A Privacy-Friendly Loyalty System Based on Discrete Logarithms over Elliptic Curves

Matthias Enzmann, Marc Fischlin*, and Markus Schneider

Fraunhofer Gesellschaft (FhG), Institute for Secure Telecooperation (SIT)
Dolivostr. 15, D-64293 Darmstadt, Germany
firstname.lastname@sit.fraunhofer.de

Abstract. Systems for the support of customer relationship management are becoming increasingly attractive for vendors. Loyalty systems provide an interesting possibility for vendors in customer relationship management. This holds for both real world and online vendors. However, beside some potential benefits of a loyalty system, customers may also fear an invasion into their privacy, and may thus refuse to participate in such programs. In this paper, we present a privacy-friendly loyalty system to be used by online vendors to issue loyalty points. The system prevents vendors from exploiting data for the creation of customer profiles by providing unconditional unlinkability of loyalty points with regard to purchases. In the proposed system, we apply the difficulty for the computation of discrete logarithms in a group of prime order to construct a secure and privacy-friendly counter. More precisely, all computations are carried out over special cryptographic groups based on elliptic curves where the decisional Diffie-Hellman problems can be solved easily while the computational Diffie-Hellman is believed to be hard.

1 Introduction

The World Wide Web has evolved to a business platform with worldwide reach and 24h/7 service for selling various kinds of goods. Presently, more than 600 millions of people have access to this business platform and thus, are potential customers for online vendors. Naturally, every online vendor's interest lies in attracting new customers and increasing the 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.

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

* Now at Department of Computer Science and Engineering, University of California, San Diego, USA. Most of work done while at Fraunhofer-Institute for Secure Telecooperation.

participate in the program, it may be worthwhile for vendors to reconsider their strategy of collecting personal data. Indeed, according to [18, 22], there are many customers that are concerned about their privacy in electronic commerce scenarios. Thus, privacy-friendly loyalty systems might be of particular interest to vendors in order to gain a competitive advantage.

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's in exchange for a reward. Usually, a reward can be obtained when a customer has reached a pre-defined number of loyalty points.

In order to enhance privacy for the customers, the vendor must not be able to generate consumer profiles by linking customers' transactions through the loyalty program. Thus, it is our goal to prevent the vendor from using loyalty points to link any two customer transactions. Hence, when points are handed in by the customer, it is not possible for the vendor to determine the purchases in which the points were obtained. Of course this is only meaningful if there is no other linking information available to the vendor 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.

The privacy-friendly loyalty system presented here uses an efficient variant of blind signatures which are based on discrete logarithms in groups of prime order. All computations are done in special cryptographic groups based on elliptic curves that allow to decide easily whether three given group elements form a Diffie-Hellman triple, while both the computational Diffie-Hellman and the Discrete Logarithm problem are conceivably intractable [19, 20]. We propose a counter-based solution in which multiplications in the elliptic curve are iteratively applied for each loyalty point that is issued. As it turns out, this yields more efficient solutions than with straightforward application of blind signatures (called token-based system), yet it also entangles the design and security analysis. Furthermore, in contrast to such a token-based system, the proposed counter-based system prevents different customers from pooling their loyalty points since values of different counters cannot be added up. In the redeem transaction, the counter which represents the loyalty points collected by a customer can be efficiently verified in one step by the vendor. The proposed loyalty system provides unconditional unlinkability of loyalty points with regard to purchases.

The paper is organized as follows. Section 2 gives some background on loyalty systems. In section 3 we define essential privacy and security requirements for loyalty systems. Section 4 provides necessary background on elliptic curves and proposes the protocols of our loyalty system. In section 5, we consider the properties of the proposed system. Related work is discussed in section 6, before we draw some conclusions.

2 Loyalty programs

A loyalty program is a structured marketing effort which rewards, and therefore encourages, loyal behaviour of customers, which is hopefully beneficial to the vendor [28]. We say that a customer is loyal if she has a strong attitude to 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. And second, they want to learn more about their customers in order to refine their business strategy.

In general, the basic conditions for loyal customer behaviour in the real world are different from the electronic world [26]. 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.

