P4R: Privacy-Preserving Pre-Payments with Refunds for Transportation Systems

Andy Rupp[1], Gesine Hinterwälder[2], Foteini Baldimtsi[3], and Christof Paar[4]

[1] Karlsruhe Institute of Technology, Germany <andy.rupp@rub.de>
[2] University of Massachusetts Amherst, USA <hinterwalder@ecs.umass.edu>
[3] Brown University, USA <foteini@cs.brown.edu>
[4] Ruhr-University Bochum, Germany <christof.paar@rub.de>

Abstract. We propose a new lightweight payment scheme for transit systems called P4R: Privacy-Preserving Pre-Payments with Refunds. In P4R a user deposits money to obtain a bundle of credentials, where each credential allows to make an arbitrary ride. The actual fare of a trip is determined on-the-fly when exiting. Overpayments are refunded where all trip refunds of a user are aggregated in a single token thereby saving memory and increasing privacy. We build on Brands' e-cash scheme to realize the pre-payment system and a new variant of blind Boneh-Lynn-Shacham signatures to implement the refund capabilities. Our construction is secure against malicious users and guarantees user privacy. We also provide an efficient implementation that shows the suitability of our scheme as future transit payment system.

Keywords: E-cash, refunds, lightweight payments, transit systems.

1 Introduction

Deploying electronic payment systems in transportation as opposed to sticking with traditional systems (like cash or ticket systems) offers important benefits like significantly reduced revenue collection costs, enhanced customer satisfaction as well as improved services such as dynamic pricing. Hence, electronic payment systems for transportation (EPST) already are and will become an even more important component of the critical infrastructure "transportation".

Currently deployed systems like the MBTA "Charlie Card" [14] or the E-ZPass [9] show the potential of electronic payment systems as a reasonable, fair, and efficient method for revenue collection. However, at the same time they are examples demonstrating the serious shortcomings of today's EPST since they lack sufficient mechanisms protecting their security and especially the privacy of their users: One problem that EPST seem to share with many other commercial systems implementing security functions is the deployment of cryptographically weak proprietary primitives as, e.g., demonstrated for the Charlie Card or Oyster Card [13]. Besides security issues, frequently concerns about the location privacy of EPST users are raised, i.e., the un-/traceability of users within the transportation system. For instance, E-ZPass and Fast Lane toll plaza records have been used by lawyers to prove that their client's cheating spouses were not where they pretended to be at a certain date and time [11] which shows that these systems do not respect locational privacy at all. However, in order to enable a large-scale deployment and broad acceptance of EPST, adequate security and privacy mechanisms are essential.

While currently deployed EPST suffer from serious privacy and security flaws there is a wealth of cryptographic payment schemes (Section 1.1) offering strong security and privacy properties. However, the unique requirements of the transportation domain, especially engineering constraints and functional requirements, prevent the use of well-established crypto like e-cash schemes.

In this paper we restrict to the consideration of a *transit* payment scenario such as payment systems for subways. Here payment devices can be fairly low-cost platforms such as RFID transponders, contactless or hybrid smart cards which are provided by the Transportation Authority (TA). Given such a device, a user can charge it at a vending machine to pay for rides in the subway system. The entry and exit points are typically physically secured by turnstiles that are equipped with readers responsible for calculating fares and conducting payment transactions with user devices. To avoid congestion in front of turnstiles, transactions have to be fast: A payment transaction should be executable within a few hundred milliseconds. Transactions at the vending machine are less time-critical but should also not take longer than several seconds.

The resource constraints of the user devices together with the realtime requirements are one of the main obstacles preventing the application of e-cash schemes. User devices are typically equipped with only a few tens of kilobytes of memory and an 8 or 16-bit microcontroller running at not more than 16 MHz. This situation greatly limits the performance achievable for a crypto primitive. Currently, on widely-used microcontrollers, operated at 16 MHz, a modular exponentiation in 1024-bit RSA with full-length exponents requires about 5 s while a point multiplication on a 160-bit elliptic curve group takes around 400 ms [10].

So clearly, it is prohibitive to do much more than a single full public-key operation on such a CPU during a payment transaction. However, almost all e-cash schemes make excessive use of exponentiations or ECC point multiplications in the spending protocol. Fortunately, by employing an ECC coprocessor as accelerator, the runtime of a point multiplication can be improved by roughly one order of magnitude, (e.g., a factor 12 for the coprocessor in [16]). Yet, due to power constraints we may only assume the usage of such a coprocessor when the payment device is in contact mode, e.g., when interacting with the vending machine, which fits our transit scenario.

