Cost-effective Payment Schemes With Privacy Regulation

[Published in K. Kim and T. Matsumoto, Eds., *Advances in Cryptology – ASIACRYPT '96*, vol. 1163 of *Lecture Notes in Computer Science*, pp. 266–275, Springer-Verlag, 1996.]

David M'Raïhi

Gemplus - Cryptography Department
1 place de la Méditerrannée, B.P. 636, F-95206 Sarcelles cedex, France
100145.2261@compuserve.com

Abstract. In this paper, we introduce a new electronic money methodology: sub-contracting the blinding to a trustee and using an Identity-based piece of information to achieve provable privacy and security. This variation on the Brickel, Gemmel and Kravitz paradigm [2] offers protection against various attacks minimizing user's computational requirement. Furthermore, our scheme offers various complexity/privacy trade-offs without endangering the issuer's overall security.

1 Introduction

Blind signatures [4] allow a message to be signed without revealing to the signer any information on the message. Therefore, this cryptographic primitive offers interesting properties for designing anonymous electronic cash schemes. In [5], Chaum, Fiat and Naor proposed the first practical electronic cash system, using the blind signature paradigm to provide privacy and security for all involved parties. Then in [1], Brands exhibited constructions based on the representation problem to increase efficiency (avoiding cut-&-choose) without giving up the main aspect of a real cash system, i.e. the honest payer privacy. Ferguson [7] proposed a scheme based on the same concept, providing also tracability after the fact.

Nevertheless, perfect anonymity obtained through the usage of blind signature could open the way to a class of attacks pointed out by von Solm and Naccache [14]. A new concept involving a trustee during withdrawal was introduced by Brickell, Gemmell and Kravitz in an escrow-like e-cash scheme. Stadler, Camenish and Piveteau proposed*fair* blind signatures as a design primitive [15] that enables the bank to remove anonymity with the help of the trustee and put it into the context of e-cash [3]. However, these schemes assume the user's ability to fast computation and do not provide any protection against certain type of active attacks against the bank.

More recently [9], Jakobsson and Yung introduced the notion of Ombudsman (a government official in charge of the customers defense against abuses) yielding an efficient electronic money system where tracing does not only depend on the bank but requires the combined endeavors of the bank and the Ombudsman in the tracing process.

In this paper we propose an alternative model where blinding is sub-contracted to trustee(s), using Identity-linked *pseudonyms*, denoted PIDs. The core ideas are:

1. compel the user $\mathcal{U}$ to register at his bank acting as a Certification Authority $\mathcal{CA}$ to obtain n public values PIDs. These PIDs are strongly linked to $\mathcal{U}$'s identity by a secret shared between $\mathcal{U}$ and $\mathcal{CA}$; without this certified information, $\mathcal{U}$ cannot initiate any communication with a trustee and therefore withdraw any electronic coin; this function is comparable to the registration at the Judge in [3] but with reduced communication and computation, simplifying also account management at the bank;
2. enable $\mathcal{U}$ to get a valid anonymous coin from the bank through a trusted party designed as Blinding Office $\mathcal{BO}$ for he performs a blinding signature protocol with $\mathcal{U}$'s bank after being convinced that $\mathcal{U}$ has previously registered at the bank; Delegate Blinding protocol enables therefore the user to get a certificate from $\mathcal{CA}$ on a value blinded by $\mathcal{BO}$; we may assume that there are different BOs, located in various places like currency exchange offices nowadays, where the users can withdraw anonymously e-cash from their accounts,
3. the privacy level is directly related to the amount of trust the user put in the $\mathcal{BO}$; furthermore, $\mathcal{BO}$ can link all payments performed with coins related to a given PID_i: privacy is regulated in n, i.e. $\mathcal{U}$'s privacy relies on his number of PIDs and therefore on his storage and computational capacity.

The new model is somewhat different from the Ombudsman concept since $\mathcal{U}$ must trust both $\mathcal{CA}$ and $\mathcal{BO}$. However, $\mathcal{CA}$ and $\mathcal{BO}$ do not need to communicate to each other during withdrawal: separating $\mathcal{CA}$ and $\mathcal{BO}$ should avoid collusions to infer private information on users. Jakobsson and Yung considered also a very strong attack on the system, the so-called bank robbery where an attacker obtains the secret keys of the bank, and showed how to prevent against this attack. The previous trustee-based schemes do not protect the bank against this kind of attack and in order to boost efficiency, neither do we in the scheme described in this paper. Nevertheless, we can observe that since $\mathcal{BO}$ delivers all the honestly withdrawn coins, it should be possible to introduce a double-verification of the coins by calling the $\mathcal{BO}$ during payment if a bank robbery has happened. $\mathcal{BO}$ must also stop immediately accepting withdrawal demands and protocols should be added in order to update securely $\mathcal{CA}$'s public keys at $\mathcal{BO}$s and refund honest users who still have coins from previous withdrawals. We decide therefore to prevent only against the attacks also considered in Brickell et al. and Stadler et al. schemes to achieve high performance saving communication and computation time.

