Using Identity-Based Public-Key Cryptography with Images to Preserve Privacy

Sebastian Pape and Nabil Benamar

Databases and Interactive Systems Research Group, University of Kassel
Wilhelmshöher Allee 73, 34121 Kassel
{pape,benamar}@db.informatik.uni-kassel.de
http://www.db.informatik.uni-kassel.de/

Abstract. We propose a public-key signature and encryption application which strongly relies on identity-based public-key cryptography. By alternately using obvious identity information like names and essential image data of the involved parties as public keys we preserve all advantages gained by identity-based public-key schemes, mainly including the absence of a public-key infrastructure [1]. On the other hand, all parties obtain only obvious and necessary information about other involved parties.

1 Introduction

The purpose of our application is to avoid tickets written on paper and particularly to remove those bondings, where a customer's name is printed on his ticket and he has to show his passport, that the controller can check the equality of the name on the ticket and the one in the passport. The controller's next step then usually is to compare the customer's appearance with the picture in his passport. In many cases the passport is only a sort of translation from the customer's name to his picture. Thus, the customer's name, address, identity number and so on are not needed here, the controller only wants to check if the person who claims a service is legitimated. Our approach aims at an portrait-based legitimation of customers with mobile devices like PDAs or cell phones. While there are several identity based applications, we found none which uses stand-alone pictures or essential parts of them to protect the customer's privacy. We give a sketch of our idea and some references in Sect. 2.1 how to derive the keys.
For the customer's purposes of course it would be desirable to have anonymous commercial transactions, e.g. by using anonymous digital credentials [2]. Otherwise there is a commercial demand to identify the customer, e.g. when charging fees for altering a booking or considering discount systems like the German Railways' one (Fig. 1). The latter costs an annual fee and grants a discount on all train tickets during this year in return. Needless to say that German Railways don't want their customers to share those discount cards. When looking at the examples above or at customer retention systems and considering today's courses of business, tickets have to be bound to a specific customer so that the transfer

of privileges, discounts or tickets is impossible. Therefor our aim was to bind tickets to a specific customer without using bureaucratic identity information like name, address, credit card number, any other customer number and so on. We explain in Sect. 2.5 why we suggest to rely on face-recognition here and do not think that the customer's face is as worthy of protection as other identity information like finger prints or name and address. To avoid reinventing the wheel we based our application solely on identity-based public-key cryptography.
Given that customers should be able to hold arbitrary devices, no tickets are stored on their mobile device(s). This design avoids unnecessary bondings to specific devices. The customer only needs to setup each of his devices once and is then able to switch them at his choice. Therefore, the tickets have to be stored in one (or more) database(s). But central ticket storage involves a drawback: Other persons – including the party providing the database – should not be able to browse the tickets of any customer. Third persons should only gain information with the customer's knowledge and control, e.g. when he proves his tickets valid to a train conductor. This leads to a database where all (most) information is encrypted with the appropriate customer's key. Since the customer has to decrypt his ticket before showing it, it has to be assured that he is not able to change or misuse the ticket's data.
As abovementioned a trivial example for our application is selling and controlling train tickets. Another example is the sale of soccer tickets. Regarding the last soccer world championchip all tickets contained RFID chips with an unique identifier which linked the ticket to the customers' identification information, e.g. name, date of birth, identity card number. Irrespective if all this information is really necessary it would be quite complicate to prove if a person belongs to a specific ticket. The guard has to read the ticket's unique number, lookup the customer's identification information in a database and then prove via the picture on the customer's identity card that he really belongs to the ticket. When looking at the current state of soccer, e.g. in Italy, there may be a need to personalise tickets, to keep hooligans out of the stadiums. But we claim that if there is really identification information necessary like name or identity card number to achieve this goal, it is needed when selling the ticket and not needed when entering the stadium. The guard does not need to know who wants to watch the soccer game, he only has to be sure, that the ticket is not passed to another person. We state more examples and wherein they differ later on.

