Biometric yet Privacy Protecting Person Authentication

Gerrit Bleumer [*]

AT&T Labs-Research Florham Park, NJ 07932, USA
bleumer@acm.org

Abstract. Many eligibility or entitlement certificates in every day life are non-transferable between persons. However, they are usually implemented by personal physical tokens that owners can easily pass around (e.g. credit card), driver's license). So there must either be negligible incentives to pass these certificates or the tokens around, or the tokens must allow to authenticate the persons who show certificates, e.g., by imprinted photographs. However, any kind of easily accessible personal identifying information threatens the owners' privacy. To solve these somehow paradoxical requirements, we assume for each owner a kind of pilot that is equipped with a tamper resistant biometric authentication facility. We draft cryptographic protocols for issuing and showing non-transferable yet privacy protecting certificates. Unforgeability of certificates relies on a well-established computational assumption, non-transferability relies upon a physical assumption and owners' privacy is protected unconditionally.

Keywords: Non-transferable certificates, Wallets-with-observer, Blind Signatures, Interactive proofs, Biometric person authentication.

1 Introduction

Many eligibilities or entitlements in every day life are bound to a person and are not intended to be transferred between persons, e.g., room/building/account access rights, driver's licenses, academic degrees, certain drug eligibilities, stock options. It is certainly desirable if not inevitable in an information society to implement such non-transferable certificates in an electronic way and many such solutions exist already, e.g. id badges, credit cards, insurance certificates, membership cards. These solutions work well if the incentives to give away, lend or sell certificates are sufficiently outweighed by the disadvantages for the respective owners. An instructive example where this is not the case are driver's licenses. Suppose electronic driver's licenses and (unmanned) electronic road checkpoints that can verify these driver's licenses. In such a scenario, lending one's driver's license to someone else would bear no disadvantage other than not being able to

[*] The research for this paper was supported by the German Research Foundation under grant DFG-Bi 311/9-1.

David Aucsmith (Ed.): Information Hiding 1998, LNCS 1525, pp. 99–110, 1998.

drive oneself at the same time. The problem is that personal information (passwords, PINs) and personal tokens (id badge, magnetic card) as such can easily be shared, lended or can even be traded. The only way to prevent this is *biometric person authentication*. In the road example, a straighforward solution is to equip the road checkpoints with video cameras peeking at the drivers. Not only would this solution render the electronic driver's licenses unnessasary, but it is almost certainly unacceptable from a privacy point of view. A smarter solution though is to equip drivers with personal devices into which biometric verification modules are implanted such that (i) drivers cannot deceive their biometric facility and (ii) road checkpoints cannot access the biometric modules (and their memory) directly, i.e., without the help of the respective owner's personal device.

In the following, we explore the latter solution. We assume each driver is equipped with a personal communication device (called *wallet* for historical reasons) that can run a trusted local process (called *observer*) [CP92]. For example, wallets and observers could be implemented by palmtops or pilots with a built-in tamper resistant chip [PPSW97]. Drivers need to trust their observers to support all legitimate operations, whereas road checkpoints need to trust observers to prevent any illegitimate operation. In order to achieve privacy, there must be no *outflow* of information from the observer to a road checkpoint. If loss or theft of observers is to be tolerated, there must also be no *inflow* of information from a driver's license issuing organization, typically the Motor Vehicle Services (MVS), to the observer. Preventing outflow and inflow requires that the observer has no communication link other than to its hosting wallet, and all communication protocols must prevent outflow from the observer (or inflow to the observer). This concept has been introduced by Chaum and Pedersen [CP92] as the *wallet with observer* architecture. Adding a biometric authentication facility to the observer has not been studied for this architecture before (Fig. 1).

