Group Discounts Compatible with Buyer Privacy

Josep Domingo-Ferrer and Alberto Blanco-Justicia

Universitat Rovira i Virgili
Dept. of Computer Engineering and Mathematics
UNESCO Chair in Data Privacy
Av. Països Catalans 26
E-43007 Tarragona, Catalonia
{josep.domingo,alberto.blanco}@urv.cat

Abstract. We show how group discounts can be offered without forcing buyers to surrender their anonymity, as long as buyers can use their own computing devices (*e.g.* smartphone, tablet or computer) to perform a purchase. Specifically, we present a protocol for privacy-preserving group discounts. The protocol allows a group of buyers to prove how many they are without disclosing their identities. Coupled with an anonymous payment system, this makes group discounts compatible with buyer privacy (that is, buyer anonymity).

Keywords: Buyer privacy, Group discounts, Cryptographic protocols, Digital signatures

1 Introduction

Group discounts are offered by vendors to encourage consumers to use their services, to promote more efficient use of resources, to protect the environment, etc. Examples include group tickets for museums, stadiums or leisure parks, discounted highway tolls or parking fees for high-occupancy vehicles, etc. It is common for the vendor to require all group members to identify themselves, but in reality this is seldom strictly necessary.

We make the assumption that the important feature about the group is the *number of its members*, rather than their identities. A secondary feature that may often (not always) be relevant for a group discount is whether group members are physically together.

Anonymously proving the number of group members and their being together is trivial in a face-to-face setting with a human verifier, who can see that the required number of people are present. However, with an automatic verifier and/or in an on-line setting, this becomes far from obvious.

In this paper, we propose a method to prove the number of people in a group while preserving the anonymity of group members and without requiring specific dedicated hardware, except for a computing device with some wireless communication capabilities (*e.g.* NFC, Bluetooth or WiFi). Also, we explore the option to include payment in our proposed system, which is necessary for group discounts. We complete the description of our method with a possible anonymous payment mechanism, *scratch cards*. The method presented here is a generalization of a specific protocol for toll discounts in high-occupancy vehicles, whose patent we recently filed [6].

The rest of the paper is structured as follows. Section 2 describes the building blocks of our method, namely a digital signature scheme, a key management scheme, an anonymous payment scheme and wireless communication technologies; the latter technologies should be short-range in applications where one wants to check that the group members are physically together. Section 3 describes our actual group size accreditation method, including the required entities and protocols. The security and the privacy of our proposal are analyzed in Section 4. In Section 5, we give a complexity estimation of our approach and describe precomputation optimizations. Finally, Section 6 summarizes conclusions and future work ideas.

2 Building Blocks

Our group size accreditation method is based on an identity-based dynamic threshold ($IBDT$) signature scheme, namely a particular case of the second protocol proposed in [7].

Threshold signature schemes are commonly based on (t, n)-threshold secret sharing schemes, such as the ones introduced in [1] and [11], and they require a minimum number t of participants to produce a valid signature. Dynamic threshold signature schemes differ from the previous ones in that the threshold t is not fixed during the setup phase, but is declared at the moment of signing. Our method takes advantage of this feature to find out how many users participated in the signature of a particular message, and consequently how many people form a group. If one wishes to prove that the signature is not only computed by at least t participants, but also that *these are together in the same place*, the above signature schemes need to be complemented with short-range communication technologies.

On the other hand, identity-based public key signature schemes, theorized by Shamir in [12] and with the first concrete protocol, based on the

Weil pairing, developed by Boneh *et al.* in [3], allow public keys pk^U to be arbitrary strings of some length, which we call *identities*. These strings are associated with a user U and reflect some aspect of his identity, *e.g.* his email address. The corresponding secret key sk^U is then computed by a trusted entity, the certification authority (CA), taking as input the user's identity and, possibly, some secret information held only by the CA, and is sent to the user U through some secure channel. Identity-based public key signature schemes offer a great flexibility in key generation and management and our method takes advantage of this feature by proposing a key management scheme that allows preserving the anonymity of the participants.

