A Practical System for Globally Revoking the Unlinkable Pseudonyms of Unknown Users

Stefan Brands[1], Liesje Demuynck[2], and Bart De Decker[2]

[1] Credentica & McGill School of Comp. Science
1010 Sherbrooke St. W., Suite 1800, Montreal, QC, Canada H3A 2R7
brands@{credentica.com,cs.mcgill.ca}
www.credentica.com
[2] K.U.Leuven, Department of Computer Science
Celestijnenlaan 200A, B-3001 Heverlee, Belgium
{Liesje.Demuynck,Bart.DeDecker}@cs.kuleuven.be
www.cs.kuleuven.be

Abstract. We propose the first single sign-on system in which a user can access services using unlinkable digital pseudonyms that can all be revoked in case she abuses any one service. Our solution does not rely on key escrow: a user needs to trust only her own computing device with following our protocols in order to be assured of the unconditional untraceability and unlinkability of her pseudonyms. Our solution involves two novel ingredients: a technique for invisibly chaining the user's pseudonyms such that all of them can be revoked on the basis of any one of them (without knowing the user's identity with the issuer) and a sublinear-time proof that a committed value is not on a list without revealing additional information about the value. Our solution is highly practical.

1 Introduction

Traditionally, most authenticated relations between users and online services are established on the basis of username and password. As users interact with more and more online services, however, passwords become increasingly vulnerable to phishing and to replay by dishonest service providers. In addition, users are struggling to remember usernames and passwords, which in turn poses a significant burden on the support systems of service providers. As a result, more and more organizations are migrating to secure *single sign-on* (SSO) systems for their users. SSO systems allow a user to access many services without having to manually authenticate more than once. In addition, SSO systems give organizations the ability to globally revoke all access privileges of users for any reason. This is desirable in intra-organizational settings where SSO is used for giving employees

This research was performed under the auspices of McGill University (School of Comp. Science) from 07-2005 until 02-2006 when the second author was visiting the first author at Credentica. Liesje Demuynck is supported by a research assistantship and travel credit from the Fund for Scientific Research, Flanders (Belgium).

online access to corporate resources: when an employee leaves a company, for example, the organization can centrally revoke all her access privileges.

The demand for secure SSO systems goes beyond organizational boundaries. In the past years, industry efforts have resulted in a number of specifications and standards aimed at *cross-organizational* SSO. However, to date very few organizations have adopted cross-organizational SSO systems, especially in consumer-facing settings. A major reason for this lack of adoption is that the current generation of cross-organizational SSO systems create potential privacy and security problems for both users and service providers. These systems revolve around a central server (also known as an *identity provider*) that sees in real time which users interact with what service providers. The identity provider can arbitrarily deny access or revoke all access capabilities of any user at any time. Furthermore, the identity provider can impersonate users and can gain access to accounts they may have established with service providers. While these powers may be desirable in intra-organizational settings, they tend to be overly invasive to autonomous users and service providers.

The SSO system proposed in this paper overcomes these problems, while preserving the ability to globally deny access to any user who abuses a service.

Outline of our Solution. Our system also relies on a central identity provider, but any unwanted powers in that provider are eliminated. The identity provider is responsible for issuing to each user a number of *digital pseudonyms*, which are a special kind of authentication tokens. Users hook their pseudonyms up with service providers and authenticate in subsequent visits by proving knowledge of a secret pseudonym key. Digital pseudonyms are unconditionally unlinkable and untraceable, even vis-à-vis collusions of service providers and the identity provider; thus, by using a different pseudonym with each service provider, each user can ensure that her account information with different service providers cannot be compiled into a super-dossier. Replay attacks are prevented, because secret pseudonym keys are never disclosed when authenticating to service providers. Assuming user devices transparently manage pseudonyms on behalf of their users, users can be given an SSO experience; for instance, a single password could locally unlock all of a user's pseudonyms for the duration of a session.

To enable the global revocation of all of a user's pseudonyms in case the user abuses any one service, the identity provider invisibly chains all of these pseudonyms. Hereto, the identity provider invisibly encodes into all of a user's pseudonyms a set of random numbers that are unique to that user (without the identity provider knowing those numbers). For each pseudonym that a service provider associates with a user, the service provider requires its user to disclose one of these encoded random numbers. By disclosing a different random number for each pseudonym, users preserve the unconditional unlinkability of their pseudonyms. At the same time, service providers can blacklist disclosed numbers in such a manner that users can efficiently prove that their encoded numbers are not blacklisted without revealing any additional information about them.

This revocation technique does not impinge on user privacy, nor does it give covert powers to service providers and the identity provider. Firstly, the encoding

of the invisible numbers into digital pseudonyms requires the cooperation of the user at issuing time. Secondly, in order to be able to blacklist a user, a service provider must ask *all* users who request access to prove that they are not on its blacklist. Thirdly, in order to compute a blacklist proof users require the blacklist as input, and so they can inspect the blacklist and sanction unreasonable requests for blacklist proofs. Fourthly, proving that one is not on the revocation list does *not* reveal any information about one's identity.

Comparison to Other Work. Blind signatures, invented in the eighties by Chaum [16, 18], allow users to authenticate using unconditionally unlinkable pseudonyms. However, when using blind signatures as pseudonyms it is impossible to revoke the pseudonyms of a fraudulent user, whether on the basis of the user's identity with the issuer or on the basis of misuse of any one service. Thus, blind signatures provide privacy for users by trading away security for service providers.

Various adaptations of blind signatures have been proposed to enable global revocation in the context of electronic cash systems, to ensure either (1) that a designated party can identify all e-coin payments of a particular account holder or (2) that all of a payer's payments can be identified if that user engages in a fraudulent payment transaction. In the context of SSO systems, these two features correspond to the ability to revoke all of a user's pseudonyms for a known user (i.e., based on the user's identity with the issuer) and of an unknown user, respectively. Various proposals to extend electronic cash systems with one or both of these features have been presented. Unfortunately, in all of these proposals, the privacy of users is in fact illusional. Namely, most techniques [7, 27, 10, 24, 23] rely on key escrow: the bank encodes into each e-coin a tracing key that its user must disclose in encrypted form at payment time, so that it can be decrypted by a designated "escrow agent (or set of parties) if needed. In e-cash systems not requiring a trusted escrow agent [12, 28], users have to settle for computational unlinkability and untraceability only.

