Formalizing Anonymous Blacklisting Systems[1,2]
(Extended Version)

Ryan Henry and Ian Goldberg
Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada N2L 3G1
{rhenry,iang}@cs.uwaterloo.ca

***Abstract*—Anonymous communications networks, such as Tor, help to solve the real and important problem of enabling users to communicate privately over the Internet. However, in doing so, anonymous communications networks introduce an entirely new problem for the service providers—such as websites, IRC networks or mail servers—with which these users interact; in particular, since all anonymous users look alike, there is no way for the service providers to hold individual misbehaving anonymous users accountable for their actions. Recent research efforts have focused on using anonymous blacklisting systems (which are sometimes called *anonymous revocation systems*) to empower service providers with the ability to revoke access from abusive anonymous users. In contrast to revocable anonymity systems, which enable some trusted third party to deanonymize users, anonymous blacklisting systems provide users with a way to authenticate anonymously with a service provider, while enabling the service provider to revoke access from any users that misbehave, *without* revealing their identities. In this paper, we introduce the anonymous blacklisting problem and survey the literature on anonymous blacklisting systems, comparing and contrasting the architecture of various existing schemes, and discussing the tradeoffs inherent with each design. The literature on anonymous blacklisting systems lacks a unified set of definitions; each scheme operates under different trust assumptions and provides different security and privacy guarantees. Therefore, before we discuss the existing approaches in detail, we first propose a formal definition for anonymous blacklisting systems, and a set of security and privacy properties that these systems should possess. We also outline a set of new performance requirements that anonymous blacklisting systems should satisfy to maximize their potential for real-world adoption, and give formal definitions for several optional features already supported by some schemes in the literature.**

***Keywords*-privacy enhancing technologies; anonymity; authentication; anonymous blacklisting; privacy-enhanced revocation.**

I. Introduction

Anonymous communications networks help to solve the real and important problem of enabling users to communicate privately over the Internet. The largest deployed anonymous communications network is a worldwide-distributed network of about 2000 volunteer-run *relays* called Tor [20], [43]. On an average day, Tor currently helps to protect between 100,000 and 300,000 privacy-conscious Internet users located in hundreds of countries around the world [32]. These users first connect to a *directory server* to obtain the list of online relays. They then form a random *circuit* (i.e., a path through some subset of the relays in the network) through which they route their communications. A typical Tor circuit passes through three relays; the last relay in the circuit is the *exit relay*. Before a packet is sent over the circuit, it is first encrypted in several layers, with each layer containing only the routing information necessary to deliver that packet to the next relay in the circuit (and eventually to its final destination). Each relay then strips off one layer of encryption and forwards the resulting packet on to the next. When a packet finally reaches the end of the circuit, the exit relay strips off the last layer of encryption and forwards the plaintext packet to its final destination. This approach, called *onion routing*, prevents a local adversary (who may be watching the user's Internet connection) from learning with which service providers—such as websites, IRC networks or mail servers—the user is interacting. It also prevents those service providers from learning the user's identity, location and IP address. The privacy that Tor provides serves many important purposes, as elaborated on the Tor Project website [44]: journalists use Tor to protect their sources; oppressed citizens use Tor to bypass government-level censorship; law enforcement agencies use Tor to protect the integrity of their investigations; ordinary Internet users use Tor to protect themselves against irresponsible corporations, marketers, and identity thieves. Nonetheless, 'a few bad onions' take advantage of this anonymity for nefarious activities such as anonymously harassing users, trolling forums and chat rooms, and committing cyber-vandalism. In response, several service providers now block access from Tor exit nodes in order to prevent anonymous users from participating in, or contributing content to, their online communities; notable examples include Wikipedia[1], Slashdot[2], and most major IRC networks [42]. This motivates the following question:

> *How can service providers on the Internet allow anonymous access while protecting themselves against abuse by misbehaving anonymous users?*

Recent research efforts have focused on using **anonymous blacklisting systems** (which are sometimes called

[1]This is the extended version of [24].

[2]Prior to 28/03/2011, this paper was titled 'A Survey of Anonymous Blacklisting Systems'.

[1]http://www.wikipedia.org/

[2]http://slashdot.org/

anonymous revocation systems) to provide an answer to this question. As opposed to revocable anonymity systems [30], which enable some trusted third party (TTP) to deanonymize users, anonymous blacklisting systems provide users with a way to authenticate anonymously with a service provider (like Wikipedia), while enabling the service provider to revoke access from any users that misbehave. (The key difference is that in anonymous blacklisting systems the user is never deanonymized; neither the service provider, nor any other party, ever learns the identity of a revoked user.) This enables the service provider to protect itself against abuse by anonymous users in much the same way as it already protects itself against abuse from nonanonymous users.

The goal of being able to blacklist *individual* anonymous users may strike the reader as a contradiction in terms; however, as we will see, given some minor assumptions this goal is in fact achievable in practice. The primary obstacle that anonymous blacklisting systems must overcome is that of Sybil resistance [21]; i.e., if a single user can assume many distinct identities in the system, then this user can easily circumvent the service provider's attempts to blacklist her. In practice, anonymous blacklisting systems solve this problem by introducing a (conceptual) *whitelist* of registered users to supplement the blacklist of revoked users. In order to authenticate anonymously with a service provider, users must somehow prove that they are indeed on the whitelist of registered users, but not on the blacklist of revoked users. Of course, users must prove this in zero-knowledge so as not to reveal any additional information to the service provider beyond the veracity of the claim. The literature proposes a variety of cryptographic techniques to do this; existing proposals employ revocable anonymous credentials, information theoretically hiding (restricted) blind signatures, and sophisticated zero-knowledge proof (ZKP) techniques. However, simply asserting that all registered users appear on a whitelist merely defines away the Sybil problem; indeed, the registration process that produces this whitelist must itself be resistant to Sybil attacks. To solve this problem, each user is associated with some 'scarce' resource, which serves as that user's identity in the system (see §V). Thus, if two different users share the same scarce resource, then they will also *share a common identity* in the system. Likewise, if a single user has access to two different resources, then this user can assume *two different identities* in the system. Anonymous blacklisting systems therefore rely on assumptions regarding the scarcity of the identifying resource to limit the disruption that any single malicious user can cause.

A recent flurry of research on anonymous blacklisting has resulted in several concrete proposals for anonymous blacklisting systems. Unfortunately, although the various schemes each intend to solve the same problem—or one of several closely related problems—they each operate under different assumptions and no unified security definitions exist. Indeed, no formal definition for anonymous blacklisting systems presently exists in the literature. This makes it difficult to compare and assess the relative merits of existing approaches.

Outline and contributions: In the present paper, we develop the first formal definition for anonymous blacklisting systems (in §I-A). We follow up this definition in §II with a set of security and privacy properties that anonymous blacklisting systems should possess to protect: 1) users' privacy against malicious service providers and third parties (including other malicious users), and 2) service providers against abuse by malicious users. Next, we propose some essential performance requirements for useful anonymous blacklisting systems, and describe some optional functionality that anonymous blacklisting systems may offer (§III and §IV, respectively). A discussion of some potential choices for scarce resources (herein referred to as *unique identifiers*), and the associated advantages and disadvantages of each, follows in §V. In §VI, we observe that all existing anonymous blacklisting systems fit into just three broad categories based on their architecture, and their security, performance, and functionality characteristics; we survey the systems in each category, comparing and contrasting the approaches they use. Paying especially close attention to the category we call *Nymble-like systems* in §VI-B, we offer our own interpretation, which includes several key observations about the architecture of these systems and some new definitions based on these observations. §VII concludes with a summary and a discussion of future research challenges in anonymous blacklisting.

A. Formal definition

We propose the following definition for anonymous blacklisting systems. Our definition makes no assumption regarding the underlying construction as we intend for it to be very general so that it may encompass the wide variety of existing approaches in the literature.

Let $\mathbf{U}$ be a set of *users*, let $\mathbf{ID}$ be a set of *unique identifiers* that differentiate users in $\mathbf{U}$, and, for any time $t \in \mathbb{N}$, let $f_t : \mathbf{U} \to \mathbf{ID}$ be the function that maps a user $\mathrm{U} \in \mathbf{U}$ to its unique identifier $id \in \mathbf{ID}$ at time t. Ideally, f_t would be injective; in practice, however, this is often not possible. For example, some schemes rely on users' IP addresses as their unique identifiers; however, IP addresses are not truly unique. We assume that all parties involved have access to a synchronized clock and that t_{cur} always denotes the current time on that clock. (Precise synchronization is not required; indeed, the ubiquitous Network Time Protocol (NTP) [36] is sufficient for our purposes.)

Definition 1 (Anonymous Blacklisting System)**.** *An **anonymous blacklisting system** is comprised of a triple of sets of participants* $(\mathbf{I}, \mathbf{R}, \mathbf{SP})$*, where*

1) $\mathbf{I}$ *is the set of **issuing authorities**,*
2) $\mathbf{R}$ *is the set of **revocation authorities**, and*
3) $\mathbf{SP}$ *is the set of **service providers**,*

and a pair of spaces $(\mathbf{C}, \mathbf{A})$*, where*

1) $\mathbf{C}$ *is the space of **access credentials**, and*
2) $\mathbf{A}$ *is the space of **authentication tokens**.*

At any given time, each service provider SP $\in$ $\mathbf{SP}$ is associated with a blacklist $\mathcal{B}_{SP} \in \mathbf{BL}$, where $\mathbf{BL}$ is the set of subsets of $\mathbf{A}$.

