# ANONUS: Anonymous Bonus Point System with Fraud Detection

Samuel Brack Stefan Dietzel Björn Scheuermann

Humboldt-Universität zu Berlin, Germany
E-Mail: samuel.brack@informatik.hu-berlin.de, stefan.dietzel@hu-berlin.de, scheuermann@informatik.hu-berlin.de

***Abstract*—Bonus point systems are widely used for rewarding customer loyalty in both traditional and electronic commerce scenarios. Merchants aim to additionally increase revenue by targeted advertising using customer data. At the same time, customers wish to protect their privacy of what they purchase. Common systems neither provide sufficient protection of privacy, nor protect the customers' identities. Anonymity makes it significantly harder to properly resolve claims of fraudulent transactions, because identities are not recorded for any purpose after a transaction has been committed. We propose an anonymous bonus system based on digital payment systems using blind cryptographic signatures. We implement a protocol extension that allows proof of bonus point possession in case of reported misbehavior, and we identify the implications on security, privacy, and performance of our proposals. Our results show that we can resolve these cases of fraud within the system without losing functionality in the bonus point protocol.**

***Index Terms*—Blind Signature; Asymmetric Cryptography; Digital Signature**

## I. Introduction

Customer loyalty programs are widely used by online and offline merchants. These systems often use some sort of quantifiable and discrete "currency", e. g., bonus points or miles. We term one unit of customer reward a "bonus point" or "points" in the remainder of this article. Bonus points are generated for purchases at the issuing merchant. After collecting a certain number of points, customers can redeem rewards and discounts or reach status levels with specific perks. Some schemes allow direct payment at the bonus network's merchants using bonus points.

Bonus point system issuers like Payback [1] in Germany provide a bonus system as a service to merchants with interchangeable bonus points in exchange for aggregated customer data over the entirety of all participating merchants. Individual profiles of customers generated through these systems are often used as the foundation for in-depth market analysis and targeted advertising. Due to the nature of these systems, there is usually full transparency of the customer for the bonus point issuer, i. e., the issuer can track the users' complete purchase histories. While privacy-preserving data mining approaches are widely known [2], existing bonus point systems typically use classic ways to conduct their analysis. From the accumulated data, even more sensitive personal information may be derived, e. g., if a customer is on a specific diet or if they are of a specific sexual orientation [3]. There are also state-level adversaries with potential access to the issuer's database. Potential customers may fear these means of data mining on their sensitive personal data [4] and decide to not participate in the bonus program, even if the system offers significant rewards and advantages.

However, the actual primary goal for a modern customer loyalty system is not the generation of highly personal profiles, but an increase in revenue by incentivizing customers to spend more money. This is usually achieved by shipping targeted advertising (and possibly special offers with lower prices) to customers. Targeted advertising contains offers that match the customer's needs deduced from their purchasing history. In current systems, this is based on the detailed knowledge of every customer. However, a customer loyalty system where the system learns about the entire market (and not about individuals) and detects trends affecting greater parts of the population can indeed generate advertising offers for selected subgroups of the participants without knowing their individual purchase histories. By shipping these offers to the relevant subgroups without actually knowing the receivers, it becomes possible to create an entirely anonymous bonus point system. With every receiver knowing their own purchasing history, a customer's device can be imagined to filter relevant promotions and adverts from a broadcasted collection of advertisement content.

Obviously, an important building block in such an architecture are the bonus points themselves. This is the aspect we focus on in this paper: we show how an anonymous bonus system can be constructed. To this end, we consider the following scenario: Alice is customer at several different merchants, including Bob's and Charlie's shops. Bob and Charlie both participate in the bonus point system operated by the system operator Debbie. Alice receives bonus points from Debbie for every purchase she conducts at either of the merchants. Bonus points are collected locally in a digital wallet on Alice's personal mobile device, e. g., on a smartphone. After collecting some of these points, she can use them to pay for a purchase at one of the participating stores. Of course, double spending (i. e., Alice paying more than once using the same points) is to be avoided, while at the same time we want to achieve that Alice does not need to reveal her identity towards any other party during the process. It should also not be possible to trace bonus points, i. e., the transaction or merchant where the used points have been generated should not be revealed.

Our system is based on blind signatures as proposed by

© 2017 IEEE doi.org/10.1109/LCN.2017.50

David Chaum [5] in the context of e-cash. As we will show, applying Chaum's ideas already get us a long way towards the above stated goals. However, there are also shortcomings, and we will point out where naïve applications of the e-cash payment protocol open up potentials for fraud. The usage of an anonymous payment system depends on reliable and instant fraud prevention, protecting both the merchants and the customers. A key contribution of this paper is that we show how these problems can be overcome.

The remainder of this paper is structured as follows. Section II discusses related work from the areas of cryptographic currencies and customer loyalty systems. Following that, Section III leads to our proposal for such a system. Fraud prevention techniques are presented in Section IV. An evaluation for the proposed strategy is given in Section V where we identify the impacts on privacy, security, and the electronic cash protocol. Finally, Section VI summarizes this paper.

## II. Related Work

In this section, we survey existing literature related to our proposal. After introducing the state of the art in electronic bonus point systems we discuss the applicability of existing decentrally and centrally operated electronic payment systems and show where our proposal differs from existing ideas.

### A. *Electronic Bonus Point Systems*

Many customer loyalty systems use an electronic backend that processes user data in order to generate market reports and targeted advertising. Apart from places like small food shops with paper-and-stamp-based loyalty cards, most of these systems either use centralized, account-backed plastic cards or mobile apps for their user management [6]. Payback [1], the biggest customer loyalty program in Germany is used by millions of people and identifies every customer over several stores. Each purchase is tracked and targeted advertising, as well as targeted offers, are distributed. While participation in this program does not necessarily require a valid address and the usage of one's real identity, a combination with credit card data or other tracking information impedes or even prevents truly anonymous usage. In contrast to our proposed system, Payback's users are not anonymous and their purchases can potentially be tracked in a central system.