2 Scenario, Terms and Our Contribution

There is a customer C who buys or receives a ticket t from a dealer D. Later on C has to prove the validity of his ticket to a Guard G. We only consider cases where C has face-to-face contact to D and G. Note that D could be any kind of salesman, e.g. for train, soccer or concert tickets or he even could be a doctor writing out prescriptions while G could be a controller or a pharmacist, respectively. An more abstract possibility would be to have personalised tokens which prove properties like "over 18", "valid driving license" or " European

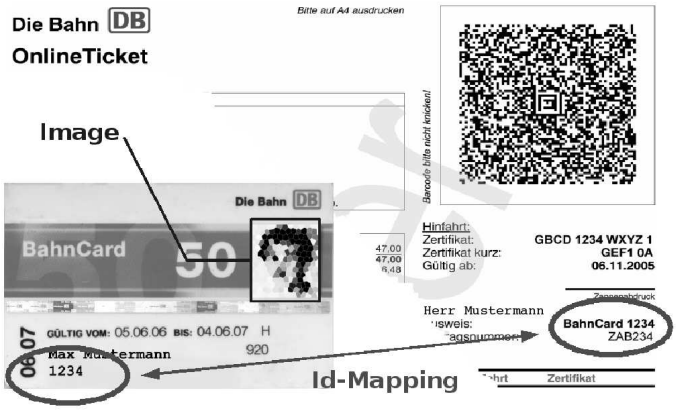

Fig. 1. Scenario example: BahnCard and OnlineTicket (German Railways)

citizen". We store the tickets in a database, but an adequate setup does not necessarily include a central database. As long as D writes to the same database G reads from, it is satisfactory to have subgroups sharing one database for each task. For example one database stores train tickets and another one contains recipes.
We would also like to emphasise, that the roles of guard and dealer are not fixed. Regarding to our first scenario of train tickets and personal discounts, the attribute "gets discount" could also be stored as a ticket at the database. Then the dealer would first act as guard and control the discount ticket, before granting the discount when selling a ticket.
The customer's public and private keys are denoted by $\mathsf{c}_{\mathsf{pub}}$ and $\mathsf{c}_{\mathsf{priv}}$, respectively. Analogous notations correspond to the dealer's and the guard's keys. Since all participants possess public-key pairs, we assume they communicate through a secure channel and do not need to consider authentication and encryption any further. The following section describes how each of the involved parties have to construct their public keys and which way their private keys are constructed by a trusted third party TTP.

2.1 Key Generation

In identity-based encryption or signature schemes the public key can be an arbitrary string. A trusted third party holds a secret master-key and then generates private keys corresponding to the respective public key string. We first describe how we construct the public keys and then suggest corresponding identity-based schemes.

Public Keys. Since it should be easy for any of C's counterparts (D or G) to get his public key, $\mathsf{c}_{\mathsf{pub}}$ is derived from the customer's face. Following [3] automatic

face-recognition involves three subtasks. The detection of faces, feature extraction and identification and/or verification. Regarding our purpose we only need the first subtasks, namely face detection and feature extraction. Despite face recognition and especially feature extraction is not perfect, enormous progress has been made. Therefore we cannot expect to get precisely the same data each time a picture of the same face is captured, but we assume that by feature extraction we receive data that for the same person remains reasonable close with each measurement. There are different efforts how to utilise this data for cryptographic purposes. Either by using fuzzy identity-based encryption [4], which has an error-tolerance property to allow decryption if and only if the sampled key is close to its original. Or by using fuzzy extractors proposed in [5] which provide the same output, even if the input changes, but remains reasonably close to the original. Dodis et al. also claim that their fuzzy extractors output is nearly distributed uniformly which renders it suitable as key in cryptographic applications. Since we will see in Sect. 2.4, that each customer needs a unique key to locate his tickets at the database we prefer the latter.
Generating the public keys for dealers and guards does not need the same effort. All dealers and guards use their obvious identity information (e.g. name, address or a symbolic name) as public key $\mathsf{d}_{\mathsf{pub}}$ respective $\mathsf{g}_{\mathsf{pub}}$. That way the customer can easily construct the dealers' and guards' public keys.

