A collection of Privacy Design Patterns

Munawar Hafiz
University of Illinois at Urbana-Champaign
e-mail: mhafiz@uiuc.edu

August 3, 2006

Abstract

The growth in computing power has enabled the storage and analysis of large volumes of data. Monitoring the Internet access profiles of millions of users has become feasible and also economically lucrative. The interesting thing here is that it is not only the crooks who are interested in privacy intrusion, but government agencies also have vested interest in profiling the population mass. This paper describes 4 design patterns that can aide the decision making process for the designers of privacy protecting systems. These design patterns are applicable to the design of anonymity systems for various types of online communication, online data sharing, location monitoring, voting and electronic cash management.

1 Background

According to Wikipedia, privacy is the ability of an individual or group to keep their lives and personal affairs out of public view, or to control the flow of information about themselves [36]. Privacy is most highly valued by people who are publicly known; but it is also coveted by people just because they do not want to turn their activities into a public spectacle. A closely related term to privacy is anonymity. However, in the real world, people do not mean to be anonymous in order to retain their privacy. Rather privacy is considered to be the choice that a person has to disclose or hide his activities.

An average Internet user performs a significant amount of communication and transactional activities daily. Users go to great length to secure their activities, *e.g.* encrypting data packets to make them confidential, adding a hash value to prove that the data packets are not tampered *etc.* These protect the application content, but still a lot of information can be harvested about a message sender (and maybe message) by monitoring his message sending habit. The growth in computing power has facilitated this activity. Other than concealing Internet activities, privacy issues are important requirements for many systems. For example, in an electronic voting system, it is imperative that a vote cannot be traced back and correlated with the voter.

The common approach to concealing information is obfuscation. However, the obfuscation mechanism would have to retain the usability of a system. For example, suppose the sender of an email uses an obfuscation mechanism to hide his identity. But the recipient of the email has to be able to reply to the sender. The obfuscation mechanism would have to be such that it allows this basic requirement.

This paper discusses these issues involving the design of privacy preserving systems. The four privacy patterns, described in this paper, are part of a larger piece of future work on privacy patterns that has nine patterns in total. These patterns are applicable to the design of anonymity solutions for various domains. The background section presents a survey of the privacy patterns, and then provides a short description of the patterns in this paper and the larger piece of work. The section also covers the basic concepts of privacy and list the conventions used in the paper.

1.1 Related Work

There has not been a lot of work on Privacy Patterns. Markus Schumacher covered two privacy patterns in the seminal paper [29]. The *Protection against Cookies* pattern describes how to control the cookies in a web client. The *Pseudonymous Email* pattern describes the mechanism of a pseudonymous email delivery system. Till Schummer, in his paper that deals with information filtering in collaborative systems [30], described patterns that block the transmission of personal information. Sadicoff *et.al* workshopped one privacy pattern in PLoP 2005 [28].

In 1997, Goldberg *et.al* [14] wrote the classic survey paper of privacy preserving systems and related issues. Goldberg followed up this work five years later [13]. Pfitzman and Waidner described the basic concepts of privacy in their 1987 paper [24]. The 1998 paper on Crowds system [26] contains a detailed description of measuring privacy.

1.2 Privacy Concepts

According to Pfitzman and Waidner [24], there are three types of anonymity properties - sender anonymity, receiver anonymity, and sender and receiver unlinkability. Sender Anonymity means that the identity of the party who sends the message is hidden. This does not require the message to be encrypted. A plaintext message that does not have any trace that can be used to link it back to the sender does not provide data confidentiality, but provides sender anonymity. Receiver anonymity means that the identity of the receiver of a message remains hidden. Sender and receiver unlinkability means that though the sender and receiver can be identified as participating in some sort of communication, they cannot be correlated to participate in a conversation (*i.e.* communicating with each other).

Pfitzman and Waidner [24] also classify the attackers against whom the privacy properties would be achieved. A local eavesdropper can observe all (and only) communication to and from the user's computer. A more powerful eavesdropper can monitor all the data traffic in a local network. A hypothetical global eavesdropper is omniscient of all the activities in the network in a global scale. Adversaries can also be classified as active, semi-honest and passive. A passive adversary just monitors the packet, it does not manipulate the data packets flowing through it like the active adversary. A semi-honest adversary follows the network protocol and appears to be honest, but the adversary manipulates the through traffic. Adversaries can work on their own, or they may be colluding.