More recently, Camenisch et al. [8, 13] and Nguyen [26] proposed credential revocation mechanisms based on *dynamic accumulators*. Dynamic accumulators enable individuals to prove list membership in constant time in the list size. The security of these accumulators relies on non-standard intractability assumptions, such as the strong RSA assumption and the q-strong Diffie-Hellman assumption. In addition, the schemes merely allow one to revoke the credentials of users on the basis of their identity with the issuer; it is not possible to revoke all of the pseudonyms of an unknown user. Finally, the proofs of knowledge in [13] are statistical zero-knowledge only and the set of accumulatable values is limited to prime numbers in a predefined interval.

An accumulator-based membership proof consists of two steps; the computation of the user's current "witness" (which is a secret value related to the user's accumulated value) and the execution of a zero-knowledge proof of knowledge. Although the latter can be executed in constant time, the former requires a time complexity which is at least linear in the number of elements deleted from the accumulator. Consider, for example, an accumulator to which no elements are added and of which n elements are removed, and assume that a small exponenti-

ation has an exponent size equal to the maximal size of an accumulated value. In this setting, the recomputation of a witness may require n small exponentiations. (In [13], this corresponds to two exponentiations with very large exponents.) In addition, the final witness can only be computed when the final blacklist is known. Hence, not all of a user's exponentiations can be precomputed.

In the context of direct anonymous attestation, Brickell et al. [6] suggest a technique in which a user provides the service provider with a pseudonym $N_V = \zeta^f$ for f a user-specific secret value and ζ a random generator of a group in which the discrete logarithm (DL) problem is hard. The purpose of N_V is twofold: (1) providing the service provider with a pseudonym and (2) enabling revocation based either on the knowledge of f or on a list of other pseudonyms $\{(N_{V'}) = (\zeta')^{f'}, \ldots\}$. The latter can be achieved by proving in zero-knowledge the relation $(\log_{\zeta'} N_{V'} \neq \log_{\zeta} N_V)$ for all $N_{V'}$ in the list [15]. This solution has two major drawbacks. Firstly, the user's unlinkability is only computational. Second, the proof that a pseudonym is not revoked based on a list of pseudonyms requires a number of exponentiations linear in the length of the blacklist.

Brands [5] proposed a practical digital credential mechanism that allows an issuer to invisibly encode into all of a user's credentials a unique number that the issuer can blacklist in order to revoke that user's credentials. This mechanism does not rely on key escrow and preserves the unconditional untraceability and unlinkability of credentials; as such, it offers the same privacy strength as our proposal. Base credentials in the system are as efficient as standard DSA signatures, and the blacklist technique (which consists of repeating a NOT-proof for each blacklist element) is provably secure under the DL assumption. However, Brands' proposal does not allow the revocation of all of the pseudonyms of an unknown user. In addition, the complexity of the cryptographic proof for showing that one's invisibly encoded number is not contained in a blacklist grows linearly in the size of the blacklist. As such, the proposal is not practical for large blacklists.

Our proposal addresses both shortcomings by extending Brands' credentials system using two new techniques: a generalization of Brands' credentials so that multiple credentials can be revoked based on something unique to any one of them, and a sublinear-time cryptographic blacklist proof that is secure under the DL assumption.

Organization of the Paper. Section 2 provides a backgrounder on Brands' credential techniques and compares the system with other credential systems. Section 3 describe our cryptographic protocols in detail and analyzes their security and privacy properties. Finally, Sections 4 and 5 analyze the practicality of the proposal and outline various extensions and variations.

2 Digital Credentials

Our new system is based on Brands' credential techniques [5]. Section 2.1 provides a backgrounder on these techniques. To motivate the choice for Brands' system, Section 2.2 compares this system with other credential techniques.

2.1 Backgrounder of Brands' Digital Credentials

In the system of Brands [5], credentials are issued by a Credential Authority (CA) that has its own key pair for digitally signing messages. When issuing a credential to a user Alice, the CA through its digital signature binds one or more attributes to a digital credential public key, the secret key of which only Alice knows. The whole package that Alice receives is called a digital credential.

Alice can show her digital credential to Bob by providing him with her digital credential public key and the CA's signature. If desired, she selectively discloses a property of the attributes in her digital credential, while hiding any other information about these attributes. Finally, to prevent Bob from replaying the digital credential, Alice digitally signs a *nonce* using her secret key.

Since Alice reveals the digital credential public key and the CA's signature when showing a digital credential, these elements must be uncorrelated to the information that the CA sees when it issues the digital credential, even if the CA tries to cheat. At the same time, the CA must be able to encode the desired attributes into the digital credential, even if Alice tries to cheat. Here is how l attributes, $(x_1, \ldots, x_l)$, are encoded in a digital credential. The tuple $(x_1 s, \ldots, x_l s, s)$ is Alice's secret key for the digital credential. Alice generates s at random from $\mathbb{Z}_q$ in the issuing protocol. Even though Alice may disclose some attributes to Bob in the showing protocol, she keeps s secret at all times; this ensure that only she knows the entire secret key. The digital credential public key is the product $h = (g_1^{x_1} \cdots g_l^{x_l} h_0)^s$. Elements $g_1, \ldots, g_l, h_0$ are random generators of a group G_q of prime order q; they are part of the CA's public key. The digital credential public key reveals no information about $x_1, \ldots, x_l$: for any public key and for any tuple $(x_1, \ldots, x_l)$, there is exactly one $s \in \mathbb{Z}_q$ that would make the match. At the same time, regardless of the choice of l and under the DL assumption in G_q, Alice cannot compute a digital credential public key for which she knows more than one secret key [5, Proposition 2.3.3]. Hence, by signing the digital credential public key the CA indirectly binds a unique attribute tuple to Alice's digital credential: the CA's signature binds Alice's public key, which in turn binds her secret key containing the attributes.