Private Keys. As already stated in identity-based public-key cryptography private keys are computed by a trusted third party TTP, which has to approve the identities of the particular party. There are two different needs for keys in our application. While C needs a pair of en- and decryption keys, D and G need signature key pairs. At first we address the customer's en- and decryption.
While there where several proposals, e.g. [6, 7], Boneh and Franklin [8] provided the first usable scheme for identity-based encryption. Their scheme relies on bilinear maps on elliptic curves, namely the Weil pairing and performs probabilistic encryption of arbitrary ciphertexts. Later research of identity-based encryption schemes is also mostly based on bilinear Weil or Tate pairings. We suggest to use their scheme not only because there already exists a well documented toolkit[1].
Let us now take a look at appropriate signature schemes for D and G. While there where quite early solutions for satisfactory id-based signature schemes [9, 10], we suggest to use the scheme from Cha and Chen [11] based on the hardness of the computational Diffie-Hellman problem since it shares the same system parameters and the same private/public key pairs with [8] and is claimed to be as efficient as Boneh's and Franklin's scheme.

2.2 Setup

Knowing how to construct private/public key pairs from the previous section, the setup for our application is quite easy. First of all the trusted third party TTP has to generate its master-key corresponding to the used cryptosystems.

[1] see http://www.voltage.com/ibe_dev/index.htm

Then each participant (C, D and G) has to get his private key from TTP (Fig. 2). The trusted third party approves that customers and dealers qualify and that the customer's public key is really derived from his face. Note that it may be possible to fully automate the process of generating c_{priv} likewise existing passport photograph automates. It is also worth mentioning, that following the previous section all participants are able to use their key pairs for signatures and en-/decryption.

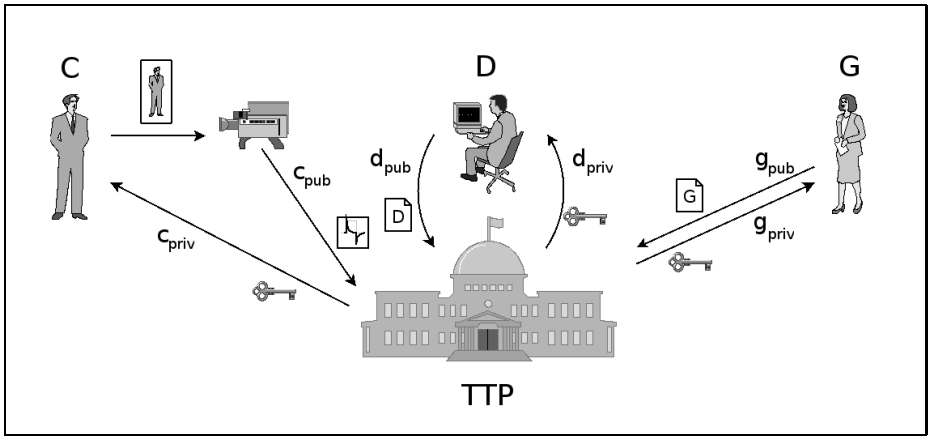

Fig. 2. Setup

2.3 Creation of Tickets

At first D has to construct c_{pub} by taking a picture of him and deriving the public key exactly as described in Sect. 2.1. As soon as D creates a ticket t, he includes c_{pub} and signs it with his private key d_{priv} and then encrypts the result with c_{pub}. Now D has to store $encr_c(sign_d(t, c_{pub}))$ in the common database as shown in Fig. 3. Note that for ticket creation the customer does not need his device. Although depending on the level of trust C has on D, D may has to prove that he really inserted the ticket in the database. Assuming D is a doctor, C might trust D will insert the ticket in the database while C may want some evidence when buying train, soccer or concert tickets. Due to the fact that no deterministic two-party contract-signing protocol can achieve fairness [12], a trusted third party may be present here. Since the usual setup probably is, that C is at D's facility and has no (straight) access to TTP, a convenient solution could be the so-called optimistic approach [13, 14]. When using optimistic protocols TTP can be regarded as offline, since TTP comes only into play if a problem appears, e.g. a technical failure or a cheating party. Thus, using the optimistic protocol for fair exchange D may return a signed receipt to C while receiving C's payment. This

