pr(R = r_{s'}) \cdot Pr(U = y_{s'} - r_{s'}) \qquad (6.13) \\ &= \frac{1}{2^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi}} \sum_{r_{s'}=0}^{2^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi-1}} Pr(U = y_{s'} - r_{s'}) \end{aligned}$$

The distribution function for $Pr(Y = y_{s'})$ can be derived analogously to the one for the y_i and so we get the following function for any $y_{s'} \in \mathbb{Z}$.

$$Pr(Y = y_{s'}) = \begin{cases} = 0 & \text{for} \quad y_{s'} < 0 \\ \leq 2^{-(\ell_n+\ell_{\mathcal{H}}+2\ell_\phi)} & \text{for} \quad 0 \leq y_{s'} < 2^{\ell_n+\ell_{\mathcal{H}}+\ell_\phi} - 1 \\ = 2^{-(\ell_n+\ell_{\mathcal{H}}+2\ell_\phi)} & \text{for} \quad 2^{\ell_n+\ell_{\mathcal{H}}+\ell_\phi} - 1 \leq y_{s'} \leq 2^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi} - 1 \\ \leq 2^{-(\ell_n+\ell_{\mathcal{H}}+2\ell_\phi)} & \text{for} \quad 2^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi} \leq y_{s'} \leq 2^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi} + 2^{\ell_n+\ell_{\mathcal{H}}+\ell_\phi} - 1 \\ = 0 & \text{for} \quad y_{s'} \geq 2^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi} + 2^{\ell_n+\ell_{\mathcal{H}}+\ell_\phi} \end{cases}$$

It remains to show that the statistical difference between Y and a random variable Y' (like the one of the simulator $\mathcal{S}$), which is uniformly distributed on $[0, \ldots, 2^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi} - 1]$, is negligible. This is shown in Estimation (6.14).

$$\begin{aligned}
\sum_{y_{s'} \in \mathbb{Z}} |Pr(Y = y_{s'}) - Pr(Y' = y_{s'})| &= 1 - \left(\frac{2^{\ell_n + \ell_{\mathcal{H}} + 2\ell_\phi} - 1 - (2^{\ell_n + \ell_{\mathcal{H}} + \ell_\phi} - 1) + 1}{2^{\ell_n + \ell_{\mathcal{H}} + 2\ell_\phi}} \right) \\
&= 1 - \left(\frac{2^{\ell_n + \ell_{\mathcal{H}} + \ell_\phi}(2^{\ell_\phi} - 1) + 1}{2^{\ell_n + \ell_{\mathcal{H}} + 2\ell_\phi}} \right) \\
&= 1 - \frac{2^{\ell_\phi} - 1}{2^{\ell_\phi}} - \frac{1}{2^{\ell_n + \ell_{\mathcal{H}} + 2\ell_\phi}} \qquad (6.14) \\
&= \frac{1}{2^{\ell_\phi}} - \frac{1}{2^{\ell_n + \ell_{\mathcal{H}} + 2\ell_\phi}} \\
&< \frac{1}{2^{\ell_\phi}}
\end{aligned}$$

Finally, we show that the value D from a real conversation is also statistically indistinguishable from the simulator's choice D'. For this, recall from Section 6.4.3 that $a_i = b^{\alpha_i} \bmod n$, for $i = 1, \ldots, m$. Hence, we have

$$\begin{aligned}
D &= \prod_{i=1}^{m} a_i^{r_i} b^{r_{s'}} = \prod_{i=1}^{m} (b^{\alpha_i})^{r_i} b^{r_{s'}} = b^{\sum_{i=1}^{m} \alpha_i r_i + r_{s'}} \bmod n \\
D' &= C^{-t'} \prod_{i=1}^{m} a_i^{y'_i} b^{y'_{s'}} = (\prod_{i=1}^{m} (b^{\alpha_i})^{x_i} b^{s'})^{-t'} \prod_{i=1}^{m} (b^{\alpha_i})^{y'_i} b^{y'_{s'}} \\
&= \prod_{i=1}^{m} b^{\alpha_i (y'_i - x_i t')} b^{y'_{s'} - s't'} \bmod n \\
&= b^{\sum_{i=1}^{m} \alpha_i (y'_i - x_i t') - s't' + y'_{s'}} \bmod n
\end{aligned}$$

Note that for this particular choice of D' the verification Equation (6.5) will hold. The quantities $\varrho := \sum_{i=1}^{m} \alpha_i r_i$ and $\xi := \sum_{i=1}^{m} \alpha_i (y'_i - x_i t') - s't'$ both appear in b's exponent and hence, we are allowed to 'reduce' them modulo $\mathrm{ord}(QR_n)$. In the following, we will denote values x reduced modulo $\mathrm{ord}(QR_n)$ by $\underline{x}$.

Now, let $\rho := \underline{\varrho} + r_{s'}$ and $\chi := \underline{\xi} + y'_{s'}$. Note that both $r_{s'}$ and $y_{s'}$ are chosen with bit-length $2^{\ell_n + \ell_{\mathcal{H}} + 2\ell_\phi} > \mathrm{ord}(QR_n)$ and that ρ, as well as χ, is eventually reduced modulo $\mathrm{ord}(QR_n)$. Specifically, we have the following distributions for the variables

$$\underline{\varrho}, \underline{\xi} \in [0, \mathrm{ord}(QR_n) - 1] \qquad \text{and} \qquad r_{s'}, y'_{s'} \in_R [0, 2^{\ell_x + \ell_{\mathcal{H}} + 2\ell_\phi} - 1].$$

Hence, the distributions of ρ and χ are identical, i.e., they are trivially also statistically indistinguishable, and consequently, also the distributions of D and D'.

In conclusion, we have shown that the distribution of the simulator's randomly chosen transcripts $T' := [C, D', t', y'_1, \ldots, y'_m, y'_{s'}]$ are statistically indistinguishable from transcripts $T := [C, D, t, y_1, \ldots, y_m, y_{s'}]$ of real conversations between a prover

and an honest verifier. Note that this would not hold for an arbitrary verifier, because in this case t might have been computed as an arbitrary secret function of C (and D) and hence, t''s distribution may largely deviate from the distribution of t, and the distribution of any y in the transcript would depend on a specific choice of C (and D). However, since in the honest verifier model we assume that t is chosen randomly, the commitment C is (in an overwhelming number of protocol runs) independent of the rest of the transcript and hence, we can conclude that the *reduced BlindSign* protocol is honest-verifier statistical zero-knowledge.

By showing that the proof of knowledge of the DLREP of C from the *BlindSign* protocol is statistical zero-knowledge, we have also shown that the signer does not gain any knowledge[4] on the message X which he is asked to sign. Hence, *BlindSign* is indeed a blind signature protocol. ■

6.6 Proof of Knowledge of a Signature

In this section, we describe the second building block for the multi-coupon system. It is a protocol used in the redemption of single coupons which are to be deducted from a multi-coupon. The protocol itself, shown in Figure 6.3, is an honest-verifier statistical zero-knowledge proof of a signature output from the *BlindSign* protocol.

The idea of this protocol is to convince a verifier $\mathcal{V}$ that a prover $\mathcal{P}$ holds a valid signature (v, e, s) on X satisfying $c = v^e a_1^{x_1} \cdots a_m^{x_m} b^s \bmod n$ without $\mathcal{V}$ learning anything of the signature but its validity. The common inputs are the same as in the *BlindSign* protocol. $\mathcal{P}$'s secret input is the message tuple X and the corresponding signature (v, e, s).

The protocol works as follows: $\mathcal{P}$ first randomises the signature components, v and s, by choosing w at random and computing $T_1 := vh^w$ and $s^* = s - ew$. In addition, $\mathcal{P}$ commits to the exponent e by computing $T_2 := g_1^e g_2^w h^r$. Then $\mathcal{P}$ sends T_1, T_2 to $\mathcal{V}$. Next, $\mathcal{P}$ proves to $\mathcal{V}$ her knowledge specified in PoK.

As discussed in Remark 2 of Section 6.4.3, from $\mathcal{V}$'s point of view, (T_1, e, s^*) is a valid signature on X, as is (v, e, s). The difference between them is that we are allowed to reveal the value T_1 to $\mathcal{V}$, but not the value v, because T_1 is different in every run of the proof protocol. Therefore, to prove the signature with $c \equiv v^e \prod_{i=1}^m a_i^{x_i} b^s$ becomes one with proving $c \equiv T_1^e \prod_{i=1}^m a_i^{x_i} b^{s^*}$. Clearly, to prove the second equation is much simpler than the first one. In addition, $\mathcal{P}$ proves knowledge of the exponent e and the auxiliary parameter w needed for the verification of the proof. PoK here performs the following simple proofs in one go:

[4] If we say "one does not gain any knowledge" in the context of statistical zero-knowledge, we mean that the chance of gaining some knowledge is negligible.

Prover $\mathcal{P}$		**Verifier** $\mathcal{V}$
Common Input:	Verification key PK	$:= (A, b, c, n, g_1, g_2, h),$ $A = (a_1, a_2, \ldots, a_m)$
	Length parameters	$\ell_x, \ell_e, \ell_s, \ell_n, \ell_{\mathcal{H}}, \ell_\phi$
Prover's Input:	Message X	$:= (x_1, \ldots, x_m)$
	Signature	(v, e, s)

choose $r, w \in_R \{0,1\}^{\ell_n + \ell_\phi}$
compute $T_1 := vh^w$;
compute $T_2 := g_1^e g_2^w h^r$ $\xrightarrow{T_1, T_2}$

Run PoK { $(\xi_1, \ldots, \xi_m, \sigma, \epsilon, \omega, \rho)$:
$c = T_1^\epsilon a_1^{\xi_1} \cdots a_m^{\xi_m} b^\sigma h^\omega \ \wedge \ T_2 = g_1^\epsilon g_2^\omega h^\rho \ \wedge$
for $i = 1, \ldots, m : \xi_i \in \{0,1\}^{\ell_x + \ell_{\mathcal{H}} + \ell_\phi + 2} \ \wedge$
$(\epsilon - 2^{\ell_e - 1}) \in \{0,1\}^{\ell_e + \ell_{\mathcal{H}} + \ell_\phi + 1}$ } $\rightarrow ind_{\mathcal{V}}$
check $ind_{\mathcal{V}} \stackrel{?}{=} accept$

Figure 6.3: *Protocol for proving knowledge of a signature:* $PoKSign$

(1) *PoKRep*: to prove knowledge of a DLREP of $c = T_1^e \prod_{i=1}^m a_i^{x_i} b^{s^*} \bmod n$ with respect to the basis $(T_1, a_1, \ldots, a_m, b)$, respectively;

(2) *PoKRep*: to prove knowledge of a DLREP of $T_2 = g_1^e g_2^w h^r \bmod n$ with respect to the basis (g_1, g_2, h), respectively;

(3) *PoKInt*: to prove the values $x_1, \ldots, x_m$ are within a right bound, i.e., for $i = 1, \ldots, m : x_i \in \{0,1\}^{\ell_x + \ell_{\mathcal{H}} + \ell_\phi + 1}$;

(4) *PoKInt*: to prove the value e is also within a right bound, i.e., $(e - 2^{\ell_e - 1}) \in \{0,1\}^{(\ell_e - 1) + \ell_{\mathcal{H}} + \ell_\phi + 1}$.

For later reference, we will denote the protocol by

$$ind \leftarrow PoKSign_{(A,b,c,n,g_1,g_2,h)}(X, v, e, s).$$

Analysis. In the following analysis, we will now show that *PoKSign* is an honest-verifier statistical zero-knowledge proof of knowledge of a CL signature. As in Section 6.5, we use a more detailed description of the protocol, in order to provide the in-depth analysis. This more detailed version of *PoKSign* is shown in Figure 6.4.

In contrast to *BlindSign*, the public key has been expanded to include 3 additional bases, g_1, g_2, h, which are used to provide unambiguous commitments for additional values of which knowledge is proven. Let $g_1, g_2 \in_R \langle h \rangle$, where h is the generator of QR_n from the introduction of the signature scheme in Section 6.4.3.

6.6.1 Completeness

For completeness, we have to show that if the prover $\mathcal{P}$ knows a signature and follows the protocol, the signer S must be able to verify this. Hence, we must show that the following verification equations hold.

$$c^t U_1 = T_1^{y_e} \prod_{i=1}^{m} a_i^{y_i} b^{y_s} h^{-z_w} \bmod n \tag{6.15}$$

$$T_2^t U_2 = g_1^{y_e} g_2^{y_w} h^{y_r} \bmod n \tag{6.16}$$

$$U_3 = T_2^{-y_e} g_1^{z_e} g_2^{z_w} h^{z_r} \bmod n \tag{6.17}$$

Assuming that the prover knows a CL signature and follows the protocol, completeness follows by evaluating Equations (6.18)–(6.20), which comprise the last steps in Figure 6.3. Note, that this is a necessary step in the overall verification of *PoKSign* but does not prove knowledge of a CL signature, yet.

$$\begin{aligned}
c^t U_1 &\stackrel{?}{=} (v^e \prod_{i=1}^{m} a_i^{x_i} b^s)^t (T_1^{r_e} \prod_{i=1}^{m} a_i^{r_i} b^{r_s} h^{-r_{ew}}) = v^{et} T_1^{r_e} \prod_{i=1}^{m} a_i^{x_i t + r_i} b^{st + r_s} h^{-r_{ew}} \\
&= v^{et} T_1^{r_e} \prod_{i=1}^{m} a_i^{y_i} b^{y_s} h^{-r_{ew}} (h^{z_w} h^{-z_w}) = v^{et} T_1^{r_e} \prod_{i=1}^{m} a_i^{y_i} b^{y_s} h^{(ewt + r_{ew}) - r_{ew}} h^{-z_w} \\
&= h^{ewt} v^{et} T_1^{r_e} \prod_{i=1}^{m} a_i^{y_i} b^{y_s} h^{-z_w} = (vh^w)^{et} T_1^{r_e} \prod_{i=1}^{m} a_i^{y_i} b^{y_s} h^{-z_w} \\
&= T_1^{et} T_1^{r_e} \prod_{i=1}^{m} a_i^{y_i} b^{y_s} h^{-z_w} \\
&= T_1^{y_e} \prod_{i=1}^{m} a_i^{y_i} b^{y_s} h^{-z_w} \bmod n
\end{aligned} \tag{6.18}$$

$$\begin{aligned}
T_2^t U_2 &\stackrel{?}{=} (g_1^e g_2^w h^r)^t (g_1^{r_e} g_2^{r_w} h^{r_r}) = g_1^{et + r_e} g_2^{wt + r_w} h^{rt + r_r} \\
&= g_1^{y_e} g_2^{y_w} h^{y_r} \bmod n
\end{aligned} \tag{6.19}$$