Reiter and Rubin [26] added a third aspect of anonymous communication: the degree of anonymity. The degree of anonymity is described informally as a continuum with the following points.

- **Absolute Privacy.** There is no way to violate the privacy of a user.
- **Beyond Suspicion.** The user is no more likely to be related to a message than any other user of the system.
- **Probable Innocence.** From the attacker's point of view, the user appears no more likely to be related than not to be related with a message. This is weaker than beyond suspicion.
- **Possible Innocence.** This is weaker than probable innocence. From the attacker's point of view, there is a non-trivial probability that the person related with a message traffic can be someone other than the person in question.
- **Exposed.** The attacker can correlate the message with the user. This is the default degree of anonymity form most transactions in the Internet.
- **Provably Exposed.** The attacker can not only identify the correlation between a user and a message traffic, but also can prove it to someone else.

1.3 Privacy Pattern Catalogue

This paper is part of a larger future project that lists nine privacy patterns. We present a summary of the patterns in the larger catalogue. Figure 1 provides an overview of the patterns and their relationship.

Anonymity Set
Hide the data by mixing it with data from other sources.

Morphed Representation
Change the representation of the data when it is passing through an anonymity providing node so that outgoing data cannot be linked with incoming data.

Hidden Metadata
Hide the meta information associated with data content that reveal information about sensitive data content.

Layered Encryption
Use a sender-initiated packet routing scheme and encrypt the data packets in multiple layers so that the intermediaries only have access to a particular layer and use that information to route the packet to the next hop.

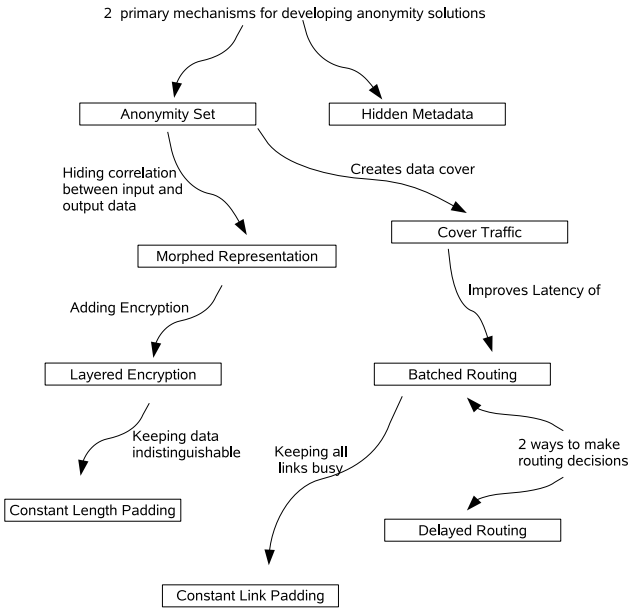

Figure 1: Privacy patterns in the catalogue and their relationship

Cover Traffic
Keep a dummy traffic flow between anonymity preserving nodes to create a decoy for actual data traffic.

Batched Routing
In a mix based system, collect the input data packets and when the collection reaches a threshold output all the data packets together.

Delayed Routing
Add random delays to the incoming data traffic of an anonymity preserving node to thwart the timing attacks.

Constant Length Padding
Add padding to data packets to make them of same length.

Constant Link Padding
Distribute data traffic equally among all the outgoing nodes from an anonymity preserving node.

1.4 Conventions used in the paper

The fictional characters Alice, Bob and Carol are used to represent parties communicating in the Internet. Server M and server N are used to denote a typical origin server in a web browsing application or a typical MTA in the message recipient's domain, in both cases the ultimate recipient of the message traffic. The characters Mallory and Eve are used to denote adversaries.

