A framework for the revocation of unintended digital signatures initiated by malicious terminals

István Zsolt BERTA Levente BUTTYÁN István VAJDA

January 15, 2005

Abstract

Human users need trusted computers when they want to generate digital signatures. In many applications, in particular if the users are mobile, they need to carry their trusted computers with themselves. Smart cards are easy to use, easy to carry, and relatively difficult to tamper with, but they do not have a user interface, therefore, the user still needs a terminal for authorizing the card to produce digital signatures. If the terminal is malicious, it can mislead the user and obtain a digital signature on an arbitrary document. In order to mitigate this problem, we propose a solution based on conditional signatures. More specifically, we propose a framework for the controlled revocation of unintended digital signatures. We also propose a solution with a special emphasis on privacy issues.

Keywords: electronic commerce, security, authorization, privacy

1 Introduction

We consider electronic commerce applications, where a mobile user – a sole human being – wishes to make business with a partner. When using cryptographic protocols, the user needs a *terminal* (e.g., a PC), which stores cryptographic keys and performs cryptographic computations on behalf of her. In addition, the terminal needs to be trusted by the user for behaving as expected, and for not compromising the security of the user (e.g., by leaking her keys). Unfortunately, most terminals cannot be called ’trusted’. Either because the party operating the terminal is not trusted by the user, or the user cannot be convinced that the terminal does not have hidden features. To prevent attacks from the terminal, smart cards are used as security measures. Although smart cards are useful for protecting cryptographic keys, they cannot verify that the message they sign was not altered by a malicious terminal, since they do not have a user interface. [1] Cryptoboxes, hardware security modules and other devices that lack a user interface are not better than smart cards against untrusted terminals: The user still needs a user interface (which is possibly malicious) to interact with these devices. Secure and tamper resistant devices manufactured with a user interface could yield a solution if they are manufactured by a trusted party. However, in these cases the user must be able to differentiate between "real" devices that are manufactured by a trusted party and between "fake" ones that are manufactured by the attacker. The protocol of Asokan et al. [2] offers a solution for such differentiation, but it is insecure against the terminal in the middle attack (or grandmaster chess attack) where the malicious terminal hijacks the user interface of a trusted terminal. (see the [2] or Annex A5 of CEN CWA 14890-1 for the description of this attack) Other solutions (like holograms on secure devices) could also provide but a limited level of protection. The ultimate solution would be a personal tamper resistant device manufactured by a trusted manufacturer that has a user interface and is small enough so the user can carry it with her. According to Rivest [3], such a device is unlikely to become feasible in the near future. He argues that user-friendly interfaces that can be customized to suit the needs of many users are unlikely to be secure. (Mobile phones and PDAs that allow the downloading of third party applications are good examples for this.) Therefore, he suggests, that digital signatures should not be considered non-repudiable proofs, but simply plausible evidence. Thus users should be given well-defined possibilities for repudiating such signatures.

In our paper follow this paradigm, and we propose a solution to the problem of untrusted terminals, our solution is based on a new concept called *conditional signature*. In our model digital signatures are not considered non-repudiable proofs, at least until a short deadline. Preliminary results related to this approach have been published in [4] and [5].

2 Related work

The problem of using untrusted terminals was addressed by Abadi et al. [6] first, by analyzing the dangers of delegation of rights to a terminal. They show that this problem could be solved with a *super smart card* (that has peripherals to communicate directly with the user), and also show secure protocols for such a device. The solution of Clarke et al. [7] uses a futuristic super smart card that is monitoring the screen of the terminal with a digital camera. In contrast to solutions based on super-smart cards, the one presented in this paper does not require the card to have any special peripheral, but can be implemented using smart cards that exist today.

Some authors have attempted to provide solutions for *realistic smart cards*. In the protocol proposed by Stabell-Kulo et al. [8] the user protects her message with a one-time-pad and a monoalphabetic substitution. We also proposed a solution where the user protects her message with a combination of biometric and algorithmic means until it is signed by her smart card [9]. The solution presented in this paper does not require the user to perform any computations.

