2008-03-12

CPG: Closed Pseudonymous Groups

Reed S. Abbott
Brigham Young University - Provo

CPG: CLOSED PSEUDONYMOUS GROUPS

by

Reed S. Abbott

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Brigham Young University

April 2008

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Reed S. Abbott

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Kent E. Seamons, Chair
Date	Daniel Zappala
Date	Dennis Ng

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Reed S. Abbott in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Kent E. Seamons
Chair, Graduate Committee

Accepted for the Department

Parris Egbert
Graduate Coordinator

Accepted for the College

Thomas W. Sederberg
Associate Dean, College of Physical and Mathematical Sciences

ABSTRACT

CPG: CLOSED PSEUDONYMOUS GROUPS

Reed S. Abbott

Department of Computer Science

Master of Science

Internet users generally feel their actions are anonymous, but this is often not the case. Users can be tracked and their actions logged for future analysis, which is not the desire of most users [12]. Software and services exist which offer anonymity on the Internet when used correctly. Anonymity on the Internet is useful for many people including whistleblowers, dissidents, law enforcement, and the security conscious, but it can be abused. A user can act maliciously under the guise of anonymity without the fear of retribution. Thus, a level of administrative control over users is desirable, even in an anonymous system. Administrative control over users in an open, anonymous system is extremely difficult, but what about a closed, pseudonymous system? Closed Pseudonymous Groups is a pseudonymous framework for a closed group of users that balances the needs of the user with those of a service administrator. Using a resource that uniquely identifies a user, the user may create a pseudonym with which they can interact with the service over the Internet. Mis-

behaving pseudonyms can be blocked from using the service, and the offending user is unable to create a new authorized pseudonym.

ACKNOWLEDGMENTS

I would like to thank my graduate advisor, Dr. Kent E. Seamons, for his analysis and guidance, and my graduate committee, Dr. Daniel Zappala and Dr. Dennis Ng, for their feedback. I would also like to thank Tim van der Horst, Andrew Harding, and other reviewers for their helpful comments.

This research was supported by funding from the National Science Foundation under grant no. CCR-0325951, prime cooperative agreement no. IIS-0331707, and The Regents of the University of California.

Table of Contents

List of Tables

List of Figures

Chapter 1 — Introduction

The Internet is generally regarded by users as an anonymous place to browse and communicate. People feel less inhibited and free to act how they please because, to quote the New Yorker, "On the Internet, nobody knows you're a dog (Figure 1.1)[14]."

Figure 1.1: From The New Yorker.

This feeling of anonymity, however, is just that; a feeling. Without an intimate knowledge of how the Internet works, the average user does not understand that their actions may be tracked and logged. Cookies track a user's surfing patterns through a site, search engine queries are mined to discover a user's interests, third-party cookies track a user between sites, and an IP address can reveal information

about a user's location. Personal information can be aggregated between sites to build up an online dossier of personal interests and habits. Server logs keep all of this information for extensive periods of time, so a user's information may be retrieved and analyzed long after an event occurs.

A user may think it unlikely that their web surfing is being monitored, but recently the FBI asked several ISPs to record all customers' online activities. The FBI's claimed purposes were terrorist defense and tracking child predators [11]. While their intentions were probably good, it is easy to imagine innocent users' information being abused. A sitting official, for instance, may use it to track a political opponent or discover spending habits of a potential donor. This is one situation in which user anonymity is desirable.

Anonymity on the Internet is useful for many purposes. For example parents of a local PTA can express concerns about a teacher without fear of retribution against their child. An abused spouse or rape victim can discuss hardships while avoiding the abuser. A student may discuss dissatisfaction with a class without apprehensions about the professor or an assistant harming the student's grades. Many more examples exist, and it's easy to see that anonymity can be beneficial, but there is a dark side to anonymity as well.

Anonymity can also be abused. A person can use the shroud of anonymity to commit malicious acts without the threat of being caught. In March of 2007 the popular tech journalist, author, and blogger Kathy Sierra was threatened multiple times by a small group of individuals over her blog [2]. The harassment included death threats and disturbing images of herself that had been altered. As a result she was forced to cancel public appearances and suspend publication of her blog. Other examples might include a disgruntled employee that reveals sensitive or humiliating

information about their employer on a blog; an angry parent of a student who threatens physical harm to a disliked teacher through an online discussion group; or a student posting inappropriate pictures on a class discussion board. In situations such as these, administrators need the ability to stop continued abuse from aberrant users.

Closed Pseudonymous Groups (CPG) is a solution for situations such as these. In the past, research has overwhelmingly concentrated on Internet-wide anonymous communication, with open enrollment where users' actions are unlinkable [6] [13] [10] [1] [7]. Solutions such as these are powerful, but moderation of misbehaving users in these systems is nearly impossible, so often the anonymizer is blocked. One example is Wikipedia, which blocks TOR users from editing articles.

CPG is unique in that it takes a narrower view and considers pseudonymity, where a user's actions are linkable, amongst closed groups. These groups might include a university, a neighborhood PTA, a church congregation, a business, or a social club. The administrator of the service decides who is allowed into the group. Authorized members of the group are then able to create a pseudonym, also known as a *nym*, which is an alternate identity of the user. The nym is used to communicate with the service without revealing the user's true identity. Nobody other than the nym's owner, including the administrator of the service, is able to link the user's nym with their true identity. The nym is persistent to allow the user to develop a reputation with the system, and if a user misbehaves the administrator is able to block the offending nym.

As a motivating scenario, consider a university professor teaching a course. The professor would like to be able to receive feedback from his students throughout the semester. In order to do this, he creates a class message board where the students can

make comments and leave feedback about the course. In order to keep the comments genuine and to protect students' identities, he wants the feedback to be anonymous. By allowing the communications to be anonymous, the professor hopes that students will be able to speak more freely and comfortably about concerns they may have with the class. The only people that are allowed to post to the message board are the students of the class. He would also like to prevent the teaching assistants, who will be managing the message board, from discovering the identities of students making the anonymous comments. If complaints are being posted, the professor wants to know if it is one person or many different people who are upset. Finally, if a student begins misbehaving on the message board (e.g., posting test questions), the professor would like to prevent that student from commenting thereafter. A message board can be created for the class simply enough, but how can it be limited to only the students in the class? If communication is only kept to students in the class, how can their identities remain anonymous? What happens if a student starts misbehaving on the message board? CPG addresses such a situation.

This paper has the following layout. Work related to CPG is summarized in section 2. Section 3 discusses the high-level design of CPG. Then in section 4 one specific implementation of CPG is presented. Section 5 details a threat analysis of CPG and its vulnerability to several attacks. In section 6 a usability study performed on CPG is described. Finally, section 7 presents our conclusions and future work.

Chapter 2 — Related Work

CPG leverages the work of several other projects to provide anonymity. Blind signatures provide the separation between a user's true identity and their pseudonym. TOR is used to obscure a user's IP address and other information which could reveal the identity of a user. Finally, SAW is used in our implementation of CPG for all our authentication needs.

2.1 Blind Signatures for Untraceable Payments

A blind signature is a cryptographic signing method in which a message can be obscured, or *blinded*, by a client before being submitting for signing by an authority [4]. Once blinded, the message appears only to be random text. After the blinded message is signed the unblinded signature may still be publicly verified against the original, non-blinded message, as with standard cryptographic signatures. This allows an authority to sign a message without the ability to read what is being signed.

Blind signatures are common in anonymous systems. Electronic cash systems use blind signatures to assert that a unit of currency is valid while still allowing the user to spend it without revealing their identity, as with real cash. Pseudonymous systems use blind signatures to provide the separation between a user's real identity and pseudonym. An authority can assert that a pseudonym is valid without the authority knowing the pseudonym.