procedure is almost equivalent to today's traditional processing. Alternatively any other fair protocol involving a trusted third party operating the database may be used instead.
Since the tickets are stored encrypted, they are stored in relation to C's public key to make it possible to recover them later on. There is also little additional (plain text) information (e.g. a date or a place) stored to reduce the number of tickets C has to decrypt later when showing his ticket (see Sect. 2.4).

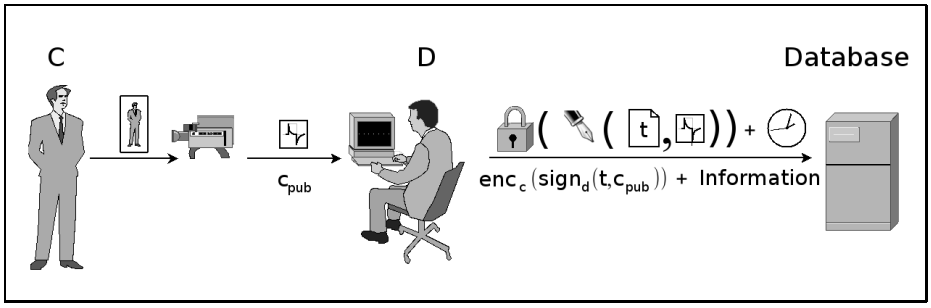

Fig. 3. Creation of tickets

2.4 Validation of Tickets

When C has to prove to G that he is the owner of a valid ticket, G first derives c_{pub} from C's face - exactly in the way D obtained C's private key in the previous section. Next G receives all tickets from the database associated with c_{pub} and the additional information and passes all matching data sets to C. Thus, C obtains a set of tickets of the form $enc_c(sign_d(t, c_{pub}))$. C is then able to decrypt the encrypted tickets and returns to G the unencrypted but signed ticket $sign_d(t, c_{pub})$ suitable for this situation. An overview of the procedure is depicted in Fig. 4. Note that C probably does not need to decrypt all tickets since he can benefit from the additional information and start decrypting the more likely tickets first.

2.5 Privacy Discussion

Using Identification Information Derived from the Customer's Face. As mentioned above we assume that the customer has face-to-face contact to the dealer and the guard. To get worse he not only has face-to-face contact to them, he usually enters their environment (shop, train, stadium, etc.). Furthermore state-of-the-art advances and produces cheaper, smaller and increasingly powerful devices. On account of this we assume it is almost impossible for the customer to prevent guard and dealer from installing hidden cameras and secretly taking