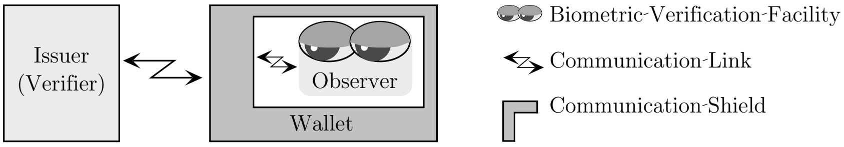

Fig. 1. Wallet with observer architecture

Each observer needs to be personalized to its respective owner in a way that is trusted by the MVS. Afterwards, they only need to verify their owner's biometric identity with sufficiently small false acceptance and false rejection rates. In principle, there are two kinds of biometric authentication. *Biometric verification* facilities need to distinguish only one particular identity from all others, whereas *biometric recognition* facilities need to distinguish a (large) number of biometric identities from each other. Recently, there is increasing scientific and commercial interest in biometric person authentication [M94,BCB97]. A growing number of easy-to-use computer peripherals is available in the consumer market [Bio]. Most of them are fingerprint recognizers with false acceptance and false rejection rates in the range of $10^{-4} \ldots 10^{-6}$ and $1 \times 10^{-2} \ldots 3 \times 10^{-2}$, respectively. The crossover error rate is typically 1–3%, but higher for elderly people and manual workers. Recognition time is typically $0.5 \ldots 1.0$ seconds. Since biometric verification is technically easier to achieve, it appears to become feasible even for small observers (and wallets) in the not too far future.

Electronic certificates can be built into wallets right at the end of the production line (*static issuing*). This is efficient for mono-purpose wallets e.g., id badges. Multi-purpose wallets, however, must allow to obtain new certificates during their entire lifetime and must also take care of dependencies among certificates (*dynamic issuing*). The concept of privacy oriented and dynamically issuable certificates is what Chaum has introduced as *credentials* [C83,C84,C85,C90,C92]. In the following, we elaborate on the additional aspect of non-transferability. We propose a cryptographic implementation of driver's licenses based on the wallet with observer architecture, where the observer has implanted some tamper resistant biometric verification facility.

2 Cryptographic Primitives

We first introduce some notation. Then we present three cryptographic primitives, two kinds of blind signature and an interactive proof of knowledge.

2.1 Notation

Since the following protocols are based on the intractability of computing discrete representations, the following definitions are useful. Let p be a k-bit prime ($k \in \mathbb{N}$), q be a large prime divisor of $p - 1$, i.e., $q \log^2 q \geq p$, and G_q be the unique subgroup of order q in $\mathbb{Z}_p^*$. Furthermore, let $g_1, \ldots, g_l$ denote elements chosen uniformly at random from $G_q \setminus \{1\}$. (Each g_i generates G_q). It is generally assumed that a discrete representation $(\alpha_1, \ldots, \alpha_l)$ of a randomly chosen element $z \in G_q$ with respect to $(g_1, \ldots, g_l)$, i.e., $z = \prod_{i=1}^{l} g_i^{\alpha_i}$, is infeasible to compute in polynomial time (in the bitlength of p). We call this the *subgroup representation assumption* (SR). In the special case of $l = 1$, the discrete representation α_1 is called the *discrete logarithm* of z with respect to g_1, and the SR assumption in this case is called *subgroup discrete logarithm assumption* (SDL). We say that two representations $(\alpha_1, \ldots, \alpha_l)$ and $(\beta_1, \ldots, \beta_l)$ are *equivalent* iff there exists a factor $d \in \mathbb{Z}_q$ such that $\alpha_i = d\beta_i$ for all $1 \leq i \leq l$.

We denote protocols in the same way as algorithms are usually denoted: by a declaration and a definition. A *protocol declaration* consists of the (i) formal output parameters, followed by (ii) an assignment arrow, followed by (iii) the protocol name and the (iv) formal input parameters in parenthesis. To enhance readability, all input and output parameters of a participant are enclosed in square brackets labeled by the participant's initial. Values of formal input parameters are called *private input* or *common input* if these parameters are given to only one or to all participants of the protocol, respectively. A *protocol definition* is denoted in matrix form. Actions of each participant are aligned in columns, and each column is labeled by its participants name. Consecutive actions are displayed in consecutively numbered rows, which are called the *steps* of the protocol.