There are different types of loyalty programs, e.g., reward systems and virtual communities. Reward systems give program members a financial incentive. They can be classified according to the time the reward is given relative to the purchase. There are immediate reward systems, e.g., price promotions or rebates through membership credit cards, and delayed reward systems, e.g., point collecting programs like frequent flyer miles or "buy 10 get one free". Virtual communities focus on social and service aspects, e.g., online discussion panels on product related problems.

There are some variants for point-based loyalty programs. 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 one can download one for free. Furthermore, we can categorize point programs according to the way points are collected. In a token-based approach, for each awarded point a token is issued, e.g., chips issued by a supermarket, while in a counter-based approach the number of points to be obtained is added to the current point balance, e.g., frequent flyer miles.

Members of loyalty programs have a greater propensity to be loyal to the vendor and also have an increased usage frequency compared to non-members [28]. Furthermore, it can be assumed that members are less willing to try offers of competing vendors, even when negative experiences with the vendor occur since these effects are moderated by the loyalty program membership [5]. According to [14], loyalty program members are also less price sensitive, spend more money 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 optimize their range of products, respectively. This means that vendors have a large consumer database 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 losing control over their personal data, since vendors may disclose their data to other parties. 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. In this paper, we propose a point-based loyalty system which may lead to increased customer loyalty due to enhanced privacy.

3 Requirements

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

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. This means, given a redeeming transaction, the vendor should be prevented from linking it to the corresponding awarding transactions and to 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 systems' properties that are necessary to achieve unlinkability. Clearly, linkability may be possible outside the loyalty system. However, preventing this 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 [16], payment systems that allow the customer to remain anonymous with respect to the vendor [12, 9], anonymity networks as in [11, 27], or privacy-friendly delivery in case of hard goods similar to the approach proposed in [15].

Security. The security requirements considered here can be summarized 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 several aspects of system integrity that need to be considered.

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 at the vendor's. Thus, we require that it must be *detectable* whether loyalty points have been spent before.

Pooling prevention. In general, vendors do not want different customers to pool their loyalty points in order to jointly achieve the redeem threshold. Thus, the loyalty system should prevent successful pooling, e.g., it should be impossible for two users to transform their individual counter values of, say, 5 into a joint counter of 10. Note that this does not address the problem of colluding customers sharing a counter; the latter cannot be prevented in systems with perfect privacy.

4 Construction of the loyalty system

In this section, we present the counter-based loyalty system. Before presenting the protocols, we introduce the specific type of elliptic curves our scheme relies on and some

important facts. Using these curves allows customers in our construction to verify the validity of issued loyalty points as will become clear later.

4.1 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 recently discovered property of some of these curves. Namely, we deploy special elliptic curves for which the *computational* Diffie-Hellman problem (given g, g^a, g^b determine g^{ab}) is believed to be intractable, whereas the *decisional* Diffie-Hellman problem (given g, g^a, g^b, g^c decide if $g^c = g^{ab}$) is known to be easy.[1] Such elliptic curves have been suggested only recently [19, 20] but have immediately gained a lot of attention because of their usefulness for the design of cryptographic protocols, e.g., [6, 7, 4, 13].

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) [20, 7];
- fast computation of the pairing functions [6, 1, 17], i.e., fast verification of putative DH triples (g^a, g^b, g^c);
- hashing into the curve [7]; that is, how to define a hash function H 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 [25].

4.2 Protocols

The loyalty scheme consists of two protocols, the *issue* and *redeem* protocol. Both protocols involve two parties, the vendor and the customer. 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.

Initialization. The system is set up as follows. The vendor chooses an appropriate elliptic curve for which the decisional Diffie-Hellman problem can be decided efficiently but for which the computational Diffie-Hellman problem is presumably hard. The order of the group 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 5.3). Let g be a generator of this curve. From now on, unless otherwise noted, it is understood that all computations are done in the curve.

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

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

Fig. 1. Initialization

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 curve) as his public key and keeps v private. The customer chooses a random serial number s from some finite set $\mathbb{S}_n$. 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 $c_0 := H(s)$, where H is some cryptographic hash function mapping to the group. This hash function should be specified and published by the vendor, too. The initialization process is depicted in Figure 1.

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 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 curve. Note that, in general, this validity test is intractable for groups like $\mathbb{Z}_p^*$. If the verification here succeeds then the customer sets $c_i := c_{i-1}^v$ and stores (i, c_i). The issue protocol is shown in Figure 2.