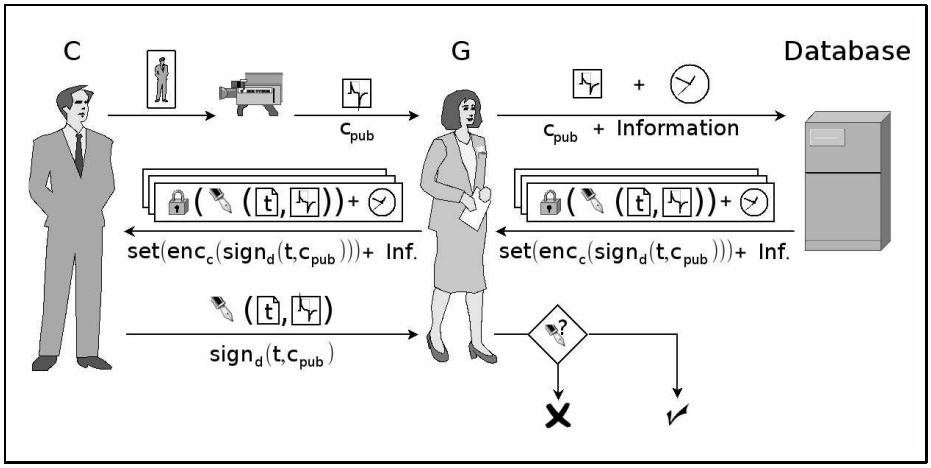

Fig. 4. Validation of tickets

pictures of the customer. We do not think that pictures or extracted essential information from them is less sensitive information than e.g. fingerprints, iris or retinal scans or gene checks. But when looking at biometric personal identification the customer can not prevent misbehaving dealers or guards from collecting information which does not rely on his cooperation or even knowledge. Besides face recognition, e.g. voice recognition and analysis of odour or gait fall in this category. Therefore we claim that we can make responsible use of such analysis since we do need regulations by law in any case - independent whether we make use of it or not. Furthermore if our application is used in conjunction with anonymous payment (like cash) and we assume misbehaving dealers or guards, they do not get information about the customer's name, address and so on like they get the traditional way. This may sound contradictory, but in our opinion the real privacy risk in face recognition systems is up to the connection of other sensitive information with the data from face recognition systems. As we explained in Sect. 2.3 almost any data at our database is stored encrypted.
Let us assume a dealer or guard misbehaves, who naturally has access to the stored data at the company's database. The public key of a customer is derived from essential information from the customer's face. Therefore an adversary can conclude how many tickets are stored for this user. He has also access to the sparse additional information stored with the tickets while all other information is stored encrypted. The adversary also may take additional pictures and store all information which has to be presented in plain text to him, namely ticket purchases or validations. The worst case is that all guards and dealers of a company are instructed to additionally store the customer's ticket information in plain text. Of course they can do that, too regarding traditional ticket creation and validation. But here furthermore the ticket is issued to a name or a unique identifier which is related to the customer's name. As already stated,

we assume a misbehaving dealer or guard is able to secretly take a picture of the customer, which he easily can link to the identifier or name of the customer assuming traditional ticket schemes. Hence with the proposed application we are at least not worse off than before, since we give only sparse information away and prevent dealers and guards from constructing a database with face recognition information linked to names or other identifiers. Although we can think of some cases where having bureaucratic identity information like name and address is far more crucial than having identity information in the form of a picture. For example when an adversary wants to collect more information it is easier to use world wide web search engines in conjunction with a name than with a picture. Even if we assume that customers are willing to disguise themselfs and we assume that face recognition can not cope with it, at least the guard has a good chance of taking a picture, when the customer has to remove his camouflage to prove that he really belongs to the picture at his identity card. Thus disguising is only helpful, if the customer dos not want to use tickets.

Additional Plaintext Information. As described above, the encrypted data stored in the database includes additional plain text information. This may be necessary if some customers hold many tickets. Since in the majority of cases G holds a mobile device and at least C's power is limited, it is useful to lower the number of tickets transferred. There is only sparse information that can be used here, because if we made C storing information he would better keep the ticket itself. However, depending on the amount of tickets it is possible to use time ranges here. Note that – independent of the kind of information stored – this is a trade-off to improve efficiency by compromising privacy since this information is stored unencrypted.

Regarding the Dealer's Anonymity. G may also be able to learn which dealer(s) C prefers since he has to verify their signature. This may be circumvented by using group signature schemes. We propose a variation of our application which makes use of identity-based group signatures in Sect.3.1. The idea of group signatures is to provide anonymity to the dealer, G is only able to verify that a member of a specific group (dealer) signed the ticket. The trusted third party acts as a group manager and is able to revoke anonymity in the case of abuse. In this case the anonymity of the dealer is valuable for C's privacy, since the guard may draw various conclusions from the fact which dealer(s) C prefers.

Compromised Private Keys. Even if the private key of a customer is compromised no one else can use his tickets, because the adversary would already fail when he has to provide the public key via the already in Sect. 2.1 explained procedure. Moreover the customer is able to get a reissued private key from the trusted third party, thus his already paid tickets are not lost. Hence compromised private keys are only relevant to the customer's privacy. Depending whether the adversary has access to the database the compromised key may let him decrypt

all tickets of the customer. In case of a lost key the customer could only aim for a re-encryption of all existing tickets with an uncompromised key. A general protection for the customer is to renew his key in a short intervall as described in Sect. 3.2.