When describing messaging systems, the terms sender and recipient, and input and output are used interchangeably. When the anonymity system is considered from an end-to-end perspective, the terms sender and recipient are used to denote message sender and message recipient respectively. When the anonymity system in considered from the perspective of the anonymity providing node, the term input is used to denote the incoming traffic of the node and the term output is used to denote the outgoing traffic of the node. The term anonymity providing node signifies the artifact that is deployed in the network to provide anonymity service.

Several figures in the paper illustrate the transformation of data traffic in the Internet. A typical data packet is shown in figure 2.

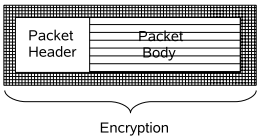

Figure 2: A typical data packet

A data packet has a header and a body. The packet header contains the identification of the sender and the recipient and is used for routing. The packet body contains the actual data, that may or may not be encrypted. Data encryption is shown with a coating around the packet. A packet encryption or a packet body shown with a different hatching pattern means the representation of the packet has changed because of new encryption/encoding.

Anonymity Set

Anonymity

Intent
Hide the data by mixing it with data from other sources.

Also Known As
Probable Suspect.

Motivation
The Athenian Mistake. In a message communication scenario, the content of the message is not always important. Merely the fact that a sender is sending a message can reveal important information. Suppose there are two regions Athens and Sparta, that are going through troubled times. The threat that one can launch a preemptive attack on the other is imminent. The Athenian army hired a veteran cryptographer who devises an unbreakable cipher. The intelligence branch of Sparta has not been able to decrypt this cipher scheme, but they have under-cover probes that let them know who is sending message to whom, a correlation that the Athenian army is not choosing to conceal.

Deep into one night, Athens decides to launch the attack the next morning. Suddenly there is a flurry of messages passing among the chain of command of the Athenian army. Spartan intelligence picks up the information that suddenly the Athenian Generals have become active late into the night. They mobilise their army that night. Athens was not prepared for the counter-strike. They loose the battle.

Protected Health Information. Health Insurance Portability and Accountability Act (HIPAA) [34] of 1996 defines the appropriate way to handle Protected Health Information (PHI). For research purpose, Cure Clinic is releasing its PHI about cancer victims to Acme Laboratories. Let us suppose that Cure Clinic was keeping the patients' name, birth date, sex, zip code and diagnostics record. To protect privacy, Cure Clinic does not release the name of the patients. The birth date, sex, zip code and diagnostics records are released. Acme Laboratories is doing the research on the probability of cancer attack on a particular age group and sexual orientation over the people of a particular locality. So all the data fields that are released are important for the research. Mallory is a malicious worker at Acme Laboratories who wants to unravel private information from this data. Mallory goes to the city council of a particular area and gets a voter list from them. The two lists are matched for age, sex and locality. Mallory finds the name and address information from the voter registration data and the health information from the patient health data.

Context
You are designing a system to protect the privacy of the users. This system will maintain sender and/or recipient anonymity in a messaging scenario. Although this is an important application area, the context is not limited to messaging only. The context also entails other scenarios like anonymity in a location tracking system, anonymous voting in an electronic voting system, or anonymity preserving data sharing in a data publishing system.

Problem
It is difficult to ensure sender and recipient anonymity during message communication. The only concern of

traditional security approaches is to protect data content. This does not hide the message path from the sender to the receiver and thus the anonymity is compromised.

In an electronic voting system or an online voting system, the system should protect the privacy of the voters by not revealing their vote.

Similarly, a user may want to hide his information from a location tracking system. This may be because the location tracking device is offering some context-aware service based on user location but the user is not interested at the moment, or may be because the user is not trusting the location tracking system, or may be because the user does not want to reveal his private location information at all.

When private datasets are released, the private data about the subjects may be exposed. The released dataset has to be sufficiently rich in order to be useful, but it also should protect the privacy of the entities.

In all the cases, the general problem is to ensure the anonymity. How can the anonymity of an entity or a personal information be retained?

Forces
The forces that need to be considered when choosing to use this pattern are as follows.

1. *User Count.* The number of users using an anonymous messaging system may vary with time. This fluctuation may depend on operational hours, user interest, *etc.* If the user flow is low, the solution does not work because there are not enough candidates to create the anonymity set.