Prover $\mathcal{P}$		**Verifier $\mathcal{V}$**
Common Input:	Commitment key PK :=	$(A, b, c, n, g_1, g_2, h)$, $A := (a_1, a_2, \ldots, a_m)$
	Length parameters	$\ell_x, \ell_e, \ell_s, \ell_n, \ell_{\mathcal{H}}, \ell_\phi$
Prover's Input:	Signature:	(v, e, s)
	Message X :=	$(x_1, x_2, \ldots, x_m)$

Choose:
$w, r \in_R [0, 2^{\ell_n+\ell_\phi} - 1]$;
$r_w, r_r \in_R [0, 2^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi} - 1]$;
$r_s \in_R [0, 2^{\ell_s+\ell_{\mathcal{H}}+\ell_\phi} - 1]$;
$r_1, \ldots, r_m \in_R [0, 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi} - 1]$;
$r_e \in_R [0, 2^{\ell_e+\ell_{\mathcal{H}}+\ell_\phi} - 1]$;
$r_{ew}, r_{er} \in_R [0, 2^{\ell_n+\ell_e+\ell_{\mathcal{H}}+2\ell_\phi} - 1]$;
$r_{ee} \in_R [0, 2^{2\ell_e+\ell_{\mathcal{H}}+\ell_\phi} - 1]$;
$T_1 := vh^w \bmod n$;
$T_2 := g_1^e g_2^w h^r \bmod n$;
$U_1 := T_1^{r_e} \prod_{i=1}^m a_i^{r_i} b^{r_s} h^{-r_{ew}} \bmod n$;
$U_2 := g_1^{r_e} g_2^{r_w} h^{r_r} \bmod n$;
$U_3 := T_2^{-r_e} g_1^{r_{ee}} g_2^{r_{ew}} h^{r_{er}} \bmod n$

$\xrightarrow{T_1, T_2, U_1, U_2, U_3}$

$\xleftarrow{t}$

Choose:
$t \in_R [0, 2^{\ell_{\mathcal{H}}} - 1]$;

$y_e := et + r_e$;
For $i = 1, \ldots, m$:
$y_i := x_i t + r_i$;
$y_s := st + r_s$;
$y_w := wt + r_w$;
$y_r := rt + r_r$;
$z_e := eet + r_{ee}$;
$z_w := ewt + r_{ew}$;
$z_r := ert + r_{er}$

$\xrightarrow{y_e, y_1, \ldots, y_m, y_s, y_w, y_r, z_e, z_w, z_r}$

Verify:
$c^t U_1 \stackrel{?}{=} T_1^{y_e} \prod_{i=1}^m a_i^{y_i} b^{y_s} h^{-z_w}$;
$T_2^t U_2 \stackrel{?}{=} g_1^{y_e} g_2^{y_w} h^{y_r}$;
$U_3 \stackrel{?}{=} T_2^{-y_e} g_1^{z_e} g_2^{z_w} h^{z_r}$;
$y_e - t2^{\ell_e - 1} \stackrel{?}{\in} \{0,1\}^{\ell_e+\ell_{\mathcal{H}}+\ell_\phi}$;
For $i = 1, \ldots, m$:
$y_i \stackrel{?}{\in} \{0,1\}^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi+1}$;

Figure 6.4: *Detailed protocol $PoKSign$*

$$
\begin{aligned}
U_3 \stackrel{?}{=} & T_2^{-r_e} g_1^{r_{ee}} g_2^{r_{ew}} h^{r_{er}} = (g_1^e g_2^w h^r)^{-r_e} g_1^{r_{ee}} g_2^{r_{ew}} h^{r_{er}} \\
&= g_1^{-er_e} g_2^{-wr_e} h^{-rr_e} g_1^{r_{ee}} g_2^{r_{ew}} h^{r_{er}} (g_1^{-eet} g_2^{-ewt} h^{-ret})(g_1^{eet} g_2^{ewt} h^{ret}) \\
&= g_1^{-e(et+r_e)} g_2^{-w(et+r_e)} h^{-r(et+r_e)} g_1^{eet+r_{ee}} g_2^{ewt+r_{ew}} h^{ret+r_{er}} \\
&= (g_1^e g_2^w h^r)^{-(et+r_e)} g_1^{z_e} g_2^{z_w} h^{z_r} \\
&= T_2^{-y_e} g_1^{z_e} g_2^{z_w} h^{z_r} \bmod n
\end{aligned} \tag{6.20}
$$

6.6.2 Knowledge Error

Again, a cheating prover $\mathcal{P}'$'s responses may pass the verification, if she correctly predicted $\mathcal{V}$'s challenge t at the time the commitments were formed. Assume that $\mathcal{P}'$ guesses t_0 and forms her commitments and responses as follows.

$$
\begin{aligned}
y'_1, \ldots, y'_m &\in_R [0, 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi} - 1] & y'_e &\in_R [0, 2^{\ell_e+\ell_{\mathcal{H}}+\ell_\phi} - 1] \\
y'_s &\in_R [0, 2^{\ell_s+\ell_{\mathcal{H}}+\ell_\phi} - 1] & y'_w, y'_r &\in_R [0, 2^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi} - 1] \\
z'_e &\in_R [0, 2^{2\ell_e+\ell_{\mathcal{H}}+\ell_\phi} - 1] & z'_w, z'_r &\in_R [0, 2^{\ell_n+\ell_e+\ell_{\mathcal{H}}+2\ell_\phi} - 1] \\
T'_1, T'_2 &\in_R QR_n & U'_1 &:= c^{-t_0} T_1'^{y'_e} \prod_{i=1}^{m} a_i^{y'_i} b^{y'_s} h^{-z'_w} \bmod n \\
U'_2 &:= T_2'^{-t_0} g_1^{y'_e} g_2^{y'_w} h^{y'_r} \bmod n & U'_3 &:= T_2'^{-y'_e} g_1^{z'_e} g_2^{z'_w} h^{z'_r} \bmod n
\end{aligned}
$$

It can be easily verified that if $\mathcal{P}'$'s guess t_0 turns out to be correct, i.e., $t = t_0$, the values above will satisfy the verification Equations (6.15)–(6.17). Since $\mathcal{P}'$ has only one try in each run of the protocol and the success probability is determined by the length of the challenge, we have a knowledge error of $2^{-\ell_{\mathcal{H}}}$.

6.6.3 Proof of Knowledge

In order to show that *PoKSign* is a proof of knowledge of a CL signature, we construct a knowledge extractor $\mathcal{K}$—more precisely a signature extractor— that retrieves the values $x_1, x_2, \ldots, x_m, e, w, r, s, e_e, w_e, r_e$ by accessing and rewinding a prover $\mathcal{P}^*$ that has probability $\pi_{\mathcal{P}^*}$ to successfully complete the proof of knowledge at hand. This knowledge extractor works like the one presented in the paragraph labelled "proof of knowledge" in Section 6.5 and therefore, its success probability will be close to $\pi_{\mathcal{P}^*}$ and it will run in expected polynomial time. Now, we condition on the event that $\mathcal{K}$ got two accepting transcripts from $\mathcal{P}^*$ for the same commitments, but different challenges t and t'. Let

$$T := [T_1, T_2, U_1, U_2, U_3, t, y_1, y_2, \ldots, y_m, y_e, y_w, y_r, y_s, z_e, z_w, z_r]$$

and

$$T' := [T_1, T_2, U_1, U_2, U_3, t', y'_1, y'_2, \ldots, y'_m, y'_e, y'_w, y'_r, y'_s, z'_e, z'_w, z'_r]$$

be the transcripts that $\mathcal{K}$ received from its interaction with $\mathcal{P}^*$. Further suppose $\mathcal{K}$ provided $\mathcal{P}^*$ with the following public key as common input

$$(a_1 := b^{\alpha_1^*}, a_2 := b^{\alpha_2^*}, \ldots, a_m := b^{\alpha_m^*}, b := h^{\beta^*}, c, n, g_1 := h^{\gamma_1^*}, g_2 := h^{\gamma_2^*}, h),$$

where the integers $\alpha_1^*, \ldots, \alpha_m^*, \beta^*, \gamma_1^*, \gamma_2^*$ have been chosen such that none has factors smaller than $2^{\ell_{\mathcal{H}}}$. From this and Equations (6.15)–(6.17), the following equations can be derived.

$$\begin{aligned} c^{\Delta t} &:= c^{(t-t')} \\ &= T_1^{(y_e - y'_e)} \prod_{i=1}^{m} a_i^{(y_i - y'_i)} b^{(y_s - y'_s)} h^{(z'_w - z_w)} \\ &=: T_1^{\Delta y_e} \prod_{i=1}^{m} a_i^{\Delta y_i} b^{\Delta y_s} h^{\Delta z_w} = T_1^{\Delta y_e} \prod_{i=1}^{m} h^{\beta^* \alpha_i^* \Delta y_i} h^{\beta^* \Delta y_s} h^{\Delta z_w} \\ &= T_1^{\Delta y_e} h^{\sum_{i=1}^{m} \beta^* \alpha_i^* \Delta y_i + \beta^* \Delta y_s + \Delta z_w} \bmod n \end{aligned} \tag{6.21}$$

$$\begin{aligned} T_2^{\Delta t} &:= T_2^{(t-t')} = g_1^{(y_e - y'_e)} g_2^{(y_w - y'_w)} h^{(y_r - y'_r)} \\ &=: g_1^{\Delta y_e} g_2^{\Delta y_w} h^{\Delta y_r} \\ &= h^{\gamma_1^* \Delta y_e + \gamma_2^* \Delta y_w + \Delta y_r} \bmod n \end{aligned} \tag{6.22}$$

$$\begin{aligned} T_2^{\Delta y_e} &:= T_2^{y_e - y'_e} \\ &= g_1^{z_e - z'_e} g_2^{z_w - z'_w} h^{z_r - z'_r} \\ &=: g_1^{\Delta z_e} g_2^{\Delta z_w} h^{\Delta z_r} = h^{\gamma_1^* \Delta z_e} h^{\gamma_2^* \Delta z_w} h^{\Delta z_r} \\ &= h^{\gamma_1^* \Delta z_e + \gamma_2^* \Delta z_w + \Delta z_r} \bmod n \end{aligned} \tag{6.23}$$

We start by arguing that Δt divides $\Delta y_e, \Delta y_w$, and Δy_r from Equation (6.22) allowing us to compute e, w, r in $\mathbb{Z}$ (see Equations (6.26)), or otherwise $\mathcal{K}$ yields an algorithm which allows to compute roots modulo n. The argument is basically the same as in Section 6.5.3 and consequently, we similarly assume that $T_1 \neq 1$ and $T_2 \neq 1$, as for these values $\mathcal{K}$ can always output the trivial DLREP $(0, 0, \ldots, 0)$. Let $u := \gamma_1^* \Delta y_e + \gamma_2^* \Delta y_w + \Delta y_r$ and assume for contradiction that $\Delta t \nmid u$. Then, by the extended Euclidian algorithm, we get values ϑ, ν such that $\Delta t \vartheta + \nu u = 1$. Next, $\mathcal{K}$ computes the value $z \in \mathbb{Z}_n^*$ such that $h = h^{\Delta t \vartheta + \nu u} = h^{\Delta t \vartheta} (h^u)^\nu = h^{\Delta t \vartheta} (T_2^{\Delta t})^\nu = (h^\vartheta T_2^\nu)^{\Delta t} =: z^{\Delta t} \bmod n$, which is a Δt-th root of h. This, however, contradicts the strong RSA assumption as this would turn $\mathcal{K}$ into an algorithm that, on input the

modulo n and an element $h \in \mathbb{Z}_n^*$, can compute a Δt-th root of h modulo n in expected polynomial time. Hence, we conclude that Δt must divide $\Delta y_e, \Delta y_w$, and Δy_r, since none of γ_1^*, γ_2^* can be divided by Δt as, by construction, none of them has a factor smaller than $2^{\ell_{\mathcal{H}}} > \Delta t$.

Using this result and Equation (6.23), we will show that Δt must also divide $\Delta z_e, \Delta z_w$, and Δz_r allowing us to compute e_e, w_e, r_e according to Equations (6.26). We already know that $\Delta t \mid \Delta y_e$, i.e., $\Delta y_e = \tilde{y}_e \Delta t$. Therefore, we can rewrite Equation (6.23) —using Equation (6.22)— as follows.

$$\begin{aligned} T_2^{\Delta y_e} &= T_2^{\tilde{y}_e \Delta t} = (T_2^{\Delta t})^{\tilde{y}_e} \\ &= (g_1^{\Delta y_e} g_2^{\Delta y_w} h^{\Delta y_r})^{\tilde{y}_e} \\ &= g_1^{\Delta z_e} g_2^{\Delta z_w} h^{\Delta z_r} \bmod n \end{aligned} \tag{6.24}$$

From Equation (6.24), it follows that either $\Delta y_e \tilde{y}_e = \Delta z_e$ or otherwise $\mathcal{K}$ could be turned into an expected polynomial time algorithm for computing roots in $\mathbb{Z}_n^*$, as in this case a multiple of $\mathrm{ord}(QR_n)$ could be retrieved. Because we already know that $\Delta t \mid \Delta y_e$ it also follows that $\Delta z_e = (\tilde{y}_e \Delta t)\tilde{y}_e$, i.e., that $\Delta t \mid \Delta z_e$. Using the same arguments, it can be seen that Δt divides z_w and z_r.