Other approaches for a distributed bonus system [7] reward customers for recommending merchants to other customers. The authors propose a mobile system where bonus points can be transferred from one person to another and redeemed at any participating merchant. However, these systems are not anonymous and users can be identified with their central accounts.

Generally, we observe that bonus points can have many properties of traditional cash. Bonus points are emitted by a set of cooperating merchants that want to redeem their customers' loyalty. They can be accumulated, (possibly) transferred to other users, and finally used for buying goods and services at participating merchants. Whether these goods and services are restricted to a certain set of items or whether the bonus points can be used as a real means of payment does not affect the technical design of the bonus system.

Anonymous bonus points also have to be untraceable, but also capable of preventing double spending. These two features are also present in cash money. Therefore, a bonus point system based on some form of electronic cash is a manifest option. The following analysis of anonymous payment systems is meant to provide an overview of potential ways to implement such a bonus system. Non-anonymous electronic payment systems like, e. g., those found in online games like World of Warcraft [8], will not be discussed, as we aim for a privacy-preserving bonus point system with anonymous participants.

### B. *Decentralized Electronic Assets*

Systems with central control are usually regulated by a state because of the immediate liability of the central issuer in its local legal framework. If a system requires to avoid central issuers in favor of a completely distributed management of assets, the ecosystem around the Bitcoin [9] protocol is a promising alternative. In Bitcoin, all transactions are stored in a distributed ledger, the so-called blockchain. This enables the system to operate without any central issuer. The assets in Bitcoin are mined by its users in a distributed proof-of-work process. Bonus points, on the other hand, have to be generated after each completed transaction of goods between a merchant and a customer. Thus, a system where merchants mine their own bonus points in a system similar to Bitcoin would need thorough adjustments so that every merchant has enough coins even in times of unusually high volume of sales. Customers, too, often hold no real interest in providing their mobile data connection to the Internet in order to contribute to a blockchain for bonus points. Proof-of-work-based systems have the additional disadvantage of requiring computing resources that cannot be used for other purposes while mining, making it infeasible for mobile devices of customers. Another disadvantage, however, are more complex regulations for merchants that plan to use Bitcoin or other cryptocurrencies legally.

Ethereum [10] is a framework for smart contract that uses its own blockchain. These contracts are publicly recorded and executed by the users of this distributed system. They are in fact turing-complete programs. A theoretical way to implement a bonus program would be to build it as a smart contract system on top of the Ethereum blockchain. However, a major concern is the slow process of transaction clearing. Blockchain transactions have to be in one of the "surviving" blocks in case of a fork, thus participants have to wait up to an hour to have relative certainty of a finished transaction. Customers of bonus point system expect instant clearing and are generally not interested in uncertain and slowly arriving bonus points.

### C. *Centrally Operated Electronic Cash Schemes*

Traditional electronic cash systems rely on a central issuer. That issuer performs a role comparable to central banks in state-issued cash currencies. The concept of electronic cash was originally introduced by David Chaum in [5], [11]. Chaum proposes a system where units of digital currency, *coins*, are stored decentrally by the users just like traditional coins and notes. These coins are identified by serial numbers; they gain their value by being digitally signed by the central issuer. Consequently, a valid coin is embodied by the tuple of a (user-selected) coin ID and the corresponding digital signature issued by the issuer. This construction ensures that no coin can be forged by a single user without breaking the issuer's private RSA [12] signing key. In order to guarantee anonymity, the signatures are conducted blindly: the issuer does not learn the coin ID which it signs. To this end, the user performs a blinding operation on the coin ID and hands the blinded ID to the central issuer. There it is signed and returned. Afterwards, the user reverts the blinding operation on both the blinded ID and the signature and thereby gains a valid digital signature for the unblinded original ID. That ID is now only known to the user and certified by the central issuer's signature. Other implementations of Chaum's scheme use other public-key systems, e. g., the discrete logarithm problem [13] or the ElGamal system [14].

When paying with a coin in such a system, the user hands over the coin ID and the signature to the receiver of the payment. The receiver can check offline whether the signature is correct (given he knows the public key of the central issuer). An online check is required to prevent double spending: the issuer is notified of the secret ID the user spends and this ID is listed as "spent" by the central issuer. If a second transaction using the same ID is attempted, the central issuer can tell the receiver to not accept the already spent coin. The downside of that approach is the requirement to have an online connection to the central issuer's system at the time of the payment.

Other mechanisms for preventing double spending are suitable in an offline scenario. The basic idea is to embed the payer's identity into the coins in such a way that at least two distinct entities need to cooperate in order to reveal a user's identity [15]. These two entities have to possess certain decryption information, which can only be obtained if the coin has been spent at that entity. Identification of the fraudulent payer can only be performed if two parties cooperate and both have received the same coin from the payer. Thus, if an honest user does not try to cheat and therefore does not spend coins twice, unwanted identification is not possible. Such systems do not need an immediate fraud detection technique, it is sufficient to deanonymize the fraudulent party at a later point and use legal means to punish them and go for compensation. Their main advantage is the possibility to be used in scenarios where the victim of the fraud is offline during the transaction and comes online later, after the fraudulent customer has left. All of these systems, however, rely on a central issuer to at least perform the signatures.

To our knowledge, there exists no bonus point system using electronic cash as a basis for operation. Its advantages of guaranteed anonymity, central control, and user-friendliness make it favorable for such a system.