The most straightforward way a user can protect her privacy is refusing to provide information that can be linked with her. However, a non-repudiable, digitally signed message is – by definition – linkable with the user. In this paper we address the problem of sending such messages from an untrusted terminal while allowing the user to retain some privacy with respect to third parties. Papers discussing the possibilities of users with limited resources in a malicious environment rarely address privacy issues simultaneously. However, we found that anonymous payment systems address a very similar privacy-problem to ours. These systems need a trusted party, usually called the bank, to issue coins, to detect double spending, to handle accounts, etc., but users would like their transactions to be untraceable by this trusted party. The foundations of some famous anonymous payment systems are introduced in [10], [11] and [12]. The trusted third party in this paper is in a position very similar to that of a bank in the above papers, so we borrow ideas from anonymous payment systems to provide privacy protection for the users. We also rely on the existence of anonymous communication channels. (see e.g. [10] or [13]).

3 Model

We consider a system where there are mobile human users who want to generate digital signatures at untrusted terminals. *User* U has limited memory and computational power. For this reason, the private key of U is stored on and the signatures are generated by smart card C in possession of user U.

Essentially, *smart card* C is a trusted personal microcomputer without direct interfaces towards U. C is connected to the terminal in front of U, and all messages between C and U must pass through the untrusted terminal. We assume that smart card C is manufactured by a trusted manufacturer and hence, it functions correctly. In particular, C does not try to leak the private key of U or to use the private key without authorization. We also assume that smart card C is able to perform cryptographic operations, like encryption or digital signature, to generate good quality pseudo-random numbers, and to store a few thousand bytes of data.

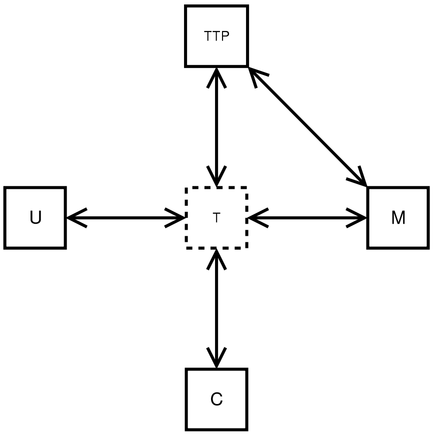

Figure 1: The entities in our model and the channels between them

We assume that *untrusted terminal* T in front of U is fully under the control of an attacker, who may have installed all kinds of malicious software on the terminal before U started to use it. This means that the attacker is able to steal and abuse any PIN code[1] typed in by U on the keyboard of the terminal, to send fake messages to U through the display of the terminal, and to modify messages that U sends to C for signing before passing them on to C. Thus, the attacker can obtain a signature from the smart card for an arbitrary message. In many applications, user U has to rely on untrusted terminal T in order to access a particular service. However, we assume that from time to time, U has access to C from a *trusted terminal* too. Such a trusted terminal could be a terminal operated by a trusted organization and believed to be tamper resistant (e.g., an ATM machine). Of course, in order to use a terminal for this purpose, it must be properly authenticated first.

We denote by M *the intended recipient of the digital signature* generated by C. M could be a service provider, a merchant, another user, etc.

TTP is a *trusted third party* in the system that both U and M trust. In Section 4 user U considers TTP to be completely trusted, but in Section 5 user U would like to retain her privacy with respect to TTP. This means that U trusts TTP only for the revocation of unintended signatures, but she would like to prevent TTP from knowing, which partners she does business with. TTP follows the given protocols, and does not try to cheat by breaking into the terminal or by intercepting messages for other parties. Neither does TTP collaborate with T or M to discover the identity of the user.

The entities of the model and their interconnections are illustrated in Figure 1.