Next, we divide Equation (6.21) by $T_1^{\Delta y_e}$ and, using $\Delta y_e = \tilde{y}_e \Delta t$, rewrite it as shown in Equation (6.25).

$$(c/T_1^{\tilde{y}_e})^{\Delta t} = h^{\sum_{i=1}^{m} \beta^* \alpha_i^* \Delta y_i + \beta^* \Delta y_s + \Delta z_w} \bmod n \tag{6.25}$$

This essentially gives us the situation of Equation (6.6) from the argument for the proof of knowledge in Section 6.5.3. Thus, using similar arguments, we can conclude that Δt divides all Δy_i and Δy_s. It follows that $\mathcal{K}$ can extract the quantities $x_1, x_2, \ldots, x_m, w, r, s, e_e, w_e$, and r_e by the following computations.

$$\begin{aligned} x_1 &:= \frac{\Delta y_1}{\Delta t} = \frac{y_1 - y_1'}{t - t'} & e &:= \frac{\Delta y_e}{\Delta t} = \frac{y_e - y_e'}{t - t'} & e_e &:= \frac{\Delta z_e}{\Delta t} = \frac{z_e - z_e'}{t - t'} \\ x_2 &:= \frac{\Delta y_2}{\Delta t} = \frac{y_2 - y_2'}{t - t'} & w &:= \frac{\Delta y_w}{\Delta t} = \frac{y_w - y_w'}{t - t'} & w_e &:= \frac{\Delta z_w}{\Delta t} = \frac{z_w - z_w'}{t - t'} \\ &\ldots & r &:= \frac{\Delta y_r}{\Delta t} = \frac{y_r - y_r'}{t - t'} & r_e &:= \frac{\Delta z_r}{\Delta t} = \frac{z_r - z_r'}{t - t'} \\ x_m &:= \frac{\Delta y_m}{\Delta t} = \frac{y_m - y_m'}{t - t'} & s &:= \frac{\Delta y_s}{\Delta t} = \frac{y_s - y_s'}{t - t'} \end{aligned} \tag{6.26}$$

This proofs knowledge of $(x_1, x_2, \ldots, x_m, s, w_e)$ which is the DLREP of (T_1^e) with respect to the basis $(a_1, a_2, \ldots, a_m, b, h)$ — note that the DLREP of T_1 cannot be proven by $\mathcal{P}$, assuming that she does not know the factorisation/order of n. Furthermore, *PoKSign* proves knowledge of the values (e, w, r) which is the DLREP

of T_2, with respect to the basis (g_1, g_2, h), and of (e_e, w_e, r_e) which in turn is the DLREP of (T_2^e), also with respect to (g_1, g_2, h). Now, we have $T_1/h^w = v$ and if $w_e = w \cdot e$, we must have $(T_1^e)/h^{w_e} = (T_1^e)/h^{we} = (T_1/h^w)^e = v^e = c/(\prod_{i=1}^m a_i^{x_i} b^s)$ — note that the last equation signifies only the first step in the verification of a CL signature. Furthermore, if e is also within the right bound —the second step in a CL signature verification— $\mathcal{K}$ has extracted a valid CL signature (v, e, s) of the message tuple $(x_1, x_2, \ldots, x_m)$.

6.6.4 Zero-Knowledge

We will now show that *PoKSign* is an honest-verifier statistical zero-knowledge proof of knowledge, i.e., we will show that the distribution of the transcript

$$T := [T_1, T_2, U_1, U_2, U_3, t, y_1, y_2, \ldots, y_m, y_e, y_w, y_r, y_s, z_e, z_w, z_r], \tag{6.27}$$

created in a conversation between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$, is statistically indistinguishable from a transcript

$$T' := [T_1, T_2', U_1', U_2', U_3', t', y_1', y_2', \ldots, y_m', y_e', y_w', y_r', y_s', z_e', z_w', z_r'], \tag{6.28}$$

created by a simulator $\mathcal{S}$ from computed quantities and values uniformly and randomly chosen from their respective intervals as follows.

$$\begin{aligned}
y_1', \ldots, y_m' &\in_R [0, 2^{\ell_x + \ell_{\mathcal{H}} + \ell_\phi} - 1] & y_e' &\in_R [0, 2^{\ell_e + \ell_{\mathcal{H}} + \ell_\phi} - 1] \\
y_s' &\in_R [0, 2^{\ell_s + \ell_{\mathcal{H}} + \ell_\phi} - 1] & y_w', y_r' &\in_R [0, 2^{\ell_n + \ell_{\mathcal{H}} + 2\ell_\phi} - 1] \\
z_e' &\in_R [0, 2^{2\ell_e + \ell_{\mathcal{H}} + \ell_\phi} - 1] & z_w', z_r' &\in_R [0, 2^{\ell_n + \ell_e + \ell_{\mathcal{H}} + 2\ell_\phi} - 1] \\
t' &\in_R [0, 2^{\ell_{\mathcal{H}}} - 1]
\end{aligned}$$

$$T_2' := h^{\vartheta} \bmod n, \quad \vartheta \in_R [0, 2^{\ell_n + \ell_\phi} - 1] \tag{6.29}$$

Using the values above, $\mathcal{S}$ computes the simulated witnesses U_1', U_2', U_3', as shown in Equations (6.30)–(6.32), where we use the quantities γ_1 and γ_2 to denote $g_1 = h^{\gamma_1} \bmod n$ and $h^{\gamma_2} \bmod n$, respectively.

$$\begin{aligned}
U_1' &:= c^{-t'} T_1^{y_e'} \prod_{i=1}^m a_i^{y_i'} b^{y_s'} h^{-z_w'} = h^{-\gamma t} T_1^{y_e'} \prod_{i=1}^m h^{\beta \alpha_i y_i'} h^{\beta y_s'} h^{-z_w'} \\
&= T_1^{y_e'} h^{\beta(y_s' + \sum_{i=1}^m \alpha_i y_i') - z_w'} \bmod n
\end{aligned} \tag{6.30}$$

$$\begin{aligned}
U_2' &:= T_2^{-t'} g_1^{y_e'} g_2^{y_w'} h^{y_r'} = h^{-\vartheta t'} h^{\gamma_1 y_e'} h^{\gamma_2 y_w'} h^{y_r'} \\
&= h^{-\vartheta t' + \gamma_1 y_e' + \gamma_2 y_w' + y_r'} \bmod n
\end{aligned} \tag{6.31}$$

$$\begin{aligned} U_3' &:= T_2^{-y_e} g_1^{z_e'} g_2^{z_w'} h^{z_r'} = h^{-\vartheta y_e} h^{\gamma_1 z_e'} h^{\gamma_2 z_w'} h^{z_r'} \\ &= h^{-\vartheta y_e + \gamma_1 z_e' + \gamma_2 z_w' + z_r'} \bmod n \end{aligned} \tag{6.32}$$

In Equations (6.33)–(6.36), we give a recap of the commitments and witnesses used in a real conversation between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$, in order to argue that these are statistically indistinguishable from the choices of the simulator $\mathcal{S}$.

$$T_2 = g_1^e g_2^w h^r = (h^{\gamma_1})^e (h^{\gamma_2})^w h^r = h^{[\gamma_1 e + \gamma_2 w] + r} \bmod n \tag{6.33}$$

$$\begin{aligned} U_1 &= T_1^{r_e} \prod_{i=1}^{m} a_i^{r_i} b^{r_s} h^{-r_{ew}} = T_1^{r_e} \prod_{i=1}^{m} b^{-\alpha_i r_i} b^{r_s} h^{-r_{ew}} \\ &= T_1^{r_e} (h^{\beta})^{r_s - \sum_{i=1}^{m} \alpha_i r_i} h^{-r_{ew}} = T_1^{r_e} h^{\beta(r_s - \sum_{i=1}^{m} \alpha_i r_i) - r_{ew}} \bmod n \end{aligned} \tag{6.34}$$

$$U_2 = g_1^{r_e} g_2^{r_w} h^{r_r} = (h^{\gamma_1})^{r_e} (h^{\gamma_2})^{r_w} h^{r_r} = h^{[\gamma_1 r_e + \gamma_2 r_w] + r_r} \bmod n \tag{6.35}$$

$$\begin{aligned} U_3 &= T_2^{-r_e} g_1^{r_{ee}} g_2^{r_{ew}} h^{r_{er}} = T_2^{-r_e} (h^{\gamma_1})^{r_{ee}} (h^{\gamma_2})^{r_{ew}} h^{r_{er}} \\ &= h^{[\gamma_1 r_{ee} + \gamma_2 r_{ew}] + r_{er}} \bmod n \end{aligned} \tag{6.36}$$

We start by arguing that T_2 and T_2' are statistically indistinguishable. For this, let $\xi := \gamma_1 e + \gamma_2 w$ (see Equation (6.33)). As in the discussion of zero knowledge from Section 6.5.4, we denote quantities x which are 'reduced' modulo $\mathrm{ord}(QR_n)$ by $\underline{x}$. First, note that the distributions of ϑ from T_2' (see Equation (6.29)) and of r from T_2 (see Equation (6.33)) are identical. Since the value $\underline{\xi}$ can be regarded as an arbitrary offset in $\mathbb{Z}_{\mathrm{ord}(QR_n)}$, it is clear that the distribution of the sum $\varsigma := \underline{\xi} + r$ and of r are identical on $\mathbb{Z}_{\mathrm{ord}(QR_n)}$. Hence, the distribution of $\underline{\vartheta}$ and of $\underline{\varsigma}$ is identical, and therefore the distribution of T_2 and T_2' is (statistically) indistinguishable.

The same arguments used for the indistinguishability of (T_2, T_2') also apply to the remaining 'pairs' $(U_1, U_1'), (U_2, U_2')$, and (U_3, U_3'). That is, for every component U, its exponent can be rewritten as a sum $\varsigma := \underline{\xi} + \rho$, where $\underline{\xi}$ takes the role of an offset in $\mathbb{Z}_{\mathrm{ord}(QR_n)}$ and ρ is the portion that makes the sum uniformly distributed on $\mathbb{Z}_{\mathrm{ord}(QR_n)}$. Now, let V be a random variable for the component from a real conversation between $\mathcal{P}$ and $\mathcal{V}$ and let V' be a random variable for the component chosen by $\mathcal{S}$. Then we have

$$Pr(V = U_i^{\xi+\rho} \bmod n) = Pr(V' = U_i'^{\xi'+\rho'} \bmod n),$$

where ξ, ρ are to be taken from the matching row of component U_i and ξ', ρ' are to be taken from the matching row of component U_i', all found in the following table.

Component	ξ / ξ'	ρ / ρ'	$Pr(R=\rho)$ / $Pr(R'=\rho')$
U_1	$\beta(r_s - \sum_{i=1}^{m} \alpha_i r_i)$	r_e	$1/2^{\ell_e+\ell_{\mathcal{H}}+\ell_\phi}$
U_1'	$\beta(y_s' + \sum_{i=1}^{m} \alpha_i y_i')$	y_e'	
U_2	$\gamma_1 r_e + \gamma_2 r_w$	r_r	$1/2^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi}$
U_2'	$-\vartheta t' + \gamma_1 y_e' + \gamma_2 y_w'$	y_r'	
U_3	$\gamma_1 r_{ee} + \gamma_2 r_{ew}$	r_{er}	$1/2^{\ell_n+\ell_e+\ell_{\mathcal{H}}+2\ell_\phi}$
U_3'	$-\vartheta y_e + \gamma_1 z_e' + \gamma_2 z_w'$	z_r'	

This means, that all pairs (U_i, U_i') are component-wise identically distributed on QR_n and hence, (statistically) indistinguishable.

Finally, we are going to argue that the remaining corresponding values from each transcript are statistically indistinguishable. As we are in the honest verifier model, $\mathcal{V}$ will choose his challenge t according to the protocol and since $\mathcal{S}$ does this as well for t', both challenges are identically distributed. For the rest of the values from the transcript T, note that every response $y_1, y_2, \ldots, y_m, y_e, y_w, y_r, y_s, z_e, z_w, z_r$ of $\mathcal{P}$ can be written as the sum of two random variables, i.e., $Y = U + R$, and hence, Y defines a random variable by itself. Following the approach from Section 6.5, the statistical indistinguishability of the distribution of $\mathcal{P}$'s responses y can be seen by determining their respective distribution functions $Pr(Y = y)$. Any response y can be written as $y = \upsilon + \rho$, where $\upsilon \in [0, 2^{\zeta} - 1]$ and $\rho \in_R [0, 2^{\zeta+\ell_\phi} - 1]$. Hence, by evaluating Equation (6.37) for any y from the table below, the distribution of any response y can be found.

Response y	Length ζ of υ
$y_1, \ldots, y_m$	$\ell_x + \ell_{\mathcal{H}}$
y_e	$\ell_e + \ell_{\mathcal{H}}$
y_w, y_r	$\ell_n + \ell_{\mathcal{H}} + \ell_\phi$
y_s	$\ell_s + \ell_{\mathcal{H}}$
z_e	$2\ell_e + \ell_{\mathcal{H}}$
z_w, z_r	$\ell_n + \ell_e + \ell_{\mathcal{H}} + \ell_\phi$

$$Pr(Y = y) = \frac{1}{2^{\zeta+\ell_\phi}} \sum_{\rho=0}^{2^{\zeta+\ell_\phi}-1} Pr(U = y - \rho) \tag{6.37}$$

Equation (6.38) summarises the distribution for any value y from the table above and Estimation (6.39) shows that the statistical difference between the distribution of any y, with respect to Y, and the distribution of a random variable Y' that is uniformly distributed on the interval $[0, 2^{\zeta+\ell_\phi} - 1]$ is a negligible quantity in the

security parameter ℓ_ϕ.

$$Pr(Y=y) = \begin{cases} = 0 & \text{for} \quad y < 0 & \text{(I)} \\ \leq 2^{-(\zeta+\ell_\phi)} & \text{for} \quad 0 \leq y < 2^\zeta - 1 & \text{(II)} \\ = 2^{-(\zeta+\ell_\phi)} & \text{for} \quad 2^\zeta - 1 \leq y \leq 2^{\zeta+\ell_\phi} - 1 & \text{(III)} \\ \leq 2^{-(\zeta+\ell_\phi)} & \text{for} \quad 2^{\zeta+\ell_\phi} \leq y \leq 2^{\zeta+\ell_\phi} + 2^\zeta - 1 & \text{(IV)} \\ = 0 & \text{for} \quad y \geq 2^{\zeta+\ell_\phi} + 2^\zeta & \text{(V)} \end{cases} \tag{6.38}$$

$$\begin{aligned} \sum_{y\in\mathbb{Z}} |Pr(Y=y) - Pr(Y'=y)| &= 1 - \left(\frac{2^{\ell_\zeta+\ell_\phi} - 1 - (2^{\ell_\zeta} - 1) + 1}{2^{\ell_\zeta+\ell_\phi}} \right) \\ &= 1 - \left(\frac{2^{\ell_\zeta}(2^{\ell_\phi} - 1) + 1}{2^{\ell_\zeta+\ell_\phi}} \right) \\ &= 1 - \frac{2^{\ell_\phi} - 1}{2^{\ell_\phi}} - \frac{1}{2^{\ell_\zeta+\ell_\phi}} = \frac{1}{2^{\ell_\phi}} - \frac{1}{2^{\ell_\zeta+\ell_\phi}} \\ &< \frac{1}{2^{\ell_\phi}} \end{aligned} \tag{6.39}$$

Hence, we have shown that the distributions of the transcripts T and T' from Equations (6.27) and (6.28), respectively, are statistically indistinguishable. Therefore, we can conclude that *PoKSign* is an honest-verifier statistical zero-knowledge proof of knowledge of a CL signature. ■

6.7 Construction of the Coupon System

In this section we propose a concrete scheme for a coupon system that allows issuance and redemption of multi-coupons. The scheme is comprised of two protocols, *Issue* and *Redeem*, which are carried out between a user U and a vendor V, and an Initialisation algorithm. Major portions of the following text had been previously published in [CES+05].