Finally, in most group discounts, a fee must be paid after proving the number of group members, so an anonymous payment method is needed. Indeed, this method should not reveal additional information about the group members to the service provider.

2.1 IBDT Signature Scheme

We outline a general identity-based dynamic threshold signature scheme, namely the second protocol proposed in [7]. Our protocol will be a slight modification of this general case; we will point out differences when needed. A general *IBDT* signature scheme consists of the following five algorithms.

IBDT 1 ***Setup*** *is a randomized trusted setup algorithm that takes as input a security parameter λ, a universe of identities $\mathcal{ID}$ and an integer n which is a polynomial function of λ and upper-bounds the possible thresholds (*i.e. *n is the maximum number of users that can participate in a threshold signature). It outputs a set of public parameters* pms *and a master key pair* msk *and* mpk*. An execution of this algorithm is denoted as*

$$(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}\,(\lambda, \mathcal{ID}, n)\,.$$

IBDT 2 ***Keygen*** *is a key extraction algorithm that takes as input the public parameters* pms*, the master key pair* msk *and* mpk*, and an identity* $\mathbf{id} \in \mathcal{ID}$*. The output is a private key $SK_{\mathbf{id}}$. An execution of this algorithm is denoted as*

$$SK_{id} \leftarrow \mathsf{Keygen}\,(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk}, \mathbf{id})\,.$$

IBDT 3 ***Sign*** *is a randomized signing algorithm that takes as input the public parameters* pms*, the master public key* mpk*, a user's secret key*

$SK_{\mathbf{id}}$*, a message* $\mathsf{Msg} \in \{0,1\}^*$ *and a threshold signing policy* $\Gamma = (t, S)$ *where* $S \subset \mathcal{ID}$ *and* $1 \leq t \leq |S| \leq n$*. Note that, in our case,* t *will be strictly equal to* $|S|$*. Sign outputs a partial signature* $\sigma_{\mathbf{id}}$*. We denote an execution of this algorithm as*

$$\sigma_{\mathbf{id}} \leftarrow \mathsf{Sign}\left(\mathsf{pms}, \mathsf{mpk}, SK_{\mathbf{id}}, \mathsf{Msg}, \Gamma\right).$$

IBDT 4 ***Comb*** *is a deterministic signing algorithm which takes as input the public parameters* pms*, the master public key* mpk*, the secret key of the combiner user* $SK_{\mathbf{id}}$*, a message* Msg*, a threshold signing policy* $\Gamma = (t, S)$ *and a specific set* S_t *of* t *partial signatures. Comb outputs a global signature* σ*. We denote the action taken by the signing algorithm as*

$$\sigma \leftarrow \mathsf{Comb}\left(\mathsf{pms}, \mathsf{mpk}, SK_{\mathbf{id}}, \mathsf{Msg}, \Gamma, \{\sigma_{\mathbf{id}}\}_{\mathbf{id} \in S_t}\right).$$

IBDT 5 ***Verify*** *is a deterministic verification algorithm that takes as input the public parameters* pms*, a master public key* mpk*, a message* Msg*, a global signature* σ *and a threshold policy* $\Gamma = (t, S)$*. It outputs* 1 *if the signature is deemed valid and* 0 *otherwise. We denote an execution of this algorithm as*

$$b \leftarrow \mathsf{Verify}\left(\mathsf{pms}, \mathsf{mpk}, \mathsf{Msg}, \sigma, \Gamma\right).$$