One implementation of blind signatures is a slight deviation from a standard RSA signature algorithm. Before a document $\boldsymbol{m}$ is submitted to an authority to be signed, the document may be blinded using a random number $\boldsymbol{r}$, called a *blinding factor*, and the authority's public key $\boldsymbol{e}$. The resulting blinded message $\boldsymbol{c}$ is computed by

$$c = m \cdot r^e (mod N)$$

The blinding factor ***r*** is not revealed to the signing authority and obscures ***m***. ***c*** is then submitted to the authority. The blinded signature ***s'*** is computed using the RSA signature scheme with the authority's private key ***d*** where

$$\begin{aligned} \text{s'} &= c^d (mod N) \\ &= (m \cdot r^e)^d (mod N) \\ &= m^d \cdot r^{ed} (mod N) \\ &= m^d \cdot r (mod N) \end{aligned}$$

After the document has been signed, ***s'*** is returned to the user. ***s'*** is unblinded to reveal the standard RSA signature ***s*** using the inverse of the blinding factor ***r*** such that

$$\begin{aligned} \text{s} &= s' \cdot r^{-1} (mod N) \\ &= m^d \cdot r \cdot r^{-1} (mod N) \\ &= m^d (mod N) \end{aligned}$$

After unblinding, ***s*** is identical to the signature that would result if no blinding had been used.

2.2 JAP: Java Anonymous Proxy

JAP is the *Java Anonymous Proxy*[1]. A client achieves anonymity by choosing from a list of groups of anonymizing proxies. A group of anonymizing proxies is called a *mix*. The mix is organized into a *cascade*. A cascade is a set order for messages to pass through the proxies. All clients' traffic enters the cascade at the same point and travels through the same series of proxies. To a webserver, all traffic from a single mix appears to come from the same proxy. Routing through the cascade prevents the client from having to trust a single source. Another positive

aspect of JAP is that it attempts to defeat timing attacks by generating filler data when a client is not active. This type of attack is discussed in section 5.2.

2.3 TOR: The Second-Generation Onion Router

TOR, short for *The Onion Router*, is an anonymizing network for TCP traffic [6]. In many instances, a user's IP address is closely tied to their identity. TOR allows a user to obfuscate the origin of network traffic by passing it through a series of anonymizing servers. It is quite similar to a circuit of anonymous email remailers but is generalized for TCP traffic.

When preparing to send data, the transmitting computer randomly chooses a number of machines from a set of available anonymizing servers. The servers are ordered by the sender to make up a circuit from the user to the data's final destination. The sender then securely establishes a unique symmetric key with each of the anonymizing servers in the circuit. The data to be sent is encrypted with each of the symmetric keys, starting with the last anonymizing server first and working back to the first server in the circuit. The encrypted data is known as an onion since many layers of encryption have been applied to the data. Once the data is encrypted with each of the keys, it starts its journey along the circuit. Each anonymizing server along the route decrypts the data, peeling off a layer of the onion, using the shared key established previously. In decrypting the data the server also reveals the next hop in the network. In this manner, the data is moved through the circuit until it reaches its destination. Anonymity is achieved since each server only knows about the previous and next nodes in the circuit. The data is encrypted and therefore unreadable for all nodes in the circuit except the final node. The final node is able to read the original data sent by the user but does not know the origin of the data.

2.4 Nym: Practical Pseudonymity for Anonymous Networks

Nym is a pseudonymous system for open systems such as Wikipedia. Users exchange a limited resource, like an IP address, for signed a token. The signed token is then exchanged for a signature on a digital certificate from a certificate authority (CA). This digital certificate acts as a user's nym. A nym can be blocked to limit the abuse of malicious users. The simplicity of Nym makes for a low barrier to adoption but problems remain.

Nym works by first having a user interact with a token server to receive a signed token. The user starts by generating a random token. The user then blinds the token and submits it to the token server. The user also submits proof of possession of a limited resource. For an IP address, the proof is implicit. If the proof is valid, the token server signs the blinded token and returns the blinded signature to the client. Once a signature has been received for a specific resource, the resource can no longer be exchanged for another signed token.

Next the client interacts with a CA to receive a signature on a client certificate. The client submits the token, unblinded signature, and client certificate to a CA. The CA examines the signature on the token to determine if it is valid. If valid, the CA signs the certificate.

The client now receives service using the signed certificate. If a user misbehaves, the service provider blocks the public key corresponding to the user. The client is unable to receive another client certificate because the limited resource has been used.

Nym has several drawbacks. Unfortunately neither email addresses nor IP addresses, the suggested limited resources, map precisely to a single individual. It is often trivial for a user to obtain another email account after being blocked. A new

IP address can often be obtained by reconnecting to the Internet service provider or using a publicly available machine. Finally, digital certificates lack portability, making it difficult to receive service from many different locations.

2.5 SAW: Simple Authentication for the Web

SAW, *Simple Authentication for the Web*, addresses the problem of the over-abundance of passwords faced by Internet users today [16]. Sometimes it feels like every site on the Internet requires a username and password. As a result, users are faced with two options: they may either generate a new password for each site, which makes password management a difficult chore, or use a single password for all sites and trust each to guard the password. Neither option is ideal.

With SAW, the observation is made that in order to reset a password, users are often issued a temporary link, which is sent to the user's email account. Proving ownership of the email account is used as an authenticator. Since this method is already used on a regular basis and proven both usable and reasonably secure, why not use this method as a primary method of authentication? SAW does just that.

SAW works in the following manner. When a user wishes to authenticate to a service, the user simply enters an email address. Two tokens are issued to the user: one through the HTTPS connection established with the service, and one in an email to the user's email address. The user must be able to retrieve the token from their email account and resubmit the two tokens to the content provider in order to authenticate.

The process of checking the user's email for the token and resubmitting the two tokens can be automated through client side software. The user needs only a single password to allow the software access to the email account. The software is then able to retrieve any tokens submitted to that email account and return them

automatically. This creates a single sign-on solution that is simple to use, easy to implement, requires very little change of either the client or the server and no change to the email provider.

SAW is impervious to eavesdropping and significantly raises the bar for active attacks.

Chapter 3 — CPG Design

Closed Pseudonymous Groups (CPG) is a framework for providing anonymity within a closed group of users.

CPG has the following design goals:

- An administrator can restrict nym creation to authorized users
- An administrator can block a nym
- A client's real identity and nym are unlinkable
- A client's separate nyms are unlinkable
- A client's actions with a nym are linkable

3.1 CPG Stages

The design of CPG is divided into three separate stages with three different entities involved: the client, the pseudonym server, and the service provider. The client is the user seeking to use a service anonymously. The pseudonym server is responsible for authenticating the client to the client's identifier and signing a client-generated registration token. Finally, the service provider must validate the signed token, authenticate the user's nym, and provide the service. In the first stage, the client interacts with the pseudonym server to authenticate and have their token signed. In the second stage, the client waits before using their newly signed token to prevent a timing attack. Finally, in the third stage, the client interacts with the service provider.

3.1.1 Stage 1: Registration Token Acquisition

The first stage of CPG is the registration token signing stage. In this stage, the pseudonym server is responsible for signing registration tokens for authorized clients. To have a token signed, a client must authenticate, be authorized, and submit a blinded token.

The first step to having a registration token signed is authentication. A client begins by submitting an identifier, the type of which is specified by the pseudonym server. The identifier uniquely identifies a client. The system must already know the identifier belongs to the client. Examples of such identifiers include a client's primary email address, an identity in a digital certificate, a student ID number, or a corporate username. The authentication method varies depending on the type of identifier. For instance, if a digital certificate is used the client must prove ownership of the associated private key. If a corporate username is used the client might be required to submit a registered password with the username. Other identifiers and methods of authentication may be used as appropriate.

The next step, authorization, serves two purposes. The first is to restrict which authenticated clients are able to obtain a signed token. The second purpose is to prevent a client from receiving multiple signed tokens. Any method of authorization appropriate to the setting may be used. For example, a simple access control list (ACL) can be created by the administrator. The pseudonym server checks the list for the client's identifier, and if on the list, the client is authorized. Once a client has a token signed, authorization to receive another is automatically removed by deleting the client's identifier from the list.