2. *User Friendliness.* Users should be able to use the privacy-enabling mechanisms with minimal alteration of their primary task. If the users have to adapt a lot to achieve anonymity, they may start judging where they should have anonymity. This way, a user's misjudgment can sometimes reveal private information.

3. *Data Usability.* An anonymous data set has to be usable. One extreme of achieving anonymity is not to release any data, but obviously this is not a usable scheme.

4. *Performance.* The privacy retaining operations should not become a performance bottleneck (*i.e.* latency, bandwidth *etc.*). For anonymous messaging, the system should be usable in low latency usage scenarios, like web browsing.

5. *Law Enforcement.* Law enforcement agencies might require that the anonymity solutions sometimes lift their anonymity cover to investigate on crime suspects. This would prevent a malicious user from abusing anonymity.

Solution
Mix the private information with other information so that the private information is not distinguishable from other information. Create a set of equally probable information and hide the user information by making it a part of the set. This set is called the anonymity set. If the size of the anonymity set is large, it will ensure stronger privacy.

For message communication, create an abstract mechanism between the message sender and the recipient that hides the correlation between the sender and the recipient. This can be done by network intermediaries

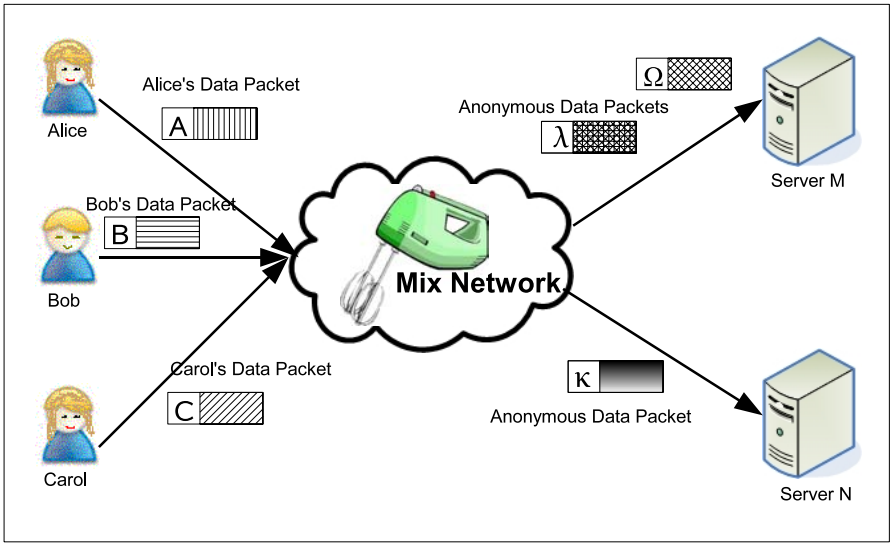

Figure 3: Sender anonymity in a mix network

called mix networks that mix the message coming from one source with messages coming from different other sources. Once the data packet from the sender passes through one of these filters, it is indistinguishable from other packets (*i.e.* sender anonymity). The anonymity set is the set of messages from different sources. Figure 3 illustrates this solution. Alice's data is collected at the mix node and mixed with Bob and Carol's data.

For recipient anonymity in message communication, broadcast the message to all users or send the message to a message pool instead of one single recipient. The recipient will view the message like everyone else but an adversary will not be able to tell who that message is for.

Use the same idea for sender anonymity in a location monitoring system. Install the abstract obfuscation mechanism in a region. Agents are identified by pseudonyms in the location tracking system. Once an agent enters the region where the obfuscation mechanism is installed, he is given a different pseudonym. If there are multiple agents in the obfuscation region and all of them adopt a different pseudonym upon entry, then the agents in the region create the anonymity set at a particular point of time. Once the agents come out of the region, their new pseudonyms cannot be correlated with the old pseudonyms.

In an anonymous voting system, a voter's vote is passed through an obfuscation mechanism where it is mixed with other votes and the generated outcome leaves no trace that can be used to link the voter with the vote.