In section 3, the usage of a PID is described. A protocol designed for subcontracting the blinding of public values is presented in section 4 and the delegation of the computation of any public information is developed in section 5. Finally, section 6 presents the implementation of an e-cash system providing privacy and security levels that depend on the user's motivation and section 7 sketches the proofs of the main security aspects of the scheme.

2 Notation

The following notation is used throughout this paper:

Symbol	*Definition*
α	primitive element of $\mathbb{Z}_p^*$, p prime
h	one-way hash function, in practice SHA can be used
f	one-way function such as exponentiation in a finite field
$\{E_{\mathcal{BO}}, D_{\mathcal{BO}}\}$	public-key encryption scheme of $\mathcal{BO}$ typically [12]
$\{S_{\mathcal{CA}}, V_{\mathcal{CA}}\}$	public-key signature scheme of $\mathcal{CA}$
$\{S_{\mathcal{U}}, V_{\mathcal{U}}\}$	public-key signature scheme with pre-processing of $\mathcal{U}$
$\{y, \hat{y}\}$	public information where $\hat{y}$ is obtained after blinding $y = f(x)$: $\hat{y} = V_{\mathcal{CA}}(r) \cdot y \Rightarrow S_{\mathcal{CA}}(\hat{y}) = S_{\mathcal{CA}}(V_{\mathcal{CA}}(r) \cdot y) = r \cdot S_{\mathcal{CA}}(y)$ with x secret and r random number known by the receiver only

We propose to use RSA for $\mathcal{CA}$'s scheme and assume that inverting the signature (or deciphering) amounts to querying a random oracle. For the user's signature scheme, Schnorr [11] seems the most suitable and secure considering recent analysis [13].

3 Model of Trust

Trust is a key concept in the system and we define hereafter the assumptions:

1. $\mathcal{U}$ trusts $\mathcal{BO}$ and $\mathcal{CA}$ not to collude in order to trace payments without a court order (or a similar legal authorization)
2. $\mathcal{U}$ trusts $\mathcal{BO}$ not to disclose information closely related to his privacy
3. $\mathcal{U}$ trusts $\mathcal{BO}$ and $\mathcal{CA}$ not to impersonate him
4. $\mathcal{CA}$ trusts $\mathcal{BO}$ to deliver information required in a tracing process

4 Pseudo-ID

The Delegate Blinding protocol simply splits the blinding ability in order to transfer from $\mathcal{CA}$ to $\mathcal{BO}$ the capability to link y to ID. To achieve privacy, since $\mathcal{BO}$ has all the public values and can infer connections between a certain y and a user ID, we must introduce a new notion of pseudo-identity, denoted PID.

The basic idea consists in performing a Diffie-Hellman [6] key exchange between $\mathcal{U}$ and $\mathcal{CA}$. $\mathcal{CA}$ will store $\mathcal{U}$'s ID and the pseudo-identities (or enough data to rebuild them upon request).

Definition 1. *$\mathcal{U}$ is pseudo-identified by an authentication session using a secret embedded in PID and therefore linked to ID.*

4.1 Pseudo-ID Generation

First, $\mathcal{U}$ presents a "physical" ID-proof to $\mathcal{CA}$; we may assume that $\mathcal{U}$ is physically present at $\mathcal{CA}$'s premises and provides a material proof such as a passport (when opening the account, for instance). $\mathcal{CA}$ delivers a certified token to enable $\mathcal{U}$ to perform pseudo identifications. $\mathcal{CA}$ generate PIDs by doing the following:

1. $\mathcal{CA}$ and $\mathcal{U}$ agree upon a Diffie-Hellman secret $s = \alpha^{ab} \bmod p$
2. $\mathcal{CA}$ sends $S_{\mathcal{CA}}(h(s))$ and stores $\{ID, s\}$
3. $\mathcal{U}$ builds $PID = E_{\mathcal{BO}}(s)|S_{\mathcal{CA}}(h(s))$ and stores $\{s, PID\}$

For a multi-pseudo generation, $\mathcal{CA}$ simply generates (in step 1) several b_is and computes the corresponding $\alpha^{b_i} \bmod p$. $\mathcal{CA}$ can expand all these secret values from a random seed $b_i = h(seed, i)$.