1.1 Related Work

E-Cash. An e-cash scheme typically consists of a bank, users, merchants and the following protocols: (1) a withdrawal protocol where a user obtains e-coins from the bank; (2) a spending protocol where the user sends coins to a merchant; (3) a deposit protocol where a merchant deposits coins obtained from a user to his bank account; and (4) other protocols for identifying malicious behavior.

In his seminal paper [7] Chaum introduced anonymous electronic cash that allows anonymous and unlinkable payments, while at the same time it ensures unforgeability of e-coins. Since then, e-cash protocols have been extensively studied. To name only a few important results: Brands [5] proposed one of the first and most famous offline anonymous e-cash schemes with the most efficient protocol when it comes to spending an e-coin. However, Brands' coins occupy a fairly large amount of memory and the withdrawal protocol is relatively expensive. To solve the memory issue, Camenisch et al. [6] proposed so-called compact e-cash but at the cost of a far less efficient spending protocol.

Payment Schemes Tailored to Transportation. Recently, EPST has started to attract the attention of the crypto community. Heydt-Benjamin et al. [12] were the first to propose an informal cryptographic framework for transit payment systems. Sadeghi et al. [19] present an RFID-based e-ticket scheme which does not expect the user's payment device to carry out too much expensive computations, but the existence of external trusted devices is assumed for the costly operations and their system only protects a user's privacy with respect to outsiders and not the transportation authority. Blass et al. [4] proposed another offline "privacy-preserving" payment system for transit applications that solely relies on a 128-bit hash function and lots of precomputed data on the backend's side. Again, a user's privacy in their system is not protected from the TA.

Popa et al. [17], Balasch et al. [1], Meiklejohn et al. [15] and Day et al. [8] proposed privacy-preserving payment systems for electronic toll collection. In their schemes, user devices are battery-powered on-board units that collect GPS location data and report this data or fare information (computed thereof) to the toll collection provider. However, these were developed for a scenario where users subscribe for a service and pay by the end of a billing period. In the transit scenario, we cannot assume that each user has access to a trusted PC to settle accounts: an untrusted vending machine is more realistic.

1.2 Our Approach and Contribution

Due to their strong security and privacy guarantees, it would be highly desirable to build a transit payment system based on e-cash. A good candidate for this purpose is Brands' scheme because of its exceptionally efficient spending protocol.[1] On the downside, Brands' coins are large and their withdrawal is expensive. Hence, it would be beneficial to limit the amount of coins that a user has to spend to pay his fare, having to spend only a single coin per trip would be ideal. However, this conflicts with the necessity of allowing flexible and dynamic prices, i.e., fares should not be flat but arbitrary monetary amounts: Setting the denomination of a coin to be 1 cent certainly allows for flexible pricing but users would need plenty of them to pay for a trip. Setting the face value to $2 reduces the number of required coins per trip but severely restricts the system of fares. To do a tradeoff and have coins for different monetary values, one would

[1] Recent results show that Brands protocol cannot be proven secure using the currently known techniques [3], however it has not been shown that the scheme is insecure.

need to deal with overpayments and change in a privacy-preserving way. This is especially difficult in EPST where bank and merchants are the same entity.

Our proposal of the transit payment system P4R addresses the issues discussed above. The idea is to "let a single coin be worth exactly the (variable) cost of an arbitrary trip in the system". More precisely, our payment system is not a typical e-cash scheme but based on the concept of pre-payments with refunds: a user has to make a deposit to get a coin worth an arbitrary ride and gets a refund if the actual fare is less than his deposit. We build our payment system using Brands' scheme where we minimize the number of coins a user needs to pay for his rides. To be precise, our approach is not limited to Brands' but also works for other schemes that can be modified in a way that coins can be shown twice without revealing the ID of a user (e.g., [2]). The refund system is realized using a new variant of blind BLS signatures and allows to aggregate refunds.

As for security, we can show that it is infeasible for malicious users to abuse the system. This includes users who try to dodge the fare or redeem higher refunds than issued. Such users will be identified by double-spending checks. Vice versa, we can show that the transportation authority cannot distinguish between trips of honest users with the same aggregated refund amounts and it cannot link rides of the same user thus providing user privacy.