The system has five probabilistic polynomial-time algorithms ($\mathtt{Reg}, \mathtt{Ext}, \mathtt{Auth}, \mathtt{Rev}, \mathtt{Aud}$) *parameterized by a security parameter κ, where*

1) $\mathtt{Reg} : \mathbf{U} \times \{0,1\}^\kappa \to \mathbf{C}$ *is the* ***registration protocol****.*
   This protocol takes as input a user $\mathrm{U} \in \mathbf{U}$ and a random bit string $r \in \{0,1\}^\kappa$; it outputs an ***access credential*** *$c \in \mathbf{C}$. This credential is valid from the current time t_{cur} until some future time $t_{exp} \leq t_{cur} + \Delta_t$, where Δ_t is a system parameter that specifies the maximum lifetime of a valid access credential.*
2) $\mathtt{Ext} : \mathbf{C} \times \mathbb{N} \times \mathbf{SP} \times \{0,1\}^\kappa \to \mathbf{A} \cup \{\perp\}$ *is the* ***token extraction protocol****.*
   This protocol takes as input an access credential $c \in \mathbf{C}$, a time $t \in \mathbb{N}$, a service provider $SP \in \mathbf{SP}$, and a random bit string $r \in \{0,1\}^\kappa$. It outputs an authentication token $a_{(c,t,SP)} \in \mathbf{A}$ if $t_{cur} \leq t \leq t_{exp}$ (where t_{exp} is the expiration time of c); otherwise, it outputs $\perp$.
3) $\mathtt{Auth} : \mathbf{A} \times \mathbf{SP} \times \mathbb{N} \to \{\mathtt{true}, \mathtt{false}\}$ *is the* ***authentication protocol****.*
   This protocol takes as input an authentication token $a_{(c,t,SP)} \in \mathbf{A}$, a service provider $SP' \in \mathbf{SP}$ and the current time $t_{cur} \in \mathbb{N}$; it outputs a boolean value $b \in \{\mathtt{true}, \mathtt{false}\}$ ($\mathtt{true}$ if and only if $t = t_{cur}$, $SP = SP'$ and $c \neq c'$ for any $a_{(c',t',SP')} \in \mathcal{B}_{SP'}$).
4) $\mathtt{Rev} : \mathbf{A} \times \mathbf{BL} \to \mathbf{BL} \cup \{\perp\}$ *is the* ***revocation protocol****.*
   This protocol takes as input an authentication token $a_{(c,t,SP)}$ and a blacklist $\mathcal{B}_{SP'} \in \mathbf{BL}$. It outputs a blacklist $\mathcal{B}'_{SP'} = \mathcal{B}_{SP'} \cup \{a_{(c,t,SP')}\} \in \mathbf{BL}$ if $SP = SP'$; otherwise, it outputs $\perp$.
5) $\mathtt{Aud} : \mathbf{C} \times \mathbf{BL} \to \{\mathtt{true}, \mathtt{false}\}$ *is the* ***blacklist audit protocol****.*
   This protocol takes as input an authentication token $a_{(c,t,SP)} \in \mathbf{A}$ and a blacklist $\mathcal{B}_{SP} \in \mathbf{BL}$. It outputs a boolean value $b \in \{\mathtt{true}, \mathtt{false}\}$, which is $\mathtt{true}$ if and only if $c = c'$ for some $a_{(c',t',SP)} \in \mathcal{B}_{SP}$.

A user $\mathrm{U} \in \mathbf{U}$ may connect to an issuing authority $I \in \mathbf{I}$ (or a subset of issuing authorities $\mathbf{I}' \subseteq \mathbf{I}$) at any time $t_{cur} \in \mathbb{N}$. Once connected, U invokes $\mathtt{Reg}(\mathrm{U}, r)$ to obtain an access credential $c \in \mathbf{C}$, which is valid until some future time $t_{exp} \leq t_{cur} + \Delta_t$. Typically, if U wishes to access the services offered by $SP \in \mathbf{SP}$ at time $t \in \mathbb{N}$ with $t_{cur} \leq t \leq t_{exp}$, she invokes $\mathtt{Ext}(c, t, SP, r)$ to obtain an authentication token $a_{(c,t,SP)} \in \mathbf{A}$. Depending on the system's architecture, this step might be run locally by U (e.g. [31], [45]–[48]), may be executed jointly by U and SP (e.g., [40], [41]), or may require U to connect to an issuing authority (e.g. [25], [26], [29], [50]–[52]). U also checks that $\mathtt{Aud}(c, \mathcal{B}_{SP})$ is $\mathtt{false}$. If not, U is revoked from SP; otherwise, she connects to SP and invokes $\mathtt{Auth}(a_{(c,t,SP)}, SP, t_{cur})$. At this point, U may perform some action at SP. In the event that SP determines that U's actions constitute abuse of its services, it can contact a revocation authority $R \in \mathbf{R}$ and invoke $\mathtt{Rev}(a_{(c,t,SP)}, \mathcal{B}_{SP})$ to have U's access token added to its blacklist; in some architectures (e.g., [45]–[48]), $R = SP$.

An anonymous blacklisting system is **secure** if it satisfies each of the security properties we propose in the following section.

II. Security notions

We formally define each security property in terms of a series of security games played by a challenger $\mathcal{C}$ against a probabilistic polynomial-time adversary $\mathcal{A}$. Each game is parameterized by a security parameter κ. A function $\epsilon(\cdot)$ is *negligible* if, for all $c > 0$ there exists a κ_0 such that $\epsilon(\kappa) < 1/\kappa^c$ for all $\kappa > \kappa_0$. An event occurs with *negligible probability* (resp. *overwhelming probability*) if the probability that the event occurs is bounded above by $\epsilon(\kappa)$ (resp. below by $1 - \epsilon(\kappa)$), where ϵ is a negligible function and κ is an appropriate security parameter.

If $\mathcal{A}$ controls a server, this means that $\mathcal{A}$ knows all of that server's secrets and may direct that server's actions. In particular, $\mathcal{A}$ may direct the server to deviate from the established protocols. If it is not explicitly stated that a particular server is under $\mathcal{A}$'s control, then it is implicitly assumed that this server behaves honestly. (Threshold entities are an exception to this rule: we always assume that $\mathcal{A}$ controls at least one fewer than the threshold number of any threshold entity.) We also consider a set $\mathbf{U}$ of users that is controlled by $\mathcal{C}$; in each game, $\mathcal{A}$ is permitted to compromise some subset of users in $\mathbf{U}$, learning all of those users' secrets and controlling their future actions.

We make no claims as the originality of the security properties in this section; each has been adapted from existing definitions in the literature. In some instances, we propose here the first formal definition of a security property, while in other instances we merely propose a definition that is more general than those already found in the literature (e.g., where the original definitions were made with concrete system architectures in mind). We believe it is important to have a universal set of formal security definitions against which one can judge several different schemes. Taken together, the security and privacy properties we propose attempt to capture the necessary and sufficient properties for anonymous blacklisting. In particular, a violation of any one property may have a detrimental effect on user privacy, or the ability to blacklist abusive users. Conversely, if all of the properties hold, then the system guarantees all of the essential properties that we intuitively desire in an anonymous blacklisting system.

A. Correctness

Definition 2 (Correctness)**.** *An anonymous blacklisting system is* ***correct*** *if, with overwhelming probability*[3]*, an honest*

[3]We cautiously define correctness in terms of overwhelming probabilities because, with negligible probability, the one-way functions used in some schemes can have collisions. Thus, an authentication request by a non-revoked user may fail because of a collision with some revoked user.

service provider will accept any authentication token from a non-revoked user, as long as it is properly generated according to the established protocols of the system.

B. Misauthentication resistance

Informally, we define *misauthentication resistance*[4] as follows: with overwhelming probability, verification should succeed *only* on those authentication tokens that are the result of a user correctly executing the established protocols. Note that our definition of misauthentication resistance does not try to capture the notion of revocability (see Definition 6). Formally, we define misauthentication resistance in terms of Security Game 1.

Security Game 1 (Misauthentication resistance)

Adversary: $\mathcal{A}$ controls SP_2 and the revocation authorities.

—

(**Probing phase**): $\mathcal{A}$ arbitrarily compromises any subset of $\mathbf{U}$, and invokes **`Reg`**, **`Ext`**, and **`Auth`** for these users as desired. $\mathcal{A}$ may invoke **`Rev`** on any authentication token output by **`Ext`**.

(**End phase**): $\mathcal{A}$ invokes $\mathtt{Auth}(a, \mathrm{SP}_1, t_{\mathrm{cur}})$ for some a. $\mathcal{A}$ wins the game if and only if **`Auth`** returns `true` and

1) a is not an output from $\mathtt{Ext}(c, t_{\mathrm{cur}}, \mathrm{SP}_1, r)$ for any c output by **`Reg`**, or
2) $t_{\mathrm{exp}} < t_{\mathrm{cur}}$, where t_{exp} is the expiration time of c.

Definition 3 (Misauthentication resistance)**.** *An anonymous blacklisting system provides **misauthentication resistance** if no probabilistic polynomial time adversary can win Security Game 1 with non-negligible probability.*

C. Backward anonymity

Informally, we define *backward anonymity*[5] as follows: given an authentication token from a member of some set of at least two users, it should be infeasible for an attacker to determine which user that authentication token was issued to, with more than negligible advantage over a random guess. Moreover, this property should hold even if some service providers later revoke any subsets of these users. Formally, we define backward anonymity in terms of Security Game 2.

[4]In several prior works, the authors rely either explicitly or implicitly on **unforgeability** (of *authentication tokens* [25], [26], [29], [40], [41], [50]–[52], *(non)-membership proofs* [6]–[9], [31], [45]–[49], or *digital signatures* [11], [15], [25], [26], [31], [34], [35], [37]–[39], [48]) to provide misauthentication resistance; the term misauthentication resistance was previously used in [45]–[47], [49], [53].

[5]In [48], [50]–[52], the authors refer to this property as **backward unlinkability**; the term backward anonymity is used in [25], [26], [29], [40]. Some schemes [9], [31], [45]–[47], [50], [51] implicitly combine the notions of backward anonymity, unlinkability (Definition 5) and revocation auditability (Definition 7) to provide a property they call **anonymity**; others [11], [37]–[39], [53] provide a similar notion of anonymity for non-revoked users only.

Security Game 2 (Backward anonymity)

Adversary: $\mathcal{A}$ controls SP_2, the issuing authorities, and the revocation authorities.

—

(**First probing phase**): $\mathcal{A}$ arbitrarily compromises any subset of $\mathbf{U}$ of size at most $|\mathbf{U}| - 2$, and invokes **`Reg`**, **`Ext`** and **`Auth`** for these users as desired. $\mathcal{A}$ may invoke **`Rev`** on any authentication token presented to SP_1 or SP_2.

(**Challenge phase**): $\mathcal{C}$ invokes **`Reg`**, **`Ext`** and **`Auth`** to authenticate each uncompromised user u_i at SP_2 (in a random order); u_i's authentication token is a_i and the set of authentication tokens is T. $\mathcal{A}$ learns T, but does not learn with which user each token in T is associated.

(**Second probing phase**): $\mathcal{A}$ may ask $\mathcal{C}$ to authenticate any user at SP_2 according to any strategy, or may revoke the owner of any authentication token presented to SP_2.

(**End phase**): $\mathcal{A}$ chooses a tuple $(u_i, a_j) \in \mathbf{U} \times T$. $\mathcal{A}$ wins the game if and only if $i = j$.

Definition 4 (Backward anonymity)**.** *An anonymous blacklisting system provides **backward anonymity** if no probabilistic polynomial time adversary can win Security Game 2 with probability non-negligibly greater than $1/2$.*

D. Unlinkability

Informally, we define *unlinkability*[6] as follows: given two or more authentication tokens from members of some set of at least two users, it should be infeasible for an attacker to distinguish authentications by the same user from those by different users, with more than negligible advantage over a random guess. This property should hold both within a single service provider and across multiple service providers. Formally, we define unlinkability in terms of Security Game 3.