## III. Anonymous Bonus Points

As discussed before, the anonymous bonus point system is based on Chaum's electronic cash [5]. We identify three essential actors: many *customers*, like Alice, buy goods and services at several *merchants*, two of which are Bob and Charlie. These merchants, in turn, participate like Alice in a customer loyalty system operated by the *bonus point issuer* Debbie. In the literature about electronic cash [11], these actors are often referred to the payers, payees, and the issuer, respectively.

### A. Bonus Point Generation

Figure 1 shows the life cycle of a bonus point. A simple, first idea of how to use Chaum's electronic cash for this use case is sketched in the following.

Bonus points are created after Alice finishes a real-world transaction paying for goods or services at Bob. The next step is the generation of a bonus point. This operation is executed between Alice and the bonus point issuer Debbie. Essentially, the customer locally generates a random number $C$ that is used as the bonus point ID. This ID has to remain secret until the bonus point is spent again. Afterwards, $C$ is blinded using a blinding function $b()$ so that the original value of $C$ is concealed. In case of RSA [12], this function utilizes Debbie's public key $(e,N)$ and another random secret number $r$ [11]

$$b(C) \coloneqq r^e \cdot C \mod N.$$

Other cryptosystems support similar constructions with different forms of the blinding function, but providing the same functionality [16].

This blind ID can be sent from Alice to Debbie. Debbie signs the blinded bonus point ID using a signature algorithm $\sigma()$—RSA in this example—using her private key $(d,N)$

$$\sigma(b(C)) \coloneqq b(C)^d \mod N = (r^e \cdot C)^d \mod N.$$

Note that $d \cdot e = 1$ in RSA where $e$ is the *private modulus* from Alice's private key.

The signature is then sent back to Alice who can apply the unblinding function $u()$

$$u\big(\sigma(C),r\big) \coloneqq r^{e \cdot d} \cdot C^d \cdot \frac{1}{r} = C^d = \sigma(C).$$

Thus, the bonus points are embossed with a unique ID $C$ only known to the customer Alice. ID collisions are theoretically possible, therefore the ID space has to be big enough to make it unlikely that two customers select the same random number. With sufficiently long IDs (128 bit or more), collisions are unlikely enough for practical applications [17].

Spending the bonus point begins by transmitting its ID $C$ and the corresponding signature $\sigma(C)$ from the customer Alice to the merchant Bob. The first step for Bob is to check if $\sigma(C)$ is indeed a valid signature for $C$. To prevent double spending, he reports $C$ to a central database hosted by the bonus point issuer Debbie. Debbie checks if this bonus point ID is so far unknown and returns the answer. The bonus point ID is stored as *spent* in the database. As a final step of the transaction, Bob relays that answer to the customer and thus finalizes the transaction. A correctly identified bonus point cannot be spent twice if the database is queried during the transaction and correctly updated after its completion. This process so far essentially follows the e-cash protocol as proposed by Chaum.

However, we make an important observation here: in the usage scenario as outlined above, consider the step where Alice hands over $C$ and $\sigma(C)$ to Bob. From this point on, Bob has knowledge of $C$—but there is no imminent guarantee for a cleared transaction for Alice. As we will see, this opens up potentials for the merchant Bob to cheat on the customer Alice, without Alice even being able to prove that this happened. We will come back to this issue soon.

### *B. Merchant Vouchers*

Before we take up the problem of the merchant being able to cheat on the customer, we consider the case of a potentially cheating customer Alice. If Alice is acting maliciously, she might send blinded bonus point IDs to Debbie that are not backed by any transaction at one of the merchants. Obviously, we need a way for Debbie to check that the bonus point creation is backed by a corresponding transaction at one of the merchants. This will go hand in hand with Debbie billing the merchants for the issued bonus points—and all that without revealing Alice's identity to any other entity.

We introduce what we call *vouchers* into the bonus point generation process. Figure 2 shows the accordingly modified bonus point protocol. During the checkout process, Alice receives a voucher $V$ for every unit of money spent. The voucher contains a serial number that is unique and linked to the merchant Bob. Vouchers are generated by or registered at Debbie. They can be created either online during the payment process or potentially also proactively in advance. If Alice is to receive one bonus point, Bob will transfer a voucher to Alice.

The voucher then serves as a form of payment for the blind signature to ensure that an eligible transaction grounds the issued blind signature. Consequently, when Alice asks Debbie to sign a bonus point, she has to provide a valid (and yet unused) voucher in exchange. Moreover, Debbie can bill Bob for the exact amount of bonus points originating at his store, simply by billing Bob for the vouchers that are issued for him.

## IV. Fraud Prevention

Coming back to the problem pointed out above, a fraudulent merchant can, however, still illicitly gain a benefit by cheating on Alice. When Alice pays at Charlie's with bonus points, Charlie may instantly decline the transaction as invalid, while still gaining knowledge of the secret bonus point ID including the corresponding signature of Debbie. Alice will only receive the response that the bonus point has not been accepted. She cannot check (let alone prove) whether the information received from the merchant is correct. That is, Alice cannot check whether the bonus point is indeed not valid, i. e., has previously been used for a payment already, or whether Charlie is lying by claiming the, actually unused, point has been used.