Once a client is authenticated and authorized, the client must generate a registration token. The registration token is a random value large enough that no two

clients are likely to generate the same value. It is recommended that the tokens be generated instead of the user entering the token's value. A token's value may only be used once in a system to prevent a token from being reused to register multiple nyms. If users were allowed to enter the token's value, multiple users may unintentionally try to use the same value. The pseudonym server is unable to prevent this since it is not allowed to see the token's value.

Before the registration token can be submitted to the pseudonym server the token must be blinded (see section 2.1). Blinding the token prevents the pseudonym server from linking a client's identity with their nym after the token is spent. Once the blinded token is received and signed by the pseudonym server, the signature is returned to the client, where it is unblinded.

3.1.2 Stage 2: Post-Creation Delay

After having a registration token signed, it is important that the client wait a period of time before using the it. If a client were to request a signature from the pseudonym server and immediately use the signed token to register their nym, a colluding pseudonym server and service provider may be able to correlate the two actions based on timing. This attack is described in more detail in section 5.2.

The waiting time varies depending on the setting in which CPG is placed. In a classroom setting, students may be given the first week of a semester to register a nym but are unable to use it. Thereafter, students may use their nym but no new nyms may be registered. In a corporate environment new clients need to be brought into the system on an irregular basis. In this case, the length of the waiting period depends on the rate in which new clients are introduced. This is an area that must be carefully considered when CPG is incorporated into a new environment and is discussed in more detail in section 3.3.1.

3.1.3 Stage 3: Nym Registration

In stage 3, a client interacts with the service provider to register a nym using the signed registration token and to receive the service.

From the client's perspective, any communication with the service provider should be performed over an anonymizing service (e.g., TOR [6], JAP [1]) to hide the client's originating IP address. It is also important that any other identifying information, such as browser type, be removed. A good anonymizing service provides this functionality.

The client can now use the registration token and signature to register a nym. If nym accounts are created and managed by the service provider, the client submits the registration token and signature along with any registration information. Information will include the desired nym and authentication information, like a password. Obviously, no identifying information should be provided when registering the nym account. If a nym account is managed by a third party (e.g., an off site email account) the client must submit proof of account ownership with the registration token and signature.

Once a nym is registered, the client must thereafter authenticate to the nym and be authorized to log in; the registration token and signature may be discarded. Any method of authentication and authorization appropriate to the setting will suffice. Authorization can be as simple as an ACL or a complex algorithm based on nym reputation, but this is an administrator's chance to block offending nyms.

3.2 Nym Selection

Care should be taken by a client when choosing a nym. The nym must not reveal any identifying information about the client. For instance, a client that is partial to cycling should not use the nym 'bikeboy'. Identifying information can be more subtle

than this example. It might be unwise for an employee of the Department of Motor Vehicles to use a work email as a nym to be used on the PTA blog. Simply having a nym from the dmv.state.gov domain reveals potentially identifying information about the owner. If this employee is the only client in the group employed by the department of motor vehicles it is trivial to link the nym with their real identity. An email address from a public email provider such as Yahoo! Mail, Gmail, or Hotmail is less likely to reveal any information about the client and can be created anonymously. If communicating with an online service when selecting the nym, as with an email nym, the client should use an anonymizing service.

3.3 Adding and Removing Clients

Adding and removing clients must be handled appropriately to maintain client anonymity. In a service such as the classroom message board in the motivating scenario, clients are added to the group at the beginning of the semester, and when the semester ends these nyms are invalidated. This process becomes more difficult when clients must be added or removed on an irregular basis.

3.3.1 Adding Clients

When adding new clients to a group, a client's anonymity is related to the size of the group with which they are added. If a single client is added to an actively participating group, it is trivial to link the new nym with the new individual in the group. If a client is added in a group of two people, the new client is only slightly more anonymous; there are two nyms which the new individual could own. The larger the group in which a client is added, the greater the anonymity each client may enjoy.

New clients may be added to a group as a batch or the entire group may all register for new nyms. With the batch method, deciding when to add new clients

to a group depends on both the activity level of the service and the frequency with which new clients are added to the group. If activity level on a service is extremely low and there are many nyms which are unknown to the active users, it may be safe to add a new client to the group immediately; a batch of one. On the other hand, if activity level is extremely high it is unsafe to add a new client until a large group of new clients may be added. If the frequency with which new clients are ready to be added to the group is high, the waiting period for a new batch is low. A threshhold method may be applied when deciding when to add new clients; for instance, no single client can be added until thirty new clients are ready to be added. It is important to remember that when a new client is added in a batch, the new client is only indistinguishable from the group of individuals with which they were added. The batch method is only reasonable if new clients need to be added on a regular basis.

If the frequency for adding new clients is low, the waiting period could be substantial. If this is the case, it may be best to invalidate all current nyms and force all clients to create a new nym. Although this method may seem inconvenient, it takes advantage of the entire group size for anonymizing each client.

3.3.2 Removing Clients

Removing a client from a group can be difficult if the identity of the client to be removed is known but their nym is not. For instance, in our classroom scenario, consider a student who decides to drop out of the class halfway through the semester. Ideally, the professor would like to prevent the student from being able to post to the discussion board, but the professor does not know the student's nym, so he is unable to revoke authorization.

One solution is for the client simply to reveal their nym to the administrator. The

client must prove ownership of the nym to the administrator. Once the administrator knows the client owns the nym, it is trivial to remove authorization for that nym. While this does remove the client from the group, it also has the unfortunate effect of revealing all actions that client has taken previously to the administrator. Once the administrator knows the nym a client has been using, the administrator may backtrack through logs to find out what actions have been taken by that client.

Invalidating a single nym may enable others to link the client with their nym in a more subtle fashion. Consider the case of the message board in our motivating scenario. If the disappearance of an extremely active commenter coincides with a student no longer coming to class, it might reveal that the two are linked.

Another option is to force all clients to create new nyms. Authorization can be removed for all current clients' nyms, and all clients that are to remain in the group are allowed to create and have new nyms signed. The client that is to be removed is unable to create a new nym and is removed from the group. This also prevents clients from linking a disappearing nym with a newly departed member of the group. As with adding new clients this way, the clearest disadvantage of this method is that it is a hassle. This is especially the case if clients are leaving the group on a frequent basis. To alleviate this problem, clients' nyms may be invalidated on a periodic basis, such as once every three months. At the very worst, a departed client can continue to use the system for one extra period.

Chapter 4 — CPG Implementation

As a proof of concept, and in order to study the usability of CPG, an implementation has been created. The service provider is an online message board which has been extended to allow for token signing and nym registration. We chose SAW as the authentication method. Two ACLs are used for authorization: one to control registration token signing and one for control of nyms. The SAW toolbar was extended to aid users in receiving and managing signed tokens and using nyms correctly. The type of nyms used in this implementation are email addresses. For simplicity, the pseudonym server and message board are hosted on the same machine and the same domain.

4.1 Token Signing

The administrator prepares the CPG message board service by creating a list of primary email addresses allowed to have a registration token signed. The message board administrator also disables the login page to prevent a client from immediately logging in after registration.

The client starts by navigating to the registration page and submitting their primary email address. The server examines the authorization ACL for the client's address. If present, three large random numbers are generated and stored with the client's email address: a random transaction identifier and two SAW tokens. The first SAW token, AT_{user}, is returned directly to the user's browser in a cookie. The cookie is of the format:

$$\mathrm{ATuser\_}transID\mathrm{=AT}_{user}$$

where *transID* is the transaction identifier.