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 $c_i := c_{i-1}^v$;		

Fig. 2. Issue protocol in the customer's i-th purchase

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

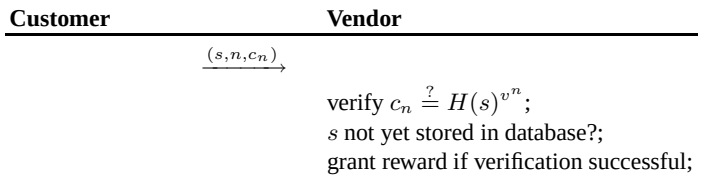

Fig. 3. Redeem protocol

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. 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 $c_n c'_n = (ss')^{v^n}$ for serial number ss'.

5 Properties

5.1 Privacy

Privacy of the customer follows easily from the fact that the element b_i in the issue protocol is uniformly and independently distributed since the values of r_i are chosen independently. This also holds if the vendor knows the serial number s and all data derived from s like $c_0 = H(s)$, $c_1 = c_0^v$, etc. This means that any two issue transactions cannot be linked by the vendor provided that there is no additional information that can be used for linking purposes. The same holds for the linkability of issue and redeem transactions. No execution of the issue protocol can be assigned to a specific customer then, even after revealing (s, n, c_n) in the redeem protocol and even if the vendor has unlimited computational power.

5.2 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 [2, 4]. 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.

Attack model. The attack model is as follows. We assume that the adversary controls several customers and coordinates their activities. Note that this covers "less malicious" cases where, say, some adversarial users act individually. The adversary is allowed to run issue protocols with the honest vendor and finally engages in a redeem protocol execution. The goal of the adversary is to claim more points than issued in total to these users.

There is a subtlety in the formalization of the adversary "redeeming more points than issued". Recall the example of two users both having already 5 individual points and then trying to combine their points to a joint counter value of 10. In this case 10 points have already been issued indeed. Hence, the two adversarial users actually do not redeem more points than earned before, yet they illicitly pool them. The definition of pooling prevention should capture such misbehavior. We therefore augment the attack model by so-called scheduled users in addition to the controlled users.

A scheduled user is basically an autonomous customer following the protocol honestly. The adversary merely schedules the actions of this user, i.e., determines when this user runs the initialization or issue protocol. More precisely, the adversary can perform three operations with scheduled users. First, the adversary can create a new scheduled user during the attack. This new user immediately follows the prescribed initialization protocol, i.e., chooses a serial number s and computes $c_0 := H(s)$. The user goes idle until the adversary wakes him up again. Second, the adversary may call a scheduled user and ask him to step the counter. In this case the user runs the issue protocol with his current counter value and returns to an idle state again. We assume that the scheduled user also stores the intermediate values in addition to the current counter value (i.e., previous blinding and counter values).[2] Third, at any time the adversary may corrupt a scheduled customer which then becomes a controlled user; the adversary gets all the previously stored information and the current counter value, and from now on coordinates all the user's activities. Note that the adversary still controls the set of corrupted users in addition to such scheduled customers.

We count the issued points as follows. For each scheduled user we individually count the number of issue protocol invocations for this user (until the user becomes corrupted or the attack ends). If a controlled customer starts an issue execution then we

[2] 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.

increment the adversary's global count instead. By this, we have an individual number n_{cust} of invocations for each customer cust (possibly $n_{\text{cust}} = 0$ if the customer has been corrupted right away or has never run the issue protocol), and a global number n_{adv} of invocations with fully controlled customers. We now say that the adversary *breaks the system* if the adversary successfully claims strictly more than $n_{\text{adv}} + \max\{n_{\text{cust}}\}$ points for some user, where the maximum is over all customers cust appearing during the attack.

Note that the lower bound itself, $n_{\text{adv}} + \max\{n_{\text{cust}}\}$, can trivially be reached by any adversary scheduling a customer $\max\{n_{\text{cust}}\}$ times and then corrupting the customer (thereby taking over its counter) and performing n_{adv} subsequent runs of the issue protocol by herself (using the corrupted customer's counter as her starting counter). Claiming more than $n_{\text{adv}} + \max\{n_{\text{cust}}\}$ points captures the cases where (a) the adversary manages to add at least one additonal point that was not issued by the vendor to some counter, (b) two or more customers manage to pool their counters, or (c) a combination of both.