To show a digital credential to Bob, Alice transmits to him the digital credential public key and the CA's digital signature. In addition, she selectively discloses a property of the attributes and digitally signs a nonce using her secret key. Alice's signature, which is derived from a *proof of knowledge*, proves not only that she knows a secret key but also that the attributes in her digital credential satisfy the particular attribute property she is disclosing to Bob. Under the DL assumption in G_q, Bob cannot compute any secret key when presented with Alice's digital credential public key, regardless of which property of $(x_1, \ldots, x_l)$ Alice discloses to him. Alice can demonstrate a wide spectrum of properties to Bob. Among others, using the notation $(x_1, \ldots, x_l) = \mathrm{rep}_{(g_1, \ldots, g_l)} h$ to refer to a *representation* $(x_1, \ldots, x_l)$ such that $h = g_1^{x_1} \ldots g_l^{x_l}$, Alice can prove any of the following properties:

- *Knowledge of a representation containing known attribute values [5, Chapter 3]:* Alice can prove knowledge of a representation $(x_1, \ldots, x_l)$ of $h \in G_q$ with

respect to any $(g_1, \ldots, g_l) \in G_q^l$, and in doing so she can disclose any subset $D \subset \{x_1, \ldots, x_l\}$. For an example subset $D = \{x_{j-1}, x_j\}$, we denote this protocol by $PK\{(\chi_1, \ldots, \chi_{j-2}, \chi_{j+1}, \ldots, \chi_l) : (\chi_1, \ldots, \chi_{j-2}, x_{j-1}, x_j, \chi_{j+1}, \ldots, \chi_l) = \mathrm{rep}_{(g_1,\ldots,g_l)} h\}$. (Greek letters represent the values that remain unknown to Bob.)

- *Knowledge and equality of discrete logarithms [17]:* Given values h_1, h_2, g_1, g_2, g_3 and g_4 in G_q, Alice can demonstrate her knowledge of a tuple (x_1, x_2, x_3) such that $h_1 = g_1^{x_1} g_2^{x_2}$ and $h_2 = g_3^{x_1} g_4^{x_3}$. We denote this protocol by $PK\{(\chi_1, \chi_2, \chi_3) : (\chi_1, \chi_2) = \mathrm{rep}_{(g_1,g_2)} h_1 \wedge (\chi_1, \chi_3) = \mathrm{rep}_{(g_3,g_4)} h_2\}$. It can be extended towards a proof of equality of arbitrary exponents using arbitrary base tuples.
- *Knowledge of discrete logarithms constituting successive powers [11, Chapter 3]:* Let $h_1, \ldots, h_n, g_1$ and g_2 be values in G_q. Alice can prove knowledge of values $x, y_1, \ldots, y_n \in \mathbb{Z}_q$ such that $h_i = g_1^{x^i} g_2^{y_i}$ for $i \in \{1, \ldots, n\}$. We denote this protocol by $PK\{(\chi, \gamma_1, \ldots, \gamma_n) : (\chi, \gamma_1) = \mathrm{rep}_{(g_1,g_2)} h_1 \wedge (\chi^2, \gamma_1) = \mathrm{rep}_{(g_1,g_2)} h_2 \wedge \ldots \wedge (\chi^n, \gamma_n) = \mathrm{rep}_{(g_1,g_2)} h_n\}$.
- *Knowledge of a discrete logarithm unequal to zero [5, Chapter 3]:* Let h be a value in G_q. Alice can demonstrate to Bob that she knows a representation (x_1, x_2) of h w.r.t. base tuple $(g_1, g_2) \in (G_q)^2$, such that $x_1 \neq 0$. We denote this protocol by $PK\{(\chi_1, \chi_2) : (\chi_1, \chi_2) = \mathrm{rep}_{(g_1,g_2)} h \wedge \chi_1 \neq 0\}$. Brands calls this a NOT proof.
- *AND connections:* All previous formulae can be combined by "AND" connectives. Given formulae $F_1(x_{1,1}, \ldots, x_{1,l_1}), \ldots, F_n(x_{n,1}, \ldots, x_{n,l_n})$ about secrets $(x_{i,1}, \ldots, x_{i,l_i})$ $(i = 1, \ldots, n)$, we denote this protocol by $PK\{(\chi_{1,1}, \ldots, \chi_{n,l_n}) : F_1(\chi_{1,1}, \ldots, \chi_{1,l_1}) \wedge \ldots \wedge F_n(\chi_{n,1}, \ldots, \chi_{n,l_n})\}$.

Under the DL assumption, all protocols are perfect honest-verifier zero-knowledge. They can be made concurrent zero-knowledge at virtually no overhead by using techniques of Damgård [21].

We briefly review the most important properties of Brands' credential system based on the Chaum-Pedersen based issuing protocol [5, Section 4.5.2].

Proposition 1. *Brands' credential system [5] satisfies the following properties.*

1. *If an honest user Alice accepts the credential issuing protocol, she retrieves a credential secret key $(x_1, \ldots, x_l, s)$, a corresponding public key h and a signature sign(h), such that $(h, sign(h))$ is uniformly distributed over the set $\{(h, sign(h)) | h \in G_q \setminus \{1\}\}$.*
2. *Assuming the Chaum-Pedersen protocol [17] is secure, it is infeasible to existentially forge a credential.*
3. *For any credential public key h and signature sign(h), for any tuple $(x_1, \ldots, x_l)$, and for any view of* CA *on a credential issuing protocol in which $p = g_1^{x_1} \ldots g_l^{x_l}$ is used as initial input (with $(x_1, \ldots, x_l)$ known by* CA*), there is exactly one set of random choices that an honest user Alice could have made during the execution of this issuing protocol such that she would have output a credential containing both h and sign(h).*
4. *Let h be a valid credential public key. Under the DL assumption and provided that $s \neq 0$, proving knowledge of a representation $(x_1^*, \ldots, x_l^*, s^*)$ of h_0^{-1}*

w.r.t. $(g_1, \ldots, g_l, h)$ *is equivalent to proving knowledge of a valid secret key* $(x_1^* s, \ldots, x_l^* s, s)$ *corresponding to* h*. Moreover, the relation* $s^* = -s^{-1}$ *holds.*

5. *Consider any number of arbitrarily interleaved executions of a showing protocol with a computationally unbounded Bob in which Alice only discloses formulae about the attributes that do not contain* s*, and in which she uses only proofs of knowledge that are statistically witness-indistinguishable. Whatever information Bob can compute about the credential attributes, he can also compute using merely his a priori information (i.e., without engaging in showing protocol executions) and the status of the requested formulae.*

Assumption 1 *Under the DL assumption, if a computationally bounded attacker* $\mathcal{A}$*, after engaging in an execution of the issuing protocol with* CA*, in which* $p = g_1^{x_1^*} \ldots g_l^{x_l^*}$ *is used as input, outputs a valid credential containing a secret key* $(x_1, \ldots, x_l, s)$*, then* $(x_1, \ldots, x_l, s) = (x_1^* s, \ldots, x_l^* s, s)$ *with overwhelming probability. This assumption remains valid even when polynomially many executions of the issuing protocol are arbitrarily interleaved.*

2.2 Comparison to Other Credential Systems

Before moving on to the new system, we compare Brands' system with the CL-based systems of Camenisch and Lysyanskaya [9, 14]. The core of the CL-based systems is a signature scheme with additional protocols for the retrieval of a signature on committed values and for the demonstration of signature possession in zero knowledge. Consequently, a credential can be shown unlinkably multiple times and all pseudonyms based on the same credential can be revoked by simply revoking the credential.