Protocol actions are denoted by usual mathematical notation and a few special symbols. Choosing an element uniformly at random from a set A and assigning it to a variable a is denoted $a \in_R A$. Evaluating an expression E and assigning the result to a is denoted by a left arrow $a \leftarrow E$. By h we denote a pseudo-random hash function [BR93] that, on input any binary string, returns a value in $\mathbb{Z}_q$. We relax the notation by allowing any number of arguments to h meaning that their binary representations are concatenated and then fed to h. Arithmetic operations are either in G_q, i.e., multiplication mod p or in $\mathbb{Z}_q$, i.e., addition and multiplication mod q. We omit the "(mod p)" and "(mod q)" whenever the modulus is clear from the context. Transmitting the value of a variable a from participant Alice to participant Bob is simply denoted by a labeled arrow $\xrightarrow{a}$ that stretches from Alice's to Bob's column[1] . A call of protocol *prot* is denoted by a similar but double headed arrow labeled by the declaration of *prot* instantiated with respective actual parameters. The phrase "proceed iff P" with P a Boolean predicate indicates that the protocol execution proceeds if and only if P holds. Otherwise, the protocol is aborted and the participants return a corresponding exception. *Polynomial composition* of protocols [FS90] means to execute a given set of protocols a polynomial number of times in arbitrarily interleaved fashion.

2.2 Restrictive Blind Signature Scheme

The signer Alice chooses her private key $x \in_R \mathbb{Z}_q$ uniformly at random and publishes the corresponding public key $y \leftarrow g^x \bmod p$. The message space is G_q. A signature σ is *valid* for message m with respect to public key y iff it satisfies the verification predicate

$$\mathit{verify}(y, m, \sigma) \ .$$

On common input a message m, Alice, on private input her signing key x, and Bob, on private input a *modifier* $\omega \in \mathbb{Z}_q$, produce a restrictive blind signature σ' for $m' = m^{\omega}$:[2]

$$([m', \sigma']^B) \leftarrow \mathit{sign}([x, m]^A, [y, m, \omega]^B) \ .$$

Prerequisite 1 *Protocol sign is (i) effective, (ii) unforgeable under the SDL assumption and (iii)* restrictive *in the sense of Brands [B93,B94], i.e., the only*

[1] We abstract here from essential fault-tolerant mechanisms like typing and (logically) time-stamping messages.

[2] We call parameter ω a modifier because it determines which output message Bob obtains relative to the common input message.

way for a cheating Bob to obtain a signature for a message m' is to choose $\alpha, \beta \in \mathbb{Z}_q$ "before" obtaining a signature for $m' = m^{\alpha} g^{\beta}$, where g is a global constant, chosen uniformly at random from $G_q \setminus \{1\}$.[3,4] *(iv) sign is unconditionally blind in the sense that Bob's results from executing protocol sign are unconditionally unlinkable to Alice's views of these executions even if Alice had unlimited computing resources.* $\diamond$

2.3 Restrictive Cascade Signature Scheme

We consider the same setup as for the restrictive blind signature scheme. The common input is a message m, Alice's private input is her signing key $\bar{x}$ corresponding to her public key $\bar{y}$, and Bobs private input is a modifier ω. Now, we allow Bob an additional private input $\sigma \in \Sigma_q$, a valid signature for the common message m with respect to some verification key y, which is usually different from Alice's verification key $\bar{y}$. Bob seeks to obtain from Alice a signature σ' for $m' = m^{\omega}$ with respect to the product $y\bar{y}$. We call this signing operation *cascade* because it can be iterated so that Bob can use his output signatures as private inputs in any subsequent execution of *cascade*:

$$([m', \sigma']^B) \leftarrow \mathit{cascade}([x, m]^A, [y, m, \sigma, \omega]^B) \ .$$