In how far do scheduled users reflect pooling attacks? In the example of two customers merging their counter values of 5, one may think of these users as scheduled users. The adversary then corrupts them and tries to redeem 10 points. In this simple attack we have $\max\{n_{\text{cust}}\} = 5$ and $n_{\text{adv}} = 0$ and, according to the definition, the adversary breaks the system if she manages to redeem 6 or more points for some user (by pooling both counter values, yielding 10 points, or by increasing the counter by at least one point using some other means).

The incremental Diffie-Hellman problem. The incremental Diffie-Hellman problem is to find $n \geq 1$ and $g^{v^{n+1}}$ 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 $n-1$ times, e.g., to compute g^{v^3} from g, g^v one may make a single call to the oracle. Specifically:

Definition 1 (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 $1 \leq n < \mathrm{ord}_{\mathbb{Z}_q^*}(v) - 1$ such that*

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

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

The upper bound on the integer n rules out trivial solutions. Else, $Z := V$ would for example be a correct claim for any multiple n of the order $\mathrm{ord}_{\mathbb{Z}_q^*}(v)$ of v in $\mathbb{Z}_q^*$ because $g^{v^{n+1}} = g^v = Z$. For our scheme we therefore choose a sufficiently large order for v; see Section 5.3 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. But any solution in the experiment will immediately give a solution for the iDH problem. We conclude that each forger for our protocol must implicitly solve the iDH problem.

In the experiment we will model the hash function H mapping serial numbers to group elements as a so-called random oracle [3]. That is, we assume that H acts as a random function: it maps inputs to uniformly and independently distributed group elements, repeating answers for previously queried inputs. Note that the idealized random oracle model merely provides some heuristic evidence that the scheme is indeed secure; refer to [10] for a discussion. Therefore, in Section 5.3 we also present a modification which completely forges random oracles but which essentially preserves the efficiency (with only a negligible loss in the initialization 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 $N_{\text{cust}} := \max\{n_{\text{cust}}\}$ of issued points for scheduled users in the upcoming experiment. This value is usually bounded by a parameter N representing the system's maximum of redeem points. Instructively, think of N as 10 or $1,000$.

To guess $N_{\text{cust}} = \max\{n_{\text{cust}}\}$ the iDH algorithm picks a uniformly distributed value between 0 and N. The forger's view in the following simulation is independent of this choice, and the iDH algorithm thus hits the right value with probability $1/(N+1)$. 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 $1/(N+1)$ compared to the forger. From now on, we condition on the event that the iDH algorithm selects the correct N_{cust}.

We describe the simulation of the forger. The iDH algorithm is given g and V and access to the oracle, and has predicted N_{cust}. It first computes $g^{v^2}, g^{v^3}, \ldots, g^{v^{N_{\text{cust}}+1}}$ by iteratively querying the oracle, starting with V. This can be done with N_{cust} 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 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.
- 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 $w_s \in \mathbb{Z}_q$ (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 N_{cust} and therefore $g^{v^{i+1}}$ must be in this list.
  - On behalf of the customer select $r_i \in \mathbb{Z}_q$ 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 $n \geq 1$ for some serial number s then the iDH algorithm computes $w_s^{-1} \bmod q$ and outputs $Z^{w_s^{-1}}$ and n and stops.[3]

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 she succeeds in the simulation with the same probability (under the condition that the iDH solver has guessed N_{cust} 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,

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

Furthermore, $n > \max\{n_{\text{cust}}\} + n_{\text{adv}}$ which implies

$$n \geq \max\{n_{\text{cust}}\} + n_{\text{adv}} + 1$$

Since the iDH algorithm has queried its oracle exactly $\max\{n_{\text{cust}}\} + n_{\text{adv}}$ times this means that $Z^{w_s^{-1}}$ and n constitute a valid solution to the iDH problem. Therefore, we have presented an algorithm solving the iDH problem whenever the forger succeeds and the 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 her 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 $\frac{1}{N+1}(\varepsilon - \frac{2}{q})$.