We compare the complexity of Brands' scheme [5, Section 4.5.2] with the optimized CL-RSA scheme of [1] and the CL-DL system of [14]. The evaluation considers communication sizes and workloads in the number of exponentiations. Multiplications and additions are neglected, as their demand on computational resources is many orders of magnitude smaller. As for the schemes, we adopt Brands' scheme for a subgroup construction with $|p| = 1600$, $|q| = 256$ and $|s| = 160$ (see Brands [5, Section 4.5.2]), a CL-RSA scheme with parameters $\ell_m = 256$, $\ell_c = 160$, $\ell_s = 80$, $\ell_e = 259$ and $\ell_n = 1600$ (see Bangerter et al. [1]), and a CL-DL scheme based on a bilinear map over elliptic curves, with $|q| = 256$, $|\mathsf{G}| \approx 2^{1600}$ and a zero-knowledge challenge length of 160 bits (see Camenisch et al. [14]). Workloads are approximated by the number of small (256-bit) exponentiations that must be performed[3]. In addition, we assume the complexity of a bilinear pairing to be competitive to that of a small exponentiation. We evaluate an issuing protocol for a credential with l user-chosen attributes which are unknown to *CA* and a showing protocol in which no properties of the attributes are demonstrated. Table 1 summarizes the results.

Compared to the CL-RSA scheme, Brands' credentials are cheaper in all aspects. With respect to CL-DL, they are much cheaper to show and slightly

[3] Larger exponentiations are reduced to small exponentiations using the guideline that an x-bit exponentiation roughly compares to x/y y-bit exponentiations.

Table 1. A comparison of complexity for different credential systems.

size of credentials

Brands	$32l + 328$ bytes
CL-RSA	$32l + 473$ bytes
CL-DL	$96l + 128$ bytes

issuing protocol

	#expon. Alice		#expon. *CA*		comm.
	offline	online	offline	online	
Brands	$2l + 6$	3	2	$l + 3$	$32l + 1116$ bytes
CL-RSA	$3l + 14$	7	-	$2l + 21$	$62l + 1405$ bytes
CL-DL	$2l + 2$	-	$2l + 3$	$l + 3$	$96l + 213$ bytes

showing protocol

	#expon. Alice		#expon. Bob		comm.
	offline	online	offline	online	
Brands	$l + 2$	-	-	$l + 7$	$32l + 748$ bytes
CL-RSA	$2l + 18$	-	-	$2l + 9$	$62l + 785$ bytes
CL-DL	$4l + 8$	-	-	$6l + 8$	$96l + 380$ bytes

more expensive to retrieve. Brands' credentials cannot be shown unlinkably. This property can however be simulated by using multiple copies of the same credential. One additional credential for identical attributes occupies 296 bytes and can be retrieved by 7 exponentiations from Alice. Hence, for l attributes, about $l/5$ Brands credentials occupy the same amount of space as one CL-DL credential. Additionally, the retrieval of $l/7 + 2$ of Brands credentials costs roughly as much for Alice as the retrieval of one CL-RSA credential.

The CL-DL scheme is based on elliptic curves and bilinear pairings. The adopted bilinear map must provide efficient computations as well as adequate security for the DL problem. Hence, the system's key setup must be chosen very carefully. In contrast, the CL-RSA scheme as well as Brands' system are very flexible in their choice of key setup.

Provided the issuer's key-setup is performed correctly, both Brands' system and the CL-DL scheme guarantee unconditional privacy for the user. In Brands' system, this key-setup can easily be checked by ensuring that p and q are prime and that $q|p - 1$. In contrast, the CL-RSA scheme provides statistical privacy only. Its procedure for checking the key-setup requires a signed proof of knowledge with binary challenges. In the setting described above, constructing the proof requires about $6(l + 1)\ell_c$ small exponentiations, while verifying it requires $7(l + 1)\ell_c$ small exponentiations.

Brands' credentials can easily be incorporated in wallets-with-observers [20, 3, 5] such that all inflow and outflow is prevented. The integration of trusted modules that can protect the security interests of the identity provider, relying parties, and/or third parties, is critical in many applications. It is not clear whether and how this can be achieved for the CL-based schemes.

Brands' system also offers other unique features, such as the ability to selectively censor user-disclosed attribute values from signed showing protocol tran-

scripts, the ability to recertify previous issued credentials without knowing their attribute values and the ability to selectively update attribute values in previously issued credentials without knowing the values themselves. Note that the latter two properties could also be achieved using the CL-based schemes. In contrast to Brands' solution, however, their proposals are highly inefficient.

Brands' scheme does not provide multi-show unlinkability but achieves unconditional privacy and highly practical showing protocols. Because of the latter property and its richer feature set, we have opted for Brands' system.

3 The New System

The principal parties in our system are a user $\mathcal{U}$, an identity provider $\mathcal{IP}$ and l service providers $\mathcal{S}_i$ $(i = 1, \ldots, l)$. $\mathcal{U}$ retrieves her pseudonyms from $\mathcal{IP}$ and uses them to authenticate to service providers. In the remainder of the paper, $\mathcal{S}_i$ refers to the service provider as well as to the provided service.

In order to obtain her pseudonyms, $\mathcal{U}$ contacts $\mathcal{IP}$ and both parties engage into a pseudonym retrieval protocol. As private output of this protocol, $\mathcal{U}$ retrieves a set of l unlinkable pseudonyms, such that each of them encodes the same random tuple $(d_1, \ldots, d_l)$. To access service $\mathcal{S}_i$, $\mathcal{U}$ authenticates herself with her i-th pseudonym and additionally discloses d_i. She also proves, for each $j \in \{1, \ldots, l\}$, that value d_j encoded in her credential is not on a blacklist L_j. Blacklists are formed as follows: for any user $\mathcal{U}$, if $\mathcal{U}$ abuses service $\mathcal{S}_j$ then $\mathcal{U}$'s value d_j is added to a public blacklist L_j.

Next, we describe the system setup and the protocols for pseudonym retrieval, pseudonym registration and subsequent authentication to service providers.