2.6 Security Aspects

First of all it has to be ensured that C is unable to forge tickets. Since all tickets are signed by D it is infeasible for C to create tickets as long as the underlying cryptosystem holds. C is also not able to pass tickets to other customers, because the tickets are bound to $\mathsf{c}_{\mathsf{pub}}$.
Due to the fact that D is able to write to the common database, D is a more sensitive party. If D wants to insert forged tickets to the database he still has the same problem as mentioned above. Entries in the database have to be signed correctly – otherwise G will not accept the ticket later. As anyone can imagine signing tickets with his own key may be no wise decision if D wants to cheat. However, D must be prevented from deleting tickets and flooding the database with invalid entries. The former can easily be achieved by adapting the database's interface. The latter would require an additional database layer. Since all entries to the database are encrypted the integrity of new entries can not be checked. By using an additional signature of the encrypted record it is possible to track which dealer inserted invalid entries to the database. When C decrypts data he is then able to complain about invalid entries and the untrustworthy dealer's license can be removed. Note that C's claim can be easily proved here, since the encrypted entry has to be stored in the database.
Accounting G's capability is quite interesting in spite of the fact he is only able to read the database. G is able to change data he read from the database before he hands it to C. Given that G is always able to decline C's legitimation – even if C turns over a valid ticket, his only intention could be accusing D of cheating. On the one hand this may easily be prevented if the database provider (or the dealer) additionally signs the set of data he sends to G. On the other hand this accusation cannot be held up for long, simply because any other honest guard can prove the opposite.
Since any combination of cheating parties that involves the guard benefits from the fact, that G is able to manipulate the legitimation test, the only combination of parties cheating in common that makes sense to consider is the pair of customer and dealer. But even if C and D make common cause with each other, the ticket still has to be signed by the dealer since G proves that later. The only way they could cheat would be if D issues a ticket to C, but instead of transmitting it to the database he hands it to C. When C has to prove to G that he has a valid ticket, he discards the set of tickets from G and shows the ticket he received from D to G. If this flaw can be exploited depends on the exact procedure charges are payed from D and is beyond the scope of this paper - not only because if paper tickets are used, D could easily print an extra ticket.
Thus, we claim our application is secure against forgery as long as the underlying cryptosystem holds and the guard really examines the tickets. The latter is

no drawback since dishonest guards or controllers cancel almost any real world ticket system.

3 Variations

As already stated in Sect. 2.5 we also propose two slight variations of our application.

3.1 Using Group Signature Schemes for Dealer and Guard

If we want to prevent the guard from learning which dealers the customer prefers, we have to use a group signature scheme. Thus all dealers or guards belonging to the same organisation use the same public key, e.g. "Dealer of German Railways" or "Soccerclub's guard". There are several approaches for id-based group signatures [15, 16] based on bilinear pairings which could be used instead of [11]. The price we have to pay here is, that we cannot share the system parameters and the private/public key pairs with [8], which might be annoying but is feasible.

3.2 Key Revocation

The main advantage of identity-based public-key cryptography is that the distribution of public keys is quite easy, because they can be derived from identity information (e.g. the customer's picture in our application) and therefore no directories with files of public keys need to be kept. But there is a price to pay. In traditional public-key schemes certification revocation lists are used to deal with the consequences of compromised keys. However when using identity-based public-key cryptography a traditional certification revocation list would give its main advantage away. The first generalised method for key revocation in identity-based public-key cryptography was described in [8]. By adding a period of time (e.g. the current year) to a public key it contains an implicit preset expiration date. The public key $\mathsf{c}_{\mathsf{pub}}^{\mathsf{rev}}$ for this variation therefore would be a concatenation of $\mathsf{c}_{\mathsf{pub}}$ and the expiration date: $\mathsf{c}_{\mathsf{pub}}^{\mathsf{rev}} = \mathsf{c}_{\mathsf{pub}} || expiration - date$. While public keys can still be derived quite easily this way, the trusted third party has to renew the private key each time the period is over. Note that this is no real key revocation since the customer has to sit an wait until his key expires and despite of whether the key should be revoked or not a regular (and frequent) key renewal is necessary. Otherwise buying tickets and getting a new private key is commutative since the customer does not need his private key when purchasing a ticket, because he only has to decrypt his ticket, when showing it to the guard. However this design of public keys necessitates a short renewal interval to reduce the impact of lost or broken keys. Thus it is inevitable to relieve the customer from the burden of receiving a private key in short time periods from the trusted third party. Dodis et al. [19–21] introduce the idea of a private key-generator-device which generates the actual private key from a secret