Prerequisite 2 *Protocol cascade is (i) effective, (ii) unforgeable under the SDL assumption, (iii)* restrictive *(see Prerequisite 1) and (iv) unconditionally blind in the sense that Bob's results from executing protocol cascade are unconditionally unlinkable to Alice's views of these executions.* $\diamond$

2.4 Diverted Proof of Knowledge

Brands [B93] has proposed an interactive proof based on work of Chaum, Evertse and van de Graaf [CEG88] where a prover P proves to a verifier V that she knows a witness, namely a representation $u = (u_1, \ldots, u_l) \in \mathbb{Z}q^l$, for a given candidate $\psi \in G_q$, i.e., $\psi = \prod_{i=1}^{l} g_i^{u_i}$ with all g_i chosen independently and uniformly at random in advance. For all $l > 1$, this interactive proof is witness indistinguishable over the predicate family W [FS90], where $W = \{W_q\}$ and $W_q = \{(\psi, w) | \psi = \prod_{i=1}^{l} g_i^{u_i}\}$. Brands has further shown in [B93] how his proof protocol can be "diverted" such that

(i) P, during the interactive proof protocol with V, has online access to a third party, called the co-prover Q, and

(ii) the witness u is shared between P and Q in a way that neither P nor Q knows u by herself.

[3] For simplicity, we assume only one such generator g. However, all following constructions work for any constant number of generators.

[4] The notion of "before" is left informal. We expect to formalize it in future work.

In contrast to the well-defined notion of divertibility by Okamoto and Ohta [OO90], the "diverted" proof protocol by Brands takes private input not only from the co-prover, but also from the prover, and there is no input common to all three parties, but only a semi-common input to Q and P and one to P and V.[5] We need an interactive proof diverted in this way with the additional property that

(iii) if P (by help of Q) can prove knowledge of a representation of an element $\psi \in G_q$ and P chooses some $\omega \in \mathbb{Z}_q$, then P (with the same help of Q), can also prove knowledge of a representation of ψ^ω.

As our last prerequisite we assume an interactive proof protocol satisfying (i), (ii) and (iii) abovel. The candidate (semi-common input to P and V) is denoted ψ. The partial witnesses of P and Q are denoted as (ω, w) and v, respectively, where $\omega \in \mathbb{Z}_q$, $v, w \in \mathbb{Z}_q^2$. The witness of ψ is $u = v\omega + w$:

$$([acc]^V) \leftarrow \mathit{prove}([v]^Q, [\psi, \omega, w]^P, [\psi]^V) \enspace .$$

Prerequisite 3 *Protocol prove is a "diverted" and witness indistinguishable proof of knowledge over $W = \{W_q\}$.* ◇

3 Driver's License Scheme

The following driver's license scheme basically consists of three protocols, namely to bind a pseudonym to a driver (Fig. 3.1), to issue a driver's license (or *license* for short) to a driver (Fig. 3.2) and to show licenses at a road checkpoint (V) (Fig. 3.3). For simplicity, we assume that the binding of new pseudonyms and the issuing of licenses is both done by the Motor Vehicle Services (MVS) I.[6] The basic idea for the following scheme is to enable an observer O to support its host wallet W in obtaining pseudonyms and licenses and in showing licenses, while not allowing O to issue new licenses. After explaining the setup of the scheme, we introduce the protocols in turn.

Observers shall authorize the pseudonym binding requests of their hosts, so there must be a *native key* (x_O, y_O) for each observer; the private part x_O is built into the observer to sign authorizations, and the public part y_O is broadcast to MVS, all road checkpoints and drivers, where wallets can look them up, too. We assume that each observer chooses its own native key.

The Motor Vehicle Services I chooses a *signing key* (x_I, y_I).[7] The public part y_I is broadcast to all drivers and road checkpoints. All observers share a

[5] Other examples of this more general notion of divertibility of proofs of knowledge have been considered by Chen [C94], but a formal definition is outstanding.