3.1 System Setup

To set up the system, $\mathcal{IP}$ decides on a group G_q of prime order q in which the DL assumption is believed to hold. She generates a keypair (sk, pk) suitable for issuing digital credentials containing $l+1$ attributes. We assume $(g_1, \ldots, g_{l+1}, h_0) \in G_q$ to be part of $\mathcal{IP}$'s public key pk. Credential public keys are of the form $g_1^{x_1} \ldots g_l^{x_l} g_{l+1}^t h_0^s$, where $(x_1, \ldots, x_l, t, s)$ is the credential secret key.

Additionally, each service provider $\mathcal{S}_i$ sets up and publishes an empty list L_i that can only be modified by $\mathcal{S}_i$. $\mathcal{S}_i$ also publishes values $a_i, b_i \in_{\mathcal{R}} G_q$ where $z_i = \log_{a_i} b_i$ is privy to $\mathcal{S}_i$.

A pseudonym is a tuple $(P, \mathit{sign}(P))$. Here, $P \neq 1$ is a credential public key. User $\mathcal{U}$ is said to be the owner of $(P, \mathit{sign}(P))$ if she knows P's secret key $(x_1, \ldots, x_l, t, s)$.

3.2 Pseudonym Retrieval

Before retrieving a set of pseudonyms, $\mathcal{U}$ authenticates her identity to $\mathcal{IP}$. Assuming $\mathcal{U}$ meets the enrollment requirements of $\mathcal{IP}$, the following protocol is then executed:

1. User $\mathcal{U}$ generates random values $d_{(1,1)}, \ldots, d_{(1,l)}, e \in_{\mathcal{R}} \mathbb{Z}_q$ and sends $p_1 = (\prod_{i=1}^{l} g_i^{d_{(1,i)}}) g_{l+1}^{e}$ to $\mathcal{IP}$.
2. $\mathcal{IP}$ retrieves p_1, picks l random values $d_{(2,1)}, \ldots, d_{(2,l)} \in_{\mathcal{R}} \mathbb{Z}_q$ and computes $p = p_1 \prod_{i=1}^{l} g_i^{d_{(2,i)}}$. She sends $d_{(2,1)}, \ldots, d_{(2,l)}$ to $\mathcal{U}$.
3. $\mathcal{U}$ creates $d_i = d_{(1,i)} + d_{(2,i)}$ for $i = 1, \ldots, l$ and computes $p = (\prod_{i=1}^{l} g_i^{d_i}) g_{l+1}^{e}$.
4. $\mathcal{IP}$ and $\mathcal{U}$ perform l instances of the credential issuing protocol of Brands [5, Section 4.5.2], using p as initial input. As a result, user $\mathcal{U}$ obtains l tuples $(P_i, \mathit{sign}(P_i))$, and l values $s_i \in \mathbb{Z}_q$, such that $P_i = (ph_0)^{s_i}$ for $i = 1, \ldots l$. (All protocol executions may be done in parallel.)

During steps 1 to 3, a random tuple $(d_1, \ldots, d_l) \in_{\mathcal{R}} (\mathbb{Z}_q)^l$ is created such that neither $\mathcal{U}$ nor $\mathcal{IP}$ can control its final value. Note that, because of the random selection of e by $\mathcal{U}$, this tuple remains unconditionally hidden from $\mathcal{IP}$. Based on $(d_1, \ldots, d_l, e)$, a list of l pseudonyms $(P_i, \mathit{sign}(P_i))$ $(1 \leq i \leq l)$ is then created for $\mathcal{U}$ during step 4.

Provided $\mathcal{U}$ has followed the protocol, the resulting pseudonyms are unconditionally unlinkable and untraceable. $\mathcal{U}$ can also compute a secret key $(d_1 s_i, \ldots, d_l s_i, e s_i, s_i)$ for each pseudonym $(P_i, \mathit{sign}(P_i))$. As a result of Assumption 1, the same tuple $(d_1, \ldots, d_l, e)$ is encoded into all of these secret keys, even when $\mathcal{U}$ tries to cheat. We will refer to $(d_1, \ldots, d_l, e)$ as the tuple encoded into P_i, and to d_j $(j \in \{1, \ldots, l\})$ as the j-th value encoded into P_i.

The following result states the infeasibility to create a pseudonym encoding a value which is the same as the value encoded into another user's pseudonym.

Proposition 2. *Under the discrete logarithm assumption in G_q and for fixed values $d \in \mathbb{Z}_q$ and $i \in \{1, \ldots, l\}$. For any attacker $\mathcal{A}$ engaging into a pseudonym retrieval protocol with $\mathcal{IP}$ and as such retrieving a valid pseudonym $(P, \mathit{sign}(P))$. With negligible probability, value d is the i-th value encoded into P.*

3.3 Pseudonym Registration with the Service Provider

To register pseudonym $(P_i, \mathit{sign}(P_i))$ with service provider $\mathcal{S}_i$, user $\mathcal{U}$ shows $(P_i, \mathit{sign}(P_i))$ and discloses value d_i encoded into P_i. $\mathcal{U}$ and $\mathcal{S}_i$ then perform (possibly in signed proof mode)

$$PK\{(\delta_1, \ldots, \delta_{i-1}, \delta_{i+1}, \ldots, \delta_l, \epsilon, \varsigma) : (\delta_1, \ldots, \delta_{i-1}, d, \delta_{i+1}, \ldots, \delta_l, \epsilon, \varsigma) = \mathrm{rep}_{(g_1, \ldots, g_l, g_{l+1}, P_i)} h_0^{-1}\}$$

$\mathcal{S}_i$ accepts the protocol if and only if she accepts this proof and if $P_i \neq 1$ and if $(P_i, \mathit{sign}(P_i))$ constitutes a valid message/signature pair. $\mathcal{S}_i$ then stores (P_i, d_i) and associates it with a new account or perhaps with a legacy account that it maintains on $\mathcal{U}$. (the latter requires a one-time legacy or out-of-band authentication step to ensure the right association is made).

As per Proposition 1 (property 4), this protocol proves Alice's ownership of $(P_i, \mathit{sign}(P_i))$ and proves that the disclosed value d_i is indeed the i-th value encoded into P_i. Furthermore, as a result of Proposition 1 (property 5), $\mathcal{S}_i$ cannot find out more information about the tuple $(d_1^*, \ldots, d_l^*, e^*)$ encoded into P_i, than what she can deduce from her previous knowledge and the fact that $d_i^* = d_i$.

