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 identity information and pictures 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 ([8]). 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 identity 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 identity is not needed here, the controller only wants to check if the person who claims a service is legitimated. Our approach aimes at an image-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 to protect the customer's privacy.
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. We state more examples and wherein they differ later on.

2 Scenario and Terms

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. Note that D could be any kind of salesman, e.g. for train or concert tickets or he even could be a doctor writing out prescriptions while G could be a controller or a pharmacist, respectively. Since we store the tickets in a database, D must have write access to this database while it is sufficient for G to have read only access. 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.
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. But first of all each of the involved parties have to get their private keys from a trusted third party TTP. The following paragraph describes the setup.

2.1 Setup

Since it should be easy to check for any of C's counterparts (D or G) that C's public key $\mathsf{c}_{\mathsf{pub}}$ really belongs to him, the public key is a picture of the customer. As known in identity-based public-key crytography C's private key $\mathsf{c}_{\mathsf{priv}}$ is computed by a trusted third party TTP, which has to make sure that C is using his own picture (Fig. 1). Given that C usually has face-to-face contact to D and G, the picture reveals only information that both of them can get anyway. Note that we intentionally use C's picture, since fingerprints, iris recognition or gene checks would disclose private information of C.
All dealers and guards use their identity information (e.g. unique name, address or a symbolic name) as public key $\mathsf{d}_{\mathsf{pub}}$ respective $\mathsf{g}_{\mathsf{pub}}$. TTP approves their identity and computes their private keys $\mathsf{d}_{\mathsf{priv}}$ and $\mathsf{g}_{\mathsf{priv}}$. Since C knows where he applies for a ticket it is easy for him to verify D's and G's identity (and public key).
As an option TTP may also have a key pair. Knowing that the trusted third party computes the private keys of all involved parties, key distribution is no problem here. This key pair enables TTP to sign the public keys of all parties - either to complicate faking of keys or to allow the use of actual ID-based systems which require additional data (see [7, p. 562]). Another option is to replace TTP by a certification authority (CA). This would render central private key computation by TTP unnecessary. Public keys then need to include the necessary identity information like C's picture as already known from PGP ([9]). As

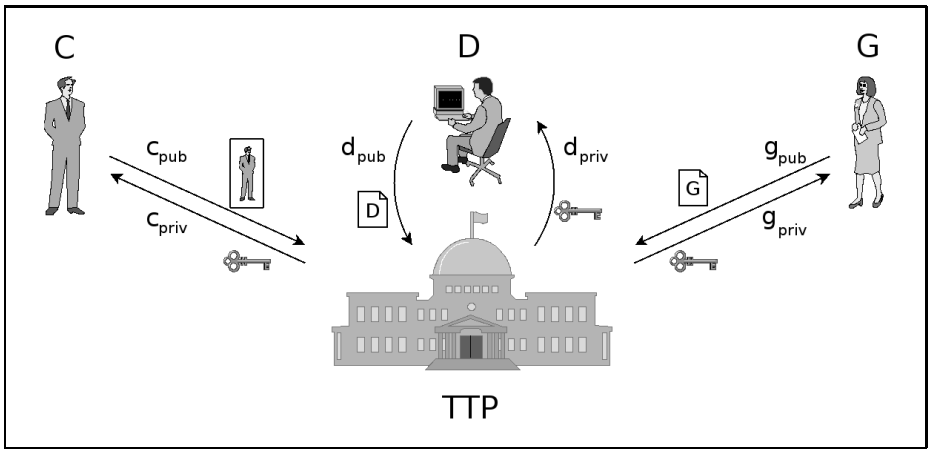

Fig. 1. Setup

before CA has to verify all participants' public information (identity or picture) and sign their public keys. A minor drawback here is, that all participants have to exchange their public keys. But since face recognition software is far from being perfect, C has to give his public key to D anyway. Only if face recognition software advances, it would be possible to use specific datasets as public keys, thus D may be able to derive C's public key by a digital camera.