Furthermore, we implemented P4R for 160-bit elliptic curves on the Moo RFID tag [20] housing a 16-bit MSP430 microcontroller. The results show that the scheme is fairly efficient even on this, not for our purposes optimized, device. Assuming a clock rate of 16 MHz for a fielded version of the device, the computations required for spending can be executed in 2 ms, getting a refund takes 340 ms, and redeeming the refund token 350 ms. Withdrawal (5.29 s) is also not far from meeting real-world requirements and could strongly be improved by making use of dedicated hardware (e.g., an ECC coprocessor in contact mode).

Due to the page limit we refer to the full version [18] for a formal description of P4R, proofs of security and privacy, a detailed discussion of the implementation, as well as interesting tradeoffs between security, privacy, and efficiency.

2 A Privacy-Preserving Transit Payment System

High-Level Description. P4R is composed of three subsystems: Trip Authorization Token (TAT), Refund Calculation Token (RCT), and Refund Token (RT) system. The TAT system is offline. Here vending machines play the role of the "bank" issuing TATs and (offline) readers at the entry turnstiles play the role of a "merchant" where tokens can be spent. The RT system is online. Here roles are reversed compared to the TAT system, i.e., readers at the exit turnstiles issue refunds and the (online) vending machines redeem these tokens.

A TAT (aka ticket) is a credential that authorizes a user to make exactly one trip. A user initially makes a deposit to obtain a number of TATs where the cost of a TAT equals the price of the most expensive trip in the system. Of course, to reduce this deposit it is also possible to have different types of TATs for different zones of the transportation system. The withdrawal of a TAT is done in a blind fashion such that a later use cannot be linked. The ID of a user is encoded in each TAT to prevent a repeated use for entering the system (double-spending detection). At the beginning of a ride a user presents an unused TAT to the reader at the entry turnstile. If it is valid and the user can show (using a zero-knowledge proof) that he knows the ID encoded in the TAT, access is granted. When leaving the system the actual fare is determined at the exit turnstile. This is done as follows: on entering, the user also received an RCT (aka stamped ticket), which contains a MAC on the TAT, the date and time, as well as on the ID of the reader. When he leaves he presents this RCT to the exit turnstile which calculates the trip cost based on this information. He also provides a blinded version of his RT (blank RT tokens are available from the vending machines) to which the reader adds the difference between the deposit for the TAT and the actual fare. To prevent re-using an RCT and thus claiming the same refund several times, the idea is to bind an RCT to the TAT which has been used on entering, and force a user to again prove the knowledge of the ID encoded into this TAT when he leaves. An RT (aka piggy bank) is used to add up several refunds instead of having a separate RT per refund. When a user decides to cash the collected refund he presents his RT to the vending machine which redeems the RT if it is not already marked as cashed in the central DB.

Some Technical Details. The TAT and RCT subsystems are loosely coupled and realized based on an extension of Brands' protocol that allows to show a TAT twice without revealing the user's identity. To setup the TAT system, the TA chooses a cyclic group $\mathbb{G}$ of prime order $q = \Theta(2^{k_1})$, group generators $g, g_1, g_2 \in \mathbb{G}$, a random number $x \in \mathbb{Z}_q^*$, and a collision-resistant hash function $H : \mathbb{G}^5 \to \mathbb{Z}_q$. The public key is $pk^{\mathsf{TAT}} = (\mathbb{G}, g, g_1, g_2, g^x, g_1^x, g_2^x, H)$ and the secret key is $sk^{\mathsf{TAT}} = x$. The user's secret ID is some number $id_{\mathcal{U}} \in \mathbb{Z}_q$. This corresponds to Brands' setting. A TAT is a tuple $\mathsf{TAT}_i = (A_i, B_i, C_i, \mathsf{sig}_x(A_i, B_i, C_i))$, where $A_i = g_1^{id_{\mathcal{U}} s_i} g_2^{s_i}$, $B_i = g_1^{x_i} g_2^{y_i}$, $C_i = g_1^{x'_i} g_2^{y'_i}$, and $\mathsf{sig}_x(A_i, B_i, C_i)$ is a Chaum-Pedersen signature. A_i encodes the user's ID and a random serial number. B_i and C_i (where C_i does not exist in Brands' original system) are commitments to random values which are later used as blinding factors for proving the ownership of a TAT (i.e., knowledge of $id_{\mathcal{U}}$ and s_i) when entering and leaving the system. If a user tries to show a TAT twice to enter or leave and is thus forced to re-use x_i, y_i or x'_i, y'_i, respectively, $id_{\mathcal{U}}$ can easily be revealed. Note that since a TAT is withdrawn in a blind fashion, only the user knows the values comprising TAT_i.