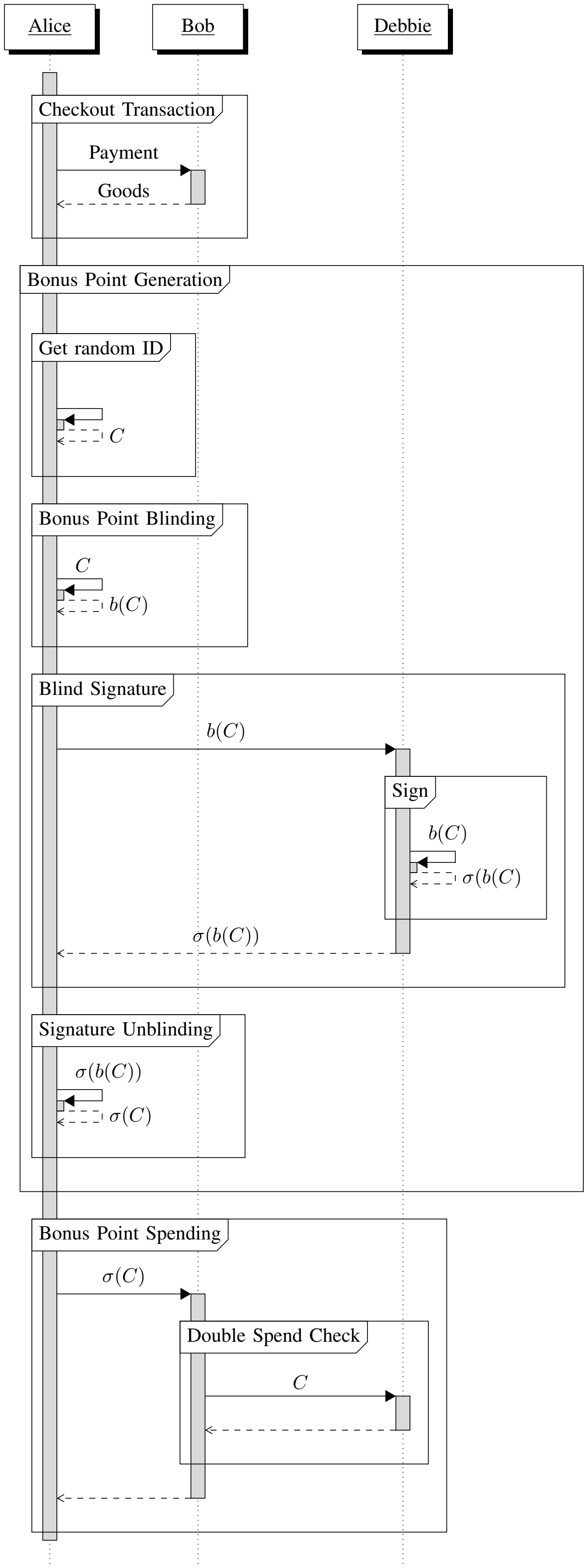

Fig. 1: The life cycle of a bonus point.

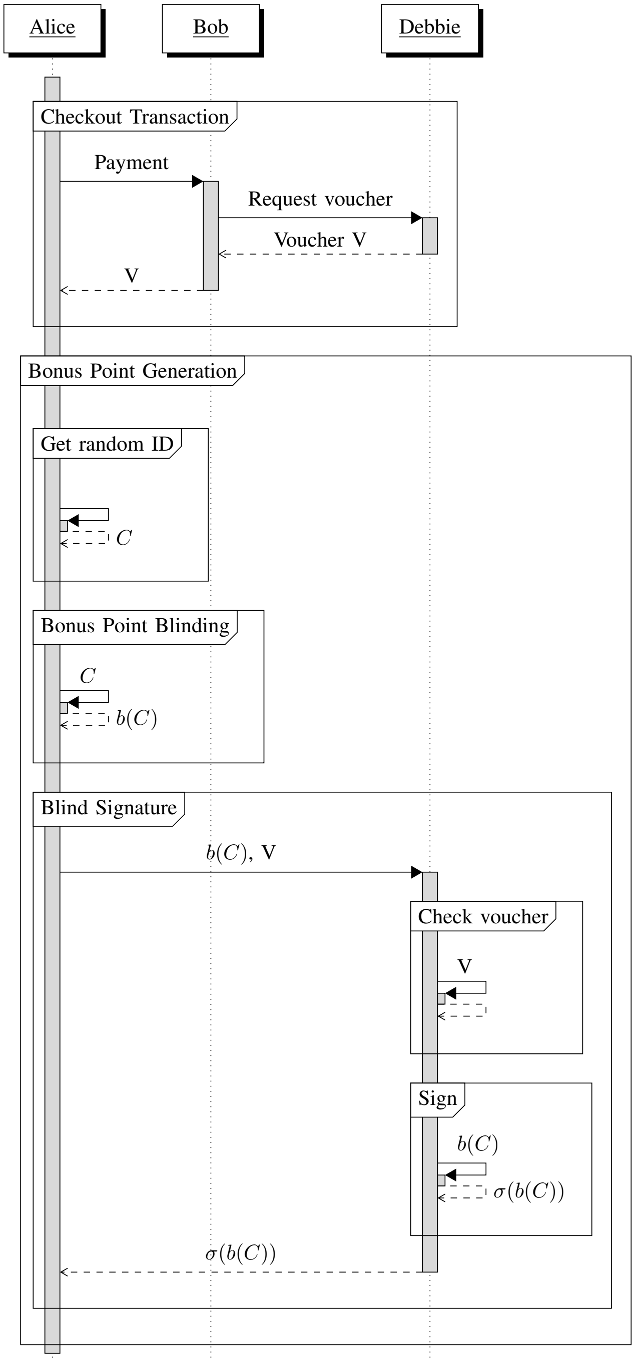

Fig. 2: Bonus points with the extension of merchant vouchers.