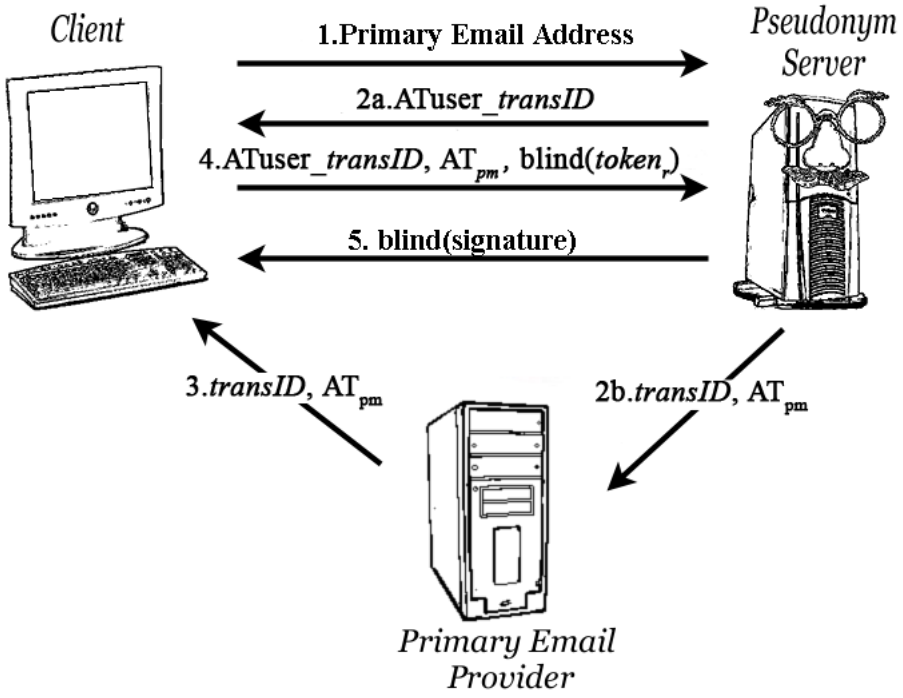

Figure 4.1: Pseudonym Registration using CPG.

The second token, AT_{pm}, and the transaction identifier are sent to the user's email account with a specially formatted title in the email. The title's special format is required for the SAW toolbar so it can find the appropriate email. The title is of the format:

$$[\text{SAW-}URL]\ \text{transID}=transID\&\text{ATpm}=AT_{pm}$$

where URL is the address of the registration token signing page and AT_{pm} is the value of the second SAW token. The body of the email is human readable, with an explanation of the email and a link to the signing page. The link has the transaction identifier and email token in the query string, similar to the email title. When a client follows the link in the email, they are taken to the signing page, and the user token in the cookie is automatically delivered to the authentication server, while

the email token and session identifier are delivered in the URL query string. These tokens are stored in hidden input fields on the token signing page.

If the primary email address is not on the ACL, a human-readable message is sent to the address explaining that an attempt was made to authenticate. The AT_{user} token is also returned to the client, even though it is not stored and the other parameters are not generated. Returning the token prevents an attacker from discovering information about the ACL. If the AT_{user} token is not returned, an attacker can try many different email addresses in order to discover which return an AT_{user} token and are thus on the ACL.

Once the client has reached the signing page, they are ready to generate and submit a registration token, $token_r$, for signing. JavaScript is used to generate a large random number for the registration token. Before the token is submitted to the pseudonym server, the token must be blinded. Another random number is generated for the blinder used to obscure the token. The blinded token is then submitted to the pseudonym server for signing along with the SAW tokens and transaction identifier.

Once the pseudonym server receives the blinded token and SAW tokens, it first checks to make sure the SAW tokens are valid with the transaction identifier received. If the tokens are valid, the primary email address associated with the tokens is retrieved and the tokens removed from the database. The primary email address is removed from the ACL to prevent a client from receiving more than one signed token. Finally, the pseudonym server signs the blinded, hashed token and returns it to the client.

The blinder must be stored by the client between submission to the pseudonym server and the return of the blinded token. Since the HTTP protocol is stateless and

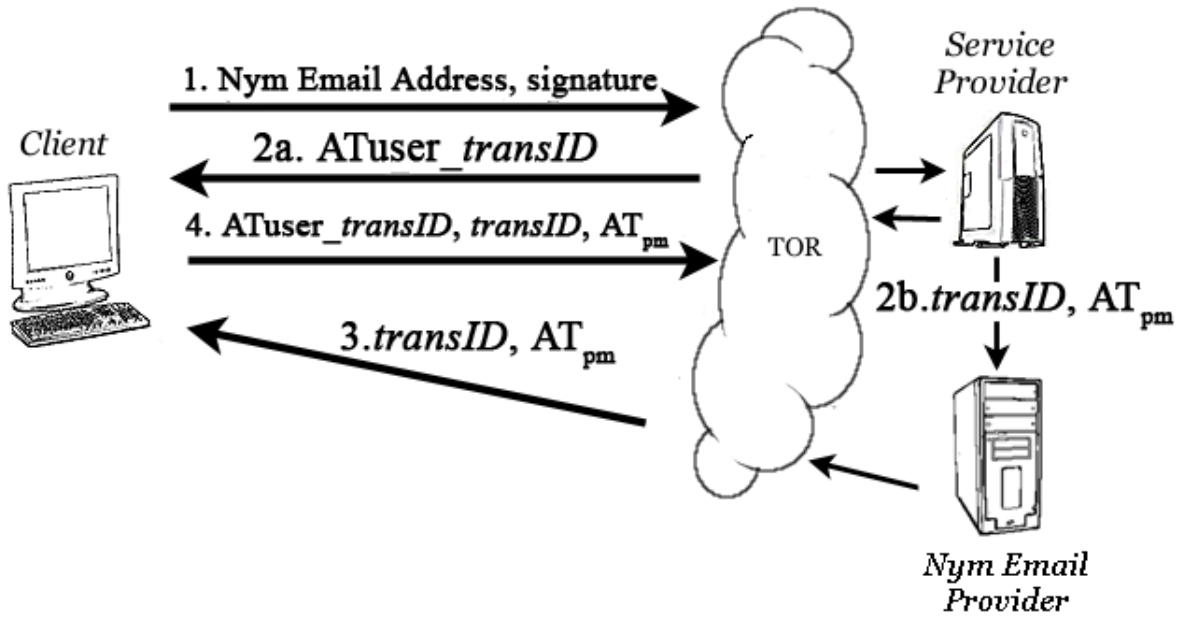

Figure 4.2: Using a nym to gain service.

JavaScript is unable to write to a file without an enormous amount of trust from the client, this presents a problem. The blinder may not be sent to the pseudonym server and returned back to the client since this would give the server the ability to unblind the token. One solution is to have the client enter random data in a text field and use this data as the blinder. The client could copy the blinder to the clipboard, and after the blinded signature is returned the client would copy the blinder back to a text field and unblind the signature. Although this solution works, it is a hassle for the client, and the client may enter easily guessed blinding values. The option chosen for this implementation is to use AJAX. Using AJAX the tokens are submitted to the server and the signature received without refreshing the web page. The blinder remains stored in a JavaScript variable without any interaction from the user and without revealing it to the pseudonym server.

4.2 Nym Registration

To begin, the client obtains a nym email account. Any email account can be used, but for this implementation users are advised to sign up for a webmail account and follow the guidelines of section 3.2. Ideally the webmail account should provide POP or IMAP service and deliver email quickly for the best user experience. Some webmail services require a user to provide their primary email account when creating a new account. These services should be avoided since it provides a link between a client and their nym. Based on these criteria, two webmail providers are recommended: GMail.com and Gawab.com. GMail provides free POP access over SSL and the POP experience is consistent. Authentication using the SAW toolbar takes about five seconds. Gawab.com also provides free POP access to their webmail accounts. The email delivery times are variable; sometimes authentication takes less than three seconds and other times it take almost thirty. Hotmail and Yahoo! Mail also provide POP service, but users are required to pay, so these were not examined for their usability with SAW.

The client can now register their nym (figure 4.2). The client begins by submitting their nym to the service provider. Once again the client must authenticate using SAW, but this time using the nym email account. An access control list is checked to see if the nym is blocked. If not, the SAW tokens are created and distributed. After authenticating to the nym account, the server checks to see if the nym is already known. If the nym is already known to the system, the user is logged in. If it is not known, the client is required to submit the registration token and unblinded signature received from the pseudonym server. Once received, the service provider checks the signature and, if valid, the client is logged in and the nym is stored.