To setup the RCT system, a random k_2-bit key K for a MAC scheme is chosen. An RCT has the form $\mathsf{RCT}_i = (\mathsf{TAT}_i, \mathrm{ts}, id_{\mathcal{R}}, \mathsf{MAC}_K(\mathsf{TAT}_i, \mathrm{ts}, id_{\mathcal{R}}))$, where ts is the current timestamp and $id_{\mathcal{R}}$ is the ID of the reader at the entry.

The RT system is based on a new variant of blind BLS signatures. For setting up this system, the TA chooses cyclic groups G, G_T of prime order $p = \Theta(2^{k_3})$ with an efficient, non-degenerated pairing $e : G \times G \to G_T$, a generator $h \in G$, and an exponent $d \in \mathbb{Z}_p$. Let W be the set of all possible single-trip refunds. Then, the public key is $pk^{\mathsf{RT}} = (G, G_T, e, h, (h^{d^w})_{w \in W})$ and the secret key is $sk^{\mathsf{RT}} = d$. Note that assuming users do not verify RTs (which is similar to not

$\mathcal{U}$		$\mathcal{V}$ (online)
	(1) User buys TATs and receives fresh RT at vending machine	
	$\xleftrightarrow{\text{PoK of } id_{\mathcal{U}}}$	
$\mathsf{TAT}_i = (A_i, B_i, C_i, \mathsf{sig}_x(A_i, B_i, C_i))$ where $A_i = g_1^{id_{\mathcal{U}} s_i} g_2^{s_i}$, $B_i = g_1^{x_i} g_2^{y_i}$ $C_i = g_1^{x_i'} g_2^{y_i'}$	$\xleftrightarrow{\text{blind sigs on } (A_i, B_i, C_i)}$	
$\mathsf{RT} = \mathsf{SN}_{\mathsf{RT}}, R = 1, v = 0$	$\xleftarrow{\mathsf{SN}_{\mathsf{RT}}}$	$\mathsf{SN}_{\mathsf{RT}} \overset{\$}{\leftarrow} G$, add $\mathsf{SN}_{\mathsf{RT}}$ to DB
	(4) User redeems RT at vending machine	
$r \overset{\$}{\leftarrow} \mathbb{Z}_p$, $\mathsf{RT}' = \mathsf{RT}^r$		
$R = Rr \bmod p$	$\xrightarrow{\mathsf{SN}_{\mathsf{RT}}, \mathsf{RT}', v, R}$	check validity of $\mathsf{SN}_{\mathsf{RT}}$ $v \overset{?}{<} p - 1$ $e(\mathsf{SN}_{\mathsf{RT}}^R, h^{d^v}) \overset{?}{=} e(\mathsf{RT}', h)$ mark $\mathsf{SN}_{\mathsf{RT}}$ as redeemed in DB

$\mathcal{U}$		$\mathcal{R}$ (offline)
	(2) User shows TAT and receives RCT when entering system	
	$\xrightarrow{\mathsf{TAT}_i}$	check validity of TAT_i
$z_1 = x_i + c(id_{\mathcal{U}} s_i) \bmod q$	$\xleftarrow{c}$	$c \overset{\$}{\leftarrow} \mathbb{Z}_q$
$z_2 = y_i + cs_i \bmod q$	$\xrightarrow{z_1, z_2}$	$g_1^{z_1} g_2^{z_2} \overset{?}{=} A_i^c B_i$ add $(\mathsf{TAT}_i, z_1, z_2, c)$ to DB
	$\xleftarrow{\mathsf{RCT}_i}$	$\mathsf{RCT}_i = (\mathsf{TAT}_i, \mathrm{ts}, id_{\mathcal{R}},$ $\mathsf{MAC}_K(\mathsf{TAT}_i, \mathrm{ts}, id_{\mathcal{R}}))$
	(3) User shows RCT and collects refund on his RT when leaving system	
	$\xrightarrow{\mathsf{RCT}_i}$	check validity of $\mathsf{RCT}_i, \mathsf{TAT}_i$
$z_1' = x_i' + c'(id_{\mathcal{U}} s_i) \bmod q$	$\xleftarrow{c'}$	$c' \overset{\$}{\leftarrow} \mathbb{Z}_q$
$z_2' = y_i' + c' s_i \bmod q$	$\xrightarrow{z_1', z_2'}$	$g_1^{z_1'} g_2^{z_2'} \overset{?}{=} A_i^c C_i$ add $(\mathsf{TAT}_i, z_1', z_2', c')$ to DB
$r \overset{\$}{\leftarrow} \mathbb{Z}_p^*$	$\xleftarrow{w}$	determine refund $w \in \mathbb{Z}_{p-1}$
$\mathsf{RT}' = \mathsf{RT}^r$	$\xrightarrow{\mathsf{RT}'}$	
$v = v + w$, $R = Rr \bmod p$, $\mathsf{RT} = \mathsf{RT}''$	$\xleftarrow{\mathsf{RT}''}$	$\mathsf{RT}'' = \mathsf{RT}'^{d^w}$