For correctness, for any security parameter $\lambda \in \mathbb{N}$, any upper bound n on the group sizes, any universe $\mathcal{ID}$, any set of public parameters and master key pair $(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk})$, and any threshold policy $\Gamma = (t, S)$ where $1 \leq t \leq |S|$, it is required that for

$$\sigma = \mathsf{Comb}\left(\mathsf{pms}, \mathsf{mpk}, SK_{\mathbf{id}}, \mathsf{Msg}, \Gamma, \{\sigma_{\mathbf{id}}\}_{\mathbf{id} \in S_t}\right),$$

$$\mathsf{Verify}\left(\mathsf{pms}, \mathsf{mpk}, \mathsf{Msg}, \sigma, \Gamma\right) = 1$$

whenever the values pms, mpk, msk have been obtained by properly executing the Setup algorithm, $|S_t| \geq t$, and for each $\mathbf{id} \in S_t$, $\sigma_{\mathbf{id}} \leftarrow \mathsf{Sign}(\mathsf{pms}, \mathsf{mpk}, SK_{\mathbf{id}}, \mathsf{Msg}, \Gamma)$ and $SK_{\mathbf{id}} \leftarrow \mathsf{Keygen}(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk}, \mathbf{id})$.

2.2 Key Management

The anonymity provided by our accreditation method is a result of our key generation protocol and management solution. As we stated above, identity-based public key cryptosystems allow using arbitrary strings as public keys. In our protocol, every user U_i is given an ordered list of public keys that depend on some unique identifier of the user, such as his national identity card number, his phone number, the IMEI number of his phone or a combination of any of them. We will call this identifier $n_{U_i} = d^i_k d^i_{k-1} \dots d^i_1$, where d^i_j is the j-th last digit of n_{U_i} and typically ranges from 0 to 9.

To generate the list of public keys from an identifier n_{U_i}, we choose a value $\ell < k$ and take the ℓ last digits of n_{U_i}. This results in a vector of public keys

$$\mathbf{PK}_{U_i} = \left\{ \mathsf{pk}_1^{d_1^i}, \dots, \mathsf{pk}_\ell^{d_\ell^i} \right\},$$

with every $\mathsf{pk}_j^{d_j^i}$ being an encoding of the digit and its position in n_{U_i}, for example:

$$\mathsf{pk}_j^{d_j^i} = j \,||\, d_j^i,$$

where $||$ is the concatenation operation. To illustrate this process, imagine $n_{U_i} = 12345678$ and $\ell = 4$. The resulting public key list would be

$$\mathbf{PK}_{U_i} = \{18, 27, 36, 45\}.$$

To prove the number of members in a group, the members will choose a common integer $j \in \{1, \dots, \ell\}$ so that the j-th public key in their list, *i.e.* $\mathsf{pk}_j^{d_j^i}$, is different for all of them. Then they will perform the required operations with these public keys and their corresponding private keys. Assuming that the values of the digits range from 0 to 9, this would provide anonymity to each of the users, since on average 10% of people will share the same public key $\mathsf{pk}_j^{d_j^i}$ for some value of j.

Note that this approach limits the size of the groups that can be certified with our method to a maximum of 10. Moreover, intuition tells us that the closer the size of the group to this maximum size, the more difficult it becomes to find a value of j for which each user has a different public key. The probability that our protocol fails depends on the number of keys each user is given, ℓ, and the size of the group n; more specifically for $n \leq 10$:

$$F(\ell, n) = \left(1 - \frac{10(10-1)\dots(10-n+1)}{10^n} \right)^\ell,$$

that is very close to 1 for values of n close to 10.

The limit on the maximum value of n can be increased by assigning $d \geq 2$ digits of n_{U_i} to each of the ℓ public keys, instead of just one digit. By doing this, the maximum value for the size of the groups becomes 10^d, and the probability of failure, for values of $n \leq 10^d$, is

$$F(\ell, n, d) = \left(1 - \frac{10^d(10^d-1)\dots(10^d-n+1)}{10^{dn}}\right)^{\ell}.$$

However, the price to be paid for choosing a larger d is a loss of anonymity, since, if more digits are associated to each public key, less users share the same public key. For example, for $d = 2$ a user would share each of his keys with only 1% of the total number of users.