4.3 Toolbar

In order to make CPG more usable and secure, the SAW toolbar for Firefox has been extended to aid users in acquiring a signed token and using a nym. The CPG-enabled SAW toolbar is able to distinguish whether a page is requesting a nym or a standard SAW email account. The toolbar also manages signed tokens for the client and warns the user of potential mistakes. Henceforth the CPG enabled SAW toolbar is simply referred to as the CPG toolbar.

When configuring an account, the CPG toolbar allows a client to distinguish the nym accounts from a non-nym account. An additional 'nym' checkbox has been added to the account configuration screen. By checking this box, the client allows the toolbar to know which accounts should only be used as nyms. This helps prevent a client from accidentally submitting an account that can be tied to the client.

In order for the CPG toolbar to recognize which type of account is being requested, the HTML page requested must contain a form with a special name. To recognize that a page is SAW enabled, and thus requesting a standard account, an HTML form must exist with the name `saw_login_form`. CPG extends upon this idea. To recognize if a page is requesting registration tokens for signing by the pseudonym server a form must exist with the name `cpg_register_form`. If this form exists the CPG toolbar displays a "Get Password" button. When a user clicks the button, the CPG toolbar is able to do the requisite blinding and submission of tokens. The signature received in response is unblinded and stored with the token value and domain that signed it.

As with registration, if a page is requesting a nym for logging in, a form must exist with the name `cpg_login_form`. When a client attempts to submit a nym, the toolbar checks to see if TOR is running. If it is not, the client is given a warning

and the chance to cancel the submission. If the nym is submitted, the signed token from the domain is submitted as well. Although not required with every log in, the signed token is submitted by the toolbar whenever a client logs in. The token can be deleted after the first successful log in, but it is difficult for the toolbar to know if the login attempt is successful, and it is harmless to submit the token with every attempt. The client is logged in if they are able to authenticate to the nym email address and the signature on the token is valid.

If the CPG toolbar is not used, the JavaScript code used to generate and blind the registration token is provided by the pseudonym server. This gives the pseudonym server the opportunity to act maliciously. The pseudonym server can easily generate a token it knows how to unblind or many other methods of leaving the option to link a client's nym and true identity. The client has the option of examining the code, but this is laborious and unreasonable. Without the toolbar, the client is forced to trust the pseudonym server. For this reason, the toolbar makes CPG both more usable and secure.

4.4 Anonymizers

For CPG to provide proper anonymity, it relies on an anonymizing service to obscure a client's IP address and scrub other information, such as a client's browser type. Three different types of anonymizing services have been tested for use with CPG: TOR, JAP, and CGI anonymizing web proxies. Each has its strengths and weaknesses, as discussed. A summary of results is presented in table B.2.

4.4.1 TOR

As discussed in section 2.3, TOR is *The Onion Router*. TOR has several attributes that make it an excellent anonymizer. First, TOR uses a circuit of en-

crypted links to obscure a client's IP address. The requested web service only sees the IP address of the last computer in the circuit, not the IP address of the client requesting information. Second, TOR leverages the scrubbing mechanisms of Privoxy to cleanse requests of identifying information such as a client's browser type and persistent cookies. Third, to make TOR more user-friendly, a Firefox extension allows for quickly enabling and disabling TOR. Finally, TOR makes use of a series of anonymizing proxies instead of a single proxy. This prevents a user from having to place a large amount of trust in a single source. These qualities provide a high level on anonymity and make TOR moderately easy to use.

TOR has several drawbacks. First, TOR's throughput is highly variable. Performance varies depending on the number of active TOR users, the quality of TOR servers, and many other factors. Some informal tests we performed showed that it is not uncommon to wait over twenty seconds for a page of 18 kilobytes to finish downloading. Without TOR this same page normally takes less than three seconds to finish. This is in line with findings in [6]. Second, TOR is not an option if a client is unable to install software (e.g., a public library computer, a corporate computer, a campus computer). Finally, it can be unclear whether or not the browser is routing through TOR. In our usability tests, one user assumed the browser was using TOR when first installed, when in fact it was not. The TOR icon appeared in the system tray so the user assumed that traffic was being routed through the TOR network.

4.4.2 JAP

JAP, the *Java Anonymous Proxy*, also has several positive traits. A positive aspect of JAP is that, like TOR, a client's trust is distributed over several anonymizing servers. No single server could be compromised to reveal a client's actions. JAP is also capable of scrubbing browser and operating system information from HTTP

headers.

Performance of JAP varies. The throughput of JAP depends on the number of clients using the cascade. A greater number of users means greater anonymity but reduced performance. Although a user must set the browser's proxy settings manually to use JAP, the JAP user interface provides a button that easily enables and disables its anonymizing capabilities.

As with TOR, the JAP interface can also be unclear on whether or not traffic is being routed through the service. A button on the JAP interface indicates if JAP is on or off, but does not indicate if a client's browser settings are correct. A meter on the interface displays the amount of activity a client is generating, but the meter does not reflect activity until the client generates some. At that point, the user may have already revealed their IP address.

4.4.3 CGI Anonymizing Web Proxy

Websites exist that offer anonymity to clients. A form is provided where a client can enter the site they wish to visit. The anonymizing site then acts as a proxy for the client, rerouting all traffic through the anonymizing page. The anonymizing website must be able to retrieve the requested site and rewrite any cookies or links to pass through the anonymizing website.

The primary advantage of an anonymizing website is that no software needs to be installed. This means that a client can attain some anonymity from any computer, provided the anonymizing website is not blocked by web filtering software. Many anonymizing web proxies also offer to block cookies, scrub identifying information, and block dynamic content. The other advantage of the anonymizing web proxy is that it is extremely easy to use. There is no configuring browser settings or fiddling with buttons to turn it on and off.

The disadvantage of an anonymizing website is that a great deal of trust is placed in a single source. Logs maintained by the anonymizing site can be used to trivially link a client's true IP address with their actions. The anonymizing website is trusted to either not keep any logs or never reveal them. Perhaps the greater danger is that the anonymizing website is acting as a trusted man-in-the-middle. Since the anonymizing website is able to see all data flowing between the client and the destination website, the anonymizing website is trusted not to abuse this information. Such information could include usernames, passwords, session cookies, credit card numbers, social security numbers, etc. In some cases, these sites can reduce security. Many do not offer a TLS-enabled connection. If a client is visiting a site that is protected by TLS, then by rerouting through the anonymizing site that does not offer TLS, all data between the client and anonymizing website is unencrypted. This leaves the information open for anyone in between to read the data.

Anonymizer	Strengths	Weaknesses
TOR	• Fast once the initial circuit is constructed • Easy to setup and use with Firefox • Trust for anonymity not placed on a single source	• Initial circuit construction may be slow • Requires ability to install software on client machine • May be unclear if browser is configured to use TOR • Inconsistent quality of service
JAP	• Easy installation • Trust for anonymity not placed on a single source	• Must configure browser manually to route through JAP • Requires ability to install software on client machine • Greater anonymity means slower transfer speeds
CGI Proxy	• No software installation • Very easy to use	• Great deal of trust placed in unknown source • Trust for anonymity placed on a single source • Non-TLS enabled proxies may add additional vulnerabilities

Table 4.1: Summary of the strengths and weaknesses of different types of anonymizers.

Chapter 5 — Threat Analysis

CPG is analyzed against well known attacks to determine if it is susceptible. The most threatening attacks are presented here.

5.1 Authentication

CPG inherits any weaknesses of the authentication method utilized. For instance, if a password is registered with a nym, the system may still be vulnerable to dictionary attacks. In the implementation created, SAW is used for authentication. SAW is susceptible to an active attack since SMTP traffic is often not secured. Assume an attacker knows of an email address on the ACL. The attacker begins by submitting the known primary email address to the pseudonym server. The pseudonym server responds by generating the requisite tokens, and the user token is delivered directly to the attacker. If the attacker is able to eavesdrop on the communication between the pseudonym server and the email provider, they will be able to intercept the email token as it is delivered. The attacker has now acquired both tokens and is able to authenticate as the victim and receive a registration token.