2.2 Creation of Tickets

If c_{pub} is not already known to D he receives it from C. Since c_{pub} is a picture of C, D is able to check easily that the received key belongs to C (if he wants to). As soon as D creates a ticket t, he signs it with his private key d_{priv} and then encrypts the result with c_{pub}. Now D has to store $encr_c(sign_d(t))$ in the common database as shown in Fig. 2.
Depending on the level of trust C has on D, D has to prove that he really inserted the ticket. Assuming D is a doctor, C might trust D will insert the ticket in the database while C may want some evicence when buying train or concert tickets. Due to the fact that no deterministic two-party contract-signing protocol can achieve fairness ([5]), 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 ([1, 3]). 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. TTP's function can be fulfilled either by a trusted database provider or by the trusted party who already provides the participant's keys. 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

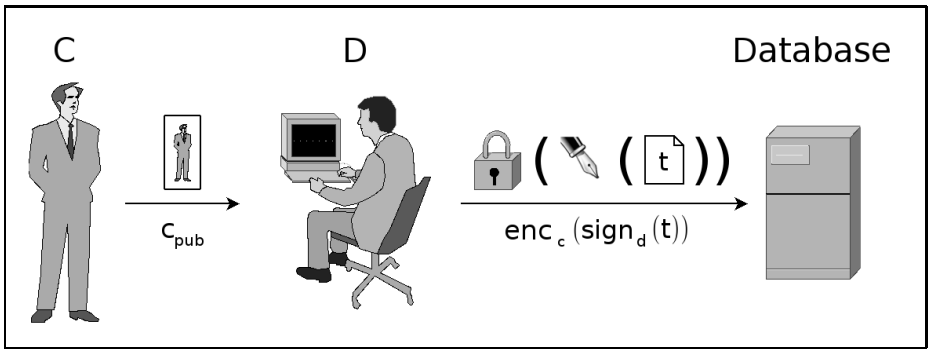

Fig. 2. creation of tickets

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. If there are lots of tickets it may be an option to include additional (plain text) information (e.g. a date or a place) to reduce the number of tickets C has to decrypt later (see Sect. 2.4).

2.3 Validation of Tickets

When C has to prove to G that he is the owner of a valid ticket, G receives all tickets from the database associated with c_{pub} and the optional, additional information. If c_{pub} is not already present, similar to the procedure when creating tickets G receives c_{pub} from C and can easily verify that the received key belongs to C. G then passes all matching data sets to C. Thus, C obtains a set of tickets of the form $enc_c(sign_d(t))$. C is then able to decrypt the encrypted tickets and returns to G the unencrypted but signed ticket $sign_d(t)$ suitable for this situation. An overview of the procedure is depicted in Fig. 3
Since C proved to be the owner of c_{priv} by decrypting $enc_c(sign_d(t))$ and C is not able to sign tickets with d_{priv}, it seems G may infer that C has received a valid ticket from D. But checking D's signature is not enough here. Since it is necessary to hand C the encrypted tickets he can not be prevented from keeping a copy. To make sure copies are useless to other persons, G has to encrypt the plain text again with c_{pub} and check wether $enc_c(sign_d(t))$ was really stored in the database. Note that G does not need to read from the database again, he simply uses the set of tickets he received earlier. Alternatively C's public key or at least its fingerprint or hash has to be included into the ticket (see also Sect. 2.4).

2.4 Privacy Issues

As described above, the encrypted data stored in the database may include additional plain text information. This may be necessary if some customers hold