In this case, $\mathcal{CA}$ stores only $\{ID, \alpha^a \bmod p, seed, n\}$ where n is the number of PIDs generated. $\mathcal{U}$ will therefore perform $n + 1$ computations to get the n secrets $s_i = \alpha^{ab_i} \bmod p$. If $\mathcal{U}$ prefers to perform n different sessions, $\mathcal{CA}$ must store n tuples. $\mathcal{U}$ must then perform $2n$ computations.

We can observe that $\mathcal{U}$ may store s only and build the PIDs dynamically, depending on the $\mathcal{BO}$ where operations are performed.

4.2 Pseudo Identification

$\mathcal{U}$ sends a PID to $\mathcal{BO}$ who can decipher $E_{\mathcal{BO}}(s)$, verify the correctness of the signature on $h(s)$ and starts an authentication session involving the secret s shared by $\mathcal{U}$ and $\mathcal{BO}$.

5 Delegating the Blinding Phase

In this section, we describe how to delegate the tracing facility of a public information y from $\mathcal{CA}$ to a $\mathcal{BO}$. $\mathcal{U}$ gets a signature $\sigma = S_{\mathcal{CA}}(y)$ from $\mathcal{CA}$, without revealing his identity, by carrying-out the following protocol:

- Blinding

1. $\mathcal{U}$ pseudo-identifies himself and sends y to $\mathcal{BO}$
2. $\mathcal{BO}$ generates a random r, computes $\hat{y}$ and sends $\hat{y}$ to $\mathcal{U}$
3. $\mathcal{U}$ identifies himself to $\mathcal{CA}$ and sends $\hat{y}$ with the $\mathcal{BO}$ references (for correct encryption)

4. $\mathcal{CA}$ replies to $\mathcal{U}$ with $e = E_{\mathcal{BO}}(S_{\mathcal{CA}}(\hat{y}))$

• Unblinding

1. $\mathcal{U}$ sends e to $\mathcal{BO}$
2. $\mathcal{BO}$ decrypts e and (knowing y's blinding factor) unblinds $\sigma = S_{\mathcal{CA}}(y)$
3. $\mathcal{BO}$ sends σ to $\mathcal{U}$

6 Sub-contracting y Computation

In [10], Naccache et al. proposed a method to delegate the computation of rs required to generate DSA [8] signatures by sharing a common secret with a trusted Authority. From this secret, the trustee can pre-compute "coupons" which can be used to generate DSA signatures, saving time and effort.

In the following, we generalize this idea to the generation of any set of public pieces of information $y_i = f(x_i)$, where $x_i = h(x, i)$ and x is some random seed. The idea is to enable the pre-computation of a set of public values (coupons or public keys) to be further used for the generation of DLP-based signatures.

6.1 Protocol

1. $\mathcal{U}$ generates a random seed x and sends $e = E_{BO}(x)$ with the number of public values n to be generated
2. $\mathcal{BO}$ decrypts e and computes $x_i = h(x, i)$ and the corresponding public values $y_i = f(x_i)$
3. $\mathcal{BO}$ sends the set $\{y_i\}_{i \leq n}$ and a certificate $c = h(\{y_i\}, x)$ so that $\mathcal{U}$ can check the set's authenticity

6.2 Security Analysis

Clearly, an eavesdropper cannot use any y_i since he does not know the corresponding x_i. He must either:

1. Break E and extract x, which is infeasible given D's nature, or
2. Find collisions in h to generate $\tilde{x} \neq x_i$ such that $y_i = f(\tilde{x})$, (also assumed infeasible), or
3. Invert f, which is assumed one-way

$\mathcal{U}$ must of course trust $\mathcal{BO}$ not to impersonate him. This solution is well suited for smart-card applications where the device at one's hand has a relatively limited computation capacity. In this case, the user can entrust the person of his choice to sub-contract his privacy-related computations.