Definition 5 (Unlinkability)**.** *An anonymous blacklisting system provides **unlinkability** if no probabilistic polynomial time adversary can win Security Game 3 with probability non-negligibly greater than $1/2$. It provides **strong unlinkability** if it provides unlinkability when $\mathcal{A}$ may additionally compromise u_0 and u_1 in the probing phase.*

Remarks.

1) Together, backward anonymity and unlinkability imply **anonymity** for non-revoked users; adding *revocation auditability* (Definition 7) makes this full anonymity for all users (revoked or otherwise).
2) Roger Dingledine raised the following question [19]: if some subset of users chooses to use the system pseudonymously (e.g., by participating in online chat rooms or logging in to a website with a persistent

[6]The term unlinkability is used in [6]–[9], [11], [37]–[41]. In [34], [35] unlinkability applies only to pseudonyms at different service providers, while in [52] this notion of unlinkability between service providers is treated separately and called **indistinguishability**.

Security Game 3 (Unlinkability)

Adversary: $\mathcal{A}$ controls SP_2 and the issuing authorities.

—

(**First challenge phase**): $\mathcal{C}$ invokes **`Reg`**, **`Ext`** and **`Auth`** for $u_0, u_1 \in \mathbf{U}$ to authenticate both users at SP_1 and SP_2; u_i's authentication token from SP_j is $a_{(i,j)}$ and the set of tokens is T. $\mathcal{A}$ learns T.

(**Probing phase**): $\mathcal{A}$ arbitrarily compromises any subset of $\mathbf{U} - \{u_0, u_1\}$, and invokes **`Reg`**, **`Ext`** and **`Auth`** for these users as desired. $\mathcal{A}$ may ask $\mathcal{C}$ to authenticate any user at SP_2 according to any strategy, and may invoke **`Rev`** for SP_1 or SP_2 on any authentication token issued to a compromised user.

(**Second challenge phase**): $\mathcal{C}$ flips two fair coins to obtain bits $b, c \in_R \{0, 1\}$ and authenticates u_b with SP_{c+1}.

(**End phase**): $\mathcal{A}$ chooses an authentication token $a_{(i,j)} \in T$. $\mathcal{A}$ wins the game if and only if $i = b$.

alias), what is the privacy impact on the other users? With our definitions, pseudonymous users can be considered to be under adversarial control; thus, if the system provides backward anonymity and unlinkability, then there is essentially no impact on user privacy.

E. Revocability

Informally, we define *revocability*[7] as follows: given an authentication token issued to some anonymous user, it should be possible for a service provider to have the user's access revoked. This mechanism should have the property that no coalition of revoked (or unregistered) users should be able to authenticate with the service provider. Revocability is related to—but distinct from—the previously introduced notion of misauthentication resistance. Formally, we define revocability in terms of Security Game 4.

Definition 6 (Revocability)**.** *An anonymous blacklisting system provides **revocability** if no probabilistic polynomial time adversary can win Security Game 4 with non-negligible probability.*

Remark. Several schemes [25], [26], [29], [45], [46], [50]–[52] use a notion called **(uncircumventable) forward linkability** to achieve revocability; in this scenario, a third party computes a trapdoor computation on an authentication token to enable the service provider to recognize future authentication tokens from the same user. This highlights the importance of our next security property: *revocation auditability*.

[7]In several prior works, the authors define a similar notion called **blacklistability** [31], [45]–[47], [49]–[51]; in other works, the authors define revocability as the ability to deanonymize a misbehaving user by recovering their unique identifier or linking their prior actions [11], [37]–[39], [53] (thus violating the backward anonymity or unlinkability properties). In [48], the authors use the terms **accountability** and revocability interchangeably.

Security Game 4 (Revocability)

Adversary: $\mathcal{A}$ controls SP_2.

—

(**Probing phase**): $\mathcal{A}$ arbitrarily compromises any subset of $\mathbf{U}$, and invokes **`Reg`**, **`Ext`** and **`Auth`** for these users as desired. $\mathcal{A}$ may invoke **`Rev`** for SP_1 or SP_2 on any authentication token issued to a compromised user. At the end of this phase, $\mathcal{A}$ authenticates each compromised and non-revoked user with SP_1; the set of authentication tokens is T.

(**Challenge phase**): $\mathcal{C}$ invokes **`Rev`** for SP_1 on each authentication token in T.

(**End phase**): $\mathcal{A}$ invokes **`Auth`** for SP_1. $\mathcal{A}$ wins the game if and only if **`Auth`** returns `true`.

F. Revocation auditability

Informally, we define *revocation auditability*[8] as follows: a user should be able to check her revocation status at a service provider before trying to authenticate. If revoked, the user can then disconnect without revealing any potentially sensitive information. This is important to avoid the situation in which a malicious service provider accepts an authentication request from a revoked user, thus reducing the size of that user's anonymity set without her knowledge. (In the extreme case, for example when a scheme achieves revocability by using uncircumventable forward linkability, this attack enables the service provider to link all of the user's actions.) Formally, we define revocation auditability in terms of Security Game 5.

Definition 7 (Revocation auditability)**.** *An anonymous blacklisting system provides **revocation auditability** if no probabilistic polynomial time adversary can win Security Game 5 with probability non-negligibly greater than* $1/2$.

G. Non-frameability

Informally, we define *non-frameability* as follows: no coalition of third parties should be able to get an honest user revoked from a service provider.[9] This definition assumes that no coalition contains a revocation authority or a malicious user that shares the victim's unique identifier; i.e., that $f_t(u_i) \neq f_t(u_j)$ for any pair (u_i, u_j) where u_i is an honest user and u_j is a coalition member. Formally, we define non-frameability in terms of Security Game 6.

Definition 8 (Non-frameability)**.** *An anonymous blacklisting system provides **non-frameability** if no probabilistic polynomial time adversary can win Security Game 6 with non-negligible probability.*

[8]This notion of revocation auditability is used explicitly in [48], [50], [51], and implicitly in [25], [26], [29], [31], [45]–[47], [52]. In [52] the authors refer to this property as **knowledgeability**. In [40], [41] the service provider periodically **audits** the clients to ensure that they have a valid subscription to its services; this idea of auditing clients is not to be confused with revocation auditability.

[9]A coalition of third parties may be any subset of the following: the issuing authorities, other service providers, and all other users.

Security Game 5 (Revocation auditability)

Adversary: $\mathcal{A}$ controls SP_2 and the issuing authorities.

—

(**First challenge phase**): $\mathcal{C}$ invokes **Reg**, **Ext** and **Auth** for $u_0, u_1 \in \mathbf{U}$ to authenticate both users at SP_2; u_i's authentication token is a_i and the set of authentication tokens is T. $\mathcal{A}$ learns T.

(**Probing phase**): $\mathcal{A}$ arbitrarily compromises any subset of $\mathbf{U} - \{u_0, u_1\}$, and invokes **Reg**, **Ext** and **Auth** for these users as desired. $\mathcal{A}$ may ask $\mathcal{C}$ to authenticate any user at SP_2 according to any strategy, and may invoke **Rev** for SP_1 or SP_2 on any authentication token.

(**Second challenge phase**): $\mathcal{C}$ flips a fair coin to obtain $b \in_R \{0, 1\}$ and attempts to authenticate u_b with SP_2.

(**End phase**): $\mathcal{A}$ chooses an authentication token a_i from the first challenge phase. $\mathcal{A}$ wins the game if and only if $i = b$ and **Aud** returns `false` for the credentials of both u_0 and u_1 at SP_2.

Security Game 6 (Non-frameability)

Adversary: $\mathcal{A}$ controls SP_2 and the issuing authorities.

—

(**First challenge phase**): $\mathcal{C}$ invokes **Reg**, **Ext** and **Auth** to authenticate each $u_i \in \mathbf{U}$ at SP_1 and SP_2; the set of tokens is T. $\mathcal{A}$ learns T.

(**Probing phase**): $\mathcal{A}$ arbitrarily compromises any proper subset of $\mathbf{U}$, and invokes **Reg**, **Ext** and **Auth** for these users as desired. $\mathcal{A}$ may ask $\mathcal{C}$ to authenticate any user at SP_1 or SP_2 according to any strategy; each resulting authentication token is added to T. $\mathcal{A}$ may revoke any authentication token presented to SP_2 and may revoke any (possibly forged) authentication token not in T from SP_1.

(**Second challenge phase**): For each uncompromised user, $\mathcal{C}$ attempts to authenticate that user with SP_1.

(**End phase**): $\mathcal{A}$ wins the game if there exists an uncompromised user u_i such that either: 1) **Aud** returns `true` for u_i's credential, or 2) **Auth** returns `false` for u_i's authentication request.

III. Performance notions

The security requirements outlined in the previous section are necessary but not sufficient for a useful anonymous blacklisting system. We believe that all anonymous blacklisting solutions should additionally possess certain crucial performance characteristics. Our requirements contain a bias towards producing an extremely lightweight component for the service provider; we do this because many service providers appear to consider the input of anonymous users to be of generally low quality, and are thus content to block access from anonymous communications networks. To maximize the system's potential for real-world adoption it is therefore important to cap the level of computational and communications overhead for the service provider; we call this property *verifier efficiency*. The system should also take care to avoid affecting the observed interaction latency between the user and the service provider by ensuring that any computation that the user must perform to authenticate is independent of the blacklist size, and by forcing significant computations to be precomputable. We call this latter property *user efficiency*.

A. Verifier efficiency

Informally, we define *verifier efficiency* as follows: the benefits that the system brings to a service provider must clearly compensate for any burden placed on the service provider. As such, the service provider's cost in terms of communications and computational complexity, storage requirements, hardware requirements, and maintenance costs, should be extremely low.

Definition 9 (Verifier efficient)**.** *An anonymous blacklisting system is **verifier efficient** if the cost of the system for a service provider is low. In particular,*

1) *verifying authentication requests and revoking abusive users have predictable running times and bandwidth requirements that are small enough so that the cost to the verifier to service a user using the blacklisting system is not much greater than the cost to service a user not using it,*
2) *the time required for revocation is short enough so that the system can keep up with the expected rate of revocations, and*
3) *the service provider does not require any specialized hardware.*

B. User efficiency

Informally, we define *user efficiency* as follows: the system should be accessible to all users and should not negatively affect users' online experiences. We note that one of Tor's target audiences is citizens of countries with oppressive leadership that censors access to certain information. Unfortunately, users from these countries may not have access to state-of-the-art computers or high-bandwidth Internet connections. Thus, requiring the user to run specialized hardware like a trusted platform module (TPM), consume large amounts of bandwidth, or to solve computational puzzles, limits the system's usefulness.