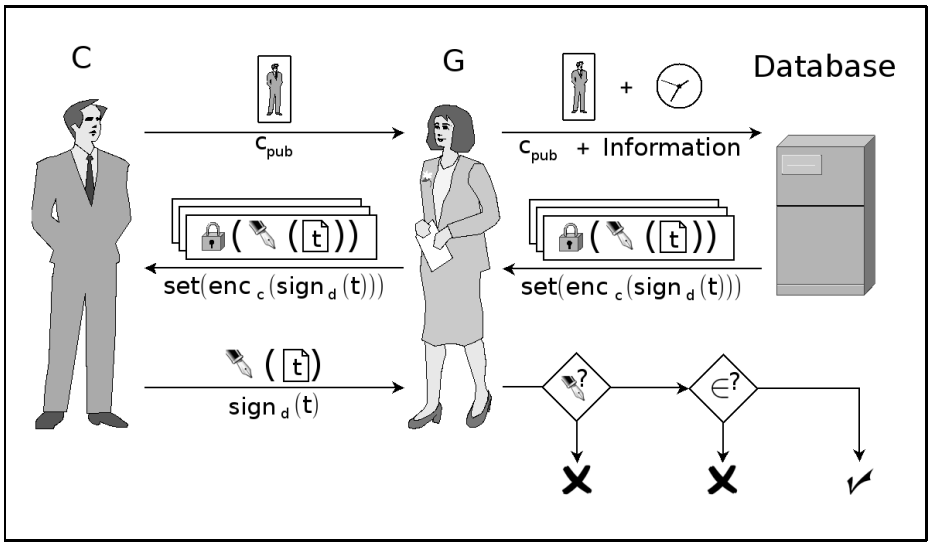

Fig. 3. validation of tickets

many tickets. Since in the majority of cases G holds a mobile device and at least C's power is limited, it may be 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. Since G compares two different encryptions of the same plaintext when validating the ticket he received from C it becomes obvious, that the underlying cryptosystem has to be deterministic. Remembering that in public-key cryptography any attacker has the capability of chosen-plaintext attacks this is again a trade-off. If improved privacy is required, a probabilistic cryptosystem ([6]) can be used, but as stated in Sect. 2.3 C's public key has to be included in the ticket then. Keep in mind that this also involves a drawback regarding the system's security (see Sect. 2.5). G has no possibility to check if the ticket he received from C was really stored in the database when using a probabilistic cryptosystem. However, an attacker would still have to forge D's signature.
G may also be able to learn which dealers C prefers since he has to verify their signature. This may be circumvented by using group signature schemes ([4], [2]). In this concept the group signature provides anonymity to the dealer, G is only able to verify that a member of a specific group signed the ticket. The trusted third party acts as a group manager and is able to revoke anonymity in the case of abuse. Thus, C's privacy is protected.

2.5 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 either the tickets have to originate from the database which is assured by G or c_{pub} is linked with the ticket.
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 group signature schemes and additional records 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 adds an other layer of signed encryption – remember c_{pub} is already available to him, since it is C's database-key. On the other hand this accusation can not be hold 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, two handicaps still exist. The ticket has to be signed by the dealer and it has to be encrypted with the customer's public key and stored in the database, since G proves both of it. Precisely because G proves if the signed ticket he receives from C is really stored in the database, C has not left any ways to change the ticket, even if he possesses the customer's private key and is able to create and sign tickets by his own. Due to the fact, the ticket is stored encrypted and G is unable to read it, the customer and the dealer do not benefit from the ability of changing the ticket later, since G is unable to read it in between anyway.
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.
An interesting instance arises concerning an underlying probabilistic cryptosys-

tem as mentioned in Sect. 2.4. Since G is not able to prove if a particular entry is in the database C and D may cheat if they pool together – or if D is also a customer. In this scenario 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. Due to the fact that re-encryption probably results in another encryption, G is unable to detect this deception. If this flaw can be exploited depends on the exact procedure charges are payed and is beyond the scope of this paper.

3 Conclusion and Drawbacks

By using the above setup implicite 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 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 also it could be useful if the customer does not trust his dealer (see Sect. 2.2).
As stated in Sect. 2.5 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. First of all many concrete proposals of identity-based systems include additional data ([7]) which modifies our setup into a slightly more complicate one. The trusted third party has to sign public keys and make its key available to all parties similar to the option replacing the trusted third party by a certification authority. 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. If the progress of face recognition software continues, it may be possible to derive the customer's key straightly from a digital camera.

4 Acknowledgments

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

References

1. N. Asokan, M. Schunter, M. Waidner: Optimistic Protocols for Fair Exchange. In 4th ACM Conference on Computer and Communications Security, pages 7–17, 1997.
2. G. Ateniese, J. Camenisch, M. Joye, G. Tsudik: A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. Lecture Notes in Computer Science, vol. 1880, pages 255–270, 2000.
3. H. Bürk, A. Pfitzmann: Value exchange systems enabling security and unobservability. In Computers and Security, Vol. 9 ,pages 715–721, 1990.
4. D. Chaum, E. van Heyst: Group signatures. In Advances in Cryptology - EUROCRYPT'91, Vol. 547 of LNCS, pages 257–265, 1991.
5. S. Even, Y. Yacobi: Relations among public key signature systems. Technical Report 175, pages 148-153, Computer Science Dept, Technion, Israel, March, 1980.
6. S. Goldwasser, S. Micali: Probabilistic Encryption. Special issue of Journal of Computer and Systems Sciences, Vol. 28, No. 2, pages 270-299, April 1984.
7. A. J. Menezes, P. C. von Oorschot, S. A. Vanstone: Handbook of Applied Cryptography. CRC Press, Boca Raton, New York, London, Tokyo, 1997.
8. A. Shamir: Identity-based cryptosystems and signature schemes. Advances in Cryptology-Crypto 84, LNCS 196, pages 47–53, Springer-Verlag, 1984.
9. P. Zimmerman: The Official PGP Users Guide. MIT Press, Cambridge, 1995.