[3] There is a very small probability that $w_s = 0$ which has no inverse in $\mathbb{Z}_q$, or that the forger successfully claims a counter value for a number s that has not been passed to the hash function before. However, both probabilities are equal to $1/q$ and we thus neglect them 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 $n \geq 1$ and $g^{v^{n+1}}$ after having made at most $n-1$ 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, $g^v \mapsto g^{v^n}$ for unknown v and some $n > 1$. This function, however, has the same power as the classical DH function for n's of order $O(\sqrt{\log q})$, refer to [24, 21].

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^n} requires n *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 [8].

In our case the constant 2 in the computation of V^{2^n} 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^n} with less than n exponentiations (using some preprocessing).

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.

5.3 Efficiency and Implementation Issues.

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 [20, 6, 7, 1, 17] for such choices. Indeed, 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 initialization step. The unforgeability now follows from the hardness of the iDH problem alone.

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 initializing a new counter. Second, requesting a serial number might be correlated with a purchase/issue transaction. This may allow the vendor to link the redeem 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 $g^{v^n} = g^{v^{n+i\cdot \mathrm{ord}_{\mathbb{Z}_q^*}(v)}}$ for any $i \geq 0$, an adversary may claim higher counter values $n + i \cdot \mathrm{ord}_{\mathbb{Z}_q^*}(v)$ instead of n. 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 $w := v^n \bmod q$ over $\mathbb{Z}_q^*$ and then $H(s)^w$ in the elliptic curve and finally compares it with the given c_n. Altogether these are only two exponentiations, and thus improves efficiency over the verification of n 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 $n = 10$. Verification of a claim then essentially boils down to a single exponentiation.

The proposed solution has an efficiency drawback in a model that allows customers to be issued more than one loyalty point in one purchase. If a customer should obtain $m > 1$ points in one purchase, the issue protocol has to be carried out m times.

6 Related Work

Much work has been done by economic and marketing experts in the field of loyalty systems, e.g., see [5, 28, 14]. Furthermore, there has been lots of work stressing the importance of privacy for electronic commerce, e.g., see [18]. 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 widthdrawal and desposit has been considered [12, 9]. In [16], a solution to establish the unlinkability of the customer's search and order phases has been proposed. 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 [23]. In this work, an infrastructure based on smart cards is proposed which allows individuals to introduce their own currencies or loyalty systems. However, they do not deal with the problem of achieving privacy in loyalty systems. Another proposal for a loyalty system was presented in [29]. In this work, the authors respect the privacy aspect. However, the goal of the system was not to provide unlinkability of transactions. The solution is based on pseudonymity, and thus provides a weaker form of privacy protection.

7 Conclusion

We have presented a privacy-friendly loyalty systems that does not allow vendors to link customers' transactions. The presented approach basically consists of a counter for loyalty points secure against forging and linking of transactions. The counter is increased in a blind signatures protocol exploiting the problem to compute discrete logarithms in groups of prime order. In the redeem phase, the counter can be verifi ed effi ciently in one step, regardless of the number of loyalty points that have been collected. Loyalty systems can provide an important strategy for vendors' customer relationship management to retain customers and to increase the incentive for repeated buying. The privacy property of our proposal may attract customers that usually refuse to become members of a loyalty program since they fear infringements of their privacy.