6.7.1 System Setup

V initialises the system by running his key generation algorithm with input 1^k, where k is a security parameter:

$$(A, b, c, n, h, p, q) \leftarrow KeyGen_m^{CL}(1^k).$$

He then randomly chooses $g_1, g_2 \in_R \langle h \rangle$ and publishes his public key $PK = (A, b, c, n, g_1, g_2, h)$ along with the length parameters ℓ_x, ℓ_e, ℓ_n, ℓ_s, $\ell_{\mathcal{H}}$ and ℓ_ϕ. In addition, he publishes a non-interactive proof that $g_1, g_2 \in \langle h \rangle$. In essence, the latter

is a simple proof of knowledge of a DLREP of g_1 and g_2 with respect to the base h that has been made non-interactive by using the Fiat-Shamir heuristic [FS87], i.e., the challenger/verifier is 'replaced' by a cryptographically strong hash function.

6.7.2 Protocols

Issue. In the issue protocol, U chooses serial numbers $x_i \in_R [0, 2^{\ell_x} - 1]$, for $i = 1, \ldots, m$, and sets $X := (x_1, \ldots, x_m)$. Then U runs with V

$$(v, e, s) \leftarrow \mathit{BlindSign}_{(A,b,c,n)}(X)$$

to obtain a blind CL signature (v, e, s) on X. The tuple $M := (X, v, e, s)$ will act as the user's multi-coupon.

Redeem. In the redeem protocol, U (randomly) chooses an unspent coupon x_j from the tuple X, sets $x := x_j$ and $X' := X \setminus (x)$. The value x then becomes a common input to the redeem protocol, i.e., it is sent to V, and subsequently plays the role of the single coupon to be deducted from the customer's multi-coupon. Next U proves to V that she is in possession of a valid multi-coupon ($\mathcal{V}$'s signature on X) containing x without revealing the signature itself.

Proving that x is the j-th element of the signed message tuple X can be easily done by using Remark 1 from Section 6.4.3: V and U compute for themselves a modified public key $PK_j := (A \setminus (a_j), b, c/a_j^x)$ and run

$$\mathit{ind} \leftarrow \mathit{PoKSign}_{(A \setminus (a_j), b, c/a_j^x, n, g_1, g_2, h)}(X', v, e, s).$$

This way, the signature (v, e, s) and X' are still kept from V, though the index j is disclosed by the public key PK_j.

Unfortunately, the disclosure of the index j violates the unlinkability requirement. To see this, simply suppose that two different coupons x and y are revealed both with respect to the base a_j from the CL signature. In this case, V immediately learns that the corresponding multi-coupons are different, since clearly $(z_1, z_2, \ldots, z_{j-1}, x, z_{j+1}, \ldots, z_m) \neq (z'_1, z'_2, \ldots, z'_{j-1}, y, z'_{j+1}, \ldots, z'_m)$, where the z_i and z'_i are the other single coupons from the multi-coupon of x and y, respectively.

So in fact, by revealing the index j, more is proven than just x_j being included in the multi-coupon's signature. It is proven that x is included in the multi-coupon's signature *and* that x is the j-th component of the message tuple X. In other words, the protocol yields an additional proof that x is the j-th element in an *ordered* set of messages $(x_1, x_2, \ldots, x_m)$, where the x_i, with $i \neq j$, are unknown to V. However, in order to retain unlinkability the index must not be disclosed and we need to prove that x is contained in an *unordered* set of messages, i.e., $x \in \{x_1, x_2, \ldots, x_m\}$ where the x_i, $i \neq j$, are unknown to V.

In order to overcome this index problem, U runs an extended version of the *PoKSign* protocol from Figure 6.3 which proves that x is part of the signature but does not disclose the index of the spent coupon. The idea for this extension is as follows. Instead of disclosing to V which concrete public key PK_i $(i = 1, \ldots, m)$ is to be used for the verification, U proves that one of the public keys PK_i is the verification key with respect to the signature (v, e, s) on the message X'. For this, the proof *PoK* is extended with the *PoKOr* protocol which adds the proof for the term $\bigvee_{i=1}^{m} C_i = T_1^{\epsilon} \prod_{l \in \{1,\ldots,m\}, l \neq i} a_l^{\xi_l} b^{\sigma} h^{\omega}$ to *PoK* (see also Section 6.4.4). — Note that the terms $C_i = c/a_i^x$ are computed by both U and V.— Using *PoKOr*, U proves that she knows the DLREP of one of the C_i with respect to its corresponding bases $(T_1, A \backslash (a_i), b, h)$ without revealing which one — since x is equal to x_j, the commitment $C_j = c/a_j^x$ must have a representation to the bases $(T_1, A \backslash (a_j), b, h)$ which is known to U. Also note that this proves knowledge of the signature $(T = vb^w, e, s^* = s - ew)$ with respect to the public key $PK_j := (A \backslash (a_j), b, c/a_j^x)$. This is according to Remark 1 the same as proving it with respect to the public key (A, b, c) and, by Remark 2, the randomised signature (T_1, e, s^*) is, from V's point of view, equivalent to the signature (v, e, s). Hence, x must be part of a valid multi-coupon, i.e., a component from a message tuple that was signed by the vendor.

Eventually, if the proof protocol yields *accept* then V is convinced that U owns a signature on an m-tuple X which contains x. At last, V checks if x is already stored in his database of redeemed coupons. If it is not, V accepts x as a valid coupon and will grant the service. □

Detailed Description. In Figure 6.5, the redeem protocol described above is shown in detail. We will only briefly go through the protocol, as most parts of it have already been discussed in Section 6.6. For the run of the protocol shown in Figure 6.5, it is assumed that U chose her j-th coupon for redemption, i.e., the discrete logarithm to the base a_j. Except for this choice in the beginning, the redeem protocol proceeds exactly as the protocol *PoKSign* from Figure 6.4, until the procedure *Prepare* is called. *Prepare* is shown in Figure 6.6.

In the procedure *Prepare*, the protocol elements for the *PoKOr* part of the protocol are computed. Essentially, *Prepare* produces $m - 1$ random transcripts of the *PoKSign* protocol, in the same way as a simulator for the protocol would do (c.f. Section 6.6.4). One such transcript is computed in each loop (see line 2–11). Each 'forged' transcript is computed with respect to a different public key $PK_i := (A \setminus (a_i), b, c/a_i)$. The only transcript that is omitted in the loop is the one for the j-th public key PK_j. This public key will be used in the main part of *Redeem* to eventually produce the only 'real' transcript, i.e., the one proving that x_j, a.k.a. x, is part of the multi-coupon. In lines 12–16, the resulting elements of the transcripts are stored in matrices for later reference. The bullets ("•") in the

User U		**Vendor** V
Common Input:	Commitment key PK :=	$(A, b, c, n, g_1, g_2, h)$, $A := (a_1, a_2, \ldots, a_m)$
	Length parameters	$\ell_x, \ell_e, \ell_s, \ell_n, \ell_{\mathcal{H}}, \ell_\phi$
User's Input:	Signature:	(v, e, s)
	Message X :=	$(x_1, x_2, \ldots, x_m)$

$x := x_j$;
Choose:
$w, r \in_R [0, 2^{\ell_n+\ell_\phi} - 1]$;
$r_w, r_r \in_R [0, 2^{\ell_n+\ell_{\mathcal{H}}+2\ell_\phi} - 1]$;
$r_s \in_R [0, 2^{\ell_s+\ell_{\mathcal{H}}+\ell_\phi} - 1]$;
$r_1, \ldots, r_m \in_R [0, 2^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi} - 1]$;
$r_e \in_R [0, 2^{\ell_e+\ell_{\mathcal{H}}+\ell_\phi} - 1]$;
$r_{ew}, r_{er} \in_R [0, 2^{\ell_n+\ell_e+\ell_{\mathcal{H}}+2\ell_\phi} - 1]$;
$r_{ee} \in_R [0, 2^{2\ell_e+\ell_{\mathcal{H}}+\ell_\phi} - 1]$;
$\mathcal{R} := (e, w, r, r_w, r_r, r_s, r_1, \ldots, r_m, r_e, r_{ew}, r_{er}, r_{ee})$;
$T_1 := vh^w \bmod n$;
$T_2 := g_1^e g_2^w h^r \bmod n$;
Call *Prepare*(T_1, T_2, x, j)
Assign its result to $(\mathcal{T}, \mathcal{Y}, \mathcal{Y}_{\text{aux}}, \mathcal{U}, \mathcal{Z})$;
$U_{j,1} := T_1^{r_e} \prod_{i=1, i\neq j}^{m} a_i^{r_i} b^{r_s} h^{-r_{ew}} \bmod n$;
$U_{j,2} := g_1^{r_e} g_2^{r_w} h^{r_r} \bmod n$;
$U_{j,3} := T_2^{-r_e} g_1^{r_{ee}} g_2^{r_{ew}} h^{r_{er}} \bmod n$;
Insert $(U_{j,1}, U_{j,2}, U_{j,3})$ in row j of $\mathcal{U}$ $\xrightarrow{T_1, T_2, \mathcal{U}}$
Choose:
$\xleftarrow{t}$ $t \in_R [0, 2^{\ell_{\mathcal{H}}} - 1]$;
$t_j := \bigoplus_{i=1, i\neq j}^{m} t_i \oplus t$;
Call *Finalise*$(\mathcal{Y}, \mathcal{Y}_{\text{aux}}, \mathcal{Z}, \mathcal{R}, t_j, j)$
Assign its result to $(\mathcal{Y}, \mathcal{Y}_{\text{aux}}, \mathcal{Z})$ $\xrightarrow{x, \mathcal{T}, \mathcal{Y}, \mathcal{Y}_{\text{aux}}, \mathcal{Z}}$
Verify:
$t \stackrel{?}{=} \bigoplus_{i=1}^{m} t_i$;
For $i = 1, \ldots, m$:
$(c/a_i^x)^{t_i} U_{i,1} \stackrel{?}{=} T_1^{y_{i,e}} \prod_{l=1, l\neq i}^{m} a_l^{y_{i,l}} b^{y_{i,s}} h^{-z_{i,w}}$;
$T_2^{t_i} U_{i,2} \stackrel{?}{=} g_1^{y_{i,e}} g_2^{y_{i,w}} h^{y_{i,r}}$;
$U_{i,3} \stackrel{?}{=} T_2^{-y_{i,e}} g_1^{z_{i,e}} g_2^{z_{i,w}} h^{z_{i,r}}$;
$y_{i,e} - t_i 2^{\ell_e - 1} \stackrel{?}{\in} \{0,1\}^{\ell_e+\ell_{\mathcal{H}}+\ell_\phi}$;
For $l = 1, \ldots, i-1, i+1, \ldots, m$:
$y_{i,l} \stackrel{?}{\in} \{0,1\}^{\ell_x+\ell_{\mathcal{H}}+\ell_\phi+1}$;

Figure 6.5: *Detailed protocol* $Redeem$*: Redemption of the* j*-th Coupon*

matrices are used to denote cells which are left empty. Finally, the matrices are returned to the caller, i.e., the main protocol.

After the call to *Prepare*, the *Redeem* protocol again proceeds in the same way as the *PoKSign* protocol, i.e., witnesses $U_{j,1}, U_{j,2}, U_{j,3}$ are computed in the same way. Then, these witnesses are stored in the matrix $\mathcal{U}$ returned from *Prepare*. Now, instead of sending just these three witnesses, as in *PoKSign*, the whole matrix $\mathcal{U}$ is sent along with T_1 and T_2. Since the *PoKOr* part of the protocol is employed to prove knowledge of a signature with respect to one out of m public keys PK_i, the customer also has to sent m tuples of witnesses $(U_{*,1}, U_{*,2}, U_{*,3})$ — one for each key. In general, *PoKOr* ensures that *at least one* of the claimed statements is true. In case of *Redeem*, the (technical) statements are "x is part of a CL signature that can be verified using public key PK_i" and *only one* of them is actually true — the statement referring to PK_j.

Next, the vendor sends his challenge t. From this challenge, U computes the 'sub-challenge' t_j which is the one used in the computations for the responses that actually prove knowledge. Note that the other challenges t_i $(i \neq j)$ from $\mathcal{T}$ had been chosen beforehand, as part of the 'forging' in *Prepare*. Also note that t_j 'corrects' the xor ("$\oplus$", exclusive or) of all forged t_i to finally 'sum up' to V's challenge t. Next, *Finalise* is called which conducts the remaining computations for the user (see Figure 6.7).