Definition 10 (User efficient)**.** *An anonymous blacklisting system is **user efficient** if the cost for the user to use the system is low; in particular,*

1) *the authentication process has a predictable running time and bandwidth requirements that do not add noticeable latency to users' interactive requests, and*
2) *the user does not require any specialized hardware.*

IV. Functionality notions

Some anonymous blacklisting systems may offer other useful features. We propose formal definitions for some optional features already found in the literature.

A. Subjective and objective blacklisting

Most anonymous blacklisting systems currently support only *subjective revocation*, wherein the service provider decides subjectively and unilaterally which users to revoke. With *objective blacklisting systems*, however, a revocation authority can only revoke a user if she violates a *contract* that specifies the service provider's terms of service [37]–[39]. In this case, there exists a function $M : \{0,1\}^* \to \{\texttt{true}, \texttt{false}, \perp\}$, called a *morality function*, which takes as input a bit string `proof` describing the user's behaviour and outputs a boolean value (or $\perp$). The output indicates if the behaviour described in `proof` violates the contract (if $\perp$ is returned, this means that subjective judgment is required to make a final determination).

Definition 11 ((Strictly) objective blacklisting)**.** *An anonymous blacklisting system is said to **enforce a contract** on a service provider if the revocation authority can only revoke access from users when given a string* `proof` *(that is provably associated with an authentication token) for which* $M(\texttt{proof}) \neq \texttt{false}$*. In this case, the system is said to support **objective blacklisting**; if the range of* M *is restricted to the set* $\{\texttt{true}, \texttt{false}\}$ *then the system is said to enforce **strictly objective blacklisting**.*

The system is said to enforce **contract-based revocation** if the enforced contract is encoded in each authentication token, and is known to and agreed upon by both the user and the service provider at the time that the token extraction protocol is run. It provides **contract auditability** if the user knows the precise semantics of the morality function (thus enabling the user to determine if a specific activity constitutes misbehaviour before deciding whether to engage in it).

Morality functions: Online forum software routinely incorporates mechanisms for automatically filtering unwanted behaviour; for example, they often filter vulgar language, scripts, external links and unsolicited advertisements. Nonetheless, nearly all major online forums also have moderators that enforce their less-tangible rules; this is because it is difficult or infeasible to write an algorithm that can effectively capture certain behaviours. Thus, constructing a useful morality function appears to be a difficult problem to solve in practice; it is therefore unclear how useful objective blacklisting is in many real-world situations.

Most existing approaches provide subjective blacklisting capabilities only; the contractual anonymity system of Schwartz *et al.* [37]–[39] and Lin and Hopper's objective blacklisting extension to Jack [31] appear to be the only exceptions in the literature. However, in [23], we demonstrate how Lin and Hopper's objective blacklisting extension to Jack can be generalized to apply to other similar schemes.

B. Rate-limited access

It is often useful for a service provider to rate limit users' access; this limits the amount of disruption a single user can cause. Many large service providers use rate limiting even for non-anonymous users. We return to the previous example of online forums: to keep spam levels reasonably low, forums often rate limit the posting of new messages.

Definition 12 (Rate limiting)**.** *An anonymous blacklisting system provides **rate limiting** if, for a given interval of time* T*, the number of pairwise mutually unlinkable authentication tokens that a user can use at a given service provider is bounded above by some monotone increasing (in the length of* T*) function.*

Two primary approaches to rate limiting are currently used in the literature:[10]

1) the interval T may be broken up into discrete time-periods $t_1, \ldots, t_k$ such that each user may authenticate once in each time period, or
2) for an interval T, the user may authenticate on any schedule they choose, up to a predefined number of times $k = f(T)$.

C. Blacklist transferability

In some instances, it may be desirable for a service provider to revoke a user's access based on their actions at some *other* service provider. For instance, Wikipedia's sister site Wiktionary[11] may wish to revoke access from users that misbehave in a session with Wikipedia. On the other hand, from a privacy point of view, it it desirable for users to be able to access both of these services *concurrently* and *unlinkably*. This observation motivates our next definition.

Definition 13 (Blacklist transferability)**.** *An anonymous blacklisting system provides **blacklist transferability**[12] if users can authenticate unlinkably and concurrently with two or more service providers, while any one of these service providers can require the user to prove that she is not revoked from another, before granting access.*

V. Unique identifiers

A user's unique identifier plays a crucial role in anonymous blacklisting systems. In most cases, the unique identifier is some scarce resource that a user can prove that she currently possesses. In order to obtain access credentials, each user must first register with an issuing authority by proving possession of her unique resource; if any issuing authority has previously issued credentials to another user with the same resource, then either the user must receive that same credential or the registration process must fail. Without basing credential issuance on a suitable identifier, an anonymous blacklisting system is susceptible to the Sybil attack [21].

This section discusses the various choices of unique identifiers described in the literature. For each choice of unique identifier, we briefly discuss how the user must prove possession of her identifier, and list some advantages

[10] The first strategy is used in [25], [26], [29], [50]–[52] while the second strategy is used in [48]; other schemes do not offer rate limiting [1], [2], [6]–[9], [11], [34], [35], [37]–[39], or propose it as an added feature [27], [31], [45]–[47] and leave details to the implementer.

[11] http://www.wiktionary.org/

[12] This idea was introduced in [45]–[47] as **blacklist (entry) sharing**.

and disadvantages to basing credential issuance on that identifier. This list of identifiers, and the lists of advantages and disadvantages associated with each entry in it, is not exhaustive; future research is likely to propose other suitable alternatives, as well as uncover further advantages and disadvantages associated with the identifiers listed herein.

In our descriptions, we assume that the registration protocol computes credentials deterministically from a user's unique identifier; some schemes (e.g., [1], [2], [6], [7], [11], [27], [31], [34], [35], [45]–[48]) propose a registration protocol that issues credentials based on user-chosen randomness. The credential issuer in this case must keep a log whenever it issues a credential to a user; it must then refuse to issue a second credential to any user who already appears in the log. In the case of a distributed credential issuer, it is important that each issuer possesses an up-to-date copy of the log at all times; indeed, care must be taken to avoid race conditions that occur if a user requests credentials from several issuers concurrently. We also note that this approach to credential issuing does not work well with using IP addresses to identify users, which is the method used by some anonymous blacklisting systems in the literature [25]–[27], [29], [50]–[52]. This is because IP addresses are neither permanent nor unique; some users may regularly obtain new IP addresses via DHCP, and some users may share the same IP address via NAT. In both instances, legitimate users are likely to encounter availability problems.[13]

A. Internet Protocol (IP) addresses (Discussed in [25]–[27], [29], [50]–[52]): A user can prove possession of an IP address by connecting directly (i.e., not through an anonymizing network) to the issuing authority.

Advantages: 1) all Internet users have an IP address, 2) issuing credentials based on IP addresses is easily automated, and 3) IP addresses can provide SPs with revocation capabilities roughly analogous to what they have with non-anonymous users.[14]

Disadvantages: 1) IP addresses are neither permanent (through DHCP users may regularly obtain a new IP address) nor necessarily unique (two or more users may use NAT to access the Internet through a single IP address)[15], and 2) the set of IP addresses is cryptographically small (there are just 2^{32} valid IP addresses in IPv4 and many of these are reserved) and thus care must be taken to protect against brute-force deanonymization attacks.

B. Telephone / SMS numbers (Discussed in [26], [50]): A user can prove possession of a telephone number by self-reporting this number to the issuing authority, who then transmits a nonce (either by placing a phone call or sending an SMS message) to that number; the user then echoes this nonce to the issuing authority to complete the registration process.

Advantages: 1) most telephone numbers are associated with a single individual (or a small group of related individuals); 2) phone numbers are typically associated with an individual for reasonably long periods of time (years, for example).

Disadvantages: 1) not all Internet users (particularly those in developing countries) have a telephone number, 2) there may be prohibitive infrastructure costs associated with this approach, and 3) pay-as-you-go SIM cards enable one to purchase a new phone number for registration purposes, then sell it with all of the minutes intact (thus recouping much of the cost).

C. e-Passports / Enhanced Driver's Licenses (Discussed in [26], [27], [39], [50], [51]): A user can prove possession of an electronic passport (e-passport) or enhanced driver's license (containing a private/public key pair) by proving knowledge of the private key.

Advantages: 1) government-issued identification is strongly bound to a single individual, 2) users have strong legal incentives not to share their private key with others, 3) laws make it difficult to obtain large quantities of passports or driver's licenses, 4) the entire registration process can be performed in zero-knowledge without revealing nontrivial information about the user's identity, and 5) if the private key is the unique identifier, then brute-force attacks are infeasible.

Disadvantages: 1) not all users have (or even live in a country where they can obtain) an electronic passport or enhanced driver's license, 2) users may be hesitant to use their government-issued identification, and 3) governments may be unwilling to issue identification that facilitates proofs of knowledge of the associated private key.

D. Public Key Infrastructure (PKI) (Discussed in [29], [31], [50], [51], [53]): A user can prove possession of the private key associated with a signed identity certificate.

Advantages: 1) the entire registration process can be performed in zero-knowledge and 2) if the private key is the unique identifier, then brute-force attacks are infeasible.

Disadvantages: 1) users must somehow obtain one and only one certificate from a certificate authority—thus, the certificate authority must resort to using one of the other unique identifiers to distinguish users.

E. Trusted Platform Modules (TPM) (Discussed in [8], [9], [29], [37]–[39], [50]–[53]): A user can prove knowledge of the private key associated with a public endorsement key from her computer's trusted platform module (TPM).

Advantages: 1) it is easy to automate the registration process and 2) this unique identifier is strongly bound to a single client machine (provided the manufacturer is

[13] As more Internet services transition to IPv6 [18], these problems may all but vanish; at present, however, they are real problems that one must consider when using IP addresses to identify users.

[14] There are some caveats with this last statement; e.g., no anonymous blacklisting systems presently support IP address blocking at the subnet granularity. In §VII, we discuss this as a potential avenue for future research.

[15] In fact, as pointed out by one of the anonymous reviewers, AfriNIC, the Regional Internet Registry for the entire continent of Africa, administers just six /8 IPv4 prefixes [28] (this yields just over 100 million IP address) to serve a population exceeding 1 billion [54].

trusted).

Disadvantages: 1) not all users (especially those in developing countries) have a trusted platform module in their computer, 2) government-level adversaries may have undue influence over the TPM manufacturers, and 3) users with more money are able to purchase more TPMs (and thus more identities on the system).

While the following two 'identifiers' (currency and proof of work) are not actually unique identifiers at all, we include them here as a potentially viable alternative to using uniquely identifiable resources. In these cases, the scarce resource (currency or computational effort) is traded for a unique identifier (e.g., a blind signature on a random value).