3.4 Accessing a Service

Upon having registered her pseudonym with $\mathcal{S}_i$, $\mathcal{U}$ may either disconnect and return later on to access the service, or proceed immediately. In either case, to access the service of $\mathcal{S}_i$, $\mathcal{U}$ and $\mathcal{S}_i$ engage in the following protocol, for blacklists $\{L_1, \ldots, L_l\}$ as defined earlier. In step 1 of the following protocol, $\mathcal{S}_i$ checks whether d_i belongs to her own blacklist L_i; in step 2, $\mathcal{U}$ proves that each j-th value d_j ($j \in \{1, \ldots, l\} \setminus \{i\}$) encoded into P_i does not belong to blacklist L_j.

1. $\mathcal{S}_i$ verifies if $d_i \in L_i$. If so, she aborts the protocol and rejects $\mathcal{U}$'s request. If not, she proceeds to step 2.
2. If all of the blacklists L_j for $j \in \{1, \ldots, i-1, i+1, \ldots, l\}$ are empty, then $\mathcal{U}$ must prove knowledge to $\mathcal{S}_i$ of her pseudonym key (assuming she is not still in the pseudonym registration session with $\mathcal{S}_i$, in which case this step can be skipped); this can be done using the standard proof of knowledge of a representation, without disclosing any attributes (d_i has already been disclosed and proven to be correct). If not all of the blacklists are empty, then the following steps are executed for each $j \in \{1, \ldots, i-1, i+1, \ldots, l\}$ for which L_j is not empty:
   (a) Both $\mathcal{U}$ and $\mathcal{S}_i$ look up $L_j = \{y_1, \ldots, y_n\}$. They set $m = \lceil \sqrt{n} \rceil$ for $n = |L_j|$ and compute the coefficients $a_{i,j} \in \mathbb{Z}_q$ ($i \in \{1, \ldots, m\}, j \in \{0, \ldots, m\}$) of the following polynomials in $\mathbb{Z}_q$.
   $p_1(x) = (x - y_1)(x - y_2) \ldots (x - y_m) = a_{1,m}x^m + a_{1,m-1}x^{m-1} + \ldots + a_{1,0}$
   $p_2(x) = (x - y_{m+1}) \ldots (x - y_{2m}) = a_{2,m}x^m + a_{2,m-1}x^{m-1} + \ldots + a_{2,0}$
   $\vdots$
   $p_m(x) = (x - y_{(m-1)m+1}) \ldots (x - y_n) = a_{m,m}x^m + a_{m,m-1}x^{m-1} + \ldots + a_{m,0}$
   (b) $\mathcal{U}$ chooses random values $r_1, \ldots, r_m \in_{\mathcal{R}} \mathbb{Z}_q$ and generates values $C_k = a_i^{d_j^k} b_i^{r_k}$ for all values $k \in \{1, \ldots, m\}$. She also computes $v_k = p_k(d_j)$, $w_k = a_{k,m}r_m + \ldots + a_{k,2}r_2 + a_{k,1}r_1$ and $C_{v_k} = a_i^{v_k} b_i^{w_k}$ for $k = 1, \ldots, m$. All values C_k, C_{v_k} ($k \in \{1, \ldots, m\}$) are sent to $\mathcal{S}_i$.
   (c) $\mathcal{S}_i$ receives C_k, C_{v_k} for all $k \in \{1, \ldots, m\}$ and checks for each $k \in \{1, \ldots, m\}$ if $C_{v_k} = (C_m)^{a_{k,m}} (C_{m-1})^{a_{k,m-1}} \ldots (C_1)^{a_{k,1}} a_i^{a_{k,0}}$. If this fails, $\mathcal{S}_i$ aborts and rejects $\mathcal{U}$'s request.
   (d) Next, the following proof of knowledge is executed. The proof makes use of the techniques described in Section 2. $\mathcal{S}_i$ accepts only if she accepts the proof.

$$
\begin{aligned}
PK\{&(\delta_1, \ldots, \delta_l, \epsilon, \varsigma, \rho_1, \ldots, \rho_m, \upsilon_1, \ldots, \upsilon_m, \omega_1, \ldots, \omega_m) : \\
&(\delta_1, \ldots, \delta_j, \ldots, \delta_l, \epsilon, \varsigma) = \mathrm{rep}_{(g_1, \ldots, g_{l+1}, P_i)} h_0^{-1} \ \wedge && (1) \\
&(\delta_j, \rho_1) = \mathrm{rep}_{(a_i, b_i)} C_1 \ \wedge \ \ldots \ \wedge (\delta_j^m, \rho_m) = \mathrm{rep}_{(a_i, b_i)} C_m \ \wedge && (2) \\
&(\upsilon_1, \omega_1) = \mathrm{rep}_{(a_i, b_i)} C_{v_1} \ \wedge \ \upsilon_1 \neq 0 \ \wedge \ \ldots \ \wedge \\
&(\upsilon_m, \omega_m) = \mathrm{rep}_{(a_i, b_i)} C_{v_m} \ \wedge \ \upsilon_m \neq 0\} && (3)
\end{aligned}
$$

We now explain what happens in step 2. In step 2a, elements in L_j are divided into subsets $L_{j,k}$ of size $m = \lceil \sqrt{|L_j|} \rceil$. The polynomials $p_k(.)$ ($k = 1, \ldots, m$)

are then constructed such as to contain only the elements of $L_{j,k}$ as roots. For each k in $\{1,\ldots,m\}$, values C_k and C_{v_k} are constructed in step 2b. C_k hides a power d_j^k of d_j, while C_{v_k} hides the mapping $p_k(d_j)$ of d_j. Note that

$$\begin{aligned}(C_m)^{a_{k,m}}(C_{m-1})^{a_{k,m-1}}\ldots(C_1)^{a_{k,1}}a_i^{a_{k,0}} \\ &= a_i^{a_{k,m}d_j^m+\ldots+a_{k,1}d_j+a_{k,0}}b_i^{a_{k,m}r_m+\ldots+a_{k,1}r_1} \\ &= C_{v_k}.\end{aligned}$$

In step 2d, $\mathcal{U}$ proves that the values hidden in $C_1,\ldots,C_k$ are consecutive powers of the same value d_j (equation 2), that this value d_j is also the j-th value encoded into P_i (equation 1), and that values $p_k(d_j)$ hidden in C_{v_k} for $k=1,\ldots,m$ differ from zero (equation 3). The latter proves that d_j is not a root of any of the polynomials p_k ($k\in\{1,\ldots,l\}$), and hence does not belong to L_j.