The service provider will choose ℓ and d depending on maximum number of keys that a user can store, the maximum allowed group size and the anonymity level to be guaranteed.

2.3 Anonymous Payment Mechanisms

Group discounts are one of the applications of our method: after proving the group size, the group members must pay a fee that depends on that size. If proving the size has been done anonymously, it would be pointless to subsequently use a non-anonymous payment protocol (such as credit card, PayPal, etc.).

Hence, we need to use an anonymous payment mechanism along with our group size accreditation protocol. The simplest option for an anonymous payment method is to use cash if the application and the service provider allow it. Unfortunately, this will not always be the case, and other payment methods have to be taken into account. Electronic cash protocols such as [4] are good candidates for this role. Nowadays, Bitcoin [9] is a well-established electronic currency and, although it is not anonymous by design [10], it can be a good solution if accompanied by careful key management policies. Also, extensions of the original protocol as Zerocoin [8] provide anonymity by design.

For completeness, we propose in this work to use a much simpler approach, based on prepaid scratch cards that users can buy at stores using cash (for maximum anonymity). Each such card contain a code $\mathsf{Pay.Code}$ which the service provider will associate with a temporary account holding a fixed credit specified by the card denomination.

2.4 Communication Technologies

Our accreditation method requires communication among the members of a group and between the members and some type of verifying device. If we want to prove not only that a group has a certain number of members, but also that these are together, the interactions with the verifying device must rely on short-range communication technologies, like NFC, Bluetooth or WiFi.

During the accreditation protocol, the users' smartphones will be detected in some way by the verifying device and a communication channel will be established. The requirements and constraints of this process depend on the type of service and verifying devices, but nonetheless it is desirable that communication establishment be fast and not too cumbersome to the user.

We propose to use Bluetooth, and in particular *Bluetooth Low Energy* [2] to communicate with the verifying device. BLE solves some of the main limitations of traditional Bluetooth, *i.e.* reduces detection and bonding times, requires much less work by the user than NFC and has a shorter range than both Bluetooth and WiFi, which is desirable in a method like ours. Finally, BLE is implemented by most major smartphone manufacturers, at least in recent models, unlike NFC.

Regarding communication between the smartphones, any of the three mentioned technologies, or a combination of them (*e.g.* Bluetooth pairing through NFC messages) seems appropriate. The choice is up to the service provider.

3 Group Size Accreditation Method

A service that implements our accreditation method includes the following elements:

- A service provider (SP) that publishes a smartphone application App_U and distributes the necessary public parameters and keys of an *IBDT* signature scheme Π to users, after some registration process.
- A smartphone application App_U for each user U which:
  - allows computing signatures with Π on behalf of U;
  - allows computing ciphertexts with a public-key encryption scheme Π' selected by SP, under SP's public key pk^{SP};
  - can be run on master or slave mode, which affects how App_U participates in the accreditation protocol.
  - includes some certificate which allows checking the validity of pk^{SP};
  - implements some communication protocol, relying in short-range communication technologies, such as NFC or Bluetooth, to interact with the applications of the rest of the members of the group and with the verifying devices.

- Prepaid payment scratch cards available at stores. Each card includes a code $\mathsf{Pay.Code}$ that the SP associates to an account with a fixed credit specified by the card denomination.
- Verifying devices installed at suitable places in the provider's infrastructures which:
  - allow verifying signatures with Π;
  - hold the SP certificates as well as the keys needed to decrypt ciphertexts produced with Π' under pk^{SP}.
  - have short-range communication capabilities and implement some protocol to communicate with the users' devices.
- Some method to penalize or prevent the misuse of the system.