When releasing private data sets for public use, change the specific values of attributes that might reveal private information to more generalized values. If the dataset has a specific gender information, change the values of gender (male/female) to more general values (person). Also partition the attribute domain space and provide partitioned information rather than exact information. For example, if the dataset has specific age information, create age groups and convert the dataset such that the specific ages are replaced with age groups. What this mechanism is doing is that it is creating an anonymity set so that one row in the dataset becomes indistinguishable from another.

Design Issues

Size of the Anonymity Set. The size of the anonymity set will determine how good the obfuscation will be.

If the set is small, then correlation between input and output of the obfuscation mechanism can be determined with higher probability. If the anonymity set has only one element, then the privacy is provably compromised.

Latency. In a messaging scenario, the mixing mechanism might stall the data traffic to wait for enough data packets to arrive so that the mixing can be done effectively. This means that the latency of the data flow increases, which might make it unusable in a low-latency messaging scenario like web browsing. Different strategies can be taken to counter the latency issue. The required degree of privacy is scenario specific, and based on that the designer can identify the trade-off between privacy and performance.

Usability of Information. In the case of dataset release, absolute data obfuscation is possible by replacing all the specific attribute values with more general values. But this way the datasets may not be useful. For example, if a research is interested in the impact of sexual orientation on cancer attacks and the dataset has been anonymized in a way that all the gender values are replaced with a more general 'person' value, then this dataset becomes useless for the research purpose. So an anonymity protection mechanism that retains the usability of data is required.

Consequences
The pattern has the following benefits.

1. *Privacy.* The obfuscation mechanism ensures that private information is not easily compromised. Not all mechanisms provide absolute privacy, but they ensure that the attacker will have to do more work to break into the system's sensitive information.

2. *Freedom from User Profiling.* Business entities are interested in user profiling to make smart advertisements. Users may not want to be bothered by these marketing suggestions. An anonymous user is free from such user annoying sales mechanisms.

3. *Minimal user involvement.* The users do not have to modify their normal activities to get anonymity service. The service is provided by proxies resident at the user end and the intermediaries in the network. Usability is improved because of this transparency.

The pattern has the following liabilities.

1. *Performance.* In the messaging domain, when the system is waiting for enough probable suspects to arrive to mix with incoming traffic, the users experience increased latency. Cover traffic can be used to create dummy probable suspects. But maintenance of a cover traffic flow is expensive in the bandwidth. Also the mix nodes might employ a batched transaction strategy that causes flush traffic out of the anonymity nodes. In that case, bandwidth becomes a big factor.

For data anonymization, it has been proven that general data obfuscation mechanism is NP-Hard [20]. In a location anonymity system, adding effective obfuscation mechanism (by introducing cover traffic) is very computation extensive.

2. *Usability of Information.* Too much data obfuscation can undermine the usefulness of data. In the case of private dataset publishing, if all the attributes of the dataset are anonymized such that they retain privacy, the resultant dataset may not be useful at all. Queries of finer granularity (that may be important for the research for which the dataset was made public initially) can not be served.

3. *Abuse of Privacy.* Anonymity systems are open to abuse by malicious users. An anonymous sender might be encouraged to send a hate mail in a public forum showing his ethnic bias. Sensitive information about a person can be posted anonymously to commit a smear attack. Terrorists might want to use the anonymity mechanism to communicate between themselves. Strong privacy guarantee for the end user makes the task of crime-fighting very difficult.

Known Uses

The Mix based networks [5] are based on the idea of mixing the incoming data traffic from one user with the data traffic coming from other users. Each mix has a public key which is used by the message senders to encrypt the message between the user and the mix. The mix accumulates these messages, decrypts them, optionally re-encrypts the messages and delivers them to the subsequent node. If there is a sufficient amount of input data packet from different sources, the mix ensures that the sources can not be linked with the data packets once the packets come out of the mix. Users should not trust only one mix. Instead, they should send their data through a cascade of mixes. In this way, weak anonymity is preserved even if some of the mixes are honest (*i.e.* not run by an adversary). The first widespread public implementation of mixes were produced by contributors of the Cypherpunks mailing list [7]. Then Mixmaster [8] and Babel [16] were based on the mix network idea to send anonymous emails. These systems are called remailers. Mixmaster was a Type II remailer (Cypherpunks remailers were Type I remailers). Mixminion [9] is a Type III remailer that addresses the problems of previous generations of remailers like the sophisticated flood and trickle attacks. The types associated with the remailers generate different generations of these remailers.