In the procedure *Finalise*, the same computations of responses are made, with respect to challenge t_j and public key PK_j, which would otherwise be computed in the protocol *PoKSign* after the prover received the verifier's challenge t (c.f. Figure 6.4). Hence, these responses are the ones allowing the vendor V to eventually evaluate the or-statement to true, as these are literally the only true responses, i.e., non-forged ones. At the end, the responses are inserted in their appropriate places in the matrices $\mathcal{Y}, \mathcal{Y}_{\text{aux}}$, and $\mathcal{Z}$ and returned to the main protocol *Redeem*.

Now, U sends the coupon to be spent x, the tuple of sub-challenges $\mathcal{T}$ and the other responses $\mathcal{Y}, \mathcal{Y}_{\text{aux}}$, and $\mathcal{Z}$ to V. Note that a row in $\mathcal{Y}, \mathcal{Y}_{\text{aux}}$ and $\mathcal{Z}$ correspond to the responses y_*, z_*, respectively, sent in the prover's last step of *PoKSign*. After the vendor received U's responses, his first step is to verify that the exclusive xor of all sub-challenges from $\mathcal{T}$ 'sums up' to his own challenge t. If that is the case, he starts verifying the user's responses, almost like in the protocol *PoKSign*. The only difference is that, instead of verifying a single tuple of responses, he now has to verify m tuples of responses, owing to the m different public keys PK_i for which the verification is done. Finally, if all verifications are successful, the vendor accepts the coupon x and stores it in his database of spent coupons (steps are not shown in Figure 6.5).

Procedure $Prepare(T_1, T_2, x, j)$:

[1] `For` $i = 1, \ldots, j-1, j+1, \ldots, m$:

[2] $t_i \in_R [0, 2^{\ell_{\mathcal{H}}} - 1]$;

[3] $y_{i,e} \in_R [0, 2^{\ell_e + \ell_{\mathcal{H}} + \ell_\phi} - 1]$;

[4] $y_{i,1}, \ldots, y_{i,m} \in_R [0, 2^{\ell_x + \ell_{\mathcal{H}} + \ell_\phi} - 1]$;

[5] $y_{i,s} \in_R [0, 2^{\ell_s + \ell_{\mathcal{H}} + \ell_\phi} - 1]$;

[6] $y_{i,w}, y_{i,r} \in_R [0, 2^{\ell_n + \ell_{\mathcal{H}} + 2\ell_\phi} - 1]$;

[7] $z_{i,e} \in_R [0, 2^{2\ell_e + \ell_{\mathcal{H}} + \ell_\phi} - 1]$;

[8] $z_{i,w}, z_{i,r} \in_R [0, 2^{\ell_n + \ell_e + \ell_{\mathcal{H}} + 2\ell_\phi} - 1]$;

[9] $U_{i,1} := (a_i^x / c)^{t_i} T_1^{y_{i,e}} \prod_{l=1, l \neq i}^{m} a_l^{y_{i,l}} b^{y_{i,s}} h^{-z_{i,w}} \bmod n$;

[10] $U_{i,2} := T_2^{-t_i} g_1^{y_{i,e}} g_2^{y_{i,w}} h^{y_{i,r}} \bmod n$;

[11] $U_{i,3} := T_2^{-y_{i,e}} g_1^{z_{i,e}} g_2^{z_{i,w}} h^{z_{i,r}} \bmod n$;

[12] $\mathcal{T} := (t_1, \ldots, t_{j-1}, \bullet, t_{j+1}, \ldots, t_m)$;

[13] $\mathcal{Y} :=$

$$\begin{pmatrix} \bullet & y_{1,2} & \cdots & y_{1,j} & y_{1,j+1} & \cdots & y_{1,m} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{j-1,1} & \cdots & \bullet & y_{j-1,j} & y_{j-1,j+1} & \cdots & y_{j-1,m} \\ \bullet & \cdots & \bullet & \bullet & \bullet & \cdots & \bullet \\ y_{j+1,1} & \cdots & y_{j+1,j-1} & y_{j+1,j} & \bullet^{\dagger} & \cdots & y_{j+1,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{m,1} & \cdots & y_{m,j-1} & \bullet & y_{m,j+1} & \cdots & \bullet \end{pmatrix};$$

[14] $\mathcal{Y}_{\text{aux}} :=$

$$\begin{pmatrix} y_{1,e} & y_{1,s} & y_{1,w} & y_{1,r} \\ \vdots & \vdots & \vdots & \vdots \\ y_{j-1,e} & y_{j-1,s} & y_{j-1,w} & y_{j-1,r} \\ \bullet & \bullet & \bullet & \bullet \\ y_{j+1,e} & y_{j+1,s} & y_{j+1,w} & y_{j+1,r} \\ \vdots & \vdots & \vdots & \vdots \\ y_{m,e} & y_{m,s} & y_{m,w} & y_{m,r} \end{pmatrix};$$

[15] $\mathcal{U} :=$

$$\begin{pmatrix} U_{1,1} & U_{1,2} & U_{1,3} \\ \vdots & \vdots & \vdots \\ U_{j-1,1} & U_{j-1,2} & U_{j-1,3} \\ \bullet & \bullet & \bullet \\ U_{j+1,1} & U_{j+1,2} & U_{j+1,3} \\ \vdots & \vdots & \vdots \\ U_{m,1} & U_{m,2} & U_{m,3} \end{pmatrix};$$

[16] $\mathcal{Z} :=$

$$\begin{pmatrix} z_{1,e} & z_{1,w} & z_{1,r} \\ \vdots & \vdots & \vdots \\ z_{j-1,e} & z_{j-1,w} & z_{j-1,r} \\ \bullet & \bullet & \bullet \\ z_{j+1,e} & z_{j+1,w} & z_{j+1,r} \\ \vdots & \vdots & \vdots \\ z_{m,e} & z_{m,w} & z_{m,r} \end{pmatrix};$$

[17] `Return` $(\mathcal{T}, \mathcal{Y}, \mathcal{Y}_{\text{aux}}, \mathcal{U}, \mathcal{Z})$

† This cell's right neighbours are not empty cells, as the dots ("...") may suggest, but continue with $y_{j+1,j+2}, y_{j+1,j+3}, \ldots$, instead.

Figure 6.6: *Procedure $Prepare$ computing 'forged' values*

Procedure $Finalise(\mathcal{Y}, \mathcal{Y}_{\text{aux}}, \mathcal{Z}, e, w, r, r_w, r_r, r_s, r_1, \ldots, r_m, r_e, r_{ew}, r_{er}, r_{ee}, t_j, j)$:
[1] $y_{j,e} := et_j + r_e$;
[2] `For` $i = 1, \ldots, j, j+1, \ldots, m$:
[3] $y_{j,i} := x_i t_j + r_i$;
[4] $y_{j,s} := st_j + r_s$;
[5] $y_{j,w} := wt_j + r_w$;
[6] $y_{j,r} := rt_j + r_r$;
[7] $z_{j,e} := eet_j + r_{ee}$;
[8] $z_{j,w} := ewt_j + r_{ew}$;
[9] $z_{j,r} := ert_j + r_{er}$;
[10] Insert $(y_{j,1}, \ldots, y_{j,j-1}, \bullet, y_{j,j+1}, \ldots, y_{j,m})$ in row j of $\mathcal{Y}$;
[11] Insert $(y_{j,e}, y_{j,s}, y_{j,w}, y_{j,r})$ in row j of $\mathcal{Y}_{\text{aux}}$;
[12] Insert $(z_{j,e}, z_{j,w}, z_{j,r})$ in row j of $\mathcal{Z}$;
[13] `Return` $(\mathcal{Y}, \mathcal{Y}_{\text{aux}}, \mathcal{Z})$

Figure 6.7: *Procedure $Finalise$ computing 'values of knowledge'*

6.7.3 Properties

In the following, we sketch how the coupon system proposed in the previous subsection satisfies the requirements from Section 6.3.1. We will analyse the security of the system assuming that the strong RSA assumption holds.

Unforgeability. The property of unforgeability of our coupon system follows from the unforgeability of the CL signature scheme. As described in the previous section, a set of multi-coupons is a single CL signature on a block of messages. As has been proven in [CL02], forging CL signatures would break the strong RSA assumption.

Resetting the number of spent coupons requires to change some component in the tuple X, e.g., replacing a redeemed coupon x_i with $x_i^* \neq x_i$, since the vendor stores each spent single coupon x_i. However, replacing x_i by x_i^*, yielding tuple X^*, must be done such that $Sign_{(\cdot)}(X) = Sign_{(\cdot)}(X^*)$. Suppose the latter can be done. Then we get $v^e \prod_{i=1}^{m} a_i^{x_i} b^s \equiv v^e \prod_{j=1}^{i-1} a_j^{x_j} a_i^{x_i^*} \prod_{j=i+1}^{m} a_j^{x_j} b^s$. Dividing by the right hand side yields $a_i^{x_i - x_i^*} \equiv 1 \bmod n$. Since $x_i \neq x_i^*$ it must be the case that $x_i - x_i^* = \alpha \cdot ord(\mathbb{Z}_n)$. Now, choose any e such that $1 < e < (x_i - x_i^*)$ and $gcd(e, x_i - x_i^*) = 1$. By the extended Euclidean algorithm we can find d such that $ed + (x_i - x_i^*)t = 1$. Using this, we can compute e-th roots in $\mathbb{Z}_n^*$. For this, let u be any value from $\mathbb{Z}_n^*$ and compute $w := u^d \bmod n$. Since $u \equiv u^{ed+(x_i - x_i^*)t} \equiv u^{ed} u^{\alpha \cdot ord(\mathbb{Z}_n) \cdot t} \equiv (u^d)^e \equiv w^e \bmod n$, the value w is an e-th root of u. This means we would have found a way to break the (strong) RSA assumption. Since this is

assumed to be infeasible, x_i cannot be replaced by $x_i^* \neq x_i$ without changing the signature (v, e, s).

Double-spending detection. If a cheating user tries to redeem an already spent single coupon x_i, she will be caught at the end of the redeem protocol, since the coupon to be redeemed must be disclosed and, thus, can easily be looked up in the vendor's database of spent coupons.

Redemption limitation. An m-redeemable coupon M cannot be redeemed more than m times (without the vendor's consent). Each multi-coupon M contains a signature on an m-tuple $(x_1, \ldots, x_m)$ of single coupons and in each run of the issue protocol a single coupon x_i is disclosed. Thus, after m honest runs with the same M all x_i will be disclosed to the vendor. As argued under *unforgeability* and *double-spending detection*, already redeemed x_i cannot be replaced by fresh x_i^* and any attempt to reuse an already disclosed x_i will be caught by the double-spending check.

Weak protection against splitting. Suppose that two users U_1 and U_2 want to share some multi-coupon $M := (X, v, e, s)$ such that U_1 receives single coupons $X_1 := \{x_1, \ldots, x_i\}$ and U_2 receives the remaining coupons $X_2 := \{x_j, \ldots, x_m\}$, $i < j$. To achieve splitting, they have to find a way to make sure that U_1 is able to redeem all $x_l' \in X_1$ while not being able to redeem any coupon $x_l'' \in X_2$, and analogously for U_2. However, in the redeem protocol it is necessary to prove knowledge of the DLREP of T_1, which essentially is X. Since proving knowledge of X while knowing only X_1 or X_2 would violate the soundness of the employed proof of knowledge *PoKRep* —and hence violate the strong RSA assumption— this is believed to be infeasible. Again, the missing part of X, either X_1 or X_2, cannot be replaced by 'fake' coupons $X'_{1|2}$ since this violates the unforgeability property of the coupon system. Hence, X cannot be split and can only be shared if both U_1 and U_2 have full knowledge of X which comes down to *all-or-nothing sharing*.

Unlinkability. For unlinkability, we have to consider two cases, unlinkability between issue and redeem protocol runs and between executions of the redeem protocol.

(1) *issue vs. redeem*: The issue protocol is identical to the protocol *BlindSign* and, hence, the vendor V, acting as signer S, does not learn anything about the message X being signed. However, V has partial knowledge of the signature (v, e, s) because at the end of the issue protocol he learns (v, e) but not s. To exploit this knowledge, he would have to recognise v or e in any run of the redeem protocol. However, since the redeem protocol is zero-knowledge, nothing is disclosed except for the fact that x_j is part of some valid signature.

(2) *redeem vs. redeem*: As argued before, since the redeem protocol is zero-knowledge, in any two runs of the redeem protocol, only the coupon to be spent in each run of the protocol is disclosed and the fact that each of them belongs to some valid multi-coupon. However, this yields no information about the corresponding multi-coupons and hence, V will not be able to tell whether the two belong to the same multi-coupon or to different ones.

From the arguments above it is clear that for any run I of the issue protocol and any two different runs of the redeem protocol, R and R', it holds that the vendor can neither decide if $\mathcal{L}(I, R)$ holds, nor that $\mathcal{L}(R, R')$ holds.

Minimum disclosure. A further consequence of the unlinkability of transactions in the coupon system, and due to the fact that no counter value is sent in any protocol, the number of unspent coupons cannot be inferred from any redeem protocol run.

6.8 Conclusion

The coupon system presented in this chapter allows vendors to issue multi-coupons to their customers, where each single coupon of such a multi-coupon can be redeemed at the vendor's in exchange for some good, e.g., an MP3 file, movie on demand, or some service, e.g., access to commercial online articles of a newspaper. Issuing coupons is advantageous to vendors since coupons effectively retain customers as long as they have coupons left to spent. However, multi-coupons might be misused by vendors to link transactions of customers in order to collect and compile information from their transactions in a profile. To protect the privacy of customers in this respect, the proposed coupon system allows customers to unlinkably redeem single coupons while preserving security requirements of vendors, such as double-spending detection and redemption limitation. In addition, the system discourages lending/sharing of multi-coupons, which we call weak protection against splitting, as this means, for the lending party, to give up control of all of the multi-coupon's remaining single coupons, as opposed to just a few single coupons, and allow the recipient to spent all of them.

The scheme underlying our coupon system may be of independent interest, as it realises m-showable credentials providing unlinkability between different showings of the credentials and discouraging lending of the credentials by employing all-or-nothing sharing.