7 Electronic Money with Privacy Regulation

We will consider two settings, beginning with the usual one in which electronic money is implemented on low-cost smart-card. The central idea is that the computational effort required from the user depends on the privacy level he wants to achieve. We will assume hereafter that the user's signature scheme is a DLP-based scheme. $\mathcal{U}$'s public key is $y = g^x \bmod p$, where p is prime and $g \in \mathbb{Z}_p^*$ has order q, a prime divisor of $p-1$.

7.1 Low Privacy

A coin is a tuple $\{y, data, \sigma\}$ where $\sigma = S_{\mathcal{CA}}(h(y|data))$ and data is any relevant information related to the coin (such as date of validity) added by $\mathcal{CA}$. $\mathcal{U}$ has a small set of public keys y certified through protocol 4.1. The idea is that payments are linkable by category (in other words $\mathcal{U}$ may be traced for some purchases related to a certain y_i but others made with y_j will not be related to $\mathcal{U}$). The maximal number of such categories is simply n. We will denote by c_i the category related to PID_i.

Definition 2. *A user holds an n-privacy or regulated privacy (in n) when his purchases are only linkable inside a category and $\mathcal{CA}$ cannot trace alone transactions outside this category*

Registration at the $\mathcal{CA}$: $\mathcal{U}$ visits $\mathcal{CA}$ and obtains n pseudonyms. This implies the computation of $n+1$ exponentiations (or $2n$ if U prefers to perform several sessions rather than one multi-session); considering [13] size of parameters and [7] techniques, such an interaction will require about $50(n+1)$ (respectively $100n$) multiplications, which is acceptable if performed once, for a reasonable value of n (say, 4 or 5).

Withdrawal: $\mathcal{U}$ interacts with a $\mathcal{BO}$ in this way:

1. $\mathcal{U}$ pseudo-identifies himself by presenting a PID of his choice
2. $\mathcal{BO}$ checks PID
3. $\mathcal{U}$ sends $\{E_{\mathcal{BO}}(x), v\}$, x random and v the number of values to be generated
4. $\mathcal{BO}$ generates a public key $y = f(h(x,0))$ and a set of pre-computed values $\left\{r_i = g^{h(x,i)}\right\}_{1 \le i \le v}$ and sends them back to $\mathcal{U}$ with a certificate c
5. $\mathcal{U}$ checks c and, if correct, starts with $\mathcal{BO}$ a Delegate Blinding[1] of y and receive $\{\sigma, data\}$, where σ is $\mathcal{CA}$'s signature on $h(y|data)$
6. $\mathcal{U}$ checks σ and stores $\{x, y, data, \{r_i\}, \sigma\}$

We observe that $\{y, data, \sigma\}$ is multi-spendable and so, it could be suitable to offer the possibility to load only rs in case y is not completely spent after v transactions. Obviously, y and $\{r_i\}$ may be completely dissociated by sending two different encrypted randoms x_1 and x_2.

[1] when $\mathcal{CA}$ sends e, he subtracts the value of the coin from $\mathcal{U}$'s account; a possible extension would be that $\mathcal{CA}$ uses different secrets corresponding to different coin values (which implies adding a tag to y).

Payment: Payment is achieved in only one multiplication for [13] (two for DSA) and a hashing by $\mathcal{U}$. The shop $\mathcal{SH}$ computational effort is about 2 exponentiations which is acceptable considering that his calculation ressources are greater that user's ones. The protocol can also support the challenge semantics introduced by Jakobson and Yung and offer extensions of the system by letting part of the challenge indicates payment-related information.

1. $\mathcal{U}$ sends a coin $\{y, data, \sigma\}$
2. $\mathcal{SH}$ checks σ and sends a message m including the amount, a random challenge and (possibly) some other application data
3. $\mathcal{U}$ generates $a = S_U(y, m)$, using a pre-computed r_i
4. $\mathcal{SH}$ makes sure that the coin has not expired and $V_{\mathcal{U}}(a, y, m) = \texttt{True}$

Deposit: $\mathcal{SH}$ sends the transactions corresponding to coins spent during a given. $\mathcal{CA}$ checks coin s correctness and performs double-deposit and overspending detections.

1. $\mathcal{SH}$ sends the transaction log $t = \{y, \sigma, data, a, m\}$ to $\mathcal{CA}$
2. $\mathcal{CA}$ checks:
   - $V_{\mathcal{CA}}(\sigma, y) = V_{\mathcal{U}}(a, y, m) = \texttt{True}$
   - the coin is still valid
   - t has not been deposited already
3. $\mathcal{CA}$ accepts the transaction and credits $\mathcal{SH}$'s account

Furthermore, $\mathcal{CA}$ will check in the already-deposited coin list if the coin was overspent. If a certain y_i was double-spent, $\mathcal{CA}$ will ask for a tracing procedure[2] which consists in $\mathcal{CA}$ and $\mathcal{BO}$ joining forces to build a link between y_i and the coin-spender's ID.