F. Currency (Discussed in [26], [50], [51]): Currency can be used to pay a fee (or make a refundable deposit) for a unique identifier.

Advantages: 1) it is easy to compute the cost for an adversary to perform a Sybil attack.

Disadvantages: 1) this approach requires infrastructure that is capable of accepting monetary payments from users, 2) users may be hesitant to use credit or debit cards, while accepting cash seems difficult, and 3) certain users might not have sufficient money to participate (especially those in developing countries). Others, of course, will have sufficient money to obtain many identities.

G. Puzzles / Proof of Work (Discussed in [26], [27], [50], [51], [53]): Users can solve puzzles, such as CAPTCHAs or computational puzzles, in exchange for a unique identifier.

Advantages: 1) it is easy to automate the registration process and 2) if the users are required to do *useful* work, this could have other fringe benefits.

Disadvantages: 1) some types of proof of work (such as CAPTCHAs) have known problems or weaknesses [10], and 2) computational asymmetry may allow some users to obtain more credentials than others.

Remark 1: It should be the case that getting one credential is plausible (if not easy), but getting two is nigh-impossible. Currency and puzzles clearly do not suffice where Sybil attacks are a realistic threat. This may also be true for TPMs or any of the other unique identifiers we discussed, given high enough stakes. We are not claiming the above identifiers are adequate, but merely that they have been considered in the literature.

Remark 2: In our own work [23], [25], we have utilized users' IP addresses as a unique resource, pointing out that many of the limitations of IP address blocking are present *even when the user is not connecting through an anonymous communications network*, yet IP address blocking is still the *de facto* standard method for revoking access from (unauthenticated) abusive users. This approach seems to work well for situations in which one wishes to provide public access to a service. For private service providers that service only registered members, a better approach is to use some form of PKI or government ID-based registration.

VI. A survey of existing approaches

This section discusses existing anonymous blacklisting systems in the literature. We divide our discussion into three separate categories: the *pseudonym systems*, the *Nymble-like systems*, and the *revocable anonymous credential systems*. In general, the schemes in each category make different security versus performance tradeoffs; at one end of the spectrum, pseudonym systems provide the weakest privacy guarantees by sacrificing unlinkability; however, their simplicity makes them easy to construct and efficient to implement. At the other end of the spectrum are revocable anonymous credential systems, which provide the strongest privacy guarantees (by providing all security properties with no trusted third parties) at the cost of high computational burden on service providers and users of the system. The Nymble-like systems occupy a middle ground between these two classes; they leverage (semi-)trusted third parties to find a compromise between the efficiency of pseudonym systems and the strong privacy guarantees and flexibility of revocable anonymous credential systems.

The rest of this section discusses a cross-section of the schemes from the literature in each of these three classes.

A. Pseudonym Systems

The first class of anonymous blacklisting systems are the **pseudonym systems**. As the name implies, pseudonym systems provide users with pseudonymity instead of full anonymity. That is, a user's identity at a service provider is not linkable back to her real identity (nor are her identities at different service providers linkable with each other), but her individual actions at a particular service provider are all easily linked with each other. Because users interact with a service provider using a persistent pseudonym, revocation is as simple as adding the pseudonym to a blacklist and denying access to any user with a pseudonym on the blacklist. Existing pseudonym systems get their security and privacy properties from one of three primary sources: 1) *private credentials*, 2) *blind signatures*, and 3) *group signatures*.

Table I summarizes the security and privacy guarantees and feature sets of the various pseudonym systems discussed in this section. Note that, by definition, pseudonym systems do not provide unlinkability. Similarly, revocation auditability does not apply to pseudonym systems, since its purpose is to protect revoked users from being granted access and having their actions secretly linked by the service provider.

Pseudonym systems are particularly well suited to situations in which service providers wish to grant pseudonymous or anonymous access to members of a closed community. Schemes based on group signatures offer additional privacy properties that are not possible with schemes based on blind signatures; i.e., non-revoked users are anonymous instead of pseudonymous. However, this added privacy for the users comes at the cost of some additional computational and communications overhead. Nonetheless, constructions in both of these classes are highly practical since no sophisticated showing protocol is required for authentication; hence, our

Scheme	Misauthentication resistance	Unlinkability	Backward anonymity	Revocability	Revocation auditability	Non-frameability	Objective vs. subjective	Rate-limiting	Blacklist transferability	Retroactive revocation	User efficient	Verifier efficient
CE [15]	✔		✔	✔		✔	S			✔		
Damgård [17]	✔		✔	✔		✔	S			✔		
Chen [16]	✔		✔	✔		✔	S			✔		
LRSW [36]	✔		✔	✔		✔	S		✔	✔		
Nym [28]	✔		✔	✔		✔	S			✔	✔	✔
CPG [2]	✔		✔	✔		✔	S			✔	✔	✔
RECAP [40]	✔			✔		✔	O			✔	✔	✔

Table I
THIS TABLE COMPARES THE PROPERTIES OF VARIOUS PSEUDONYM SYSTEMS.

opinion is that in most situations the additional privacy afforded by schemes based on group signatures is worth the additional overhead.

1) Schemes based on private credentials: Chaum [13] proposed pseudonym systems as a way for users to control the transfer of information about themselves between different organizations. To enable this, he proposed that a user first establish a pseudonym with every organization with which she wishes to interact. Then, to transfer information about herself from one organization to another, the user obtains a *credential on a pseudonym* from the first organization, which encodes this information. It then transforms this credential into the "same" credential on one of its other pseudonyms. This enables the user to prove to a second organization that the first organization has certified the information encoded in the credential, without necessarily revealing information about her pseudonym at the first organization. Chaum and Evertse presented the first construction of a pseudonym system based on RSA in the year following Chaum's proposal [14]. Shortly thereafter, Damgård proposed a provably secure (assuming the existence of claw-free functions)—though impractical—scheme based on zero-knowledge proofs [17]. Later, Chen proposed a practical construction for Damgård's model based on discrete logarithms [16].

In her Master's thesis, Lysyanskaya presented a new model for pseudonym systems that incorporates the ability for an organization to revoke access to a credential on a pseudonym [34], [35]; thus, her model makes blacklist transferability possible. That is, SP_1 can require the user to show a credential indicating that she has authorized access to SP_2. If SP_2 later revokes the user, then her credential will also be revoked, thus preventing her from showing it to SP_1 in the future. Moreover, by having SP_1 verify that the user possesses a non-revoked access credential *for* SP_1 *itself*, a fully anonymous scheme can be built by allowing each user to *rerandomize* its pseudonyms between showings. In [11], Camenisch and Lysyanskaya extend the idea to do just that, resulting in the first of the *revocable anonymous credential systems* (see §VI-C).

2) Schemes based on blind signatures: The first use of pseudonym systems specifically as a revocation mechanism appears to be by Holt and Seamons [27]. They proposed Nym as a way to enable the popular collaborative website Wikipedia to revoke access from misbehaving Tor users. Nym does away with much of the sophisticated functionality offered by [11], [14], [16], [17], [34], [35] to build an extremely simple mechanism for users to establish pseudonyms with a service provider. Their scheme was the first to associate each user with a unique identifier (they recommend her IP address or email address). In Nym, users prove possession of their unique identifiers in exchange for blind RSA signatures on (user-specified) random nonces. They later exchange the unblinded signatures for client certificates, which, because of the unconditional unlinkability of blind RSA signatures, are completely unlinkable to their real identities. Abbot *et al.* describe a similar system, called Closed Pseudonymous Groups (CPG), wherein members of some 'real-world' group (for example, students of the same class or subscribers to a service) register pseudonyms to participate in a closed online community [1], [2]. Since pseudonymous access in their system is restricted only to members of a certain 'real-world' group, Abbot *et al.* discuss approaches to revoking a user based on her real-world actions (for example, if she drops the class or lets her membership to the service lapse).

3) Schemes based on group signatures: In 1991, Chaum and van Heyst proposed group signatures, wherein each member of a group can sign any message *on behalf of the group*. Anyone can verify a group signature using the

group's public key, but only a special entity known as the **Revocation Manager** (RM) can determine which group member produced a particular group signature [15]. (Sometimes, the RM is the same entity that distributes private keys, in which case its name is the **Group Manager** (GM).) If the RM is trusted, then group signatures make it easy to construct a closed community in which non-revoked users are fully anonymous (within the anonymity set of all non-revoked members of the group). If a user misbehaves, then the RM can link her past and future actions and thus revoke her anonymity.[16]

Schwartz *et al.* proposed contract-based revocation in their Contractual Anonymity papers [37]–[39]. They leverage ideas from trusted computing to construct a contract-based revocation system, called RECAP, using group signatures as the underlying primitive. In particular, they use remote attestation to allow the user to confirm that the software running on the RM will only deanonymize her in the event that she violates a pre-agreed-upon (by the user and the SP) contract. Their reliance on trusted computing means that user privacy is not arbitrarily entrusted to the RM.

B. Nymble-like Systems

The second class of anonymous blacklisting systems are the **Nymble-like systems**. The Nymble-like systems leverage (semi-)trusted third parties to provide stronger privacy guarantees than the pseudonym systems, without introducing significant overhead to the protocols. They are particularly well suited to situations in which a service provider wishes to make anonymous access to its services publicly available, but has no strong economic incentives to invest significant resources into doing so. However, systems in this class rely on the existence of some additional semi-trusted infrastructure to support them; hence, our opinion is that Nymble-like systems are best deployed in tandem with an anonymous communications network, such as Tor, since this allows the supporting infrastructure to be decoupled from the—and shared among several—service providers.

The Nymble-like system category gets its name from Nymble [29], [50]–[52]. Since Nymble's proposal in 2006 [52], there have been three additional proposals for Nymble-like systems in the literature. However, before discussing these systems we briefly examine a predecessor to Nymble called Unlinkable Serial Transactions, proposed by Syverson *et al.* in 1997 [40], [41].

Unlinkable Serial Transactions (UST) is a protocol for online subscription services that prevents the service provider from tracking the behaviour of a subscriber, while protecting it from abuse due to simultaneous active sessions by a single subscription. UST introduced the concept of having the user authenticate with temporally related—but mutually unlinkable—authentication tokens. In the scheme, the user and the service provider negotiate a blind authentication token that the user later exchanges for services from the service provider. At the end of a user's session, she and the service provider negotiate a new blind authentication token for her next session. Thus, at any given time, the user possesses just one valid and unused token; this prevents a second anonymous user from accessing a user's subscription at the same time as that user. If the user is judged by the service provider to have misbehaved (for example, by attempting to use the same token twice concurrently), then the service provider can revoke the user's access by ending her session without issuing a new authentication token. However, due to UST's token generation method, the scheme provides no way for the service provider to revoke a user if it discovers her misbehaviour *after* her session has ended. This makes UST an unsuitable blacklisting system for many real-world applications.