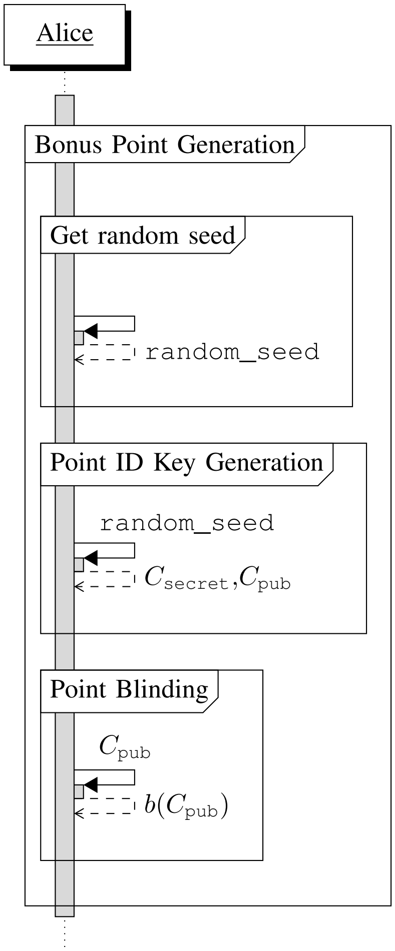

Fig. 3: Bonus point generation with asymmetric cryptographic signatures.

Charlie, in turn, is able to take the role of a customer at another merchant—say, Bob—and use the illegally obtained bonus point to purchase goods himself. When Alice attempts to spend her bonus point after the fraudulent merchant has done so already, the transaction will (rightfully) be declined as double spending. In this scenario, the bonus points are effectively stolen without Alice being able to prove this.

The goal of any process handling this problem is to create an atomic transaction that, if aborted, does not result in any useful knowledge gain at either side. A feasible approach, that we will demonstrate in our proposed protocol, is to use the principle of cryptographic knowledge proofs.

### *A. Proofs of Point Ownership*

Asymmetric cryptographic keys can be used to implement a proof system where customers prove their knowledge of the bonus point secret to Debbie, the central database operator. Proving knowledge of the point secret can be used by legitimate point owners to solve disputes where merchants maliciously reject unspent points, thereafter claiming their value themselves. To implement this fraud prevention, we propose to change the point creation process to introduce the point secret. More specifically, we replace the random bonus point ID by an asymmetric key pair, for example from the ElGamal cryptosystem [18] or the RSA cryptosystem [12]. The bonus point ID consists of the public key component $C_{\texttt{pub}}$ of an asymmetric key pair, as displayed in Figure 3, while Alice keeps the associated secret key $C_{\texttt{secret}}$. Alice generates this key pair when she wants to generate a bonus point. The secret key remains secret at all times and is held by the wallet on Alice's device at least until the spending transaction is cleared by Debbie's database.

When creating a bonus point, Alice no longer blinds and transfers the actual secret ID $C_{\texttt{secret}}$ to Debbie. Instead, the ID subject to blinding is the public key $C_{\texttt{pub}}$. The key pair is generated with sufficiently long keys so that an attacker keen on guessing $C_{\texttt{secret}}$ does not succeed in a reasonable amount of time [19]. We propose to use at least 2048 bit for RSA keys. By that, we also ensure that the probability of a random collision is still low enough, even though not every number is prime and thus can not be the private modulus of an RSA key pair.

After unblinding, Alice holds a signature of the public key $C_{\texttt{pub}}$, forming the bonus point ID $\sigma(C_{\texttt{pub}})$. When spending a bonus point she can reveal the blindly provided signature $\sigma(C_{\texttt{pub}})$ and the public key component $C_{\texttt{pub}}$ of the secret. The merchant Bob then proceeds with verifying the signature's validity and forwards the public key $C_{\texttt{pub}}$ to the central issuer Debbie.

Should Alice's spending be rejected and she suspects fraud committed by Bob, Alice files a claim at Debbie, as displayed in Figure 4. In the next step Debbie authenticates herself and sends a random challenge to Alice and asks her to sign it using the private key $C_{\texttt{secret}}$ associated with the signed public key $C_{\texttt{pub}}$ that was used in the disputed transaction. Alice is supposed to append a random nonce to the challenge before signing it in order to prevent a chosen plaintext attack [20]. Generally, a chosen plaintext attack is an attack where the attacker forces the victim to sign or encrypt a specific message in order to gain knowledge on the structure of the victim's private key. If Alice can provide a valid signature to the challenge, it is safe to assume she is in fact in possession of the secret key and has been cheated on by a merchant.

A particular benefit of our key-pair-based proof protocol is that Alice never reveals her secret bonus point ID. In particular, the secret is not even revealed when a fraud claim is filed. Keeping the point ID secret makes our approach resistant against attacks where the merchant Bob and the issuer Debbie collude to extract Alice's point secrets.

Albeit, detecting which merchant is the one cheating is still an unsolved problem: Debbie relies on Alice's report of where she has unsuccessfully tried to spend the stolen bonus point.

After Alice's proof of point ownership, Debbie can reimburse the customer. Any other participant in the system cannot provide the signed response to the challenge without breaking the underlying crypto-system. Additionally, customers still cannot generate additional bonus point IDs, because valid bonus points still carry Debbie's signature.

### B. Authorization Receipts

While the fraud resolution protocol we discussed in Section IV-A is necessary to protect legitimate customers, it opens up an additional attack vector that we address in the following. Namely, a fraudulent *customer* might invoke the disputation process, claiming that the merchant has tried to commit fraud while, in fact, the customer tries to achieve double spending.