Chapter 7

Conclusion

In this work, we had been primarily concerned with the privacy of online customers. Our focus was on moving away from hardly verifiable privacy claims of vendors on to methods providing privacy protection that is verifiable by customers, rather than by a third party of the vendor's choice, as in seal programs.

The abstract model developed in this work can support vendors in actively transforming their businesses into more privacy-friendly ones by helping them to understand and identify the different types of data links created in regular transactions. Such links, between or within transactions, often create the threat to privacy and their type and existence eventually make up the difference between a privacy-enhanced store and a regular one. The novel approach in our model is to consider not only data conveyed between transactions but also between individual phases of a transaction.

The benefit of considering phases for privacy protection is that by ensuring unlinkability of phases, privacy is ensured even if the transactions that include these phases can be related, as the unlinked phase is decoupled from all other data. Thus, by ensuring unlinkability of phases, one can add more privacy safeguards which will help to limit the collection of personal data even more.

As an example for the decoupling of phases, we presented how the search phase of an online transaction can be decoupled from its subsequent order, payment, and delivery phase. The *decoupling component* developed for this purpose was initially based on mobile agents but only served as an example, as other implementations are certainly possible. Another possibility for such an implementation, making use of Web services, had also been sketched herein.

The other contributions of this work in the area of privacy and security had been made in the relatively new field of digital incentive and loyalty systems. Such systems are already known from the real world and are critically eyed by privacy advocates for their potential of privacy invasion.

Loyalty and incentive systems for the Internet may gain momentum in the near future, as more and more services are offered which can be completely handled online, e.g., music downloads, movie-on-demand, ebook downloads, etc. In such scenarios, customers on the one hand may still welcome the benefits of a loyalty program membership, but on the other hand may not want to give up privacy for it.

Our works on loyalty and incentive systems allow both program membership and privacy protection. The systems developed are the first ones usable in online commerce that allow customer retention while providing verifiable privacy protection for customers and security for vendors employing the systems. In particular, the privacy-preserving property of the systems may even attract privacy-sensitive customers who otherwise would never consider to become members of a regular loyalty program because of privacy concerns.

The loyalty systems are based on different cryptographic assumptions but share the idea of an *anonymous counter* which is a novel concept introduced in this work. Herein, such counters are realised as a two party protocol, where one party secretly holds a counter that can only be increased by the other party. The protocol ensures that the party holding the counter cannot manipulate it and the party who is to increase the counter does not learn its value until the holder discloses it.

The last contribution of this work are what we call *multi-coupons*. Such coupons are issued once and can subsequently be spent for a predetermined number of times, e.g., to access some online service. What distinguishes multi-coupons from batches of single tokens is the fact that our coupons cannot be partitioned such that every owner of a part from the multi-coupon can spent only her own part and none of the other ones. Multi-coupons are basically limited-show credentials which allow their holders to anonymously prove some statement, as long as they do not overspent. That is, the credentials become linkable when they are shown more often then specified by their issuer.

Today's privacy problems arise not only because more data is being collected and processed than ever before but also because this data is interlinked and related to natural persons. Apart from creating concerns for individual freedom and democracy, unchecked data collection also hinders the proliferation of online commerce because privacy-concerned users may yet go online but often choose not to buy from the Internet. This creates an opportunity for vendors investing in privacy enhancing technology as they may attract privacy-concerned users which have neglected online commerce so far or which are looking for privacy-friendly alternatives. The results of this work can help vendors to provide such alternatives.

References

[Acq02] Alessandro Acquisti. Privacy and security of personal information: Economic incentives and technological solutions. In *Workshop on the Economics of Information Security (WEIS 02)*, 2002.

[AG04] Alessandro Acquisti and Jens Grossklags. Privacy and rationality: Preliminary evidence from pilot data. In *3rd Annual Workshop on the Economics of Information Security (WEIS 04)*, 2004.

[AG05a] Alessandro Acquisti and Jens Grossklags. Privacy and rationality in individual decision making. In *IEEE Security & Privacy Magazine*, volume 2, pages 26–33, January/February 2005.

[AG05b] Alessandro Acquisti and Jens Grossklags. Uncertainty, ambuiguity and privacy. In *4th Annual Workshop on the Economics of Information Security (WEIS 05)*, 2005.

[AHG03] Paul Ashley, Satoshi Hada, and Günter Karjoth and Calvin Powers and Matthias Schunter. Enterprise Privacy Authorization Language (EPAL). Research Report, RZ3485 (#93951), IBM Research, March 2003.

[AJJ+00] Robert M. Arlein, Ben Jai, Markus Jakobsson, Fabian Monrose, and Michael K. Reiter. Privacy-preserving global customization (extended abstract). In *Proceedings of the 2nd ACM conference on Electronic Commerce (EC'00)*, October 2000.

[AV05] Alessandro Acquisti and Hal R. Varian. Conditioning prices on purchase history. *Marketing Science*, 24(3):367–381, 2005.

[BC89] Gilles Brassard and Claude Crépeau. Sorting out zero-knowledge. In *Advances in Cryptology - EUROCRYPT '89 – International Conference on the Theory and Application of Cryptographic Techniques, Proceedings*, number 434 in LNCS. Springer Verlag, 1989.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37, 1988.

[BCC04a] Ernie Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *Proceedings of the 11th ACM conference on Computer and communications security*, October 2004.

[BCC04b] Ernie Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. `http://eprint.iacr.org`, February 2004.

[BCvRE03] Tom Bellwood, Luc Clément, and Claus von Riegen (Eds.). Universal Description, Discovery, and Integration (UDDI) Version 3.0.1. OASIS Standard, (http://uddi.org), October 2003.

[BD01] Feng Bao and Robert Deng. Privacy protection for transactions of digital goods. In *Information and Communications Security (ICICS 2001), Third International Conference, Proceedings*, number 2229 in LNCS. Springer Verlag, November 2001.

[BDF01] Feng Bao, Robert H. Deng, and Peirong Feng. An efficient and practical scheme for privacy protection in the e-commerce of digital goods. In *Information Security and Cryptology - ICISC 2000 —Third International Conference, 2000, Proceedings*, number 2015 in LNCS. Springer Verlag, 2001.

[BDS03] Federal Data Protection Act (Bundesdatenschutzgesetz, BDSG) of December 20, 1990 (BGBl.I 1990 S.2954), amended by law of September 14, 1994 (BGBl. I S. 2325), law of December 16, 1997 (BGBl. I S. 2325), law of December 17, 1997 (BGBl. I S. 2325), and of May 23, 2001 (BGBl. I S. 904), last amendment by law of January 14, 2003 (BGBl. I S. 66), January 2003.

[Ben99] Paola Benassi. TRUSTe: An online privacy seal program. *Communications of the ACM*, 42(2), February 1999.

[BF01] Dan Boneh and Matthew Franklin. Identity based encryption from the Weil pairing. In *Advances in Cryptology - CRYPTO 2001 – 21st Annual International Cryptology Conference, Proceedings*, number 2139 in LNCS, pages 213–229. Springer Verlag, 2001.

[BFK00] Oliver Berthold, Hannes Federrath, and Stefan Köpsell. Web MIXes: A system for anonymous and unobservable Internet access. In H. Fed-

errath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, number 2009 in LNCS, pages 115–129. Springer Verlag, July 2000.

[BG93] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *Advances in Cryptology - CRYPTO '92 – 12th Annual International Cryptology Conference, Proceedings*, number 740 in LNCS. Springer Verlag, 1993.

[BGH+95] Mihir Bellare, Juan A. Garay, Ralf Hauser, Amir Herzberg, Hugo Krawczyk, Michael Steiner, Gene Tsudik, and Michael Waidner. iKP – A Family of Secure Electronic Payment Protocols. In *First USENIX Workshop on Electronic Commerce*, July 1995.

[BGK95] Ernie Brickell, Peter Gemmell, and David Kravitz. Trustee-based tracing extensions to anonymous cash and the making of anonymous change. In *Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'95), Proceedings*, 1995.

[BH00] Kathy Bohrer and Bobby Holland. Customer Profile Exchange (CPExchange) Specification, Version 1.0, October 2000.

[BKB00] Ruth N. Bolton, P. K. Kannan, and Matthew D. Bramlett. Implications of loyalty programs and service experiences for customer retention and value. *Journal of the Academy of Marketing Science*, 28(1), 2000.

[BKLS02] Paulo S.L.M. Baretto, Hae Y. Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. In *Advances in Cryptology - CRYPTO 2002 – 22nd Annual International Cryptology Conference, Proceedings*, number 2442 in LNCS. Springer Verlag, 2002.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In *ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings*, number 2248 in LNCS, pages 514–532. Springer Verlag, 2001.

[BN00] Dan Boneh and Moni Naor. Timed commitments (extended abstract). In *Advances in Cryptology - CRYPTO 2000 – 20th Annual International Cryptology Conference, Proceedings*, number 1880 in LNCS. Springer Verlag, 2000.

[BNPS01] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. In *Financial Cryptography – 5th International Conference (FC 2001), Proceedings*, number 2339 in LNCS. Springer Verlag, 2001.

[BNPS03] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. *Journal of Cryptology*, 16(3), 2003.

[Bol03] Alexandra Boldyreva. Efficient threshold signatures, multisignatures and blind signatures based on the Gap-Diffie-Hellman-group signature scheme. In *Public Key Cryptography (PKC) 2003*, number 2567 in LNCS, pages 31–46. Springer Verlag, 2003.

[Bou00] Fabrice Boudot. Efficient proofs that a commited number lies in an interval. In *Advances in Cryptology - EUROCRYPT 2000 – International Conference on the Theory and Application of Cryptographic Techniques, Proceedings*, number 1807 in LNCS. Springer Verlag, 2000.

[BP97] Nico Barić and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology - EUROCRYPT '97 – International Conference on the Theory and Application of Cryptographic Techniques, Proceedings*, number 1233 in LNCS. Springer Verlag, 1997.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on computer and communications security (CCS '93)*, November 1993.

[Bra93] Stefan Brands. An efficient off-line electronic cash system based on the representation problem. CWI Report, CS-R9323, Centrum voor Wiskunde en Informatica (CWI), 1993.

[Bra99] Stefan Brands. *Rethinking Public Key Infrastructure and Digital Certificates – Building in Privacy.* PhD thesis, Eindhoven Institute of Technology, The Netherlands, 1999.

[BSG00] Philippe Boucher, Adam Shostack, and Ian Goldberg. Freedom systems 2.0 architecture. White paper, Zero Knowledge Systems, Inc., December 2000.

[BVe83] BVerfGE 65, 1 - Volkszählung: Urteil des Ersten Senats vom 15. Dezember 1983 auf die mündliche Verhandlung vom 18. und 19. Oktober 1983 - 1 BvR 209, 269, 362, 420, 440, 484/83 in den Verfahren über die Verfassungsbeschwerden. `http://www.datenschutz-berlin.de/gesetze/sonstige/volksz.htm`, December 1983.

[Cam98] Jan Camenisch. *Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem.* PhD thesis, ETH Zürich, Switzerland, 1998. Dissertation ETH No. 12520, Vol. 2 of ETH-Series in Information Security an Cryptography, ISBN 3-89649-286-1, Hartung-Gorre Verlag, Konstanz.

[CCMW01] Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. Web Services Description Language (WSDL) 1.1. W3C Note, March 2001.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology - CRYPTO '94 – 14th Annual International Cryptology Conference, Proceedings*, number 839 in LNCS. Springer Verlag, 1994.

[CES+05] Liqun Chen, Matthias Enzmann, Ahmad-Reza Sadeghi, Markus Schneider, and Michael Steiner. A privacy-protecting coupon system. In *Financial Cryptography and Data Security – 9th International Conference (FC 2005), Proceedings*, number 3570 in LNCS, February 2005.

[CFN90] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In *Advances in Cryptology - CRYPTO '88 – 8th Annual International Cryptology Conference, Proceedings*, number 403 in LNCS. Springer Verlag, 1990.

[CFT98] Agnes Chan, Yair Frankel, and Yiannis Tsiounis. Easy come - easy go divisible cash. In *Advances in Cryptology - EUROCRYPT '98 – International Conference on the Theory and Application of Cryptographic Techniques, Proceedings*, number 1403 in LNCS. Springer Verlag, 1998.

[CG04] Jan Camenisch and Jens Groth. Group signatures: Better efficiency and new theoretical aspects. In *4th Conference on Security in Communication Networks - SCN '04*, number 3352 in LNCS. Springer Verlag, 2004.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing (STOC 1998)*, pages 209–218. ACM Press, 1998.

[CGH06a] Sébastien Canard, Aline Gouget, and Emeline Hufschmitt. A handy multi-coupon system. In *Proceedings of the 4th International Conference on Applied Cryptography and Network Sercurity, ACNS 2006, Singapore, June 6-9, 2006*, number 3989 in LNCS. Springer Verlag, June 2006.

[CGH06b] Sébastien Canard, Aline Gouget, and Emeline Hufschmitt. A handy multi-coupon system. `http://eprint.iacr.org/2006/231`, 2006.

[Cha81] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2), February 1981.

[Cha83] David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology - CRYPTO '82, Proceedings*. Plenum Press, 1983.

[Cha89] David Chaum. Privacy protected payments: Unconditional payer and/or payee untraceability. In *Smart Card 2000, Proceedings*. North Holland, 1989.

[Che96] Lidong Chen. Access with pseudonyms. In *Cryptography: Policy and Algorithms, International Conference, Brisbane, Australia, July, 1995, Proceedings*, number 1029 in LNCS. Springer Verlag, 1996.

[CHL05] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. Number 3494 in LNCS. Springer Verlag, 2005.

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Advances in Cryptology - EUROCRYPT 2001 – International Conference on the Theory and Application of Cryptographic Techniques, Proceedings*, number 2045 in LNCS. Springer Verlag, 2001.

[CL02] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In *Third Conference on Security in Communication Networks – SCN'02*, number 2576 in LNCS. Springer Verlag, 2002.

[Cla88] Roger Clarke. Information technology and dataveillance. *Communications of the ACM*, 31(5), May 1988.

[Cla99] Roger Clarke. Internet privacy concerns confirm the case for intervention. *Communications of the ACM*, 42(2), February 1999.