References

1. Paulo S.L.M. Baretto, Hae Y. Kim, Ben Lynn, Michael Scott. Efficient algorithms for pairing-based cryptosystems. In *Advances in Cryptology - CRYPTO 2002 – 22th Annual International Cryptology Conference, Proceedings*, LNCS 2442. Springer Verlag, 2002.
2. M. Bellare, C. Namprempre, D. Pointcheval, M. Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. In *Journal of Cryptology*, Vol. 16, No. 3, 2003.
3. Mihir Bellare, 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.
4. Alexandra Boldyreva. Efficient threshold signatures, multisignatures and blind signatures based on the Gap-Diffie-Hellman-group signature scheme. In *Public Key Cryptography (PKC) 2003*, LNCS 2567. Springer Verlag, 2003.
5. Ruth N. Bolton, P. K. Kannan, 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.
6. Dan Boneh, Matthew Franklin. Identity based encryption from the Weil pairing. In *Advances in Cryptology - CRYPTO 2001 – 21st Annual International Cryptology Conference, Proceedings*, LNCS 2139. Springer Verlag, 2001.
7. Dan Boneh, Ben Lynn, 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*, LNCS 2248. Springer Verlag, 2001.
8. Dan Boneh, Moni Naor. Timed commitments (extended abstract). In *Advances in Cryptology - CRYPTO 2000 – 20th Annual International Cryptology Conference, Proceedings*, LNCS 1880. Springer Verlag, 2000.
9. Jan Camenisch, Jean-Marc Piveteau, Markus Stadler. An efficient fair payment system. In *1st ACM Conference on Computer and Communications Security (CCS'96)*. ACM Press, 1996.
10. Ran Canetti, Oded Goldreich, Shai Halevi. The random oracle methodology, revisited. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing (STOC 1998)*. ACM Press, 1998.
11. David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2), February 1981.
12. David Chaum. Privacy protected payments: Unconditional payer and/or payee untraceability. In *Smart Card 2000, Proceedings*. North Holland, 1989.

13. Yevgeniy Dodis. Efficient construction of (distributed) verifiable random functions. In *Public Key Cryptography (PKC) 2003*, LNCS 2567. Springer Verlag, 2003.
14. Grahame R. Dowling, Mark Uncles. Do customer loyalty programs really work? *Sloan Management Review*, 38(4), 1997.
15. Matthias Enzmann, Claudia Eckert. Pseudonymes Einkaufen physischer Güter. In *Sichere Geschäftsprozesse, Tagungsband zur Arbeitskonferenz Elektronische Geschäftsprozesse*. IT Verlag für Informationstechnik, 2002.
16. Matthias Enzmann, Thomas Kunz, 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)*, LNCS 2455. Springer Verlag, September 2002.
17. Steven D. Galbraith, Keith Harrison, David Soldera. Implementing the Tate pairing. In *Algorithmic Number Theory, 5th International Symposium, ANTS-V, Proceedings*, LNCS 2369. Springer Verlag, 2002.
18. Donna L. Hoffman, Thomas P. Novak, Marcos Peralta. Building consumer trust online. *Communications of the ACM*, 42(4), April 1999.
19. Antoine Joux. A one round protocol for tripartite Diffie-Hellman. In *Algorithmic Number Theory, 4th International Symposium, ANTS-IV, Proceedings*, LNCS 1838. Springer Verlag, 2000.
20. A. Joux, K. Nguyen. Separating Decision Diffie-Hellman from Diffie-Hellman in cryptographic groups. Cryptology ePrint Archive, Report 2001/003, 2001. http://eprint.iacr.org/.
21. 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*, LNCS 2247. Springer Verlag, 2001.
22. Alfred Kobsa. Tailoring privacy to users's needs. In *User Modeling 2001 (UM 2001), 8th International Conference, Proceedings*, LNAI 2109. Springer Verlag, 2001.
23. David P. Maher. A platform for privately defined currencies, loyalty credits, and play money. In *Financial Cryptography, Second International Conference (FC'98), Proceedings*, LNCS 1465. Springer Verlag, 1998.
24. Ueli M. Maurer, Stefan Wolf. Diffie-Hellman oracles. In *Advances in Cryptology - CRYPTO '96 – 16th Annual International Cryptology Conference, Proceedings*, LNCS 1109. Springer Verlag, 1996.
25. Alfred J. Menezes. *Elliptic Curve Public Key Cryptosystems*, volume 234 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, 1993.
26. Gina Colarelli O'Connor, Robert O'Keefe. The Internet as a new marketplace: Implications for consumer behaviour and marketing management. In M. Shaw, R. Blanning, T. Strader, A. Whinston, (eds), *Handbook on Electronic Commerce*. Springer Verlag, 2000.
27. Michael G. Reed, Paul F. Syverson, 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.
28. Byron Sharp, Anne Sharp. Loyalty programs and their impact on repeat-purchase loyalty patterns. *International Journal of Research in Marketing*, 14(5), December 1997.
29. Arrianto Mukti Wibowo, Kwok Yan Lam, Gary S.H. Tan. Loyalty program scheme for anonymous payment systems. In *Electronic Commerce and Web Technologies*, LNCS 1875. Springer Verlag, 2000.