master key in non id-based cryptography. Hierarchical identity-based encryption schemes [17, 18] however allow the trusted third party do delegate key generation to some lower-level trusted party. As the name hierarchical indicates, there can be several levels of delegation and when a key-generator of a specific level is compromised higher-level key generators are not put at risk. [22] then combines hierarchical identity-based cryptosystems with the idea of private key generators and suggest that, e.g. each time a mobile phone's battery is recharged it recharges its stored private keys from such a private key generator. That way it is possible to have short renewal intervals, without making the customer revisiting the trusted third party daily.
If we use $\mathsf{c}_{\mathsf{pub}}^{\mathsf{rev}}$ as customer's public key in combination with short renewal intervals, our application changes slightly, because we can omit the additional plain text information stored with each ticket.

4 Conclusion and Drawbacks

By using the above setup implicit key management is given as known by identity-based public-key systems and almost no unnecessary information is revealed to any party. Since the customer knows at least the symbolic identity of salesmen, doctors, controllers, pharmacists and so on he easily derives the corresponding public keys without gaining additional knowledge. Vice versa because the customer's public key is derived from a picture of him all groups mentioned above learn nothing more about him than they could see anyway when negotiating face-to-face. Note that TTP is only involved when setting up the system. The trusted third party is not needed during the communication phase although it could be useful if the customer does not trust his dealer (see Sect. 2.3).
As stated in Sect. 2.6 none of the participating parties is able to cheat and as long as the underlying cryptosystem holds our application can be regarded as secure.
However, there are some drawbacks. Given that both dealer and guard need the ticket's plain text information it is impossible to prevent them from keeping their own records. Nevertheless, this is not a major drawback since today's real world scenario already allows that. Depending on the situation the customer may even want to keep them informed (e.g. doctor, pharmacist).
Finally the proposed application removes the bonding between a customer's name and a service and makes it possible to bind tickets to a picture, so the customer reveals no more information than obvious in face-to-face communication. Even if an undesirable face-to-identity dictionary exists this application may be usefull, since access to this dictionary hopefully will be restricted to governmental authorities. In conjunction with anonymous payment [23] our application gives consideration to the user's privacy needs while also including commercial issues and provides personally bound electronic tickets which do at least not reveal more information than todays transactions.

5 Acknowledgements

We would like to thank all our colleagues for helpful discussions, especially Heiko Stamer for his numerous annotations.