Onion Routing [15, 33] systems are based on mixes but they leverage the idea of mixes and add layered encryption. Onion Routing systems have better latency values than mix networks and therefore are more applicable in a web browsing scenario. Crowds [26] is another system that provides low-latency data anonymization. Its approach is different from onion routing or mix schemes. Here, every node in the network is similar and every node can either forward the data packet to another node in the network or send the request to the end server based on a probabilistic coin toss. Thus every node is a probable suspect of being the originator. In this way, Crowds provides plausible deniability for the message sender.

Hordes [31] uses multicast routing where every responder gets every message. This provides recipient anonymity.

The Votegrity system based on by Chaum's secret-ballot receipts [6] and the VoteHere system based on Neff's secret shuffle algorithm [21] use the concept of mix networks for mixing the votes.

Mix zone [3, 4] is the same concept of mix networks taken into the domain of location anonymity. The mix zone concept was added with the Active Bat [35] system to provide location anonymity.

The principle of k-Anonymity [32] was introduced by Latanya Sweeney for publishing of secret data. The sensitive information in a dataset is obfuscated by replacing them with a more general information.

Related Patterns

MORPHED REPRESENTATION is used with ANONYMITY SET to hide the correlation between incoming and outgoing traffics.

Morphed Representation

Sender Anonymity

Intent

Change the representation of the data when it is passing through an anonymity providing node so that outgoing data cannot be linked with incoming data.

Also Known As

Werewolf, Gate of Heaven, Dr. Jekyll and Mr. Hyde, Amoeboid Shape.

Motivation

The unsuccessful mix. Alice, Bob and Carol are using a mix based system to communicate over the Internet. Alice sends her data through a node in the network where it gets mixed with the data coming from other sources (*e.g.* Bob, Carol, *etc.*). Figure 4 shows that Mallory is a passive observer of the mix network and she wants to find out all the correspondences of Alice.

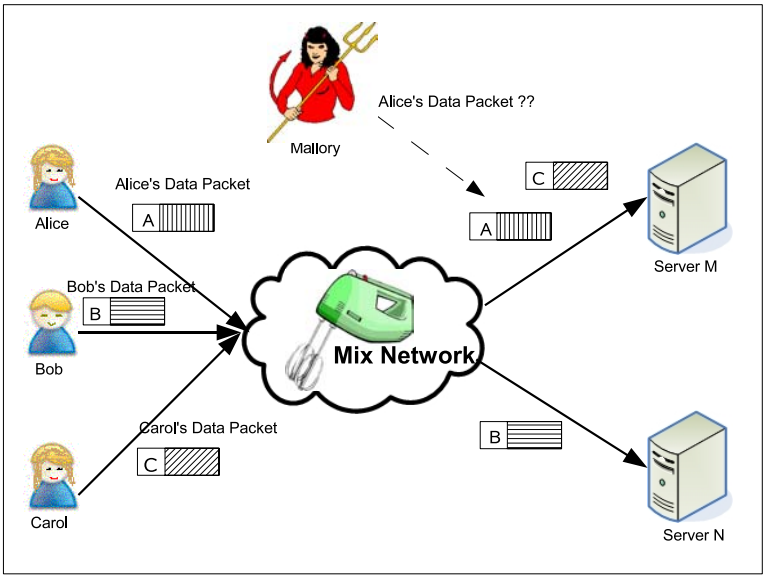

Figure 4: The unsuccessful Mix network

Alice encrypts her data with the public key of the recipient (in this case server N) to keep it confidential. The mix network receives Alice's packet, waits for other packets to arrive and then releases a bunch of packets together. However, the incoming and outgoing packets have the same data fields, hence it is easy for Mallory to find out who is sending which packet. Mallory can profile everyone's messaging habit very easily.