The complete accreditation protocol runs as follows:

Protocol 1 ***System setup protocol.***

1. *SP chooses the user identifier to be used as n_U and appropriate values for ℓ and d.*
2. *SP generates the parameters of the* IBDT *signature scheme Π as per Algorithm* $\mathsf{IBDT.Setup}$*;*
3. *SP generates the parameters of the public-key encryption scheme Π'.*

Protocol 2 ***Registration protocol.***

1. *A user U with identifier n_U authenticates himself to the service provider, face-to-face or by some other means. The user receives a PIN code* pin_U*.*
2. *The service provider associates to U a vector of public keys of Π,* $\mathbf{PK_{id}}$ *as described in Section 2.2.*
3. *The service provider computes the secret keys associated to* $\mathbf{PK_{id}}$ *as per Algorithm* $\mathsf{IBDT.Keygen}$*:*

$$\mathbf{SK_{id}} = \left(sk_1^{d_1^{\mathbf{id}}}, \ldots, sk_\ell^{d_\ell^{\mathbf{id}}} \right).$$

4. *The user downloads the smartphone application* App_U *and, using the PIN code* pin_U*, completes the registration protocol and receives the system parameters and keys, as well as the public key* pk^{SP}*.*

Protocol 3 ***Credit purchase.***

1. *A user buys a prepaid card for the system,* e.g. *a scratch card, from a store.*

2. *The card includes some code* $\mathsf{Pay.Code}$ *which has to be introduced in the smartphone application.*

Protocol 4 ***Group setup protocol.***

1. *Some user* U^**, among the group of users* $U_1, \ldots, U_t$ *who want to use the service, takes the leading role. This user will be responsible for most of the communication with the verifying device.* U^* *sets his smartphone application to run in master mode and the others set it to work in slave mode.*
2. *The users agree on a value* $j \in \{1, 2, \ldots, \ell\}$ *such that the value of the* j*-th public key in* $\mathbf{PK_{id}}$ *is different for every user.*

Protocol 5 ***Group size accreditation protocol.***

1. *A verifying device detects the users' devices and sends them a unique time-stamped ticket* T *that may include a description of the service conditions and options.*
2. *Each user* U_i *runs Algorithm* $\mathsf{IBDT.Sign}$ *to compute a partial signature with* Π *under his secret key* $\mathsf{sk}_j^{d_j^i}$ *on message*

$$\mathsf{Msg} = \left\langle \mathsf{T} \,||\, \mathsf{pk}_j^{d_j^1} \,||\, \ldots \,||\, \mathsf{pk}_j^{d_j^t} \right\rangle ,$$

   for the threshold predicate $\Gamma = (t, \{\mathsf{pk}_j^{d_j^1}, \ldots, \mathsf{pk}_j^{d_j^t}\})$*. It sends the resulting partial signature* $\sigma_\mathbf{i}$ *to* U^**.*
3. U^* *receives* $(\sigma_1, \ldots, \sigma_t)$ *and runs Algorithm* $\mathsf{IBDT.Comb}$ *to combine these signatures and output a final signature* σ *on behalf of* $U_1, \ldots, U_t$*.* U^* *sends to the verifying device*

$$\mathsf{Msg}' = \langle \mathsf{Msg}, \sigma \rangle .$$

4. *The verifying device checks the validity of the signature by running*

$$\mathsf{IBDT.Verify}(\mathsf{Msg}, \sigma, \mathsf{pk}_j^{d_j^1} || \ldots || \mathsf{pk}_j^{d_j^t}, t).$$

   Note that this signature will only be valid if all users $U_1, \ldots, U_t$ *have collaborated in computing it, and thus it proves that the group of users is composed of at least t people. If the signature is not valid, the group will be penalized in an application-dependent way,* e.g. *with access denial, group discount denial, etc. Otherwise, the service provider grants access to the group of users and tells the group the amount* amount_t *they have to pay depending on the group size.*

Protocol 6 ***Payment.***

1. *Each group member U in the (sub)set P of group members who want to collaborate in paying the bill sends to the verifying device via Bluetooth or WiFi his payment code encrypted under SP's public key:*

$$C_U = \mathsf{Enc}_{\mathsf{pk}^{SP}}(\mathsf{T}||\mathsf{Pay.Code}_U),$$

where $\mathsf{Pay.Code}_U$ is the code which user U obtained from a prepaid scratch card and where Enc is the public-key encryption algorithm of scheme Π'.