[CLM+02] Lorrie Cranor, Marc Langheinrich, Massimo Marchiori, Martin Presler-Marshall, and Joseph Reagle. The Platform for Privacy Preferences 1.0 (P3P1.0 Specification). W3C Recommendation 16 April 2002, April 2002.

[CM99a] Jan Camenisch and Markus Michels. Proving in zero-knowledge that a number n is the product of two safe primes. In *Advances in Cryptology - EUROCRYPT '99 – International Conference on the Theory and Application of Cryptographic Techniques, Proceedings*, number 1592 in LNCS. Springer Verlag, 1999.

[CM99b] Jan Camenisch and Markus Michels. Separability and efficiency for generic group signature schemes. In *Advances in Cryptology - CRYPTO '99 – 19th Annual International Cryptology Conference, Proceedings*, number 1666 in LNCS. Springer Verlag, 1999.

[CM01] Mary J. Culnan and George R. Milne. The Culnan-Milne survey on consumers & online privacy notices: Summary of responses, December 2001.

[Com04] Computerworld. FTC: Identity theft, online scams rose in '04. `http://www.computerworld.com/managementtopics/ebusiness/story/0,10801,99429,00.html`, February 2004.

[Con06] Yvonne Conrad. Kundenkarten und Rabattsysteme. *Datenschutz und Datensicherheit (DuD)*, 30(7), 2006.

[CPS96] Jan Camenisch, Jean-Marc Piveteau, and Markus Stadler. An efficient fair payment system. In *3rd ACM Conference on Computer and Communications Security (CCS'96)*. ACM Press, 1996.

[CS97] Jan L. Camenisch and Markus A. Stadler. Efficient group signature schemes for large groups. In *Advances in Cryptology - CRYPTO '97 – 17th Annual International Cryptology Conference, Proceedings*, number 1294 in LNCS. Springer Verlag, 1997.

[DF02] Ivan B. Damgård and Eiichihiro Fujisaki. A statistically hiding integer commitment scheme based on groups with hidden order. In *Advances in Cryptology - ASIACRYPT '02, International Conference on the Theory and Applications of Crypotology and Information Security 2002, Proceedings*, number 2501 in LNCS. Springer Verlag, 2002.

[DMS04] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[Dod03] Yevgeniy Dodis. Efficient construction of (distributed) verifiable random functions. In *Public Key Cryptography (PKC) 2003*, number 2567 in LNCS, pages 1–17. Springer Verlag, 2003.

[DSCP02] Claudia Díaz, Stefaan Seys, Joris Calessens, and Bart Preneel. Towards measuring anonymity. In *Privacy Enhancing Technologies (PET 2002)*, number 2482 in LNCS. Springer Verlag, 2002.

[dSdCPY94] Angelo de Santis, Giovanni di Crescenzo, Giuseppe Persiano, and Moti Yung. On monotone formula closure of SZK. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, November 1994.

[DTG06] Stelios Dritsas, John Tsaparas, and Dimitris Gritzalis. A generic privacy enhancing technology for pervasive computing environments. In *Trust and Privacy in Digital Business (TrustBus 2006)*. Springer-Verlag, September 2006.

[DU97] Grahame R. Dowling and Mark Uncles. Do customer loyalty programs really work? *Sloan Management Review*, 38(4):71–82, 1997.

[DY05] Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In *Public Key Cryptography (PKC) 2005*, number 3386 in LNCS, pages 416–431. Springer Verlag, 2005.

[EAASS06] Julia B. Earp, Annie I. Antón, Lynda Aiman-Smith, and William H. Stufflebeam. Examining internet privacy policies within the context of user privacy values. *IEEE Transactions on Engineering Management*, 52(2):227–237, May 2006.

[EE02] Matthias Enzmann and Claudia Eckert. Pseudonymes Einkaufen physischer Güter. In *Sichere Geschäftsprozesse, Tagungsband zur Arbeitskonferenz Elektronische Geschäftsprozesse*. IT Verlag für Informationstechnik, 2002.

[EEOS05] Claudia Eckert, Matthias Enzmann, Susanne Okunick, and Markus Schneider. Kundenbindung durch ein anonymes Rabattsystem. In *D-A-CH Security*, March 2005.

[EFS04] Matthias Enzmann, Marc Fischlin, and Markus Schneider. A privacy-friendly loyalty system based on discrete logarithms over elliptic

curves. In *Financial Cryptography – 8th International Conference (FC 2004), Proceedings*, number 3110 in LNCS, February 2004.

[EKS02a] Matthias Enzmann, Thomas Kunz, and Markus Schneider. A new infrastructure for user tracking prevention and privacy protection in internet shopping. In *Infrastructure Security Conference (InfraSec'02)*, number 2427 in LNCS. Springer Verlag, October 2002.

[EKS02b] Matthias Enzmann, Thomas Kunz, and Markus Schneider. Privacy protection through unlinkability of customer activities in business processes using mobile agents. In *3rd International Conference on Electronic Commerce and Web Technologies (EC-Web 2002)*, number 2455 in LNCS. Springer Verlag, September 2002.

[EKS02c] Matthias Enzmann, Thomas Kunz, and Markus Schneider. Schutz der Privatsphäre beim Online-Einkauf durch Verwendung mobiler Agenten. In *Von e-Learning bis e-Payment — Das Internet als sicherer Marktplatz, Leipziger Informatiktage (LIT'02)*. Akademische Verlags-Gesellschaft, Berlin, September 2002.

[EKS02d] Matthias Enzmann, Thomas Kunz, and Markus Schneider. Using mobile agents for privacy amplification in the trade with tangible goods. In *6th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2002)*, July 2002.

[EKS03] Matthias Enzmann, Thomas Kunz, and Markus Schneider. Datenschutzfreundlicher Online-Einkauf durch Reduktion personenbezogener Daten. In *Trust in the Network Economy, Evolaris Volume 2*. Springer Verlag, August 2003.

[ER02] Matthias Enzmann and Alexander Roßnagel. Realisierter Datenschutz für den Einkauf im Internet. *Computer & Recht*, 18(2):141–150, February 2002.

[ES04] Matthias Enzmann and Markus Schneider. A privacy-friendly loyalty system for electronic marketplaces. In *IEEE International Conference on e-Technology, e-Commerce, and e-Service 2004 (EEE 2004)*. IEEE Computer Society Press, 2004.

[EU06] European Parliament and the Council of the European Union. Directive of the European Parliament and of The Council on the retention of data generated or processed in connection with the provision of publicly available electronic communications services

or of public communications networks and amending Directive 2002/58/EC. http://register.consilium.europa.eu/pdf/en/05/st03/st03677.en05.pdf, February 2006.

[Fed00] Federal Trade Commission. FTC Sues Failed Website, Toysmart.com, for Deceptively Offering for Sale Personal Information of Website Visitors. http://www.ftc.gov/opa/2000/07/toysmart.htm, July 2000.

[Fel67] William Feller. *An Introduction to Probability Theory and Its Applications*. John Wiley & Sons, 1967.

[Fer93] Niels Ferguson. Extensions of single term coins. In *Advances in Cryptology - CRYPTO '93 – 13th Annual International Cryptology Conference, Proceedings*, number 773 in LNCS. Springer Verlag, 1993.

[FF00] Marc Fischlin and Roger Fischlin. Efficient non-malleable commitment schemes. In *Advances in Cryptology - CRYPTO 2000 – 20th Annual International Cryptology Conference, Proceedings*, number 1880 in LNCS. Springer Verlag, 2000.

[FFSS04] Joan Feigenbaum, Michael J. Freedman, Tomas Sander, and Adam Shostack. Economic barriers to the deployment of existing privacy technologies (position paper). *3rd Annual Workshop on the Economics of Information Security (WEIS 04)*, 2004.

[FG06] Martin Fisch and Christoph Gscheidle. ARD/ZDF-Online Studie 2006, Onliner 2006: Zwischen Breitband und Web 2.0 — Ausstattung und Nutzungsinnovation. Media Persperktiven 8/2006, 2006.

[FKH00] Batya Friedman, Peter H. Kahn, and Daniel C. Howe. Trust online. *Communications of the ACM*, 43(12), December 2000.

[FKZ02] Fred M. Feinberg, Aradhna Krishna, and Z. John Zhang. Do we care what others get? A behaviorist approach to targeted promotions. *Journal of Marketing Research*, 39(3), August 2002.

[FM02] Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, DC, November 2002.

[FO97] E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology - CRYPTO '97 – 17th Annual International Cryptology Conference, Proceedings*, number 1294 in LNCS. Springer Verlag, 1997.

[FPSS96] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11), November 1996.

[Fri90] Charles Fried. Privacy: A rationale contex. In M. David Ermann, Mary B. Williams, and Claudio Gutierez, editors, *Computers, Ethics, and Society*. Oxford University Press, New York, 1990.

[FS87] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO '86 – 6th Annual International Cryptology Conference, Proceedings*. Springer-Verlag, 1987.

[FU96] Usama Fayyad and Ramasamy Uthurusamy. Data mining and knowledge discovery in databases. *Communications of the ACM*, 39(11), November 1996.

[Ful98] Joseph S. Fulda. AI watch: From data to knowledge: implications of data mining. *ACM SIGCAS Computers and Society*, 28(1), March 1998.

[Ful99] Joseph S. Fulda. Solution to a philosophical problem concerning data mining. *ACM SIGCAS Computers and Society*, 29(4), December 1999.

[Gav95] Ruth Gavison. Privacy and the limits of the law. In Deborah G. Johnson and Helen Nissenbaum, editors, *Computers, Ethics & Social Values*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[GCKR97] Robert Gray, George Cybenko, David Kotz, and Daniela Rus. Agent Tcl. In William Cockayne and Michael Zypa, editors, *Itinerant Agents*. Manning Publishing, 1997.

[GGK+99] Eran Gabber, Phillip B. Gibbons, David M. Kristol, Yossi Matias, and Alain Mayer. Design and implementation of the lucent personalized web assistant (LPWA). *Communications of the ACM*, 42(2), February 1999.

[GHS02] Steven D. Galbraith, Keith Harrison, and David Soldera. Implementing the Tate pairing. In *Algorithmic Number Theory, 5th International Symposium, ANTS-V, Proceedings*, number 2369 in LNCS, pages 324–337. Springer Verlag, 2002.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Computing*, 18(1), February 1989.

[Gol02] Ian Goldberg. Privacy-enhancing technologies for the Internet, II: Five years later. In *Privacy Enhancing Technologies (PET 2002)*, number 2482 in LNCS. Springer Verlag, 2002.

[Got99] Donald Gotterbarn. Privacy lost: The net, autonomous agents, and 'virtual information'. In *Ethics and Information Technology*, number 1. Kluwer Academic Publishers, 1999.

[GRS99] David Goldschlag, Michael Reed, and Paul Syverson. Onion routing for anonymous and private internet connections. *Communications of the ACM*, 42(2), February 1999.

[GWB97] Ian Goldberg, David Wagner, and Eric Brewer. Privacy-enhancing technologies for the internet. In *IEEE Compcon'97*. IEEE Computer Society, 1997.

[Hei06] Heise. AOL veröffentlichte Suchanfragen von über 500.000 Mitgliedern. `http://www.heise.de/newsticker/meldung/76474`, August 2006.

[HNP99] Donna L. Hoffman, Thomas P. Novak, and Marcos Peralta. Building consumer trust online. *Communications of the ACM*, 42(4), April 1999.

[Int99] Harris Interactive. IBM Multi-National Consumer Privacy Survey, October 1999.

[JN01] Antoine Joux and Kim Nguyen. Separating Decision Diffie-Hellman from Diffie-Hellman in cryptographic groups. Cryptology ePrint Archive, Report 2001/003, 2001. http://eprint.iacr.org/.

[Joh85] Deborah G. Johnson. *Computer Ethics*. Prentice-Hall, 1985.

[Joh97] Deborah G. Johnson. Ethics online. *Communications of the ACM*, 40(1), January 1997.

[Jou00] Antoine Joux. A one round protocol for tripartite Diffie-Hellman. In *Algorithmic Number Theory, 4th International Symposium, ANTS-IV, Proceedings*, number 1838 in LNCS, pages 385–394. Springer Verlag, 2000.

[Jue01] Ari Juels. Targeted advertising ... and privacy too. In *Progress in Cryptology — CT-RSA 2001, The Cryptographers' Track at RSA Conference 2001 San Francisco, Proceedings*, number 2020 in LNCS. Springer Verlag, 2001.

[Jup02] Jupiter Media Metrix. Seventy percent of us consumers worry about online privacy, but few take protective action (survey summary). `http://www.jmm.com/xp/jmm/press/2002/pr_060302.xml`, June 2002.

[JY96] Markus Jakobsson and Moti Yung. Revokable and versatile electronic money. In *ACM Conference on Computer and Communications Security (CCS'96)*. ACM Press, 1996.

[KCLC07] Ponnurangam Kumaraguru, Lorrie Faith Cranor, Jorge Lobo, and Seraphin B. Calo. A survey of privacy policy languages. Symposium On Usable Privacy and Security (SOUPS), July 2007.

[Ken01] John Kennan. Repeated bargaining with persistent private information. *The Review of Economic Studies*, 68(4):719–755, October 2001.

[KGG+98] David M. Kristol, Eran Gabber, Phillip B. Gibbons, Yossi Matias, and Alain Mayer. Design and implementation of the lucent personalized web assistant (LPWA). Manuscript, 1998.

[Kil01] Eike Kiltz. A tool box of cryptographic functions related to the Diffie-Hellman function. In *INDOCRYPT 2001, Second International Conference on Cryptology in India, Proceedings*, number 2247 in LNCS. Springer Verlag, 2001.

[KLO98] Günter Karjoth, Danny B. Lange, and Mitsuru Oshima. A security model for aglets. In G. Vigna, editor, *Mobile Agents and Security*, number 1419 in LNCS. Springer Verlag, 1998.

[KM97] D. Kristol and L. Montulli. HTTP State Management Mechanism. RFC 2109, February 1997.

[KMW07] Balachander Krishnamurthy, Delfina Malandrino, and Craig E. Wills. Measuring privacy loss and the impact of privacy protection in Web browsing. Symposium On Usable Privacy and Security (SOUPS), July 2007.