Context

You are designing a mix based system to protect the privacy of the users. You want to have sender anonymity and sender and receiver unlinkability for the communicating parties. The mix based system can be a mix based filter in the Internet messaging domain that is used for email messaging or web browsing.

Problem
Mix networks combine the data from a sender with the data coming from multiple other sources and send them together. The incoming data packets carry the data content. They also have meta-characteristics associated like the time of packet generation, ingress order of packets *etc.* The mix network obfuscates these meta-characteristics by adding random delays to the ingress packets, or by batching a number of ingress packets before releasing the packets. However, if the mix concepts do not obfuscate the data content, the incoming and outgoing data packets have the same representation and they can be correlated trivially. This compromises sender anonymity as the packets can be linked to the packet generator if an adversary has enough capability to trace the packet path back to the sender. Also the packets can be traced to the recipient and sender and receiver unlinkability is compromised.

How can the representation of the data be obfuscated?

Forces
This pattern addresses the following forces.

1. *Packet Characteristics.* The size and content of the data packet separates one packet from another. It is highly improbable that the size and content of two packets would be the same because timestamps are associated with packets. Even if the data inside the packet is protected by encryption, the encrypted content and size of packets reveal the correlation of the outgoing packets with the incoming packets.

2. *Scalability.* The Mix networks should be scalable. A PKI infrastructure should be established between the participating nodes (*i.e.* mix nodes and end nodes) such that symmetric encryption keys can be exchanged during data transfer.

3. *Confidentiality of Data.* Data flow in the network is encrypted to retain confidentiality of content.

4. *Data Corruption.* The mix nodes should not change the data content, only change the representation of the data content to achieve unlinkability.

5. *Type of Adversary.* A global, passive adversary monitors the data traffic in the network and does not manipulate the data content. Active adversaries may control the mix nodes. Mix nodes can also be controlled by passive (or semi-honest) adversaries that adhere to the mix protocol but only monitor the data content passing through the node.

6. *Performance.* The privacy retaining operations should not become a performance bottleneck. For anonymous messaging, the system should be usable in low latency usage scenarios, like web browsing.

Solution
Change the representation of the incoming data packet such that the outgoing packets look different from incoming packets. The incoming packets are encrypted using a key shared between the sending node and the mix node. At the mix node, decrypt the packet and then re-encrypt it with the key shared between the mix node and the subsequent node.

Figure 5 shows the message transfer between Alice and Server N. Alice encrypts her packet with a shared key between her and the mix node. The mix node decrypts the packet and then re-encrypts it with the shared key between the mix and server N. Mallory is monitoring the network and she cannot identify the packets because

the packet do not look the same.

Use symmetric keys for encryption to avoid the expensive primary key operations. The nodes set up a key share with all its neighbors during the setup phase.