4 A solution based on conditional signatures

In order to detect attacks, we propose a framework that allows mobile users to sign messages on untrusted terminals with the help of their smart cards, review the signatures later in a trusted

[1] Although PIN codes are useful against e.g. card theft, they provide little protection against the threat of untrusted terminals, so their use is not discussed further in this paper.

environment, and revoke fake ones. This is made possible by using conditional signatures.

Conditional signatures were introduced by Lee and Kim [14], who used this concept for solving fair exchange problems without expensive cryptographic primitives like verifiable escrow. A conditional signature of U on a message m is U's ordinary signature $sig_U(m, c)$ on m and a description of a condition c. If $sig_U(m, c)$ is correct and condition c is true, then $sig_U(m, c)$ is considered to be equivalent with $sig_U(m)$, U's ordinary digital signature on m. However, if c is false, then U is not responsible for m. Intuitively, U's conditional signature is U's commitment: *'I signed m, but if c is not true, then my signature on m is not valid.'*

Since it is not possible to prevent the terminal from obtaining a signature from the card on an arbitrarily chosen document, we propose that C *generates a conditional signature* such that it is guaranteed that the condition cannot become true before a certain amount of time has passed. This leaves time for the user to move to a trusted terminal for checking the signatures generated by the card, and for enforcing that the conditions of the fake signatures can never become true.

The conditional signature finalizes the user's offer towards recipient M. Neither U, nor M can modify the signed offer anymore, but both of them can withdraw from it (e.g. if an attacker had modified the offer before it was signed). M may choose to reject the offer and not to provide any service to the user. The user may invalidate the offer and claim the she did not intend to sign that particular document. However, if the user does not revoke the signature until a certain deadline (contained in the condition), then she cannot do anything about the validity of her signature anymore. If she revokes an intended signature, M can use the revoked signature as plausible evidence for proving that the user initiated a transaction. Still, the user may question this evidence, so one revoked signature may not be enough for bringing a user to court, but if a user revokes too many signatures at too many partners, either she can be blacklisted or a court may consider the evidence to be sufficient. Conditional signatures should be used in systems where a particular service can be accessed only via terminals not trusted by the user, and a user is likely to take part in many transactions, and transactions have rather a small value so it is not worth for the user to spoil her reputation by revoking a single intended signature.

Our framework follows the philosophy of existing credit based payment systems. Within this paper we are proposing neither a new payment system, nor means for a merchant to collect payments. We propose a solution for protecting mobile user from malicious terminals.

These thoughts lead to the following generic protocol. Note that steps 1-4 below happen at an untrusted terminal, steps 5 and 6 are performed using a trusted terminal and via secure channels.

Protocol 1 (the generic protocol).

Step 1: $U \rightarrow T$: m

Step 2: $T \rightarrow C$: m

Step 3: $C \rightarrow T$: $c, sig_U(m, c)$

The card logs m in its internal memory[2], computes the conditional signature $sig_U(m, c)$ of U on m, where c is a condition that includes (among other things) deadline t, and outputs $(c, sig_U(m, c))$ to the terminal. The intention is that the signature $sig_U(m, c)$ will not be valid before t; in addition, it will become valid after t if and only if the other conditions in c hold.

Step 4: $T \rightarrow M$: $(m, c, sig_U(m, c))$

Step 5: $C \rightarrow U$: M, m, c

Later, but before the deadline t, U reviews the list of messages logged by C at a trusted terminal.

Step 6: For each message m the user intended to sign, she ensures that the condition c becomes true; for the rest of the messages, U ensures that the condition becomes false. This might involve additional steps and further communication with M or TTP. (See Protocol 2 for an example.) In order to verify a conditional signature, the verifier needs to check if the digital signature $sig_U(m, c)$ of the card is correct and condition c is true.