[6] In a more integrated system, the issuers of different kinds of certificates could rationalize this approach by establishing a pseudonym binding center and relying upon the same binding procedure.

[7] This signing key serves to issue one kind of driver's license. For different kinds of licenses, e.g., basic, commercial, agricultural, boat, different signing keys must be used.

co-signing key x_O, whose corresponding co-verification key y_O is also broadcast to all road checkpoints and is built into all wallets.

Since observers shall authorize the license showing requests of their hosts, each observer also needs to have a *co-signing key* (x^*, y^*). Again, the private part x^* is built into the observer, and the public part y^* is broadcast to all road checkpoints and wallets. The initialization of observers and their tamper-resistance must be trusted by the Motor Vehicle Services and road checkpoints.

Before observers can be used, each one must be personalized for the particular biometric identity of its owner. Once personalized, there is no way to re-personalize it, and from that on, we assume it verifies its owner with sufficiently small false acceptance and false rejection rates. Now, we let each driver personalize its observer, and insert the observer into his or her personal wallet. Once the wallet has mounted the observer, that driver is prepared to execute any of the following protocols.

In the following, the generators g_1, g_2 are chosen uniformly at random and independently of each other and of the generator g in the definition of restrictiveness of the blind and cascade signature primitives (Prerequisites 1 and 2).

3.1 Binding a Pseudonym to a Driver

The MVS I, a wallet W[8], and an observer O, on input its private native key and a biometric identity $\odot\odot$ (read "face"[9]) execute the binding protocol. If the observer verifies the biometric identity successfully, then I obtains the driver's new *source pseudonym* ψ; so does the driver's wallet and, in addition, the wallet and the observer each obtain their respective partial witnesses (ω, w) and v that are later needed to prove ψ.

3.2 Issuing a Driver's License

The MVS I inputs the private signing key x_I and both, I and the wallet W input W's source pseudonym ψ. In addition, W inputs the partial witness (ω, w) for ψ, and the observer O inputs its own partial witness v plus a biometric identity $\odot\odot$. If the observer verifies $\odot\odot$ successfully, then W obtains a license χ for the *interim pseudonym* ψ' and the corresponding partial witness (ω', w') for ψ'. O obtains the interim pseudonym too.

3.3 Showing a Driver's License

The wallet W inputs a license χ for interim pseudonym ψ' plus the corresponding partial witness (ω', w). The observer O inputs its partial witness v plus a biometric identity $\odot\odot$. If the observer verifies $\odot\odot$ successfully, then the road checkpoint obtains a valid license χ' for *target pseudonym* ϕ. The wallet also obtains its target pseudonym for further reference.

[8] Here and in the following, we do not mention the public keys being input

[9] the parameter $\odot\odot$ contains a binary string representation of the biometric identity of the driver performing this action. This string representation is only visible within the tamper resistant observer, so we may regard it as O's private input.

$$([\psi]^I, [\psi, \omega, w]^W, [v]^O) \leftarrow \mathit{bind}([y_O]^I, [y_O]^W, [x_O, \odot\!\odot]^O)$$

	MVS (I)		**Wallet**		**Observer**
(1)			$\omega \in_R \mathbb{Z}_q, w \in_R \mathbb{Z}_q^2$		$v \in_R \mathbb{Z}_q^2$
(2)			$\psi_W \leftarrow g_1^{\frac{w_1}{\omega}} g_2^{\frac{w_2}{\omega}}$		Proceed iff $\mathit{verify}(\odot\!\odot)$
				$\xrightarrow{\psi_W}$	
(3)					$\psi_O \leftarrow g_1^{v_1} g_2^{v_2}$
				$\xleftarrow{\psi_O}$	
(4)			$\psi' \leftarrow \psi_W \psi_O$		$\psi' \leftarrow \psi_W \psi_O$
(5)			$\xleftrightarrow{[\psi, \sigma]^W \leftarrow \mathit{sign}([x_O, \psi']^O, [y_O, \psi', \omega]^W)}$		
		$\xleftarrow{\psi, \sigma}$			
(6)	$\xleftrightarrow{[acc]^I \leftarrow \mathit{prove}([v]^O, [\psi, \omega, w]^W, [\psi]^I)}$				
(7)	Accept ψ iff $acc \wedge \mathit{verify}(y_O, \psi, \sigma)$				