2. *The verifying device decrypts the ciphertexts $\{C_U : U \in P\}$ to obtain the payment codes of the users in P.*
3. *The verifying device substracts the quantity amount_t divided by the cardinal of P to the accounts associated with the received payment codes.*

4 Security and Privacy Analysis

Security and privacy are offered by design in our proposal:

- The chosen IBDT scheme ensures unforgeability of signatures under chosen message attacks even when an attacker can choose arbitrarily the threshold signing policy. In this case, this means that, for any $t \geq 2$, no group of less than t buyers is able to deceive the service provider by producing a threshold signature with threshold t. Complete security proofs can be found in the original paper [7].
- No more than the pseudonyms and the number of participants of a group is revealed to the service provider during the execution of the protocol. Buyer anonymity is guaranteed by the key management scheme described in Section 2.2 *within the community of buyers sharing the same public key.* For example, if each public key is associated to a combination of d decimal digits, then on average this public key is shared by a community containing $10^{-d} \times 100\%$ of the total number of users.
- When payment is completely anonymous, whatever anonymity level achieved by key management is preserved after payment. For our given method, this is ensured when a given $\mathsf{Pay.Code}$ cannot be linked to a specific buyer. This can be achieved, for example, if the scratch card containing the $\mathsf{Pay.Code}$ is purchased using cash.

5 Performance Analysis

Our group size accreditation method is to be run by service providers, specialized verifying devices and the users' smartphones. Therefore, it is important that the computations of the underlying cryptographic protocol be as fast as possible, especially the algorithms that are executed by the smartphones, which have limited computational capabilities and rely on batteries.

In this section, we analyze the performance of the underlying *IBDT* signature scheme. This scheme is a pairing-based cryptographic protocol and as such, the required operations are performed in elliptic curve groups. We analyze its performance by counting the number of point multiplications, point exponentiations and pairings, which are the most costly operations.

Table 1 shows the number of these operations for each of the algorithms in the *IBDT* signature scheme. The number of operations is counted as a function of the maximum number of possible participants in a signature, n, and the size of the signing group t. As we stated previously, $t \leq n$.

Table 1. Operations required per algorithm

	Multiplications	**Exponentiations**	**Pairings**
Setup	0	$n+4$	1
Keygen	$2n$	$4n$	0
Sign	$2n+6$	$2n+5$	0
Comb	$2n-t+1$	$2n-t$	0
Verify	$n+2$	$n+1$	4

Note that the Sign and Comb algorithms, that are intended to be executed in the users' smartphones during the **group size accreditation protocol** (5), present what seems to be quite a high number of operations. This might be a problem if the devices in which these algorithms are to be executed do not have enough computational power. Moreover, these two algorithms should precisely be most efficient, since they are run most often, and possibly with time constraints. Therefore, it would be interesting if we could precompute some of their operations.

The Sign algorithm is a probabilistic protocol, that is, it has some random values in it that have to be refreshed each time it is executed.

This limits the amount of operations in the algorithm that can be precomputed. On the other hand, most of the operations depend on static values, *e.g.* keys and threshold policies Γ. Threshold policies contain the number of signers that will participate in a signature and their public keys. We assume that groups of users will be quite stable, *i.e.* users will generally use services together with the same group members, or at least with a limited set of different groups. We can exploit this assumption by precomputing operations that only depend on static values and threshold policies.

The Comb algorithm obviously depends on the output of Sign, but it is a deterministic algorithm and some of its operations depend on static values and also on the threshold policies. Therefore, by the same assumption as before, we can precompute some of the operations.