For this reason, the authentication method chosen for an implementation should be appropriate for the task. SAW is recommended for low to medium security situations and should be deployed in such settings. For higher security settings, digital certificates or two-factor authentication methods should be considered.

5.2 Timing

Timing is a delicate issue in online anonymity that is convenient to overlook but must be considered. Some possible threats are examined here related to timing.

As mentioned previously, it is important that a client wait after having a registration token signed before using it with the service provider. If a pseudonym

server and service provider are able to compare logs, it may be possible to link a client's primary email address with their nym. Consider a pseudonym server that logs when a client has a registration token signed and a service provider that logs when a nym signs in. The pseudonym server does not know the value of the registration token but knows which client has had a token signed and at what time. The service provider knows when a nym has been used to log in but does not know which primary identifier was used in exchange for signing the token. If a client were to have a token signed and immediately use it to gain service, the combined logs would show a primary identifier being exchanged for signing a nym and a nym being used to gain service a moment later. While on an extremely active pseudonym server this may not be a problem, on a less active server there would be a strong indication that the nym belonged to the owner of the primary identifier. For this reason it is recommended that a client wait a period of time before using a newly signed token.

Another attack uses packet timing to discover a client's identity. Consider the case where the service provider is also able to log packets coming out of a client's computer. This might happen on a corporate network. An employee connects to the corporate network to access a CPG enabled service hosted by the corporation. Even though the employee has been careful to enable TOR, the corporation's network administrator is able to correlate packets leaving the client's machine into the TOR network with packets entering the server hosting the service from the TOR network. JAP attempts to thwart this attack by generating filler traffic when a client is idle, but this may not be enough to prevent raising suspicion. For these reasons it is recommended that a separate network be used to access a service from that where it is hosted.

5.3 Cookies

Cookies can be a threat to revealing a client's identity. A pseudonym server in the same domain as the service provider, or a third party colluding with the pseudonym server and service provider, can place a cookie on the client's machine when the client registers in an attempt to reveal their identity. The cookie can be set with information used to identify the client; the primary email address used, the time the token was signed, or an identification number created specifically for that client. When the client returns to use their nym, the cookie is sent to the service provider or third party allowing them to reveal the identity of the nym owner.

To prevent this attack, the client should refuse all third-party cookies and destroy all cookies from the pseudonym server's domain after having a token signed. All web browsers allow a client to delete cookies manually. In some browsers (e.g., Firefox) cookies can be destroyed automatically at the end of every session. This setting is recommended if it is available. Turning cookies off completely thwarts this attack, but many websites require cookies for functionality.

Chapter 6 — Usability

Two separate usability studies were conducted on the implementation of CPG to determine its ease of use and improve usability. The usability studies were performed on two separate classes over periods of two weeks each. Both classes were Brigham Young University computer security courses, so students were all computer savvy individuals who were familiar with the basics of computer security. Participants were directed to an online set of instructions detailing how to set up and use CPG. Students were to complete the instructions in their spare time without help. In the instructions, the unblinded signature is referred to as the user's password. This was done to help users to more easily understand how to use the system. Each student was also informed through email or a handout which email address was to be used for registration. This email address was associated with the student's class registration, so it was assumed the address was correct and usable. They were given the first week to obtain a signed registration token and the following week they could sign in with their nym and post to the message board. After the two week period was over, students were given a set of questions to answer about their experience using CPG. A summary of the results can be found in the appendix in section B.2 and B.3.

6.1 Study 1

The CPG design used by 5 students in the first usability study was slightly modified from the final design presented above. In the former design, a client's nym email address was hashed, blinded, and signed by the pseudonym server instead of a registration token (see section 3.1.1). This required students to create a nym prior to having it signed. Later, in stage 3 (see section 3.1.3), clients were authenticated to the nym and required to submit the signature on their nym when logging in for

the first time. This proved to be problematic as demonstrated by two users.

The first user, instead of having their nym email account signed, made up an unrelated, non-email nym and had it signed instead. Then when the user went to log in, they entered in the non-email nym and submitted it. The pseudonym server was unable to complete authentication since the service had nowhere in which to send the email SAW token.

Another user mistyped their nym email address when configuring the nym account in the CPG toolbar. The user's misspelled nym was then submitted to the pseudonym server and signed. When the user went to log in for the first time, the incorrectly spelled email was not on the ACL so an error message was sent to the address. The user was also unable to check the email account for messages. After realizing the mistake, the user attempted to have the correctly spelled nym signed, but their primary email address had been removed from the ACL preventing them from having another nym signed.

There is evidence that the toolbar will prevent user error. One user preferred not to use Firefox, so he was unable to use the CPG toolbar. This student made a mistake that would have been corrected by the toolbar. The student attempted to use his nym email account in stage 1 instead of his primary account. Since the nym email account was not on the authorization list, the user received an error message. As a result the user was, "having all sorts of trouble with this." If the user had been using the toolbar, he would have been unable to submit their nym account.

The lessons learned in the usability study resulted in improvements to the CPG design to its current form. Clients had problems with signing their nym because they were not required to prove ownership before having it signed. The pseudonym server was unable to examine the nym to verify its validity since it had been blinded.

As a result, these users had the wrong nym signed. They were then incapable of authenticating to the nym and were unable to have the correct nym signed since their primary email address had been removed from the ACL. By signing a registration token instead of the nym, the client must authenticate to the nym before spending the registration token.

6.2 Study 2

A similar study of the improved CPG design was conducted involving 10 students. This study highlighted three main problem areas.

The first problem area is the instructions. Four of the users complained that the instructions needed improvement. One user complained the instructions were "fairly long and tedious." Another user complained the instructions needed to include a general overview of how the instructions would proceed. The user commented, "I didn't understand why I was doing something, even though I knew exactly how to do it."

The second problem area is with TOR. The majority of users found the inconvenience of TOR to be moderate or great (see figure B.3). Users complained responses from TOR were either too slow or were never received. The poor response times drove two users to disable it and proceed without it, jeopardizing their anonymity.

SAW showed room for improvement. Three users complained of long delays while waiting for the toolbar to retrieve the email. When asked how difficult it was to get their password and sign in using the toolbar, one user responded, "Very Difficult. This has actually been preventing me from logging in. I let it go for one or two minutes, but it won't always sign me in." When confronted with this problem, users unfamiliar with SAW did not know how to proceed.

The toolbar was shown to make it easier for users to get their password and sign

in than without the toolbar. When asked how difficult it was to get their password and sign in with the toolbar, eleven of the fifteen respondents found it either "easy" or "very easy" (see figure B.3). Of the users that tried to get their password and sign in without the toolbar, two of the three found it either "difficult" or "very difficult" (see figure B.3).

Chapter 7 — Conclusions and Future Work

CPG is a novel framework that balances client anonymity with the needs of a service administrator. Clients can act anonymously, but administrators can block nyms that act maliciously. As evidenced by Wikipedia editing being unavailable to TOR users, administrators need a level of control to stop abuse or many services will not be available for anonymous users.

CPG provides a strong separation between a client's nym and their true identity. If used correctly even the administrator of a service is unable to discover a nym's true identity. If an administrator wishes to keep a user's identity anonymous, it also prevents outside sources from strong-arming an administrator into revealing a client's identity.

CPG provides the mechanics to remain anonymous, but it can not prevent a user from disclosing their identity through behavior. A student who complains relentlessly about the same topic in class as on a CPG-enabled class message board leaves little doubt about their identity. The following is a list of assumptions pertinent to maintaining a user's anonymity and can serve as a checklist to a client when using CPG.