Fig. 2. Binding a pseudonym to a Driver

Proposition 1. *The proposed driver's license scheme is (i) effective, (ii) the licenses are unforgeable under the SDL assumption, (iii) the licenses are non-transferable between drivers unless observers are broken, and (iv) the views of Motor Vehicle Services, observers and road checkpoints after any polynomial composition of protocols bind, issue and show are mutually unconditionally unlinkable.* $\Diamond$

Proof. (Proposition 1)

Effectiveness: Let the observer's native key be (x_O and y_O), the MVS's signing key be (x_I, y_I) and the observers co-signing key be (x^*, y^*). Furthermore, let a driver D personalize its observer O with biometric identity $\odot\!\odot$ and mount it by wallet W. Then, if I binds the source pseudonym ψ to D,

$$([\psi]^I, [\psi, \omega, w]^W, [v]^O) \leftarrow \mathit{bind}([y_O]^I, [y_O]^W, [x_O, \odot\!\odot]^O) \enspace ,$$

and I issues a license χ for intermediate pseudonym ψ' to D,

$$([\psi', \chi, \omega', w']^W, [\psi']^O) \leftarrow \mathit{issue}([x_I, \psi]^I, [y_I, \psi, \omega, w]^W, [v, \odot\!\odot]^O) \enspace ,$$

and D then shows its license χ for interim pseudonym ψ' to road checkpoint V,

$$([acc, \phi, \chi']^V, [\phi]^W) \leftarrow \mathit{show}([y_I, y^*]^V, [y_I, y^*, \psi', \chi, w', \omega']^W, [y_I, x^*, v, \odot\!\odot]^O) \enspace ,$$

then V sees a license χ' valid for target pseudonym ϕ. This holds for subsequent shows for further target pseudonyms $\phi', \phi'', \ldots$ to road checkpoints $V', V'', \ldots$.

$$([\psi', \chi, \omega', w']^W, [\psi']^O) \leftarrow \mathit{issue}([x_I, \psi]^I, [y_I, \psi, \omega, w]^W, [v, \odot\odot]^O)$$

	MVS (I)	**Wallet**	**Observer**
(1)		$\xleftarrow{\psi}$	Proceed iff $\mathit{verify}(\odot\odot)$
(2)		$\xleftrightarrow{[acc]^I \leftarrow \mathit{prove}([v]^O, [\psi, \omega, w]^W, [\psi]^I)}$	
(3)	Proceed iff acc	Choose $\omega' \in_R \mathbb{Z}_q$	
(4)	$\xleftrightarrow{[\psi', \chi]^W \leftarrow \mathit{sign}([x_I, \psi]^I, [y_I, \psi, \omega']^W)}$		
(5)		$w' \leftarrow \omega w; \omega' \leftarrow \omega' \omega$	
		$\xrightarrow{\psi'}$	

Fig. 3. Issuing a driver's license

Before each show, the wallet prepares its observer by passing pseudonym ψ', so that O knows which partial knowledge v to input to protocol *show*.

Unforgeability of licenses under the SDL assumption follows from Prerequisite 2 because licenses are defined as cascade signatures.