[Kob01] Alfred Kobsa. Tailoring privacy to users's needs. In *User Modeling 2001 (UM 2001), 8th International Conference, Proceedings*, number 2109 in LNAI. Springer Verlag, 2001.

[Kob07] Alfred Kobsa. Privacy-enhanced personalization. *Communications of the ACM*, 50(8), August 2007.

[KSW02] Günter Karjoth, Matthias Schunter, and Michael Waidner. Platform for enterprise privacy practices: Privacy-enabled management of customer data. In *Privacy Enhancing Technologies (PET 2002)*, number 2482 in LNCS. Springer Verlag, 2002.

[Lau96] Kenneth C. Laudon. Markets and privacy. *Communications of the ACM*, 39(9), September 1996.

[Lea02] Rob Leathern. Jupiter research consumer survey (04/02). FTC Security Workshop: Security and Privacy Data, May 2002.

[LO98] Danny B. Lange and Mitsuru Oshima. *Programming and Deploying Java Mobile Agents with Aglets.* Addison-Wesley, 1998.

[LPSW00] Gérard Lacoste, Birgit Pfitzmann, Michael Steiner, and Michael Waidner, editors. *SEMPER — Secure Electronic Marketplace for Europe.* Springer, 2000.

[Mah98] David P. Maher. A platform for privately defined currencies, loyalty credits, and play money. In *Financial Cryptography, Second International Conference (FC'98), Proceedings*, number 1465 in LNCS. Springer Verlag, 1998.

[Mau95] Ueli M. Maurer. Fast generation of prime numbers and secure public-key cryptographic parameters. *Journal of Cryptology*, 8(3), 1995.

[MD03] Trevor T. Moores and Gurpreet Dhillon. Do privacy seals in e-commerce really work? *Communications of the ACM*, 46(12), December 2003.

[Men93] Alfred J. Menezes. *Elliptic Curve Public Key Cryptosystems*, volume 234 of *The Kluwer International Series in Engineering and Computer Science.* Kluwer Academic Publishers, 1993.

[Moo97] James H. Moor. Towards a theory of privacy in the information age. *ACM SIGCAS Computers and Society*, 27(3), September 1997.

[Moo05] Trevor Moores. Do consumers understand the role of privacy seals in e-commerce? *Communications of the ACM*, 48(3), March 2005.

[MOV97] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography.* CRC Press series on discrete mathematics and its applications. CRC Press, 1997. ISBN 0-8493-8523-7.

[MPT06] Marco Casassa Mont, Siani Pearson, and Robert Thyne. A systematic approach to privacy enforcement and policy compliance checking in enterprises. In *Trust and Privacy in Digital Business (TrustBus 2006)*. Springer-Verlag, September 2006.

[MV97a] MasterCard and Visa. The SET Standard Book 1: Business Description. www.setco.org, May 1997.

[MV97b] MasterCard and Visa. The SET Standard Book 2: Programmer's Guide. www.setco.org, May 1997.

[MV97c] MasterCard and Visa. The SET Standard Book 3: Formal Protocol Definitions. www.setco.org, May 1997.

[MW96] Ueli M. Maurer and Stefan Wolf. Diffie-Hellman oracles. In *Advances in Cryptology - CRYPTO '96 – 16th Annual International Cryptology Conference, Proceedings*, number 1109 in LNCS, pages 268–282. Springer Verlag, 1996.

[Ngu06] Lan Nguyen. Privacy-protecting coupon system revisited. In *Financial Cryptography and Data Security – 10th International Conference (FC 2006), Proceedings*, number 4107 in LNCS, February 2006.

[Nis98] Helen Nissenbaum. Protecting privacy in an information age: The problem of privacy in public. *Law and Philosophy*, 17(5), November 1998.

[Nis99] Helen Nissenbaum. The meaning of anonymity in an information age. *The Information Society*, 15, 1999.

[Nis04] Helen Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 71(1), 2004.

[Odl03] Andrew Odlyzko. Privacy, Economics, and Price Discrimination on the Internet. In *5th International Conference on Electronic Commerce (EC 2003)*. ACM Press, 2003.

[OO90] T. Okamoto and K. Ohta. Disposable zero-knowledge authentications and their applications to untraceable electronic cash. In *Advances in Cryptology - CRYPTO '89 – 9th Annual International Cryptology Conference, Proceedings*, number 435 in LNCS. Springer Verlag, 1990.

[OO00] Gina Colarelli O'Connor and Robert O'Keefe. The Internet as a new marketplace: Implications for consumer behaviour and marketing

management. In M. Shaw, R. Blanning, T. Strader, and A. Whinston, editors, *Handbook on Electronic Commerce*. Springer Verlag, 2000.

[Pea04] Robert Pear. Electronic cards replace coupons for food stamps. New York Times, June 23, 2004.

[PK01] Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity — A proposal for terminology. In *Anonymity 2000*, number 2009 in LNCS. Springer Verlag, 2001.

[PMPS00] Jaroslaw Pastuszak, Dariusz Michalek, Josef Pieprzyk, and Jennifer Seberry. Identification of bad signatures in batches. In *Public Key Cryptography (PKC 2000), Third International Workshop on Practice and Theory in Public Key Cryptosystems, Proceedings*, number 1751 in LNCS. Springer Verlag, 2000.

[Poi99] David Pointcheval. New public key cryptosystems based on the dependent-RSA problems. In *Advances in Cryptology - EUROCRYPT '99 – International Conference on the Theory and Application of Cryptographic Techniques, Proceedings*, number 1592 in LNCS. Springer Verlag, 1999.

[Pol07] Irene Pollach. What's wrong with online privacy policies? *Communications of the ACM*, 50(9), September 2007.

[PRW03] Arnold Picot, Ralf Reichwald, and Rolf T. Wigand. *Die grenzenlose Unternehmung, 5. Auflage*. Gabler Verlag, 2003.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3), 2000.

[PV04] Pino Persiano and Ivan Visconti. An efficient and usable multi-show non-transferable anonymous credential system. In *Financial Cryptography – 8th International Conference (FC 2004), Proceedings*, number 3110 in LNCS. Springer Verlag, February 2004.

[PW86] Andreas Pfitzmann and Michael Waidner. Networks without user observability — Design options. In *Advances in Cryptology - EUROCRYPT '85 – International Conference on the Theory and Application of Cryptographic Techniques, Proceedings*, number 219 in LNCS. Springer Verlag, 1986.

[Ram05] Anita Ramasastry. Web sites change prices based on customers' habits. `http://www.cnn.com/2005/LAW/06/24/ramasastry.website.prices/index.html`, 2005.

[Reb00] Michael Rebstock. Elektronische Geschäftsabwicklung, Märkte und Transaktionen – eine methodische Analyse. *HMD Praxis in der Wirtschaftsinformatik*, 37(215), 2000.

[Res05] Forrester Custom Consumer Research. Understanding consumers' needs for internet security. Survey commissioned by the Business Software Alliance, `http://www.bsa.org/uk/upload/Forrester%20Study%202005%20BSA%20All%20Countries%20FINAL.pdf`, November 2005.

[Ros00] Linda Rosencrance. Amazon charging different prices on some DVDs. Computerworld, `http://www.computerworld.com/industrytopics/retail/story/0,10801,49569,00.html`, September 2000.

[Roß02] Alexander Roßnagel, editor. *Datenschutz beim Online-Einkauf*. Vieweg Verlag, 2002.

[RP02] Marc Rennhard and Bernhard Plattner. Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2002)*, Washington, DC, USA, November 2002.

[RP04] Marc Rennhard and Bernhard Plattner. Practical anonymity for the masses with morphmix. In Ari Juels, editor, *Proceedings of Financial Cryptography (FC '04)*, number 3110 in LNCS. Springer Verlag, February 2004.

[RR98] Michael K. Reiter and Aviel D. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1), 1998.

[RSA78] Ron L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), February 1978.

[RSG96] Michael G. Reed, Paul F. Syverson, and David M. Goldschlag. Proxies for anonymous routing. In *Proceedings of 12th Annual Computer Security Applications Conference (ACSAC'96)*. IEEE Press, December 1996.

[RSG98] Michael G. Reed, Paul F. Syverson, and David M. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications — Special Issue on Copyright and Privacy Protection*, 16(4), 1998.

[Sch03] Philip Scholz. *Datenschutz beim Internet-Einkauf*. Nomos Verlagsgesellschaft, Baden-Baden, 2003.

[SD02] Andrej Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In *Privacy Enhancing Technologies (PET 2002)*, number 2482 in LNCS. Springer Verlag, 2002.

[SDP06] Arun Sen, Peter A. Dacin, and Christos Pattichis. Current trends in Web data analysis. *Communications of the ACM*, 49(11), November 2006.

[Ser04] Andrei Serjantov. *On the Anonymity of Anonymity Systems*. PhD thesis, University of Cambridge, Department of Computer Science, July 2004.

[SL98] Beat F. Schmid and Markus A. Lindemann. Elements of a reference model for electronic markets. In *Proceedings of the 31st Hawaii International Conference on System Sciences (HICSS'98)*, volume 4. IEEE Press, January 1998.

[SRG97] Paul F. Syverson, Michael G. Reed, and David M. Goldschlag. Private web browsing. *Journal of Computer Security — Special Issue on Web Security*, 5(3), 1997.

[SS97] Byron Sharp and Anne Sharp. Loyalty programs and their impact on repeat-purchase loyalty patterns. *International Journal of Research in Marketing*, 14(5), December 1997.

[SS03] Ahmad-Reza Sadeghi and Markus Schneider. Electronic payment systems. In *Digital Rights Management*, number 2770 in LNCS. Springer Verlag, 2003.

[SSG99] Stuart G. Stubblebine, Paul F. Syverson, and David M. Goldschlag. Unlinkable serial transactions: Protocols and applications. *ACM Transactions on Information and System Security*, 2(4), 1999.

[STZ07] Xuehua Shen, Bin Tan, and Cheng Xiang Zhai. Privacy protection in personalized search. *ACM SIGIR Information Retrieval Forum*, 41(1), June 2007.

[Sul03] Bob Sullivan. The darkest side of ID theft. MSNBC News, `http://www.msnbc.msn.com/id/3078488/`, March 2003.

[Syv03] Paul Syverson. The paradoxical value of privacy. *2nd Annual Workshop on the Economics of Information Security (WEIS 03)*, 2003.

[SZ95] Greg Shaffer and Z. John Zhang. Competitivec coupon marketing. *Marketing Science*, 14(4), 1995.

[Tav99a] Herman T. Tavani. Informational privacy, data mining, and the internet. *Ethics and Information Technology*, 1(2):137–145, June 1999.

[Tav99b] Herman T. Tavani. KDD, data mining, and the challenge for normative privacy. *Ethics and Information Technology*, 1(4):265–273, December 1999.

[Tav00] Herman T. Tavani. Privacy and security. In Duncan Langford, editor, *Internet Ethics*. St. Martin's Press, 2000.

[Tay02] Curtis R. Taylor. Private demands and demands for privacy: Dynamic pricing and the market for customer information. Technical Report, Department of Economics, Duke Universtiy, September 2002.

[Tim06] New York Times. Face Is Exposed for AOL Searcher No. 4417749. `http://www.nytimes.com/2006/08/09/technology/09aol.html?_r=1&oref=slogin`, August 2006.

[TM00] Gaurav Tewari and Pattie Maes. Design and implementation of an agent-based intermediary infrastructure for electronic markets. In *Proceedings of the 2nd ACM conference on Electronic Commerce (EC'00)*, October 2000.

[TM01] Herman T. Tavani and James H. Moor. Privacy protection, control of information, and privacy-enhancing technologies. *ACM SIGCAS Computers and Society*, 31(1), March 2001.

[Ved99] Anton Vedder. KDD: The challenge to individualism. In *Ethics and Information Technology*, number 1. Kluwer Academic Publishers, 1999.

[Ver01] Eric R. Verheul. Self-blindable credential certificates from the Weil pairing. In *Advances in Cryptology - ASIACRYPT '01, International Conference on the Theory and Applications of Crypotology and Information Security 2001, Proceedings*, number 2248 in LNCS. Springer-Verlag, 2001.

[vGF04] Birgit van Eimeren, Heinz Gerhard, and Beate Frees. ARD/ZDF-Online Studie 2004, Internetverbreitung in Deutschland: Potenzial vorerst ausgeschöpft? Media Persperktiven 8/2004, 2004.

[Wal99] Kathleen A. Wallace. Anonymity. *Ethics and Information Technology*, 1(1):21–31, March 1999.

[WB90] Samuel D. Warren and Louis D. Brandeis. The right to privacy. *Harvard Law Review*, 4, 1890.

[Wei06] Thilo Weichert. Verbraucher-Scoring meets Datenschutz. *Datenschutz und Datensicherheit (DuD)*, 30(7), 2006.

[Wes67] Alan F. Westin. *Privacy and Freedom*. Anatheum Press, New York, 1967.

[WLT00] Arrianto Mukti Wibowo, Kwok Yan Lam, and Gary S.H. Tan. Loyalty program scheme for anonymous payment systems. In *Electronic Commerce and Web Technologies*, number 1875 in LNCS. Springer Verlag, 2000.

[WLW98] Huaiqing Wang, Matthew K.O. Lee, and Chen Wang. Consumer privacy concerns about internet marketing. *Communications of the ACM*, 41(3), March 1998.

[WS79] H.C. Williams and B. Schmid. Some remarks concerning the M.I.T. public-key cryptosystem. *BIT*, 19(4), 1979.

[WSUK00] Dirk Westhoff, Markus Schneider, Claus Unger, and Firoz Kaderali. Protecting a mobile agent's route against collusions. In *Selected Areas in Cryptography, 6th Annual International Workshop (SAC'99)*, number 1758 in LNCS. Springer Verlag, 2000.

[XZCW07] Yabo Xu, Benyu Zhang, Zheng Chen, and Ke Wang. Privacy-enhancing personalized web search. In *Proceedings of the 16th WWW Conference*, May 2007.

[Yan02] Song Y. Yan. *Number Theory for Computing*. Springer Verlag, 2002.