There are two possibilities for determining the value of condition c. One possibility is that condition c becomes true unless the user revokes her signature (default accept). The other possi-

[2] In order to make the presentation easier, we assume that the card can log the entire message. In [4] we show that it is enough for the card to receive the hash of the message, and the message itself can be logged by an external log server that needs to be trusted only by the user.

bility is that the condition becomes false unless the user confirms her signature (default deny). In [4] we show an example for both approaches. Protocols following the latter approach are simple, but they require user U to explicitly confirm each signature; this means that M cannot do business with users who forget to confirm their signatures. Protocols supporting the default accept approach seem more practical, but in these protocols the conditional signature is in the hands of the untrusted terminal (or in the hands of recipient M) after Step 3, and it becomes valid after a certain deadline automatically, unless the user revokes it. If the user has to revoke the signature, then she would like that the revocation becomes known to all parties in the system. The party responsible for publishing the revocation should be trusted both by the user (e.g. for the correct handling of the revocations and for making them available to everyone) and by recipient M (e.g. for refusing to accept late revocations). Therefore, it seems that practical protocols require the help of a trusted third party (TTP).

Below, we propose a specific example where the interpretation of condition c is the following: *"My signature on the above message is valid if and only if deadline t has passed and TTP countersigned it."* Whenever user U leaves a trusted terminal she sets deadline t on her card to a point of time when she is likely to be using a trusted terminal again.

Protocol 2 (the condition is a simple deadline).

U signs message m at an untrusted terminal:

Step 1: $U \rightarrow T$: m

Step 2: $T \rightarrow C$: m

Step 3: $C \rightarrow T$: $t, TTP, sig_U(m, t, TTP)$

Step 4: $T \rightarrow M$: $m, t, TTP, sig_U(m, t, TTP)$

U reviews signed messages at a trusted terminal:

Step 5: $C \rightarrow U$: M, m, t, TTP

Step 6: If U did not intend to sign message m and deadline t has not passed, then:

$U \rightarrow TTP$: *'I revoke my signature $sig_U(m, t, TTP)$.'*

Otherwise, U does not need to act.

Step 7: $M \rightarrow TTP$: $t, TTP, sig_U(m, t, TTP)$

Step 8: If U did not revoke the signature at TTP before t, then:

$TTP \rightarrow M$: $sig_{TTP}(sig_U(m, t, TTP))$

In order to verify the conditional signature, the verifier needs to check if the digital signatures $sig_U(m, t, TTP)$ and $sig_{TTP}(sig_U(m, t, TTP))$ are correct.

5 A solution to protect the user's privacy

While user U may trust TTP for signature revocation, perhaps she does not want TTP to know, where, when and what messages she wanted to sign. Therefore, we enhance our former protocol for signature revocation to allow U to retain her privacy with respect to TTP. We reckon that if the protocol prevented TTP from linking the user with recipient M, more organizations would qualify to be a TTP.

During a protocol run, user U would like to prevent TTP from obtaining any information that can differentiate her from other users. In particular, she would like to hide id_U (her user name or identifier), and message m. Moreover, she would also like to prevent TTP from obtaining any information that can be linked with these too. It is clear that user U does not want to protect this information against M, because she intends to send message m to recipient M. She cannot hide m from T either, because she types the message using the keyboard of the terminal. If there are n users who rely on TTP for signature revocation (and e.g. subscribe to this service), TTP has at least a $\frac{1}{n}$ chance of selecting the particular user who took part in the protocol. Therefore, our aim is to develop protocols, where TTP can suspect each user with probability close to $\frac{1}{n}$. Intuitively, this means that TTP's probability distribution of any user sending the message is uniform, and hence, no user is more likely to be the sender than any other. [15].

Our next protocol follows the generic concepts of Protocol 1 but also protects the privacy of the user. The first deviation appears in Step 3, when the smart card outputs two cryptograms.