Non-Transferability between Drivers: Protocol *bind* ensures that the wallet does not have any Shannon information about any representation of its source pseudonym ψ. Note, that in step (4) of *bind*, the observer enforces ψ to contain a random factor ψ_O that is not chosen by the wallet. Moreover, the SR-assumption ensures that the wallet cannot compute any representation on its own. Therefore, different drivers will represent their pseudonyms inequivalently with overwhelming probability. Furthermore, the restrictiveness of *sign*

$$([\phi, \chi']^V, [\phi]^W) \leftarrow \mathit{show}([y_I, y^*]^V, [y_I, y^*, \psi', \chi, \omega', w']^W, [y_I, x^*, v, \odot\odot]^O)$$

	Checkpoint (V)	**Wallet**	**Observer**
(1)		Choose $\omega'' \in_R \mathbb{Z}_q$	Proceed iff $\mathit{verify}(\odot\odot)$
		$\xrightarrow{\psi', \chi}$	
(2)		$\xleftrightarrow{[\phi, \chi']^W \leftarrow \mathit{cascade}([x^*, \psi']^O, [y^*, \psi', \chi, \omega'']^W)}$	
		$\xleftarrow{\phi, \chi'}$	
(3)	$\xleftrightarrow{[acc]^I \leftarrow \mathit{prove}([v]^O, [\phi, \omega' \cdot \omega'', w']^W, [\phi]^I)}$		
(4)	Proceed iff $acc \wedge$ $\mathit{verify}(y_I y^*, \phi, \chi')$		

Fig. 4. Showing a Driver's License

(Prerequisite 1) and of *cascade* (Prerequisite 2) ensures that the intermediate pseudonym ψ' and all target pseudonyms $\phi, \phi', \ldots$ derived from ψ are equivalent to ψ. To see this, observe first that the generators g_1, g_2 in protocols *bind*, *issue* and *show* are chosen uniformly at random and independently of any pseudonyms and constants of the underlying blind signature and cascade signature schemes. (In particular, they are chosen independently of g.) From the restrictiveness of the blind and cascade signature schemes, we have that the wallet knows (i) a representation (α, β) of the intermediate pseudonym such that $\psi' = \psi^{\alpha} g^{\beta}$ and (ii) a representation (γ, δ) of the target pseudonym such that $\phi = {\psi'}^{\gamma} g^{\delta}$. Therefore, the wallet also knows a representation of the target pseudonym ϕ with respect to ψ and g, namely $(\alpha\gamma, \beta\gamma + \delta)$.

So we conclude, that a wallet has no Shannon information about the representations of any its intermediate and target pseudonyms. And therefore, the wallet has only but negligible chance to succeed in executing *issue* or *show* without the help of its observer. We assume now that the observer is neither broken (in which case its partial witness v might have been compromised) nor is its biometric verification facility bypassed or tampered with. Then we conclude that the three drivers (i) to whom the source pseudonym ψ is bound, (ii) who obtains a license for the intermediate pseudonym ψ', and (iii) who shows this license under a target pseudonym $\phi, \phi', \ldots$ are all the same (in terms of biometric identity).

Unconditional Unlinkability of the views of MVS, observers and road checkpoints holds for two reasons: Firstly, all pseudonyms related to the same license, i.e., $\psi, \psi', \phi, \phi', \ldots$ look statistically independent to anyone but the wallet W who chooses the modifiers $\omega, \omega', \omega'', \ldots$ to derive its pseudonyms. Secondly, the views of MVS, observers and road checkpoints on the wallet contain—except for those pseudonyms—only the subviews produced by the subprotocol *prove*, which the wallet uses to prove (its knowledge of representations of) its pseudonyms. From Prerequisite 3 we have that protocol *prove* is witness indistinguishable for predicate W_q, and according to Feige, Shamir [FS90] this also holds for any polynomial composition of protocol *prove*. Hence, follows the claim. □

4 Conclusions and Open Questions

We have proposed an efficient implementation for electronic driver's licenses that drivers can freely carry around in small personal devices. Our proposal is based on the wallet with observer architecture proposed by Chaum and Pedersen [CP92]. Since transferability of driver's licenses can only be prevented by some kind of biometric driver authentication, we propose to implant some biometric verification facility into the observer. We have shown how the drivers' privacy can then still be protected even against coalitions of MVS, road checkpoints and observers, which have access to unlimited computing resources. The solution can easily be adapted to other kinds of electronic certificates that must not be tranferred between individuals. Our proposal is based upon restrictive blind signatures and a new primitive called restrictive cascade signatures. Implementations for these primitives will be published soon.