Nymble [29], [50]–[52] combines three main ideas from prior schemes to solve the problem of allowing extremely efficient, retroactive blacklisting of anonymous users. In particular, it builds on 1) Nym's approach of issuing pseudonyms based on unique identifiers (a trusted platform module in an early version, and an IP address in later incarnations), 2) the group signatures approach of having a trusted entity responsible for revocation, and 3) UST's idea of issuing temporally related, mutually unlinkable use-once authentication tokens to users. The construction used in Nymble results in an extremely lightweight solution for all parties involved (most notably, for the service provider). It does this, however, by placing a lot of trust in third parties.

In particular, Nymble uses two trusted third parties called the Pseudonym Manager (PM) and the Nymble Manager (NM). The PM issues users with a pseudonym called a **Nym** that it computes by applying an HMAC to the user's IP address. Incorporating some way for a service provider to authenticate the validity of the pseudonyms that are associated with it would result in an efficient pseudonym system that binds users to their unique identifiers without requiring the issuing authorities to maintain a centralized list of users. However, Nymble-like systems take this one step further in order to provide users with full anonymity.

When a user wants to connect to a service provider, the NM takes the user's Nym and the **canonical name** of the service provider, and uses this to generate a set of mutually unlinkable use-once authentication tokens called **nymbles**. Each nymble is valid for some **time period** in the current **linkability window**; moreover, each nymble contains a trapdoor that allows the NM to, given that nymble, compute all *subsequent* nymbles in the same linkability window. (This is how blacklisting is accomplished.) The NM can always compute a user's last nymble of the linkability window from any of their other nymbles; we therefore call this last nymble the user's **SP-specific pseudonym**. On the other hand, the NM cannot "go backwards" to compute earlier nymbles.

At the start of each new linkability window, a change in system parameters causes all subsequent nymbles and SP-specific pseudonyms to change unlinkably (even to the

[16] We place schemes based on group signatures in the class of pseudonym systems because, upon revocation, all actions of the user at the service provider become linkable, thus degrading her anonymity to pseudonymity.

Scheme	Misauthentication resistance	Unlinkability	Backward anonymity	Revocability	Revocation auditability	Non-frameability	ZK-verinym	ZK-pseudonym	Objective vs. subjective	Rate-limiting	Blacklist transferability	Retroactive revocation	User efficient	Verifier efficient
UST [41]	✔	✔	✔	✔	✔	✔	∞ [3]	✔	S	✔			✔	✔
Nymble [30]	✔	✔	✔	✔	✔	✔			S [1]	✔		✔	✔	✔
Nymbler [26]	✔	✔	✔	✔	✔	✔	k [4]	✔	S [1]	✔	✘ [2]	✔	✔	✔
BNymble [34]	✔	✔	✔	✔	✔	✔	∞ [3]		S [1]	✔		✔	✔	✔
Jack [32]	✔	✔	✔	✔	✔	✔	∞ [3]	✔	S/O		✘ [2]	✔		✔

[1] Support for objective revocation can be added with the extension from [22]
[2] Support for blacklist transferability can be added with the extension from [22]
[3] This scheme supports the strong ZK-verinym property
[4] This scheme supports the ZK-verinym property for threshold k with a distributed k-of-n VI

Table II
THIS TABLE COMPARES THE PROPERTIES OF VARIOUS NYMBLE-LIKE SYSTEMS.

NM). Thus, at the start of each linkability window, all service providers must reset their blacklists and forgive all prior misbehaviour. This brings *dynamism* and *forgiveness* to Nymble; that is, it ensures that a user's misbehaviour can be (and eventually will be) forgiven without the NM having the ability to subsequently track the user. On the other hand, from the perspective of the service provider, it also limits the flexibility of the system. Thus, the future work sections of existing literature have proposed the investigation of mechanisms by which service providers can enforce revocation that spans linkability windows. (We propose a solution to this problem in [23].)

Note that Nymble's security relies heavily on strong assumptions about the noncollusion of its two TTPs. In particular, if a malicious PM and NM collude then they can easily determine with which service providers a user is interacting; further, if a malicious NM colludes with any number of service providers then they can easily link all of a user's actions at those service providers. Indeed, if a malicious PM, NM, and service provider combine these attacks, then the user is completely deanonymized.

Thus, in our own Nymbler scheme [25], [26], we modify Nymble to encode a user's Nym into an anonymous credential and use zero-knowledge proofs to remove the requirement of non-collusion between the trusted third parties. We also propose to distribute the PM (which we call the Credential Manager (CM)) so that no single party, other than the user herself, can learn a user's Nym during registration. This prevents brute-force attacks wherein a malicious entity computes the pseudonyms associated with large numbers of IP addresses in order to match them against users observed at some service provider. The construction used in Nymbler takes great care to maintain extremely low computational costs at the service provider.

Lin and Hopper's Jack [31] weakens Nymble's trust assumptions using a different approach. First, the scheme does away with the idea of basing pseudonyms on unique resources; instead, pseudonyms are based on user-chosen randomness, but are handed out only once to any given unique resource. It also reduces the role of the NM in an attempt to prevent attacks wherein the service provider colludes with a malicious NM. However, it places a much larger computational burden on the service provider.

Nymble's sensitivity to attacks involving the PM exists because the PM computes a user's Nym deterministically from a verinym[17] (her IP address). Although a one-way function maps IP addresses to Nyms, the space of valid IP addresses is cryptographically small; hence, given a user's Nym and knowledge of the PM's one-way function, an adversary can easily determine the user's identity with a brute-force attack. The output of this one-way function is, therefore, also a verinym for the user. For the remainder of this paper we shall thus refer to a user's Nym as her **verinym**, and to the entity that issues verinyms to users as a **Verinym Issuer** (VI).

To protect against attacks like the one above, we suggest a distributed (threshold) VI, and the following property, as security requirements of future Nymble-like systems.

Definition 14 ((Strong) ZK-verinym)**.** *A Nymble-like system satisfies the **zero-knowledge verinym** (ZK-verinym) property for threshold $k > 1$ if:*

1) *no fewer than k colluding VIs can compute the verinym associated with a given IP address, and*

[17] A **verinym** is any piece of information—such as an IP address, credit card number, etc.—that can be used to identify a user. [22]

2) *provided the user correctly follows the protocols, no (possibly colluding) set of third parties can extract nontrivial information about the user's verinym by observing her interactions with the NI or any subset of service providers.*

The system satisfies the ***strong zero-knowledge verinym*** *(strong ZK-verinym) property if it satisfies the ZK-verinym property for all thresholds* $k \in \mathbb{N}$.

The goal of this property is to minimize the potential for information leakage due to the use of verinyms.

Jack [31] satisfies the strong ZK-verinym property, while Nymbler [25] satisfies the ZK-verinym property for threshold k when a distributed k-of-n threshold VI is used. On the other hand, the original Nymble [29], [50], [51] does not satisfy the definition because, for example, a single PM computes the user's verinym using an HMAC. In this case, a malicious VI is able to link a user's SP-specific pseudonyms to her real identity, thus violating her anonymity.

Recently, Lofgren and Hopper proposed BNymble (i.e., Blinded Nymble) [33], which modifies the original Nymble design only slightly in order to satisfy the strong ZK-verinym property. Similar to Jack and Nym, BNymble replaces pseudonyms based deterministically on users' IP addresses with blind RSA signatures on user-chosen randomness. They point out that, naively implemented, this approach introduces a new privacy risk to the system. In particular, the system seems to require that the VI maintain a list of all users' IP addresses, a situation that is clearly undesirable. (Even if the VI is trusted to behave honestly, there is always the potential for the VI to be compromised and the list of IP addresses to be subsequently leaked to an adversary.) They solve this by proposing an additional third party that applies a one-way function (an HMAC with a key that is unknown to the VI) to each user's IP address prior to that user connecting to the VI. The VI then maintains a list of HMAC outputs instead of IP addresses. Since the secret HMAC key is unknown to the VI, the privacy of users on this list is maintained even if the VI is compromised. (Note, however, that the reduction in availability mentioned in §V still applies.) This modification allows BNymble to guarantee significantly stronger privacy guarantees than Nymble, at the cost of only a very minor level of additional overhead ($\approx 11\%$ additional computation for each nymble that is issued) and the aforementioned availability issues.

However, the Nymble framework is also sensitive to attacks involving the NM. The ZK-verinym property reduces this threat moderately, because it prevents a malicious NM from linking a user's actions at one service provider with her actions at any other service provider. On the other hand, the ZK-verinym property does nothing to prevent the NM from linking all of a user's actions at a single service provider. For this reason, we also suggest the following complementary property as a security requirement of future Nymble-like systems.

Definition 15 (ZK-pseudonym)**.** *A Nymble-like system satisfies the* ***zero-knowledge pseudonym*** *(ZK-pseudonym) property if:*

1) *during nymble acquisition, no party other than the user herself can learn any nontrivial information about the nymbles issued to a user, and*
2) *no entity is capable of extracting nontrivial information about a user's SP-specific pseudonym from a nymble without a priori knowledge of the NM's secret trapdoor key.*

The goal of this property is to minimize the potential for proactive deanonymization by a malicious NM. If the ZK-pseudonym property is satisfied, then the NM must extract a user's pseudonym from a nymble revealed by the service provider before it is possible for the NM to link the user's actions. Both Nymbler and Jack satisfy the ZK-pseudonym property.

At this point, we observe that the NM in Nymble plays two related—but distinct—roles: on the one hand, the NM is responsible for issuing nymbles to users, while on the other hand, the NM is responsible for revoking access using a trapdoor computation. Indeed, these two roles are logically orthogonal, and two distinct servers can fill them. Although never explicitly stated, this observation has indeed been used in the literature; for example, Jack completely eliminates the role of the nymble issuer. Instead, a user computes her own nymbles on demand using Camenisch and Shoup's Verifiable Encryption of Discrete Logarithms [12]. Thus, we shall replace the NM by two separate entities: the **Nymble Issuer** (NI), which is responsible for issuing nymbles to users, and the **Pseudonym Extractor** (PE), which is responsible for extracting SP-specific pseudonyms from nymbles to revoke misbehaving users.

The ZK-pseudonym property protects against a malicious NI, but it does not protect against a malicious PE that has sufficient resources to extract the pseudonyms from nymbles in real time.