One of them is the conditional signature encrypted with a random symmetric key k, and the other one is encrypted with the public key of TTP and contains condition c along with a random revocation token r and key k. This protocol follows the spirit of bit commitment protocols: user U commits herself to her signature to M, and reveals her signature if c is true, i.e. if the signature is not revoked before deadline t. Unlike in the protocol described in Section 4, terminal T is unable to verify the signature in this step. Thus, we need to refine our assumptions about the smart card: Henceforth, smart card C is assumed to be trustworthy and tamper-resistant[3], so *all other parties* (U and M and TTP) *consider C a trusted party*. Thus, in Step 3 terminal T knows that card C follows the protocol, and is not sending garbage. The cryptogram $E_{TTP}(r, k, c)$ that the smart card outputs in Step 3 is forwarded to M in Step 4 and later to TTP in Step 7. The user receives revocation token r from the card via a trusted terminal, and may repudiate her signature by submitting r to TTP in Step 6 via an *anonymous channel*. We assume that such an anonymous channel exists. TTP decrypts the cryptogram that was sent by M in Step 7, and enforces condition c to become true (in Step 8) unless the revocation token r inside the cryptogram was submitted before. Based on r, TTP is unable to link U with M. (Note that the identity of M is not hidden from TTP.) While TTP needs to store revocation token r, it may not be necessary to store it forever. This problem could be solved e.g. by introducing a lapse time, so TTP could refuse to validate very ancient conditional signatures. In this case, condition c is the following string: *"My signature on the above message is not valid before deadline t."* The protocol looks as follows:

Protocol 3 (protecting the user's privacy).

Step 1: $U \rightarrow T$: m

[3] Tamper-resistance means, it is impossible to alter its behavior, reverse engineer it or extract information from it. Smart cards are considered to be tamper-resistant in commercial applications only, so well-funded attackers with a state-of-the-art semiconductor laboratory might be able to penetrate their defenses. Sometimes, experts figure out low cost attacks [16] that do allow less funded (but highly skilled) adversaries to mount certain attacks on smart cards. However, until now, literature was able to propose countermeasures against most low-cost attacks (e.g. [17], [18]). Thus we consider the tamper-resistance of smart cards a justified assumption.

Step 2: $T \rightarrow C$: m

C generates random symmetric key k and revocation token r.

Step 3: $C \rightarrow T$: $c,\ E_k[sig_U(m)],\ E_{TTP}(r, k, c)$

Step 4: $T \rightarrow M$: $m,\ c, E_k[sig_U(m)],\ E_{TTP}(r, k, c)$

Later, at a trusted terminal:

Step 5: $C \rightarrow U$: M, m, c, r

If user U would like to repudiate the signature on message m then

Step 6: $U \rightarrow TTP$: r (via an anonymous channel)

After deadline t:

Step 7: $M \rightarrow TTP$: $E_{TTP}(r, k, c)$

If deadline t has passed, and r was not submitted to TTP, then:

Step 8: $TTP \rightarrow M$: k

Step 9: M decrypts $E_k[sig_U(m)]$ using k and obtains $sig_U(m)$.

One important merit of this protocol that a third party needs to have m and $sig_U(m)$ only in order to verify the conditional signature of U. Since, this conditional signature is not different from a regular one, its verification requires the same procedure too. The other important merit of Protocol 3 is that *the user is able to retain a provable degree of privacy with respect to TTP.*

Assume that user U comes from a large community $\mathcal{U}$ of users, the so-called anonymity set. She would like to conceal her identity $id_U \in \mathcal{U}$ from TTP, so user U would like to prevent TTP from finding out which user in this community sent a particular message. TTP does not have any a priori knowledge on which user is sending a message, so the distribution of random variable id_U is uniform in the eyes of TTP. If user U sends revocation token r directly to TTP via an anonymous channel (in Step 6), TTP may obtain additional $anonch_U \in \mathcal{U}$ information on the identity of U. Random variable $anonch_U$ can be viewed as a decision of TTP regarding the identity of user U. This decision is made based on various fragments of information TTP can collect from the network protocols that constitute the anonymous channel used. As r is generated independent from id_U, random variables $anonch_U$ and r are independent. If the anonymous channel is perfect, the distribution of random variable $anonch_U$ (i.e. the sender anonymity probability distribution) is uniform. The paradigm of using the distribution (and the entropy) of a random variable (that expresses the attacker's knowledge on the identity of the user) for measuring the user's anonymity in a system was introduced by Serjantov and Danezis [15] who also showed examples for calculating the distribution of $anonch_U$ in some specific systems.