5 Acknowledgement

I thank Birgit Pfitzmann and Joachim Biskup for their continued encouragement of this work. Matthias Schunter contributed looks from several interesting angles, and Matt Franklin kindly but firmly urged me to write down this work.

References

[B93] Stefan Brands: An Efficient Off-line Electronic Cash System Based On The Representation Problem; Centrum voor Wiskunde en Informatica, Computer Science/Departement of Algorithmics and Architecture, *Technical Report CS-R9323*, March 1993.

[B94] Stefan Brands: Untraceable Off-line Cash in Wallet with Observers; *Crypto '93*, LNCS 773, Springer-Verlag, Berlin 1994, 302-318.

[BCB97] Josef Bigün, Gérard Chollet, Gunilla Borgefors (eds.): *Audio- and Video-based Biometric Person Authentication* (AVBPA) '97, LNCS 1206, Springer-Verlag, Berlin 1997

[Bio] Biometric fingerprint readers:

BioMouse	http://www.abio.com
PC-Lockdown	http://users.ids.net
SecureTouch	http://www.biometricaccess.com
TouchSafe	http://www.identix.com
U.are.U	http://www.digitalpersona.com
Veriprint	http://www.biometricID.com

[BR93] Mihir Bellare, Phillip Rogaway: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols; *1st ACM Conference on Computer and Communications Security*, ACM Press, New York 1993, 62-73.

[C83] David Chaum: Blind Signature System; *Crypto '83*, Plenum Press, New York 1984, 153.

[C84] David Chaum: A New Paradigm for Individuals in the Information Age; *1984 IEEE Symposium on Security and Privacy*, IEEE Press, Washington 1984, 99-103.

[C85] David Chaum: Security without Identification: Transaction Systems to make Big Brother Obsolete; *Communications of the ACM* 28/10 (1985) 1030-1044.

[C90] David Chaum: Showing credentials without identification: Transferring signatures between unconditionally unlinkable pseudonyms; *Auscrypt '90*, LNCS 453, Springer-Verlag, Berlin 1990, 246-264.

[C92] David Chaum: Achieving Electronic Privacy; *Scientific American* (August 1992) 96-101.

[C94] Lidong Chen: Witness Hiding Proofs and Applications; *PhD Thesis DAIMI PB-477*, Computer Science Department Aarhus University, August 1994.

[CEG88] David Chaum, Jan.-Hendrik Evertse, Jeroen van de Graaf: An improved protocol for demonstrating possession of discrete logarithms and some generalizations; *Eurocrypt '87*, LNCS 304, Springer-Verlag, Berlin 1988, 127-141.

[CP92] David Chaum, Torben Pryds Pedersen: Wallet Databases with Observers. *Crypto '92*, LNCS 740, Springer Verlag, Berlin 1993, 89-105.

[FS90] Uriel Feige, Adi Shamir: Witness Indistinguishable and Witness Hiding Protocols; *22nd Symposium on Theory of Computing (STOC) 1990*, ACM Press, New York 1990, 416-426.

[M94] Benjamin Miller: Vital signs of identity; *IEEE spectrum* 31/2 (1994) 22-30.

[OO90] Tatsuaki Okamoto, Kazuo Ohta: Divertible zero-knowledge interactive proofs and commutative random self-reducibility; *Eurocrypt '89*, LNCS 434, Springer-Verlag, Berlin 1990, 134-149.

[PPSW97] Andreas Pfitzmann, Birgit Pfitzmann, Matthias Schunter, Michael Waidner: Trusting Mobile User Devices and Security Modules; *Computer* 30/2 (1997) 61-68.