7.2 High Privacy

Privacy can be regulated according to the user's motivation by increasing the number of PIDs to an optimum which is one per payment. This implies in practice a high-storage device such as a PCMCIA card or an electronic wallet. In this case, the withdrawal will only consist in a pseudo-identification followed by the delegate blinding of y.

8 Security

We will sketch the proofs of some security aspects of the new system.

[2] this should include Judge's control to avoid unauthorized privacy disclosure.

8.1 Pseudo-Identification

An observer cannot get any information on s unless he breaks the Diffie-Hellman protocol. We can state the following lemmas:

Lemma 1. *Only $\mathcal{U}$ and $\mathcal{CA}$ know s.*

Proof (Sketch). Follows directly from the Diffie-Hellman assumption. □

Lemma 2. *Assuming that $\mathcal{CA}$ follows honestly protocol 4.1, it is always possible to relate PID to ID.*

Proof (Sketch). From Lemma 1, and assuming that $\mathcal{CA}$ will not try to impersonate $\mathcal{U}$, only $\mathcal{U}$ will be able to present a PID_i corresponding to a single s_i. $\mathcal{CA}$ can retrieve $\mathcal{U}$'s ID from this *Pseudo-ID* by querying his database (or re-computing the different s_i in case of storage optimization). □

Theorem 1. *Only $\mathcal{U}$ can pseudo-identify himself to $\mathcal{BO}$ using one of his PID_i*

Proof (Sketch). $\mathcal{U}$ sends a PID_i to $\mathcal{BO}$ who deciphers it to get a s_i. $\mathcal{BO}$ verifies $S_{\mathcal{CA}}(h(s_i))$ to make sure that $\mathcal{CA}$ previously stored a tuple related to this PID_i. From Lemma 1 and 2, it follows that $\mathcal{BO}$ can authenticate $\mathcal{U}$ using the secret s_i and therefore build a link to his identity ID. □

8.2 Delegate Blinding

As stated by Theorem 2 the main feature of this protocol is the impossibility for the user to get a signature on y without the help of $\mathcal{BO}$. We will sketch the proof hereafter:

Theorem 2. *$\mathcal{U}$ cannot obtain a valid signature on y without $\mathcal{BO}$'s help.* □

Proof (Sketch). $\mathcal{U}$ may try not to interact with $\mathcal{BO}$ and generate a couple $\{y, \hat{y}\}$ of his own by computing $\hat{y} = V_{\mathcal{CA}}(r) \cdot y$, where r is a random generated by $\mathcal{U}$. He must nevertheless send $\hat{y}$ to $\mathcal{CA}$ who replies with $e = E_{\mathcal{BO}}(\hat{y})$; then prior to unblinding $S_{\mathcal{CA}}(\hat{y})$, which is possible since $\mathcal{U}$ knows the blinding factor r, $\mathcal{U}$ must invert $\mathcal{BO}$'s encryption scheme (E) which is assumed infeasible. □

We can also observe that if $\mathcal{U}$ colludes with a $\mathcal{BO}$, $\mathcal{CA}$ can infer the $\mathcal{BO}$'s identity since if the user sends bad $\mathcal{BO}$'s references, decryption of e is not possible anymore.

8.3 Payment Scheme

Lemma 3. *The proposed scheme achieves overspending detection.*

Proof (Sketch). Would a set of transactions t_i, corresponding to a single y, exceed the coin-value, $\mathcal{CA}$ can assume overspending because of user's signature scheme property (only $\mathcal{U}$ can sign new messages). $\mathcal{CA}$ can ask $\mathcal{BO}$ to trace $\mathcal{U}$ (the control by a judge in the course of a legal procedure might be added). $\mathcal{BO}$ will double-check t_i and that the total amount exceeds the coin-value and if confirmed, will reveal PID. □

Theorem 3. *The proposed scheme achieved regulated privacy w. r. t. to $\mathcal{BO}$'s honesty in the tracing process.*