Let's consider the datablocks that appear during a protocol run. Message m is a random variable that may contain information linkable with id_U We assume, TTP perceives that messages have a uniform distribution. Condition c consists of a fixed string constant and a deadline t, which is not a random variable but a constant in the eyes of TTP, so condition c cannot be linked with id_U. Datablocks r and k are one-time random numbers of uniform distribution. Digital signature $sig_U(m)$ is a possibly randomized transformation of m. Using the public key of a certain user, it is possible to check if a signature was calculated by that particular user. Thus, $sig_U(m)$ can be linked with id_U. We assume that without having the corresponding secret key k, it is not possible to link datablock $E_k[sig_U(m)]$ with the identity of U, so we assume, the distribution of $E_k[sig_U(m)]$ is uniform in the eyes of TTP. As neither datablocks r, k and c nor the private key of TTP can be linked with U, datablock $E_{TTP}(r, k, c)$ cannot be linked with the user.

Let $H(X)$ denote the entropy of random variable X and let $I(X, Y)$ denote the mutual information between random variables X and Y. By σ we denote the set of datablocks TTP receives during a protocol run, and by ω we denote the set of datablocks user U needs to conceal with respect to TTP. In our case:

$$\omega = (id_U, m, sig_U(m))$$

Theorem 1. *In Protocol 3 U to retains the following degree of privacy with respect to TTP:*

(a) If all parties behave honestly, user U has unconditional privacy. Formally: $I(\sigma_a; \omega) = 0$, where $\sigma_a = (r, k, c)$.

(b) If user U decides to revoke the signature, she has the degree of privacy provided by the anony-

mous channel. Formally: $I(\sigma_b;\omega) = I(anonch_U;\omega)$*, where* $\sigma_b = (\sigma_a, anonch_U)$*.*

Proof:

(a) If all parties behave honestly, Step 6 of the protocol is not executed, so the only message TTP receives is $E_{TTP}(r,k,c)$ in Step 7. Based on this message, TTP can compute (r,k,c). Thus, $\sigma_a = (r,k,c)$, so $I(\sigma_a,\omega) = 0$, because σ_a and ω are generated independently.

(b) If Step 6 is executed, TTP receives r via the anonymous channel. TTP already knows r, so $\sigma_b = (\sigma_a, anonch_U)$.

$$I(\sigma_b;\omega) = I(\sigma_a, anonch_U;\ \omega) = H(\omega) - H(\omega|\sigma_a,\ anonch_U)$$

Since $I(\sigma_a;\ \omega, anonch_U) = 0$,

$$H(\omega) - H(\omega|\sigma_a,\ anonch_U) = H(\omega) - H(\omega|anonch_U) = I(anonch_U;\omega). \quad \blacksquare$$

6 Conclusion

In this paper we proposed a framework for the protection of mobile users from untrusted terminals. Our framework allows the user to generate conditional digital signatures at untrusted terminals, to review signed documents form a trusted terminal before a certain deadline, and to revoke unintended signatures. We showed that in case of practical protocols, the user has to rely on a trusted third party. Therefore, we enhanced our protocol for signature revocation to allow the user to retain her privacy towards this trusted third party.