- A pseudonym should not reveal information about its owner
- A client computer should be on a separate network from the service provider
- A client should wait a sufficient time between receiving and using a registration token
- Cookies from the pseudonym server should be deleted after receiving a regis-

tration token

- An anonymizer should be on and configured correctly when interacting with the service provider

- A client should not reveal identity information to the service provider

CPG is flexible enough that it can be incorporated into an existing service with only minor changes. CPG has the ability to make use of existing authentication and authorization infrastructures within an organization. It has also been shown that CPG is applicable in many situations, and, with the aid of the CPG toolbar, CPG is fairly easy to use correctly. In response to the questionnaire, all but four users answered that it was either "very easy" or "easy" to get their password and sign in with the toolbar (see figure B.3).

A natural conflict exists between smaller sized closed groups suggested by this thesis and achieving anonymity. It's well known that better anonymity is achieved in larger groups. A study to discover the minimal size of a group while still achieving useful anonymity is left for future work.

The usability of CPG can still be improved. The process of installing an anonymizer and toolbar, configuring the toolbar, signing up for a separate email account, and then following one type-written page of instructions remains too tedious for most users. To simplify the process, the toolbar could be bundled with the anonymizer and installed concurrently. Using the toolbar, it might also simplify registration by authenticating and requesting a signed token in a single step. The user will configure the toolbar with their primary email address and navigate to the registration page. The toolbar will then perform SAW authentication and, if the authentication succeeds, the toolbar will automatically generate and have the registration token

signed as well.

The usability of CPG as implemented is also highly dependant on the usability of SAW. Free email accounts that allow POP or IMAP access and are suitable for anonymous use remain sparse. The email providers that fit these criteria may have spotty response time making them frustrating to use. When the SAW toolbar supports authentication through instant messaging services (e.g. Google Talk, MSN Messenger, Yahoo! Messenger, AIM), the number of services and response times should increase dramatically.

A long term usability study is left to future work. This type of study may reveal new ways in which users surrender their anonymity. For instance, is the student who frequently talks in class also the student who frequently posts on the message board? Using CPG in production may also provide new insights in adding and removing clients in persistent environments. The current solution of adding and removing clients in batches or invalidating all current nyms works but is inconvenient.

Finally, our results identify a challenge to the developers of TOR and JAP. Our usability study has shown that it can be unclear to users whether or not traffic is being routed through the anonymizer. Anonymizing software should be as easy to use as a CGI proxy anonymizer, but with the anonymity afforded by systems like TOR and JAP.

References

[1] JAP Anonymity & Privacy. JAP Anonymity & Privacy, 2006.

[2] Brad Stone. A Call for Manners in the World of Nasty Blogs. Website, April 2007. http://www.nytimes.com/2007/04/09/technology/09blog.html?ex=1333857600&en=8df0ef9fe934fc04&ei=5124.

[3] Business Week/Harris Poll: A Growing Threat. Business Week/Harris Poll: A Growing Threat. Website, March 2000. http://www.businessweek.com/2000/00_12/b3673010.htm.

[4] David Chaum. Blind Signatures for Untraceable Payments. In *Advances in Cryptology: Proceedings of CRYPTO '82*, Plemum, New York, 1983.

[5] Privoxy Developers. Privoxy - home page. Website, 2006. http://www.privoxy.org.

[6] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the Thirteenth USENIX Security Symposium*, San Diego, CA, Aug 2004.

[7] Ian Goldberg. Privacy-enhancing technologies for the Internet,II:Five years later. *Lecture Notes in Computer Science*, 2482, 2003.

[8] David Gray. Anonymous Coursework Submission. In *Proceedings of the 4th International Conference on Computer Systems and Technologies: e-Learning*, Rousse, Bulgaria, 2003.

[9] Jason E. Holt and Kent E. Seamons. Nym: Practical Pseudonymity for Anonymous Networks. Website, June 2006. `http://isrl.cs.byu.edu/pubs/isrl-techreport-2006-4.pdf`.

[10] Aviel D. Rubin Marc Waldman and Lorrie Faith Cranor. Publius: A robust, tamper-evident, censorship-resistant, web publishing system. In *Proc. 9th USENIX Security Symposium*, pages 59–72, August 2000.

[11] Declan McCullagh. FBI director wants ISPs to track users. Website, October 2006. `http://news.com.com/2100-7348_3-6126877.html`.

[12] Opinions on Internet Privacy Graphs. Opinions on Internet Privacy Graphs. Website, April 1997. `http://www.gvu.gatech.edu/user_surveys/survey-1997-04/graphs/privacy/Opinions_on_Internet_Privacy.html`.

[13] Michael K. Reiter and Aviel D. Rubin. Anonymous Web Transactions with Crowds. *Communications of the ACM*, 42(2), February 1999.

[14] Peter Steiner. On The Internet, Nobody Knows You're a Dog. *The New Yorker*, 69(20), July 1993.

[15] Joseph Turow, Lauren Feldman, and Kimberly Meltzer. Open to Exploitation: American Shoppers Online and Offline. Website, June 2005. `http://www.annenbergpublicpolicycenter.org/Downloads/Information_And_Society/Turow_APPC_Report_WEB_FINAL.pdf`.

[16] Timothy W. van der Horst and Kent E. Seamons. Simple Authentication for the Web, 2006.

Appendix A — Instructions

The following is the set of instructions given to students during the usability study.

What is CPG?

CPG provides user anonymity while preventing continued abuse from malicious users. CPG allows a user to communicate with a website through a personal pseudonym rather than using their true identity. If a user begins misbehaving, a website administrator is able to block the malicious pseudonym, but is unable to link the pseudonym to the user's real identity.

Prerequisites

1: Download and install Firefox

Hopefully you already have Firefox. If not get it and install it here:

http://getfirefox.com

2: Download and install the SAW toolbar

Download and install the SAW toolbar from the link below.

http://websaw.org/sawtoolbar

Install the toolbar by opening the file in Firefox and following the instructions.

3: Create a new email account for your nym.

You will use a new email address as your pseudonym, or nym. Any email account with POP or SMTP access will work, such as GMail, but the account should be unlinkable to yourself.

Unless you know better, it is recommended you create your nym with one of these free email providers:

http://mail.google.com/mail/signup

When creating your nym email address:

A. Do not have your name or any identifying information in the address. If your name is Reed, reed@cs.byu.edu would be a poor choice for a nym.

B. Do not use an email account where the domain could identify you. If you own porkbelly.com, using an email address from that domain would be a poor choice.

C. Do not use an email account other people know you own. It's best to create an entirely new account that is unlinkable to yourself.

4: Install TOR

Your IP address can be used to link your pseudonym to your identity, so TOR helps hide your IP address. Download and install TOR from here:

http://tor.eff.org/download.html.en

Configure Your Nym SAW Account

Now let's configure the SAW toolbar so it can access your nym email account. Your nym email account is the new account you just created that is unlinkable to yourself.

To configure SAW, follow these steps:

1. Start Firefox

2. The SAW toolbar should be visible. Click 'Configure Accounts...'. If the toolbar does not say 'Configure Accounts...', you may have to click the down arrow on the toolbar to access it.

3. Click 'New Account'. Fill out the fields with your account information.

4. Click on the 'Nym?' checkbox to let the toolbar know that this is your nym. If using a GMail account, you MUST enable POP access from within your GMail account!

Get Password

Once your account is created, you're now ready to get your password.

1. Navigate to

https://webcpg.org/register.php

2. Enter and submit your primary email address. This is the email address you have registered with the class. This is not the email address you just created.

3. Check for a new email in the account you just submitted. Follow the link in the email. If it says you are not authorized, you may have submitted the wrong email address. If you are sure you submitted the correct email, contact the TA.

4. If you are authorized, you will be taken to the password page. The SAW toolbar will recognize that a password is available for you. Just click on the 'Get Password' button on the toolbar.