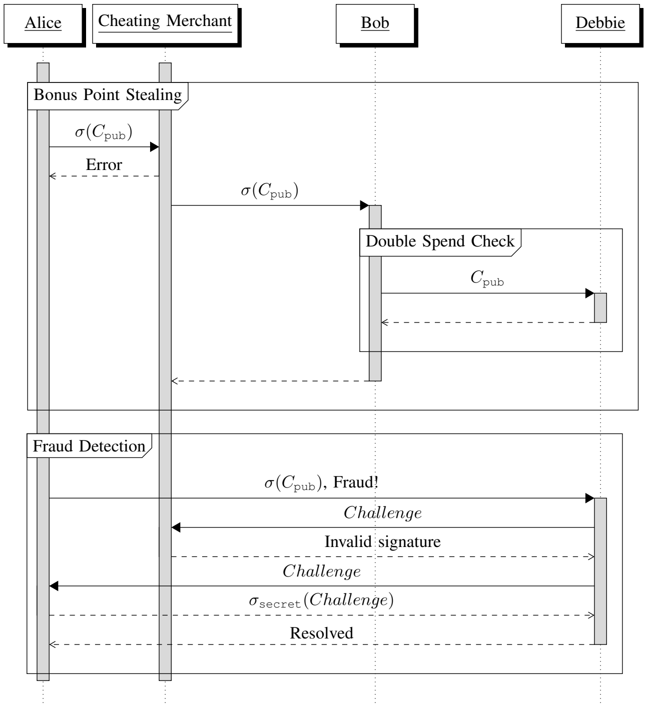

Fig. 4: Fraud and its resolution in the asymmetric signature case.

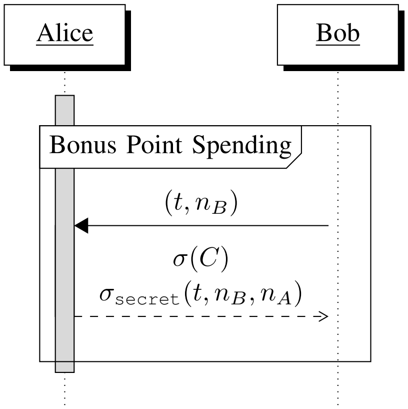

Fig. 5: Payment protocol with authorization receipts.

To illustrate the attack, consider a fraudulent customer Evan, two honest merchants Bob and Charlie, and the central issuer Debbie. Evan purchases a good at Bob's store using an already spent bonus point represented by the signed public key component $C_{\texttt{pub}}$. Bob consequently fails to validate the bonus point with Debbie and denies the transaction. After that, Evan contacts Debbie via the fraud resolution system and claims he has been cheated on by a third, uninvolved merchant Charlie. Evan is able to provide the bonus point's secret key component to Debbie and, thereby, confirms ownership of the bonus point. The information presented to Debbie looks exactly like in the case of actual fraud against Evan by the unsuspecting merchant Charlie. We call this the *reverse customer fraud* attack.

The reverse customer fraud attack relies on an uninformed merchant that can be framed as an unlawful adversary trying to withhold the transaction. To counter the attack, we add an additional receipt to the point spending process, which proves the customer's intent to exchange points for goods. The additional receipt can then be used by merchants to uncover malicious users. Figure 5 shows the necessary changes in the payment protocol.

In addition to receiving the actual bonus point, the merchant poses a challenge with a random nonce, a current time stamp, and the merchant ID. The customer adds an own nonce to the tuple and signs all values using the bonus point's secret key in order to proceed. The merchant's nonce and timestamp ensure freshness and tie the receipt to the current transaction without the customer having to verify individual transaction details. The customer's nonce prevents a chosen-plaintext attack [20] mounted by the merchant. The merchant can verify the client's signature using the bonus point's public key component. This signed challenge is called an *authorization receipt* and can be stored and later used by the merchant to prove that the customer had knowledge of the secret key and the intent to pay at the merchant's shop. Note that with these receipts it is no longer necessary for the issuer to rely on the customer when identifying cheating merchants.

## V. Evaluation

To evaluate our protocol, we define privacy and security requirements, and we demonstrate how our protocol fulfills these requirements. First, we identify incentives and capabilities for each group of participants in our system, focusing on requirements for user privacy and system integrity. We then discuss how each requirement is fulfilled by the proposed algorithms.

### A. System Participants

To define our adversary model, we identify several scenarios where different parties can conduct fraud on other participants. Each participant has different interests in the system, and we assume that each might use their capabilities to gain an unfair advantage.

- *Customers* may seek to gain bonus points without having paid for goods in an eligible transaction. They also gain an advantage if they can mount double spending attacks using their bonus points. Additionally, customers might try to abuse the fraud report infrastructure by trying to frame innocent merchants.
- *Merchants* may try to decline handing out bonus points, because the issuer bills them for every bonus point generated at a transaction from their store. Such misbehavior can only be penalized directly by the issuer if customers complain about that merchant. In addition, merchants may try to decline valid bonus points to gain a financial advantage. Further to that, merchants may themselves try to redeem bonus points they obtained but denied from a customer.
- The *issuer* has an interest that every party in the system is forced to comply with the protocol's rules. The bonus point system is offered as a service to merchants, and if either customers or merchants perceive that another party easily gains an unfair advantage, they no longer have an interest in participating. Maintaining trust between all parties is thus the most important target of the issuer. Consequently, a cheating issuer acts against its own interests, because merchants and customers would no longer employ the issuer as a service provider should they detect such fraud.

We discuss our protocols properties in terms of customer and merchant fraud and an attacker model for the issuer in the following.