The conditional signature is the finalization of the user's offer towards the recipient. After the message is protected by a conditional signature, neither side can alter it, but both sides can withdraw from it. According to the paradigm of Rivest, the conditional signature is not a non-repudiable proof, but it can be used as plausible evidence (e.g. for proving that the user was present and she initiated a transaction with her card). However, if the user does not revoke the signature until a certain deadline, she cannot do anything about the validity of her signature, and it can be considered as a regular, non-repudiable signature.

References

[1] B. Schneier and A. Shostack, “Breaking up is Hard to do: Modelling security threats for smart cards.” USENIX Workshop on Smart Card Technology, Chicago, Illinois, USA, 1999.

[2] N. Asokan, H. Debar, M. Steiner, and M. Waidner, “Authenticating Public Terminals.” Computer Networks, 1999.

[3] R. Rivest, “Issues in Cryptography.” Computers, Freedom, Privacy 2001 Conference `http://theory.lcs.mit.edu/~rivest/Rivest-IssuesInCryptography.pdf`, 2001.

[4] I. Berta, L. Buttyán, and I. Vajda, “Mitigating the Untrusted Terminal Problem Using Conditional Signatures.” Proceedings of International Conference on Information Technology ITCC 2004, IEEE, Las Vegas, NV, USA, April, 2004.

[5] I. Z. Berta, L. Buttyán, and I. Vajda, “Privacy protecting protocols for revokable signatures.” Cardis2004, Toulouse, France, 2004.

[6] M. Abadi, M. Burrows, C. Kaufman, and B. Lampson, “Authentication and Delegation with Smart-cards.” Theoretical Aspects of Computer Software: Proc. of the International Conference TACS’91, Springer, Berlin, Heidelberg, 1992.

[7] D. Clarke, B. Gassend, T. Kotwal, M. Burnside, M. v. Dijk, S. Devadas, and R. Rivest, “The Untrusted Computer Problem and Camera-Based Authentication,” 2002.

[8] T. Stabell-Kulo, R. Arild, and P. Myrvang, “Providing Authentication to Messages Signed with a Smart Card in Hostile Environments.” Usenix Workshop on Smart Card Technology, Chicago, Illinois, USA, May 10-11, 1999., 1999.

[9] I. Z. Berta and I. Vajda, “Documents from Malicious Terminals.” SPIE Microtechnologies for the New Millenium 2003, Bioengineered and Bioinspired Systems, Spain, 2003.

[10] D. Chaum, “Untraceable electronic mail, return addresses and digital pseudonyms.” Communications of the ACM, v24, n.2 pp.84-88, 1981.

[11] S. Brands, “Untraceable off-line cash in wallets with observers.” In Crypto’93 Springer-Verlag, LNCS 773 pp. 302-318, 1994.

[12] M. Franklin and M. Yung, “Towards provably secure efficient electronic cash.” Columbia Univ. Dept. of CS TR CSUCS-018-92, 1992.

[13] P. Syverson, D. Goldschlag, and M. Reed, “Anonymous Connections and Onion Routing.” IEEE Symposium on Security and Privacy, Oakland, California, 1997.

[14] B. Lee and K. Kim, “Fair Exchange of Digital Signatures using Conditional Signature.” SCIS 2002, Symposium on Cryptography and Information Security, 2002.

[15] A. Serjantov and G. Danezis, “Towards an Information Theoretic Metric for Anonymity.” Privacy Enhancing Technologies, PET2002, 2002.

[16] R. Anderson and M. Kuhn, “Low Cost Attacks on Tamper Resistant Devices.” M Lomas et al. (ed.), Security Protocols, 5th International Workshop, Paris, Proceedings, Springer LNCS 1361, pp 125-136, ISBN 3-540-64040-1., 1997.

[17] W. Fung, M. Golin, and J. Gray, III., “Protection of Keys against Modification Attack.” HKUST Theoretical Computer Science Center Research Report 2001-04.

[18] W. Rankl and W. Effing, “Smart Card Handbook.” John Wiley & Sons, 3rd edition, 2003.