Proposition 3. *Under the discrete logarithm assumption, provided that $P_i \neq 1$, the subprotocol in step 2 is a perfect honest-verifier zero-knowledge proof that for all $j\in\{1,\ldots,l\}\setminus\{i\}$, the j-th value encoded into P_i does not belong to blacklist L_j.*

Proposition 4. *Consider a computationally unbounded service provider $\mathcal{S}_i$ and an honest user $\mathcal{U}$. Consider any number of arbitrary interleaved executions of step 2 for a pseudonym $(P, sign(P))$ with $P\neq 1$ and for the same or different lists L_j ($j\in\{1,\ldots,l\}\setminus\{i\}$). Whatever information $\mathcal{S}_i$ can compute about $(d_1,\ldots,d_l)$ encoded into P, $\mathcal{S}_i$ can also compute it using merely her a-priori information and the shown formulae $d_j\notin L_j$ ($\forall j\in\{1,\ldots,l\}\setminus\{i\}$).*

For a detailed proof of Propositions 3 and 4, we refer to our technical report [4].

The following result can now easily be seen to hold, based on Propositions 1, 3 and 4.

Proposition 5. *Under the DL assumption, the following holds for any registered pseudonym $(P_i, sign(P_i))$ and d_i that $\mathcal{S}_i$ has accepted, assuming $\mathcal{S}_i$ accepts $\mathcal{U}$'s blacklist proof. With overwhelming probability, $\mathcal{U}$ is the owner of a valid pseudonym $(P_i, sign(P_i))$ which has not been revoked and for which d_i is the i-th value encoded into P_i. Furthermore, $\mathcal{S}_i$ cannot find out any more information about the values encoded into P_i than what she can deduce from her a-priori information, the fact that $(P_i, sign(P_i))$ has not been revoked and the fact that d_i is the i-th value encoded into P_i.*

We also have the following result.

Proposition 6. *Given non-empty sets $D_1,\ldots,D_l\subset\mathbb{Z}_q$, for any pseudonym $(P, sign(P))$ such that P encodes a tuple $(d_1,\ldots,d_l)\in D_1\times\ldots\times D_l$, for any view of $\mathcal{IP}$ in an execution of a retrieval protocol and for any $j\in\{1,\ldots,l\}$. There are exactly $(\prod_{i=1}^{l}|D_i|).(q-1)^{l-1}q^{2(l-1)}\neq 0$ sets of random choices that an honest user $\mathcal{U}$ could have made during the execution of this retrieval protocol, such that she would have output $(P, sign(P))$ as her j-th pseudonym.*

That is, a computationally unbounded $\mathcal{IP}$ cannot link a pseudonym $(P, sign(P))$ to its retrieval protocol, even if she would know the tuple $(d_1, \ldots, d_l)$ encoded into P. This is an immediate result of Proposition 1 (property 3) and the specifications of the credential issuing protocol ([5, Section 4.5.2]). Namely, there are exactly $\prod_{i=1}^{l}(|D_i|)$ tuples $(d_1, \ldots, d_l, e)$ such that p (and hence P) will be correctly formed. Furthermore, only 1 set of random choices remains during the j-th instance of the credential issuing protocol, and $q^2(q-1)$ sets of choices during each other instance $i \in \{1, \ldots, l\} \setminus \{j\}$.

4 Efficiency Analysis

The retrieval protocol is executed only once between $\mathcal{U}$ and $\mathcal{IP}$. It requires $\mathcal{U}$ to perform $9l + 1$ exponentiations in G_q, of which $3l + 1$ exponentiations can be precomputed. $\mathcal{IP}$ in turn performs $3l + 1$ exponentiations. A total of $2l + 2$ elements in G_q, and $3l$ elements in $\mathbb{Z}_q$ are communicated. By way of example, if we take $l = 100$, and if we set G_q to be the unique q-order subgroup of the multiplicative group $\mathbb{Z}_p^*$ for primes p and q of 1600 and 256 bits respectively, this amounts to 901 exponentiations for $\mathcal{U}$, 301 exponentiations for $\mathcal{IP}$, and 49kB of transferred data.

With regard to the service access protocol, we take into account the following optimizations:

1. The proofs of knowledge of step 2d, for all $j \in \{1, \ldots, l\} \setminus \{i\}$, can be collapsed into a single proof protocol. As a result, equation 1 has to be performed only once.
2. The check in step 2c can be sped up using the batch verification techniques [2]. $\mathcal{S}_i$ hereto chooses random values $o_1, \ldots, o_m$ in a set $V \subset \mathbb{Z}_q$, and checks the following equation: $\prod_{k=1}^{m} C_{v_k}^{o_k} \stackrel{?}{=} a_i^{\sum_{k=1}^{m} a_{k,0} o_k} \prod_{i=1}^{m} C_i^{\sum_{k=1}^{m} a_{k,i} o_k}$. If this check succeeds, the probability that $\mathcal{S}_i$ correctly accepts step 2c is at least $1 - 1/|V|$.
3. $\mathcal{S}_i$ can complement blacklists using whitelists. A whitelist $L'_j \subset L_j$ represents the set of values for which $\mathcal{U}$ has already passed the blacklist proof. A tuple $(L'_1, \ldots, L'_l)$ of whitelists is stored, both by $\mathcal{U}$ and by $\mathcal{S}_i$, for each credential $(P, sign(P))$. Assuming the elements in L_j are ordered chronologically, it is sufficient for $\mathcal{U}$ and $\mathcal{S}_i$ to only store the last value that passed the proof. Whenever $\mathcal{U}$ requests access to $\mathcal{S}_i$, she merely needs to perform a blacklist proof with respect to the "delta-blacklists" $L_j^* = L_j \setminus L'_j$ for $j = \{1, \ldots, l\}$.
4. All of $\mathcal{U}$'s exponentiations can be precomputed using a variation of Brands' error correction factors technique [5, Section 5.4.2]. A detailed description of this protocol can be found in our technical report [4]. Note that these precomputation can be performed even before the final blacklist is known. All that is needed is an upper bound on $\sqrt{n}$ for n the size of the blacklists.
5. By employing her private value z_i, $\mathcal{S}_i$ can collapse her multi-exponentiations $a_i^x b_i^y$ into one exponentiation of the form $a_i^{x+z_i y}$.