Nymble, BNymble and Jack are all highly susceptible to linking attacks by a malicious third party. In Nymble and BNymble, the malicious NI just sends all nymbles it computes to the SP, which results in instant deanonymization of all users. In Jack, the attack works in reverse: the service provider just sends all nymbles it receives to the PE to have their SP-specific pseudonyms extracted. (Note that this second attack is not specific to Jack; indeed, it also applies to Nymble/BNymble and, to a lesser extent, to Nymbler.) While the attack in Jack is more costly than in Nymble and BNymble, an adversarial PE can still deanonymize all Jack users in real time. This is the case because extracting an SP-specific pseudonym in Jack takes about 26 ms of computation [31]; thus, a malicious PE can deanonymize about 2,300 authentications *per minute* using a single processor. On the other hand, the work required by the service provider to verify the zero-knowledge proofs that accompany each nymble is nearly 18 times that which is required for the PE to extract SP-specific pseudonyms from those nymbles. This means that an SP can only support about 128 authentications

per minute on the same hardware.

To combat this threat, Nymbler uses a trapdoor computation with tuneable computational cost; in particular, given t, the desired wall-clock time per trapdoor computation, as input, the PE's public and private key can be chosen so that the trapdoor function takes t time to evaluate on average. If the PE normally revokes at most K users per L minutes, then t can be set to just under L/K minutes. This renders wide-scale deanonymization economically infeasible for SPs with sufficiently high traffic volumes. We propose here that such trapdoor functions should be chosen so that the PE can prove in zero-knowledge that a random problem instance really does require t time to solve on average. (Nymbler uses discrete logarithms in a trapdoor discrete logarithm group as their trapdoor; it is indeed possible to prove statements about the difficulty of this function in zero-knowledge, under usual cryptographic assumptions.)

In Nymble/BNymble, the NI always issues a user with the entire sequence of nymbles for her verinym, right up to the end of the current linkability window. In Nymbler, the user is issued a credential that encodes her verinym and an expiration time. The verinym can only be used to obtain nymbles corresponding to time periods prior to its expiration time. For our generalized Nymble framework, we propose to subdivide each linkability window into smaller intervals called **verinym validity periods** (VVPs). (The number of time periods in a linkability window will be a multiple of the number of VVPs.) When the VI issues the user a verinym, this verinym is only valid until some future VVP; no nymble will be issued to any user for any time period in a VVP after her verinym expires. This capability allows greater flexibility in the choice of system parameters. For example, if a user obtains her IP address through DHCP, then she may receive a new address daily; an SP, however, may wish to use one-week linkability windows. With VVPs, the service provider can use one-week linkability windows, but still require the user to renew her verinym each day. Note that by changing input parameters to the function that maps verinyms to nymble seeds, particular SPs are able to choose a duration for their own linkability windows that is *shorter* than the duration set by the VIs (but not longer).

The cost of authentication (that is, verifying that a nymble is valid and checking its revocation status) at the service provider is constant in each of the Nymble-like schemes (and also in UST). In Nymble and BNymble, the service provider computes an HMAC to verify that the nymble is valid, and consults a hash map (with constant amortized lookup time) to ensure that the user's SP-specific pseudonym is not on the blacklist. In Nymbler, the service provider checks a *verifier-efficient restricted blind signature* to verify that the nymble is valid, and consults a hash map (much like Nymble) to ensure that user's SP-specific pseudonym is not on the blacklist. Similarly, in UST the service provider verifies a blind signature and consults a hash map to ensure that the signed value has never been seen before. (Note that in UST the service provider also *issues* a blind signature at the end of each session, thus incurring some additional cost shortly after each successful authentication.) In Jack, the service provider verifies two zero-knowledge proofs: one proof certifies that the nymble is valid (i.e., that the user is on the whitelist), and the second certifies that the user's SP-specific pseudonym does not appear on the blacklist. Computation for the user is essentially zero in Nymble and BNymble; it is constant—though somewhat higher than in Nymble and BNymble—in Nymbler. (However, essentially all of this computation is precomputation in Nymbler.) In Jack, the user must perform $O(|\Delta\mathcal{B}|)$ modular multiplications and exponentiations for each authentication, where $\Delta\mathcal{B}$ is the set of updates to the service provider's blacklist since the user's last authentication.

Table II summarizes the security and privacy guarantees and feature sets of the various Nymble-like systems discussed in this section.

C. Revocable Anonymous Credential Systems

The final class of anonymous blacklisting systems are the **revocable anonymous credential systems**. These schemes take a heavyweight approach to security and privacy by completely replacing TTPs with ZKPs. Unfortunately, the high computational overhead associated with them means that they are often of theoretical interest only.

As noted in our discussion of pseudonym systems based on private credentials, the first scheme in this class is the anonymous credential system of Camenisch and Lysyanskaya [11]. Since its introduction, a number of other general-purpose anonymous credential systems with revocation capabilities have appeared in the literature. Our focus here is only on those that specialize specifically as anonymous blacklisting systems.

Brands *et al.* constructed an anonymous blacklisting system for the setting of single-sign-on systems [6], [7] using Brands' private credentials [4], [5]. As Brands' credentials are not re-randomizable, and thus different showings of the same credential are linkable, the system calls for each user to obtain a set of credentials upon registration; each credential in the set can then be used for one authentication. The idea behind Brands' scheme is to have each service provider maintain a list of blacklisted credentials. To prove that she is not on the blacklist, a user in their scheme sends the service provider a zero-knowledge proof that none of her credentials is on the blacklist. The crux of the system is a novel construction of this zero-knowledge proof that enables both the prover and the verifier to do this using a number of exponentiations that scales with the square root of the list size (as opposed to linearly, as is the case with the naive approach). The batch verification techniques of Bellare *et al.* [3] make this possible. By design, this approach makes blacklist transferability particularly simple.

Tsang *et al.* (in fact, most of the Nymble authors and Au) proposed Blacklistable Anonymous Credentials (BLAC) [45]–[47] in the following year. BLAC removes the trust assumptions from Nymble by eliminating the role of the NM entirely. Similar to [6], [7], authentication with

Scheme	Misauthentication resistance	Unlinkability	Backward anonymity	Revocability	Revocation auditability	Non-frameability	Objective vs. subjective	Rate-limiting	Blacklist transferability	Retroactive revocation	User efficient	Verifier efficient
CL [11]	✔	✔	✔	✔	✔	✔	S			✔		
BDD [7]	✔	✔	✔	✔	✔	✔	S		✔	✔		
BLAC [46]	✔	✔	✔	✔	✔	✔	S		✔	✔		
EPID [8]	✔	✔	✔	✔	✔	✔	S		✔	✔		
PEREA [49]	✔	✔	✔	✔	✔	✔	S			✔[1]		✔

[1] Subject to the revocation window size (k)

Table III
THIS TABLE COMPARES THE PROPERTIES OF VARIOUS REVOCABLE ANONYMOUS CREDENTIAL SYSTEMS.

an SP in BLAC requires U to prove that her credential is not present on the blacklist. Unfortunately, BLAC is impractical for most real-world applications because the non-membership proof scales linearly in the size of the blacklist. (For each blacklist entry, the proof takes about 1.8 ms for U to prepare and 1.6 ms for the SP to verify [45].) If the blacklist grows to just one thousand users, then several hundred kilobytes of communication and several seconds of computation are required (per access) to prove that U is not on the blacklist [51]. For large SPs with many users (such as Wikipedia), the performance of this approach is unacceptable.

Concurrently and independently, Brickell and Li proposed Enhanced Privacy ID (EPID) [8], [9]. EPID is similar in spirit to BLAC, but is specially designed to enable a TPM device, with an embedded private key, to authenticate anonymously, while enabling the SP to revoke access from compromised TPMs. The non-membership proof in EPID is slightly faster than that of BLAC, but the scheme requires clients to have specialized hardware and is still prohibitively expensive since the computational overhead still scales linearly in the size of the blacklist.

Privacy-Enhanced Revocation with Efficient Authentication (PEREA) [48] is the second revocable anonymous credential system proposed by the authors of Nymble. It uses a cryptographic accumulator to replace the linear-time (at the service provider) non-membership proof with a constant-time non-membership proof. To facilitate this, the system uses an *authentication window*, which is similar in concept to that of a linkability window, except that it specifies the *maximum number of subsequent connections* a user may make before it becomes impossible to block them due to behaviour during a previous session, instead of *the maximum duration of time* that can elapse. However, while the accumulator approach makes the cost of verification at the service provider constant, the client is still required to perform linear work to compute non-membership witnesses.

Table III summarizes the security and privacy guarantees of the various revocable anonymous credential systems, as well as the features provided by each scheme.

VII. Conclusion

So far in this paper, we have formally defined anonymous blacklisting systems and proposed a set of definitions about their security and privacy properties, performance characteristics and functionality. We examined different ways to uniquely identify users and discussed their strengths and weaknesses, demonstrating that existing approaches are far from perfect. We developed a simple taxonomy for anonymous blacklisting systems and surveyed the literature, classifying each existing scheme into one of three categories: pseudonym systems, Nymble-like systems and revocable anonymous credential systems. We briefly described each of the anonymous blacklisting systems in the literature and compared those in the same category. We especially focused on the category of Nymble-like systems, since this relatively new approach to anonymous blacklisting systems has recently received a lot of attention from the research community. We believe that new schemes in this class are likely to be proposed in the near future (indeed, Lofgren and Hopper's BNymble [33] was accepted for publication just days before the submission of this paper) and we hope that our definitions and observations will assist in these endeavours. We conclude by discussing some open research problems in anonymous blacklisting.

As first mentioned by Tsang *et al.* in their Nymble paper [29], a useful feature for systems that use IP addresses as a unique identifier would be to provide service providers with the ability to block *entire subnets* in addition to just individual IP addresses. This would enable them to effectively block malicious users that have access to several different IP addresses in a small range. While there exist straightforward

modifications to some constructions that would make this possible, these modifications would negatively affect user privacy; thus, we leave it to future work to develop a privacy-friendly solution to this problem. In particular, we envision a design in which users in the same /24 prefix are indistinguishable, yet revoking some threshold number of IP addresses in the same /24 prefix results in the entire prefix being blocked.

On the other hand, there are also situations in which it would be desirable to give *certain IP addresses* the ability to misbehave a threshold number of times before revoking access to the users behind these addresses. For example, large institutions (such as universities) often have many users who share a single IP address through NAT; in such cases, it might be useful to allow the institution to run its own internal issuing authority that issues credentials based on internal IP addresses. The semantics here are similar to subnet blocking; users behind the same NAT address obtain credentials from the internal issuing authority to access services concurrently and unlinkably. However, if more than some threshold number of internal users have their access revoked from a service provider, then this would result in all users behind that IP address being blocked.

Another useful enhancement would be to provide service providers with the ability to detect repeat offenders and revoke these users' access for longer durations of time. (For example, Wikipedia's blocking policy states that administrators should consider "the severity of the behavior; [and] whether the user has engaged in that behavior before" when deciding on the duration of a block [55, "Duration of blocks"].) Indeed, this is a trivial extension when the misbehaviours all occur in a single session and are mutually linkable; however, detecting repeat offenders whose prior offenses have already been 'forgiven' seems to require the ability for some party to link users' actions to some extent, which is clearly undesirable.