These precomputations will divide the Sign and Comb algorithms in two phases each, one for precomputing values, which will be executed during the **group setup protocol** (4), and the other one performed during the **group size accreditation protocol** (5). The resulting number of operations in each of these phases is shown in Table 2.

Table 2. Precomputed and non-precomputed operations of the Sign and Comb algorithms (PC stands for precomputed)

	Multiplications	**Exponentiations**	**Pairings**
Sign PC	$2n + 2$	$2n + 1$	0
FastSign	2	4	0
Comb PC	$2n - 2t$	$2n - 2t$	0
FastComb	$3t + 1$	$3t$	0

6 Conclusions and Future Work

We have presented a privacy-preserving mechanism for group discounts. The method is built upon an *IBDT* signature scheme, a concrete key generation and management solution, short-range communication technologies and anonymous payment mechanisms. Our complexity analysis and initial tests show that the method is usable in practice.

Future work will consist of implementing the protocol, testing it and developing a generic app for privacy-preserving group discounts that can be easily customized for specific applications.

Acknowledgments

The following funding sources are acknowledged: Google (Faculty Research Award to the first author), Government of Catalonia (ICREA Acadèmia Prize to the first author and grant 2014 SGR 537), Spanish Government (project TIN2011-27076-C03-01 "CO-PRIVACY"), European Commission (FP7 projects "DwB" and "Inter-Trust") and Templeton World Charity Foundation (grant TWCF0095/AB60 "Co-Utility"). The authors are with the UNESCO Chair in Data Privacy. The views in this paper are the authors' own and do not necessarily reflect the views of Google, UNESCO or the Templeton World Charity Foundation.

References

1. G. R. Blakley, "Safeguarding cryptographic keys," *Proceedings of the National Computer Conference*, pp.313–317, New York: AFIPS Press, 1979.
2. Bluetooth SIG, "Specification of the Bluetooth System," 2013. Available in https://www.bluetooth.org/en-us/specification/adopted-specifications.
3. D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *Advances in Cryptology–CRYPTO 2001*, LNCS 2139, pp. 213–229, Springer, 2001.
4. D. Chaum, A. Fiat and M. Naor, "Untraceable electronic cash," *Advances in Cryptology–CRYPTO 88*, LNCS 403, pp. 319–327, Springer, 1990.
5. A. De Caro, V. Iovino, "jPBC: Java pairing based cryptography," *Computers and Communication (ISCC), 2011 IEEE Symposium on*, pp. 850–855, IEEE, 2011. Available in http://gas.dia.unisa.it/projects/jpbc/.
6. J. Domingo-Ferrer, C. Ràfols and J. Aragonès-Vilella, *Method and system for customized contactless toll collection in carpool lanes* (in Spanish "Método y sistema de cobro sin contacto, por el uso de una vía, para vehículos de alta ocupación"), Spanish patent ref. no: P201200215. Date filed: February 28, 2012. Patent owner: Universitat Rovira i Virgili.
7. J. Herranz, F. Laguillaumie, B. Libert and C. Ràfols, "Short attribute-based signatures for threshold predicates," *Topics in Cryptology–CT-RSA 2012*, pp. 51–67, Springer, 2012.
8. I. Miers, C. Garman, M. Green and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," *Security and Privacy (SP), 2013 IEEE Symposium on*, pp. 397–411, IEEE, 2013.
9. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Consulted*, vol. 1, 2008. Available in http://www.bitcoin.org/bitcoin.pdf.
10. F. Reid and M. Harrigan, "An analysis of anonymity in the Bitcoin system," in *Security and Privacy in Social Networks*, eds. Y. Altshuler *et al.*, pp. 197–223, Springer, 2013.
11. A. Shamir, "How to Share a Secret," *Communications of the ACM*, 22:612–613, 1979.
12. A. Shamir, "Identity based cryptosystems and signature schemes," *Advances in Cryptology–CRYPTO 1984*, LNCS 196, pp. 47–53, Springer, 1985.