Using these optimizations, $\mathcal{U}$ performs $8\sum_{j=1,j\neq i}^{l}(\lceil\sqrt{|L_j|}\,\rceil)+l+2$ exponentiations in G_q and $\mathcal{S}_i$ performs $7\sum_{j=1,j\neq i}^{l}(\lceil\sqrt{|L_j|}\,\rceil)+2l+6$ exponentiations. A total of $4\sum_{j=1,j\neq i}^{l}(\lceil\sqrt{|L_j|}\,\rceil)+3$ elements in G_q and $5\sum_{j=1,j\neq i}^{l}(\lceil\sqrt{|L_j|}\,\rceil)+(l+5)$ elements in $\mathbb{Z}_q$ are communicated. For an example value $l=100$ and regardless of the construction of G_q. For blacklists $L_1,\ldots,L_l$ of more that 20 entries each, our blacklist technique is more efficient than the parallel execution of a NOT proof [5, Section 3.4.1] for each list entry.

5 Extensions and Variations

Abuse of any one service in practice may not necessitate banning the abuser from the entire system. In some cases, it may suffice to ban the abuser either from accessing just that service or from accessing a subset of all services. The former can be accommodated by blacklisting the user's public key, the latter by giving users different batches of pseudonyms for use at different service providers. Furthermore, users can be banned only temporarily by deleting their blacklisted numbers from the blacklists at a later stage.

By employing Brands' issuing protocol [5, Section 4.5.2], we enable the so-called refreshing of credentials [5, pp 190-191]. For example, if $\mathcal{U}$ loses the secret key of some of her pseudonyms, she could get a fresh set of pseudonyms with the same encoded values by refreshing one of her previous pseudonyms; in order to avoid linkability at this time, one of her old pseudonyms could be set aside to allow the bootstrapping of other pseudonyms with the same encoded values.

The complexity of steps 2a-2d of our blacklist protocol[4] is linear only in the number of multiplications for calculating the coefficients $a_{i,j}$. The number of exponentiations and the size of the communication are sublinear in the length of the blacklist. More precisely, $\mathcal{U}$ performs $8\lceil\sqrt{|L_j|}\,\rceil$ exponentiations in G_q and $\mathcal{S}_i$ performs $7\lceil\sqrt{|L_j|}\,\rceil+3$ exponentiations. A total of $4\lceil\sqrt{|L_j|}\,\rceil$ elements in G_q and $5\lceil\sqrt{|L_j|}\,\rceil+2$ elements in $\mathbb{Z}_q$ are communicated. On top, the protocol can be transformed in an equally efficient protocol for proving than an element is on a whitelist. For this, equation 3 of step 2d is replaced by the following equation: $((0,\omega_1)=\mathrm{rep}_{(a_i,b_i)}C_{v_1}\ \vee\ \ldots\ \vee\ (0,\omega_m)=\mathrm{rep}_{(a_i,b_i)}C_{v_m})$.

Our blacklisting technique can be adapted to fit any homomorphic commitment scheme for which similar zero-knowledge proofs are available. Among others, it can be used with the RSAREP scheme of Brands [5, Section 2.3.3] and the integer commitment scheme of Damgård and Fujisaki [22]. Note that the latter does not support Brands' NOT-proof. Instead, the NOT relation must be demonstrated by proving a statement $[(x\geq 1)\vee(x\leq -1)]$. This can be achieved in constant time using well-known techniques [25, 19]. In both cases, the resulting zero-knowledge proof protocols require $O(|L|^{1/2})$ exponentiations from both parties and $O(|L|^{1/2})$ communicated values.

[4] Equation 1 of step 2d can be omitted for a proof that $d\notin L$ without d having to be encoded into a credential.

References

1. E. Bangerter, J. Camenisch, and A. Lysyanskaya. A cryptographic framework for the controlled release of certified data. In *IWSP*, 2004.
2. Mihir Bellare, Juan A. Garay, and Tal Rabin. Fast batch verification for modular exponentiation and digital signatures. In *EUROCRYPT*, pages 236–250, 1998.
3. S. Brands. Untraceable off-line cash in wallets with observers. In *CRYPTO*, 1993.
4. S. Brands, L. Demuynck, and B. De Decker. A pract. system for globally revoking the unlinkable pseudonyms of unknown users. Technical report, K.U.Leuven, 2006.
5. S. A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, Cambridge, MA, USA, 2000.
6. E. F. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *ACM Conference on Computer and Communications Security*, pages 132–145, 2004.
7. E. F. Brickell, P. Gemmell, and D. W. Kravitz. Trustee-based tracing extensions to anonymous cash and the making of anonymous change. In *SODA*, 1995.
8. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT*, 2001.
9. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *SCN*, pages 268–289, 2002.
10. J. Camenisch, U. M. Maurer, and M. Stadler. Digital payment systems with passive anonymity-revoking trustees. *Journal of Computer Security*, 5(1):69–90, 1997.
11. Jan Camenisch. *Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem*. PhD thesis, ETH Zurich, 1998.
12. Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In *EUROCRYPT*, pages 302–321, 2005.
13. Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *CRYPTO*, pages 61–76, 2002.
14. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO*, pages 56–72, 2004.
15. Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *CRYPTO*, pages 126–144, 2003.
16. D. Chaum. Blind signatures for untraceable payments. In *CRYPTO*, 1982.
17. D. Chaum and T. Pedersen. Wallet databases with observers. In *CRYPTO*, 1992.
18. David Chaum. Blind signature system. In *CRYPTO*, page 153, 1983.
19. R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO*, pages 174–187, 1994.
20. Ronald Cramer and Torben P. Pedersen. Improved privacy in wallets with observers (extended abstract). In *EUROCRYPT*, pages 329–343, 1993.
21. Ivan Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *EUROCRYPT*, pages 418–430, 2000.
22. Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *ASIACRYPT*, pages 125–142, 2002.
23. George I. Davida, Yair Frankel, Yiannis Tsiounis, and Moti Yung. Anonymity control in e-cash systems. In *Financial Cryptography*, pages 1–16, 1997.
24. Markus Jakobsson and Moti Yung. Distributed "magic ink" signatures. In *EUROCRYPT*, pages 450–464, 1997.
25. Helger Lipmaa. Statistical zero-knowledge proofs from diophantine equations.
26. L. Nguyen. Accumulators from bilin. pairings and applications. In *CT-RSA*, 2005.
27. Markus Stadler, Jean-Marc Piveteau, and Jan Camenisch. Fair blind signatures. In *EUROCRYPT*, pages 209–219, 1995.
28. Victor K. Wei. More compact e-cash with efficient coin tracing. Cryptology ePrint Archive, Report 2005/411, 2005. `http://eprint.iacr.org/`.