References

1. A. Shamir, Identity-based cryptosystems and signature schemes, Advances in Cryptology-Crypto 84, LNCS 196, pages 47–53, Springer-Verlag, 1984.
2. D. Chaum, Security without identification: transaction systems to make big brother obsolete. Communications of the ACM, 28, 1030-1044, 1985.
3. W. Zhao, R. Chellappa, J. Phillips, A. Rosenfeld, Face Recognition: A Literature Survey ACM Computing Surveys, pages 399–458, 2003.
4. A. Sahai and B. Waters, Fuzzy Identity Based Encryption. In Advances in Cryptology – Eurocrypt, volume 3494 of LNCS, pages 457–473. Springer Verlag, 2005.
5. Y. Dodis, L. Reyzin, A. Smith, Fuzzy extractors: How to generate string keys from biometrics and other noisy data, In Proceedings of the International Conference on Advances in Cryptology (EUROCRYPT '04), Lecture Notes in Computer Science, Springer Verlag, 2004.
6. H. Tanaka, A Realization Scheme for the Identity-Based Cryptosystem CRYPTO '87: A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology, Springer-Verlag, pages 340–349, 1988.
7. S. Tsuji and T.Itoh, An ID-based cryptosystem based on the discrete logarithm problem, IEEE Journal of Selected Areas in Communications, Vol.7, No.4, pp.467–473, 1989.
8. D. Boneh, M. Franklin, Identity-Based Encryption from the Weil Pairing, SIAM Journal on Computing, vol. 32, issue 3, pages 586–615, 2003.
9. A. Fiat, A. Shamir, How to prove yourself: practical solutions to identification and signature problems Proceedings on Advances in cryptology, In Proceedings of CRYPTO '86, Springer-Verlag, pages 186–194, 1987.
10. U. Fiege, A. Fiat, A. Shamir, Zero knowledge proofs of identity In Proceedings of the nineteenth annual ACM conference on Theory of computing (STOC '87), ACM, pages 210–217, 1987.
11. J. C. Cha, J. H. Cheon, An Identity-Based Signature from Gap Diffie-Hellman Groups, PKC '03: Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography, Springer-Verlag, pages 18–30, 2003.
12. S. Even, Y. Yacobi, Relations among public key signature systems, Technical Report 175, pages 148-153, Computer Science Dept, Technion, Israel, March, 1980.
13. N. Asokan, M. Schunter, M. Waidner, Optimistic Protocols for Fair Exchange, In 4th ACM Conference on Computer and Communications Security, pages 7–17, 1997.
14. H. Bürk, A. Pfitzmann, Value exchange systems enabling security and unobservability, In Computers and Security, vol. 9 ,pages 715–721, 1990.
15. Z. Chen, and J. Huang, and D. Huang, and J. Zhang and Y. Wang, Provably secure and ID-based group signature scheme, In Proceedings of the 18th International Conference on Advanced Information Networking and Applications (AINA 2004), volume 2, 384-387, 2004.

16. V. K. Wei, T. H. Yuen, F. Zhang, Group Signature Where Group Manager, Members and Open Authority Are Identity-Based, In Proceedings of the Information Security and Privacy, 10th Australasian Conference, ACISP 2005, Springer Verlag, pages 468–480, 2005.
17. C. Gentry, A. Silverberg, Hierarchical ID-Based Cryptography, Advances in Cryptology – Asiacrypt'2002, Lecture Notes on Computer Science 2501, Springer-Verlag, pages 548–566, 2002.
18. J. Horwitz, B. Lynn, Towards Hierarchical Identity-Based Encryption, Advances in Cryptology – Eurocrypt'2002, Lecture Notes on Computer Science 2332, Springer-Verlag, pages 466–481, 2002.
19. Y. Dodis, J. Katz, S. Xu, and M. Yung, Key-insulated public key cryptosystems, Proc. Eurocrypt'02, LNCS 2332, pages 65-82, Springer-Verlag, 2002.
20. Y. Dodis, M. Franklin, J. Katz, A. Miyaji, and M. Yung, Intrusion-resilient public-key encryption, Proc. CT-RSA'03, LNCS 2612, pages 19-32, Springer-Verlag, 2003.
21. Y. Dodis, M. Franklin, J. Katz, A. Miyaji, and M. Yung, A generic construction for intrusion-resilient public-key encryption, Proc. CT-RSA'04, LNCS 2964, pages 81-98, Springer-Verlag, 2004.
22. Y. Hanaoka, H. Hanaoka, J. Shikata, H. Imai, Identity-Based Hierarchical Strongly Key-Insulated Encryption and Its Application, Cryptology ePrint Archive, Report 2004/338, 2005.
23. David Chaum, Blind signatures for untraceable payments, Advances in Cryptology, Proceedings of CRYPTO '82 (David Chaum, Ronald L. Rivest, and Alan T. Sherman, eds.), Plenum Press, 1983.