Of course, each of the categories has some outstanding research problems of its own. Our own paper [23], published concurrently with this one, solves several previously open problems for Nymble-like systems. Further work is also warranted on revocable anonymous credential systems; the design of revocable anonymous credentials that are both user-efficient and verifier-efficient remains an attractive prospect for the future. Finally, further investigation into suitable unique identifiers and Sybil resistance in general will have a significant impact on the security and practicality of the next generation of anonymous blacklisting systems.

Acknowledgements: We thank Roger Dingledine, the anonymous reviewers, and our shepherd Dan Wallach, for their helpful input. This work is supported by NSERC, OGS, MITACS, and a David R. Cheriton Graduate Scholarship.

REFERENCES

[1] R. S. Abbott, "CPG: Closed Pseudonymous Groups," Master's thesis, Computer Science Department, BYU, 2008.

[2] R. S. Abbott, T. W. van der Horst, and K. E. Seamons, "CPG: Closed Pseudonymous Groups," in *Proceedings of WPES 2008*, Alexandria, VA, October 2008.

[3] M. Bellare, J. A. Garay, and T. Rabin, "Fast Batch Verification for Modular Exponentiation and Digital Signatures," in *Proceedings of EUROCRYPT 1998*, Espoo, Finland, May 1998.

[4] S. Brands, "Restrictive Blinding of Secret-Key Certificates," in *Proceedings of EUROCRYPT 1995*, Saint-Malo, France, May 1995.

[5] S. A. Brands, *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, 2000.

[6] S. A. Brands, L. Demuynck, and B. D. Decker, "A Practical System for Globally Revoking the Unlinkable Pseudonyms of Unknown Users." Department of Computer Science, K.U.Leuven, Technical Report CW472, 2006.

[7] S. A. Brands, L. Demuynck, and B. D. Decker, "A Practical System for Globally Revoking the Unlinkable Pseudonyms of Unknown Users," in *Proceedings of ACISP 2007*, Townsville, Australia, July 2007.

[8] E. Brickell and J. Li, "Enhanced Privacy ID: A Direct Anonymous Attestation Scheme with Enhanced Revocation Capabilities," in *Proceedings of WPES 2007*, Alexandria, VA, October 2007.

[9] E. Brickell and J. Li, "Enhanced Privacy ID from Bilinear Pairing," Cryptology ePrint Archive, Report 2009/095, 2009.

[10] E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell, and D. Jurafsky, "How Good Are Humans at Solving CAPTCHAs? A Large Scale Evaluation," in *Proceedings of IEEE S&P 2010*, Oakland, CA, May 2010.

[11] J. Camenisch and A. Lysyanskaya, "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation," in *Proceedings of EUROCRYPT 2001*, Innsbruck, Austria, May 2001.

[12] J. Camenisch and V. Shoup, "Practical Verifiable Encryption and Decryption of Discrete Logarithms," in *Proceedings of CRYPTO 2003*, Santa Barbara, CA, August 2003.

[13] D. Chaum, "Security Without Identification: Transaction Systems to Make Big Brother Obsolete," *Communications of the ACM*, vol. 28, no. 10, pp. 1030–1044, October 1985.

[14] D. Chaum and J.-H. Evertse, "A Secure and Privacy-protecting Protocol for Transmitting Personal Information Between Organizations," in *Proceedings of CRYPTO 1986*, Santa Barbara, CA, August 1986.

[15] D. Chaum and E. van Heyst, "Group Signatures," in *Proceedings of EUROCRYPT 1991*, Brighton, UK, April 1991.

[16] L. Chen, "Access with Pseudonyms," in *Proceedings of CPA 1995*, Brisbane, Australia, July 1995.

[17] I. B. Damgård, "Payment Systems and Credential Mechanisms with Provable Security Against Abuse by Individuals," in *Proceedings of CRYPTO 1988*, Santa Barbara, CA, August 1988.

[18] S. Deering and R. Hinden, "RFC 2460 – Internet Protocol, Version 6 (IPv6) Specification," 1998.

[19] R. Dingledine, Personal communication, August 2010.

[20] R. Dingledine, N. Mathewson, and P. F. Syverson, "Tor: The Second-Generation Onion Router," in *Proceedings of USENIX Security 2004*, San Diego, CA, August 2004.

[21] J. R. Douceur, "The Sybil Attack," in *Proceedings of IPTPS 2002*, Cambridge, MA, March 2002.

[22] I. Goldberg, "A Pseudonymous Communications Infrastructure for the Internet," Ph.D. dissertation, UC Berkeley, 2000.

[23] R. Henry and I. Goldberg, "Extending Nymble-like Systems," in *Proceedings of IEEE S&P 2011*, Oakland, CA, May 2011.

[24] R. Henry and I. Goldberg, "Formalizing Anonymous Blacklisting Systems," in *Proceedings of IEEE S&P 2011*, Oakland, CA, May 2011.

[25] R. Henry, K. Henry, and I. Goldberg, "Making a Nymbler Nymble using VERBS," in *Proceedings of PETS 2010*, Berlin, Germany, July 2010.

[26] R. Henry, K. Henry, and I. Goldberg, "Making a Nymbler Nymble using VERBS (Extended Version)." Centre for Applied Cryptographic Research, UWaterloo, Technical Report CACR 2010-05, 2010.
[27] J. E. Holt and K. E. Seamons, "Nym: Practical Pseudonymity for Anonymous Networks." Internet Security Research Lab, BYU, Technical Report 2006-4, 2006.
[28] IANA, "IANA IPv4 Address Space Registry." [Online]. Available: http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml
[29] P. C. Johnson, A. Kapadia, P. P. Tsang, and S. W. Smith, "Nymble: Anonymous IP-Address Blocking," in *Proceedings of PETS 2007*, Ottawa, ON, June 2007.
[30] S. Köpsell, R. Wendolsky, and H. Federrath, "Revocable Anonymity," in *Proceedings of ETRICS 2006*, Freiburg, Germany, June 2006.
[31] Z. Lin and N. Hopper, "Jack: Scalable Accumulator-based Nymble System," in *Proceedings of WPES 2010*, Chicago, IL, October 2010.
[32] K. Loesing, "Measuring the Tor Network: Evaluation of Client Requests to the Directories." The Tor Project, Technical Report, 2009.
[33] P. Lofgren and N. Hopper, "BNymble (A Short Paper): More Anonymous Blacklisting at Almost No Cost," in *Proceedings of FC 2011*, St. Lucia, February 2011.
[34] A. Lysyanskaya, "Pseudonym Systems," Master's thesis, Department of Electrical Engineering and Computer Science, MIT, 1999.
[35] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf, "Pseudonym Systems," in *Proceedings of SAC 1999*, Kingston, ON, August 1999.
[36] D. L. Mills, "Network Time Protocol Version 4 Reference and Implementation Guide." University of Delaware, Technical Report 06-6-1, 2006.
[37] E. J. Schwartz, "Contractual Anonymity," Master's thesis, Information Networking Institute, Carnegie Mellon, 2009.
[38] E. J. Schwartz, D. Brumley, and J. M. McCune, "Contractual Anonymity." School of Computer Science, Carnegie Melon, Technical Report CMU-CS-09-144, 2009.
[39] E. J. Schwartz, D. Brumley, and J. M. McCune, "A Contractual Anonymity System," in *Proceedings of NDSS 2010*, San Diego, CA, February 2010.
[40] S. G. Stubblebine, P. F. Syverson, and D. M. Goldschlag, "Unlinkable Serial Transactions: Protocols and Applications," *ACM Transactions on Information and System Security (TISSEC)*, vol. 2, no. 4, pp. 354–389, November 1999.
[41] P. F. Syverson, S. G. Stubblebine, and D. M. Goldschlag, "Unlinkable Serial Transactions," in *Proceedings of FC 1997*, Anguilla, British West Indies, February 1997.
[42] The Tor Project Inc., "TheOnionRouter/BlockingIrc – Tor Bug Tracker & Wiki." [Online]. Available: https://trac.torproject.org/projects/tor/wiki/TheOnionRouter/BlockingIrc
[43] The Tor Project Inc., "TorStatus - Tor Network Status." [Online]. Available: http://torstatus.cyberphunk.org/
[44] The Tor Project Inc., "Who uses Tor?" [Online]. Available: http://www.torproject.org/about/torusers.html.en
[45] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith, "Blacklistable Anonymous Credentials: Blocking Misbehaving Users Without TTPs," in *Proceedings of CCS 2007*, Alexandria, VA, October 2007.
[46] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith, "Blacklistable Anonymous Credentials: Blocking Misbehaving Users Without TTPs (Extended Version)." Computer Science Department, Dartmouth College, Technical Report TR2007-601, 2007.
[47] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith, "BLAC: Revoking Repeatedly Misbehaving Anonymous Users Without Relying on TTPs." Computer Science Department, Dartmouth College, Technical Report TR2008-635, 2008.
[48] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith, "PEREA: Towards Practical TTP-free Revocation in Anonymous Authentication," in *Proceedings of CCS 2008*, Alexandria, VA, October 2008.
[49] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith, "BLAC: Revoking Repeatedly Misbehaving Anonymous Users without Relying on TTPs," *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 4, October 2010, Article No. 39.
[50] P. P. Tsang, A. Kapadia, C. Cornelius, and S. W. Smith, "Nymble: Blocking Misbehaving Users in Anonymizing Networks." Computer Science Department, Dartmouth College, Technical Report TR2008-637, 2008.
[51] P. P. Tsang, A. Kapadia, C. Cornelius, and S. W. Smith, "Nymble: Blocking Misbehaving Users in Anonymizing Networks," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, vol. 8, no. 2, pp. 256–269, March 2011.
[52] P. P. Tsang, A. Kapadia, and S. W. Smith, "Anonymous IP-address Blocking in Tor with Trusted Computing (Short Paper: Work in Progress)," in *Proceedings of WATC 2006 (Fall)*, Tokyo, Japan, November 2006.
[53] P. P. Tsang and S. W. Smith, "PPAA: Peer-to-Peer Anonymous Authentication," in *Proceedings of ACNS 2008*, New York, NY, June 2008.
[54] United Nations (Department of Economic and Social Affairs, population division), "World Population Prospects: The 2006 Revision." [Online]. Available: http://esa.un.org/unpp/
[55] Wikimedia Foundation, "Wikipedia:Blocking policy — Wikipedia, the free encyclopedia." [Online]. Available: http://en.wikipedia.org/wiki/Wikipedia:Blocking_policy