Fig. 1. Main P4R protocols executed by users $\mathcal{U}$, vending machines $\mathcal{V}$, and readers $\mathcal{R}$

counting change for small amounts) pk^{RT} does not need to be stored on user devices. An RT holding the refund value $v \in \mathbb{Z}_{p-1}$ corresponds to the multi-signature $\mathsf{RT} = \mathsf{SN}_{\mathsf{RT}}^{d^v}$, where $\mathsf{SN}_{\mathsf{RT}}$ is a random serial number chosen by the TA. Clearly, a refund $\mathsf{RT} = \mathsf{SN}_{\mathsf{RT}}^{d^v}$ can be increased to $v + w$ by raising it to d^w. To blind an RT before collecting a refund RT is raised to some random $r \in \mathbb{Z}_p^*$. There is actually no need to remove the old blinding r before blinding RT again using r' to collect another refund, so blinding factors can be aggregated as $R = rr'$.

More details on P4R are given in Figure 1 showing the main interactions between users $\mathcal{U}$, vending machines $\mathcal{V}$, and readers $\mathcal{R}$.

3 Security and Privacy

TA Security. P4R guarantees that the TA does not lose any money: users are not able to receive reimbursements which exceed the overall deposit for TATs minus the overall fare of their trips without being identified by the TA. In order for a user to cheat the TAT subsystem he would need to either forge a TAT, reuse it or use a foreign (eavesdropped) one. Fortunately, all these possibilities are ruled out by the (assumed) security of Brands' scheme: unforgeability, double-spending security (restrictiveness), and soundness. A user could also try to cheat the RCT subsystem by creating RCTs himself, use the same RCT twice or use a "foreign" RCT not corresponding to his current trip. Security against those attacks can be reduced to MAC unforgeability and Brands' restrictiveness and soundness under the assumption that we exclude physical attacks (i.e., users jumping over turnstiles) and users adhere to the protocol schedule. Finally, a user cannot cheat the RCT system: RTs cannot be redeemed twice due to online checks. Also, a user cannot forge RTs or claim higher values on his RT since this would require breaking the $\sum$-Incremental DH-Assumption [18].

User Security. Individual users are protected in the sense that a passive adversary is not able to steal tickets or refunds from a user. In particular, an adversary neither can use "foreign" TATs or RCTs since he would have to prove knowledge of the encoded $id_{\mathcal{U}}$ and s_i nor he can redeem an eavesdropped RT since he would need to know the aggregated blinding factor R. In the full version [18] we also consider security against more powerful adversaries.

User Privacy. A transportation system should provide *location privacy* for its users, i.e., it should not be possible to trace the movements of individual users within the system. Unfortunately, in P4R redeeming refunds leaks some information on the sequence of trips a user did. Luckily, we can show that this sequence is still hidden within the set of all possible trip sequences leading to the same total refund amount and argue that theoretically this set should be pretty large (equals the number of integer partitions of the refund value). However, one should also be aware of the limits of this approach: In practice, we cannot guarantee that during a certain period of time many such sequences actually appear in the records of the TA. For instance, it could happen that since the issue date of a RT nobody but the owner of this token did trips resulting in a particular refund amount (though in theory many sequences lead to this amount). Hence, the exact level of location privacy provided by P4R depends on the characteristics of the transportation system and user behavior. Nevertheless, we believe that for real-world transportation systems these limits are no real issues.