5. A popup shows the password when it is returned, and the toolbar stores it for you.

6. Clear your cookies. In Firefox, go to Tools-¿Clear Private Data. Select 'Cookies' and click 'Clear Private Data Now'.

Wait

Before signing in with your new password with your nym account, you need to wait. Typically a minimum of 1 day is recommended. The waiting period is used to prevent a timing attack that allows a website to link your true identity to your nym. See the document proposal.pdf for more details.

Use Your Nym

Now that you have your password and the waiting period is over, you may now use your password to log in with your nym.

1. Start Firefox

2. Enable TOR by clicking on the TOR button in the bottom-right hand corner of your Firefox browser.

3. Navigate to

https://webcpg.org/login.php

4. Using the SAW toolbar, click on the nym you would like to use to log in.

5. If you did not save your password when configuring your SAW account, a dialog box will request your email account password. Enter your password and click 'OK' to log in.

6. The SAW authentication progress window will appear as it tries to authenticate you. If it runs for more than 30 seconds, you may not be authorized or you may have a problem with your account configuration.

7. Voila! If authentication succeeds, you have successfully logged into the site. You may now post messages to the message board.

Appendix B — Questionnaire

The following is the questionnaire students received after participating in the usability study. A summary of responses is provided.

B.1 Questions

CPG Questionnaire

1. How difficult was it to get your password?
   A. Very Easy
   B. Easy
   C. Moderate
   D. Difficult
   E. Very Difficult
   F. Did not do it
2. When accessing the message board, how often did you use an anonymizer (such as TOR)?
   A. Never
   B. Rarely
   C. Sometimes
   D. Often
   E. Always
   F. Did not Use It
3. How much of an inconvenience was the anonymizer?
   A.None
   B. Small

C. Moderate

D. Great

E. Extreme

F. Did not Use It

4. How difficult was it to get your password and sign in using the toolbar?

A.Very Easy

B.Easy

C.Moderate

D. Difficult

E. Very Difficult

F. Never Tried

5. How difficult was it to get your password and sign in without the toolbar?

A. Very Easy

B. Easy

C. Moderate

D. Difficult

E. Very Difficult

F. Never Tried

6. Did anyone else find out your anonymous identity?

1.Yes

2. No

If yes, was it deliberate or accidental?

1. Deliberate

2. Accidental

7. Was the ability to access and use the site anonymously useful?

1. Yes

2. No

Why or Why not?

8. What did you find difficult or inconvenient about CPG?

9. How might we improve the process?

10. Additional Comments:

B.2 Response Table

The following is an ordinal summary of the number of user responses to each multiple choice question in the usability questionnaire. The first study had 5 participants and the second had 10.

	Study 1							Study 2					
	A	B	C	D	E	F		A	B	C	D	E	F
Question 1	0	1	2	2	0	0		0	4	4	0	1	1
Question 2	0	0	0	1	3	1		0	1	0	2	4	3
Question 3	0	0	1	3	0	1		0	0	3	4	0	3
Question 4	1	3	0	0	0	1		4	3	0	0	2	1
Question 5	0	0	0	1	0	4		0	0	1	0	1	8
Question 6	1	4	-	-	-	-		0	10	-	-	-	-

B.3 Response Charts

This section compares the percentage of each response for each question. Study 1 is on the left, and study 2 is on the right.

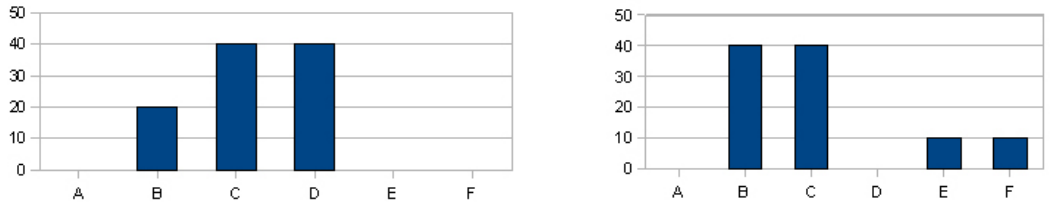

Figure B.1: Question 1

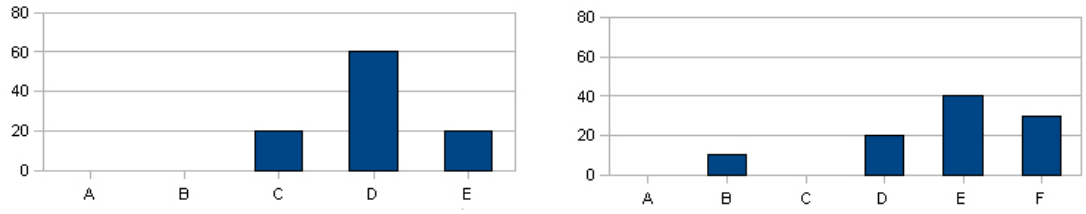

Figure B.2: Question 2

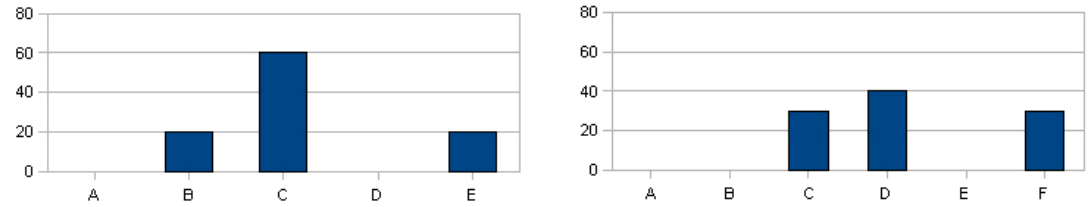

Figure B.3: Question 3

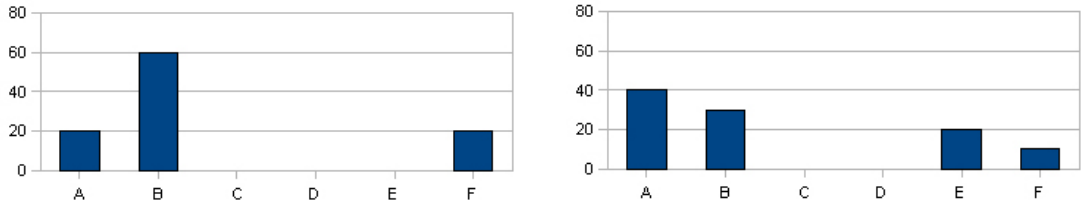

Figure B.4: Question 4

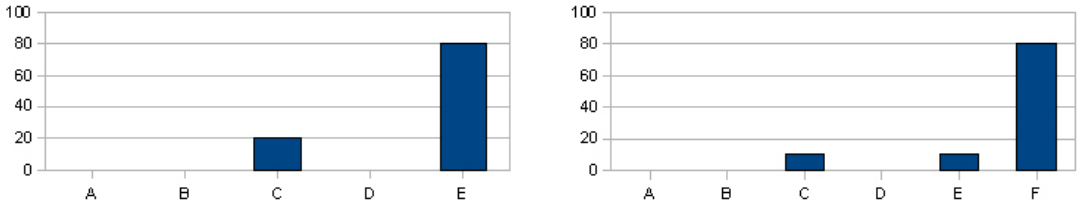

Figure B.5: Question 5

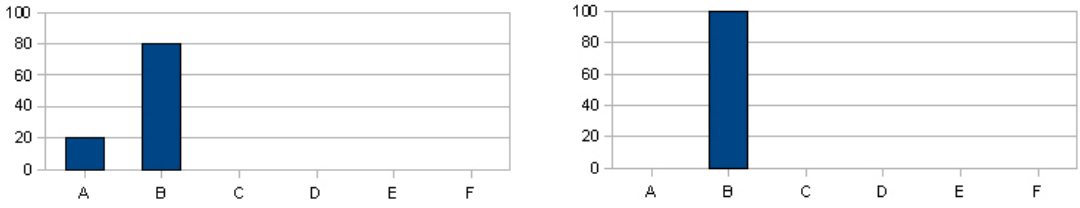

Figure B.6: Question 6