### B. Customers

Customers are effectively not able to commit double spending, because merchants do not accept a bonus point if the issuer's database indicates prior usage of that point. Gaining illicit bonus points without paying in an eligible transaction is excluded by our newly introduced system of vouchers. Vouchers ensure that for every blind signature there is a merchant who accounts for its value. Counterfeiting a valid voucher without possessing it from an eligible transaction is practically impossible, as the serial number is very long. By choosing a length of at least 128 bit a guessing attacker has to try an infeasible amount of options before generating a valid voucher by chance. The last remaining attack vector for customers is the abuse of our fraud resolution system. The question whether a customer intended to pay at the merchant can be decided by using the proof of point ownership in conjunction with authorization receipts, as discussed in Section IV. This combination prevents customers from creating unbacked bonus points and from gaining advantage by wrongly reporting unsuspect merchants as fraudulent. Thus, customers are no longer able to gain an unfair advantage.

A customer's privacy is not threatened in any constellation in our system. The underlying system of electronic cash guarantees privacy when regarding all transactions present in that protocol. At first glance, it may seem that the introduced merchant vouchers reveal to the issuer the merchant that a customer has conducted business with. However, if communication between customer and issuer is anonymous itself (e. g., by using a proxy server), the issuer does not learn anything about the customer's identity. The asymmetric-key-ID extension of the original protocol does not leak any more private information than the random IDs from the original approach if the assumption of hard reversibility of key generation [21] holds. Like in the original protocol, the asymmetric key pair for the bonus point has to be generated using a truly random source and not contain identifying information like a MAC address or other system-specific parameters. When the claim resolution process is activated, information about the inner structure of the private key still does not leak. In this case, only the public key, which is itself derived from the secret key, is leaked. Authorization receipts also do not convey any usable information on the customer's identity, as the signing key is different for each bonus point and not linked to a certain customer. The same effect can be observed in case of the challenge-response protocol during fraud resolution. Thus, no information is released that could endanger the user's privacy.

### C. Merchants

The main attack vector in the naïve protocol implementation is for a merchant to steal a bonus point when a customer tries to pay with that point. After gaining knowledge of the secret bonus point ID, the merchant denies the fulfillment transaction and uses the point itself. Our proposed extension of asymmetric key pairs serves as a method to prove being the original generator of a bonus point without revealing the secret itself. With that, fraud remains impossible as long as the secret key for a specific bonus point is not leaked, because of the challenge posed by the central issuer that has to be freshly signed by the suspected creator of the bonus point. However, the merchant trying to commit said fraud can only be identified if he tries to spend the gained bonus point in a later transaction. Otherwise, a fraudulent merchant can only be identified using complaint reports of customers to the system operator.

Combining asymmetric keys as IDs and authorization receipts leads to a system where neither merchants nor customers are able to claim being cheated on due to the ability of the other side to prove their intent and knowledge at the time of the transaction.

### D. Attacker Model for the Issuer

On one hand, the issuer occupies a central role in the bonus point system and might seek opportunities to cheat on customers and on merchants. On the other side, the issuer's business model is based on the successful operation of the anonymous and trusted bonus point system. In contrast, one might argue that merchants use the bonus point system as secondary aid to increase revenue in their primary business and thus might have the objective to unfairly maximize the effectiveness of this tool. If a merchant is caught and subsequently excluded from the system, they might already have gained more advantages than they would have by remaining in the system for a longer term.

The issuer however performs its role exclusively in the bonus system and gains income only if the system itself performs well and without complaints by either customers or merchants. Thus, the stakes of losing customers' or merchants' trust are high and substantiated reports of fraud have the possibility of killing the issuer's entire business model instantly. The customers' main assets that have to be protected are their privacy and the integrity of their earned points. In contrast, the merchants' goals are usually an uninterrupted operation of the system and the prevention of fraud through customers against them. Both parties put significant trust into the issuer while constantly observing its operations in order to check if the system still performs towards their goals.

Therefore, we assume an honest-but-curious attacker model for the issuer because we are convinced that a malicious issuer would not be operating for long. As established above, merchants and customers can stop participating as soon as they are convinced the operator is no longer acting according to their interests. Using this model, we can exclude attacks where the issuer issues bonus points without receiving a valid voucher. The protocol requires that the vouchers are always checked and every blindly signed bonus point can be accounted for. On the other end of the protocol the issuer could deny a valid deposit of a valid bonus point spent by an honest customer at an honest merchant. Again, the chosen model prevents this attack as the issuer is assumed to be acting according to the protocol. Subsequently, the issuer can gain no advantage in the system without damaging its own source of revenue.

### E. Performance Considerations

Next, we discuss the effects of our proposals when considering performance. For that, we term $n$ the number of bonus points in a transaction.

Our proposed bonus point payment protocol does not remove any of the original functionality found in Chaum's electronic cash system. Vouchers are used both for accounting purposes in the system and for preventing customers to generate more bonus points than they are entitled to. They bear unique serial numbers, but these are of fixed length and can be managed centrally by the issuer. Therefore, only a constant message overhead is necessary to transmit them from merchants to customers when committing a bonus-point-eligible transaction. The transmission of vouchers from customers to the issuer during bonus point generation adds a constant overhead to every blinded bonus point, too. A typical voucher serial number is 16 bytes long, thus introducing 32 bytes of overall overhead when considering both transmissions from the merchant to the customer and from the customer to the issuer. The number of messages is either unaltered if the vouchers are generated by the merchants themselves or one message more per point than in the original approach if merchants request them on the fly from the issuer. Thus, the number of messages with or without vouchers is in $\mathcal{O}(n)$ with message size $\mathcal{O}(1)$.