Proof (Sketch). Assuming a transaction $t = \{y, \sigma, a, m\}$ deposited at $\mathcal{CA}$; $\mathcal{CA}$ saw only $\hat{y}$ during withdrawal. Consequently, he cannot link the transaction to the ID received at this time. $\mathcal{BO}$ can only link t to the PID_i used by $\mathcal{U}$ during withdrawal. In case of overspending, $\mathcal{BO}$ and $\mathcal{CA}$ can join forces and link t to the ID which corresponds to the faulty PID_i. They can also link all payments related to this PID_i (in other words, $\{t_j\} \in c_i$). But $\mathcal{CA}$ cannot get any information on other transactions related to the other PIDs since $\mathcal{BO}$ helps to trace only the transaction in c_i. Furthermore, $\mathcal{CA}$ will only be able to link transactions related to a given y. From that, he cannot trace a honest user without the help of $\mathcal{BO}$ but only suspect that some user performed transactions in a certain category c_i. Eventually, it follows from Lemma 3 that $\mathcal{CA}$ cannot falsely engage $\mathcal{BO}$ in a tracing procedure. □

Finally, let us underline that $\mathcal{U}$ and/or $\mathcal{BO}$ cannot create electronic money without breaking $\mathcal{CA}$'s signature scheme.

9 Conclusion

This paper presented a new simple and efficient payment scheme which combines the usage of a pseudonym, strongly linked to user's identity, with the delegation of public-key blinding. The scheme enables a user to trade-off privacy against computational complexity.

Furthermore, by introducing a user-representative, which is fully trusted by the user, we can provide a possible direction for low-cost device oriented applications, such as simple smart-cards, where privacy-level relies on the user-only decision.

10 Acknowledgments

It is a pleasure to thank Jacques Stern for advice and helpful comments during this research work, Markus Jakobson for valuable comments and suggestions and an anonymous referee for useful remarks.

References

[1] S. Brands, "Untraceable Off-line Cash in Wallet with Observers", Advances in Cryptology - CRYPTO '93, LNCS **773**, pp. 302-318.

[2] E. Brickell, P. Gemmel and D. Kravitz, "Trustee-based Tracing Extensions to Anonymous Cash and the Making of Anonimous Change", Proceedings of 6th. annual Symposium on Discrete Algorithm (SODA), 1995, ACM Press, pp. 457-466.

[3] J. Cammenish, J-M. Piveteau and M. Stadler, "An Efficient Fair Payment System", Third ACM Conference on Computer and Communications Security, 1996, ACM Press, pp. 88-94.

[4] D. Chaum, "Blind Signatures for Untraceable Payments", Advances in Cryptology - Proceedings of Crypto 82, Plenum, NY, pp. 199-203.

[5] D. Chaum, A. Fiat and M. Naor, "Untraceable Electronic Cash", Advances in Cryptology - CRYPTO '88, LNCS **403**, pp. 318-327.

[6] W. Diffie and M. Hellman, "New Directions in Cryptography", IEEE Tans. Info. Theory IT-**22**, Nov. 1976, pp. 644-654.

[7] N. Ferguson, "Extensions of Single Term Coins", Advances in Cryptology - CRYPTO '93, LNCS **773**, pp. 292-301.

[8] FIPS PUB 186, February 1, 1993, Digital Signature Standard.

[9] M. Jakobsson and M. Yung, "Revokable and Versatile Electronic Money", Third ACM Conference on Computer and Communications Security, 1996, ACM Press, pp. 76-87.

[10] D. Naccache, D. M'Raïhi, S. Vaudenay and D. Raphaeli, "Can DSA be improved ? - Complexity trade-offs with the Digital Signature Standard", Advances in Cryptology - EUROCRYPT '94, LNCS **950**, pp. 77-85.

[11] D. Pointcheval, J. Stern, "Security Proofs for Signature Schemes", Advances in Cryptology - EUROCRYPT '96, LNCS **1070** pp. 387-398.

[12] R. Rivest, A. Shamir and L. Adleman, "A method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, v. **21**, n. 2, Feb 1978, pp. 120-126.

[13] C. Schnorr, "Efficient identification and signatures for smart-cards", Advances in Cryptology - EUROCRYPT '89, LNCS **765**, pp. 435-439.

[14] S. von Solms and D. Naccache, "On Blind Signatures and Perfect Crimes", Computers and Security, **11** (1992) pp. 581-583.

[15] M. Stadler, J-M. Piveteau and J. Cammenish, "Fair Blind Signatures", Advances in Cryptology - EUROCRYPT '95, LNCS **921**, pp. 209-219.