Acknowledgments. We would like to thank Marc Fischlin for his valuable input to the security proofs and the anonymous reviewers for helpful comments.

References

1. Balasch, J., Rial, A., Troncoso, C., Preneel, B., Verbauwhede, I., Geuens, C.: PrETP: Privacy-preserving electronic toll pricing. In: USENIX Security Sympo-

sium. pp. 63–78. USENIX Association (2010)

2. Baldimtsi, F., Lysyanskaya, A.: Anonymous credentials light. IACR Cryptology ePrint Archive 2012, 298 (2012)
3. Baldimtsi, F., Lysyanskaya, A.: On the security of one-witness blind signature schemes. IACR Cryptology ePrint Archive 2012, 197 (2012)
4. Blass, E.O., Kurmus, A., Molva, R., Strufe, T.: PSP: private and secure payment with rfid. In: Al-Shaer, E., Paraboschi, S. (eds.) WPES. pp. 51–60. ACM (2009)
5. Brands, S.: An efficient off-line electronic cash system based on the representation problem. Tech. Rep. CS-R9323, CWI (1993)
6. Camenisch, J., Hohenberger, S., Lysyanskaya, A.: Compact e-cash. In: Cramer, R. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 3494, pp. 302–321. Springer (2005)
7. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) CRYPTO. pp. 199–203. Plenum Press, New York (1982)
8. Day, J., Huang, Y., Knapp, E., Goldberg, I.: SPEcTRe: spot-checked private ecash tolling at roadside. In: Chen, Y., Vaidya, J. (eds.) WPES. pp. 61–68. ACM (2011)
9. E-ZPass Interagency Group: E-ZPass. http://www.ezpass.com/
10. Gura, N., Patel, A., Wander, A., Eberle, H., Shantz, S.C.: Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. In: Joye, M., Quisquater, J.J. (eds.) CHES. Lecture Notes in Computer Science, vol. 3156, pp. 119–132. Springer (2004)
11. Hager, C.: Divorce lawyers using fast lane to track cheaters. http://msl1.mit.edu/furdlog/docs/2007-08-10_wbz_fastlane_tracking.pdf
12. Heydt-Benjamin, T.S., Chae, H.J., Defend, B., Fu, K.: Privacy for public transportation. In: Danezis, G., Golle, P. (eds.) Privacy Enhancing Technologies. Lecture Notes in Computer Science, vol. 4258, pp. 1–19. Springer (2006)
13. de Koning Gans, G., Hoepman, J.H., Garcia, F.D.: A practical attack on the MIFARE Classic. In: Grimaud, G., Standaert, F.X. (eds.) CARDIS. Lecture Notes in Computer Science, vol. 5189, pp. 267–282. Springer (2008)
14. Massachusetts Bay Transportation Authority: CharlieCards & Tickets. http://www.mbta.com/fares_and_passes/charlie/
15. Meiklejohn, S., Mowery, K., Checkoway, S., Shacham, H.: The phantom tollbooth: Privacy-preserving electronic toll collection in the presence of driver collusion. In: USENIX Security Symposium. USENIX Association (2011)
16. Park, J., Hwang, J.T., Kim, Y.C.: FPGA and ASIC implementation of ECC processor for security on medical embedded system. In: ICITA (2). pp. 547–551. IEEE Computer Society (2005)
17. Popa, R.A., Balakrishnan, H., Blumberg, A.J.: VPriv: Protecting privacy in location-based vehicular services. In: USENIX Security Symposium. pp. 335–350. USENIX Association (2009)
18. Rupp, A., Baldimtsi, F., Hinterwalder, G., Paar, C.: Efficient and privacy preserving payments in transit systems: Cryptographic theory meets practice. http://cs.brown.edu/~foteini/papers/P4R.pdf (2013)
19. Sadeghi, A.R., Visconti, I., Wachsmann, C.: User privacy in transport systems based on RFID e-tickets. In: Bettini, C., Jajodia, S., Samarati, P., Wang, X.S. (eds.) PiLBA. CEUR Workshop Proceedings, vol. 397. CEUR-WS.org (2008)
20. Zhang, H., Gummeson, J., Ransford, B., Fu, K.: Moo: A batteryless computational RFID and sensing platform. Tech. Rep. UM-CS-2011-020, Department of Computer Science, University of Massachusetts Amherst, Amherst, MA (Jun 2011), https://web.cs.umass.edu/publication/docs/2011/UM-CS-2011-020.pdf