Point generation has no effects on the performance of the protocol. While Chaum proposes to use a truly random number, our proposal of applying a public key generation function to produce the key pair with secret $C_{\texttt{secret}}$ also produces a seemingly random numerical public key $C_{\texttt{pub}}$. That number is treated exactly the same way as the random bonus point ID has been used in the original version of the protocol. Thus, while message size is slightly increased (in case of RSA we observed 2048 bit instead of 128 bit per message), it is still a constant overhead per bonus point.

Spending a bonus point is slightly altered in our concepts. The transmission of a bonus point from a customer to a merchant contains an ID $C_{\texttt{pub}}$ and its corresponding signature $\sigma(C_{\texttt{pub}})$, equivalent to $C$ and $\sigma(C)$ in the first approach. So, message size is still in $\mathcal{O}(1)$. When a bonus point ID is entered into the issuer's central database it does again not matter if the reported ID is a public key derived from the actual random ID or the ID itself. The issuer has to store these numbers indefinitely in both cases and merchants can check for their validity when accepting bonus point payments, so that database is growing within $\mathcal{O}(n)$.

However, small differences can be noticed in case of authorization receipts. At the time of transaction every bonus point has to be accompanied by an authorization receipt, i. e., one more message exchange per point between merchant and customer for generating the receipt. This leads to a number of exchanged messages in $\mathcal{O}(n)$, i. e., a constant message overhead per bonus point. When processing a transaction using authorization receipts, every merchant has to store such a receipt for any point involved, thus holding a database with linear growth.

## VI. Conclusion

In contrast to traditional bonus point systems, that generally provide no privacy at all, we propose a completely decentralized and anonymous scheme where users actually generate bonus points themselves. This enables users to remain entirely anonymous while participating in a customer loyalty program.

Additionally, we identified a weakness in the concept of the electronic cash system, where a user could cheat. This is fixed by altering the bonus point generation algorithm so that it is possible to prove committed intent of transaction on one side and identity of point generation on the other side. By that we eliminate a specific fraud scenario and thus incentivize bonus system operators to switch to such a system.

## Acknowledgement

We would like to express our gratitude to the BMBF (German Federal Ministry of Education and Research), who funded this research in the context of the *Goodcoin* project.

## References

[1] PAYBACK GmbH, "About Payback," last accessed 21st April 2017. [Online]. Available: https://www.payback.net/en/about-payback/

[2] C. C. Aggarwal and S. Y. Philip, "A General Survey of Privacy-Preserving Data Mining Models and Algorithms," in *Privacy-preserving data mining*. Springer, 2008, pp. 11–52.

[3] C. Clifton and D. Marks, "Security and Privacy Implications of Data Mining," in *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, 1996, pp. 15–19.

[12] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[4] R. T. Rust, P. Kannan, and N. Peng, "The customer economics of internet privacy," *Journal of the Academy of Marketing Science*, vol. 30, no. 4, pp. 455–464, 2002.

[5] D. Chaum, "Blind Signatures for Untraceable Payments," in *Advances in cryptology*. Springer, 1983, pp. 199–203.

[6] S. Bowie, "Computer program and system for credit card companies for recording and processing bonus credits issued to card users," 2001, US Patent 6,195,644.

[7] T. Straub and A. Heinemann, "An Anonymous Bonus Point System For Mobile Commerce Based On Word-Of-Mouth Recommendation," in *Proceedings of the 2004 ACM symposium on Applied computing*. ACM, 2004, pp. 766–773.

[8] European Central Bank, *Virtual Currency Schemes*, Frankfurt-on-Main, 2012.

[9] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[10] G. Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," *Ethereum Project Yellow Paper*, vol. 151, 2014.

[11] D. Chaum, A. Fiat, and M. Naor, "Untraceable Electronic Cash," *Proceedings on Advances in cryptology*, pp. 319–327, 1990.

[13] V. R. Shen, Y. F. Chung, T. S. Chen, Y. A. Lin, and others, "A Blind Signature Based on Discrete Logarithm Problem," *International Journal of Innovative Computing Information and Control*, vol. 7, no. 9, pp. 5403–5416, 2011.

[14] E. Mohammed, A.-E. Emarah, and K. El-Shennaway, "A Blind Signature Scheme Based on ElGamal Signature," in *Radio Science Conference, 2000. 17th NRSC'2000. Seventeenth National*. IEEE, 2000, pp. C25–1.

[15] S. Brands, "Untraceable Off-line Cash in Wallet with Observers," in *Advances in Cryptology — CRYPTO' 93: 13th Annual International Cryptology Conference Santa Barbara, California, USA August 22–26, 1993 Proceedings*, D. R. Stinson, Ed. Springer Berlin Heidelberg, 1994, pp. 302–318.

[16] A. Boldyreva, "Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme," in *International Workshop on Public Key Cryptography*. Springer, 2003, pp. 31–46.

[17] P. Jesus, C. Baquero, and P. S. Almeida, "ID Generation in Mobile Environments," in *CSMU 2006: Proceedings of the Conference on Mobile and Ubiquitous Systems*, 6 2006.

[18] T. Elgamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, Jul 1985.

[19] A. K. Lenstra and E. R. Verheul, "Selecting Cryptographic Key Sizes," *Journal of Cryptology*, vol. 14, no. 4, pp. 255–293, Sept. 2001. [Online]. Available: http://link.springer.com/10.1007/s00145-001-0009-4

[20] Y. Desmedt and A. M. Odlyzko, "A Chosen Text Attack on the RSA Cryptosystem and Some Discrete Logarithm Schemes," in *Conference on the Theory and Application of Cryptographic Techniques*. Springer, 1985, pp. 516–522.

[21] C. Percival and S. Josefsson, "The scrypt Password-Based Key Derivation Function," RFC 7914 (Informational), RFC Editor, Fremont, CA, USA, pp. 1–16, Aug. 2016.