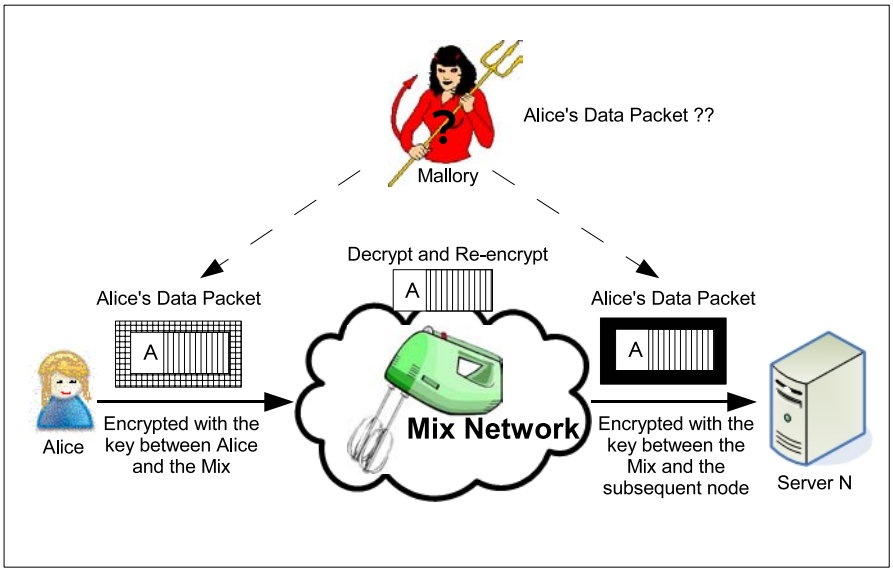

Figure 5: Data packet morphing at a Mix node

Design Issues

Key Sharing. The nodes in the network have to establish a symmetric key share with their neighbors. This symmetric key share can be established using public key certificates. However, the deployment of a global PKI infrastructure is an additional overhead for the scheme to be successful. To avoid the use of public key based key share establishment, lightweight secret sharing schemes like Diffie-Hellman key exchange [10] can be used.

End-to-end encryption. Since the packets are decrypted and re-encrypted in the mix nodes, confidentiality might be compromised if the data is in plaintext after the decryption. In that case even a semi-honest adversary running the mix node can compromise the privacy of sender and recipient and the confidentiality of the data. To avoid this, data has to be encrypted end-to-end. The sender encrypts the plaintext content with the public key of the ultimate recipient and then uses the symmetric key share to route it through the intermediaries.

Size of Neighbor Set. The scheme depends on all the nodes keeping a symmetric key share with their neighbors. A large list of neighbors (*i.e.* a large anonymity set) would ensure better anonymity because the node has many options to choose from for the next hop. However, a large list would add maintenance overhead of key share tables.

Consequences

The pattern has the following benefits.

1. *Privacy.* The sender enjoys improved privacy because the representation of the data changes at every intermediate node. A single adversary can break the anonymity if he can observe the network globally which

is fundamentally infeasible. The mechanism is also safe from colluding adversaries unless they are distributed globally and control the whole network. As long as there is one honest mix, it will obfuscate the correlation between input and output data traffic.

The pattern has the following liabilities.

1. *Performance Overhead.* Performance overhead comes from two things - overhead of creating symmetric key shares and overhead of cryptographic operations at each mix node. The key sharing is often done beforehand to avoid the overhead during data transfer. There are several trade-offs to consider to determine the lifetime of the symmetric key share. If the symmetric keys are used for a long time then the system becomes vulnerable to brute force attack on the key. If the keys have a short lifetime then the key setup overhead would be considerably high. Public keys can be durable, but they would involve a high computational overhead.

2. *Denial of Service.* The active adversaries controlling the mix can drop the packets and create a denial of service scenario. Without the presence of a network management component, it would be very difficult to find the misbehaving node.

Known Uses
The Mix based networks [5] are based on the idea of mixing the incoming data traffic from one user with the data traffic coming from other users. To hide the correlation, the incoming data in the mix network is decrypted and then re-encrypted so that the egress traffic and the ingress traffic cannot be matched. Remailers based on the mix network principle, *e.g.* the Cypherpunks mailing list [7], Mixmaster [8], Babel [16], Mixminion [9] *etc.*, follow this pattern to hide the correlation between incoming and outgoing packets.

Onion Routing [15, 33] systems are based on the concept of mixes but they have better latency values than mix networks and are therefore more applicable in a web browsing scenario. Private web browsing systems for peer-to-peer communication that provide anonymity following this pattern include Morphmix [27], Tarzan [12] *etc.*

Related Patterns
MORPHED REPRESENTATION is used with ANONYMITY SET to hide the correlation between incoming and outgoing traffic. Sometimes the data traffic is encrypted with LAYERED ENCRYPTION so that MORPHED REPRESENTATION does not compromise data confidentiality.

Hidden Metadata

Sender Anonymity

Intent
Hide the meta information associated with data content that reveal information about sensitive data content.

Also Known As
Header Manipulation, Anonymization Proxy, Anonymization Gateway, Blurred Identity, Pseudonym Hopping.

Motivation
Exposure. Alice is suffering from a medical condition and she wants to find some information about it. She visits the website `www.dishonest-medical-website.org` that is controlled by Mallory (figure 6). While Alice is visiting the website, Mallory secretly gathers Alice's email address, geographical location, computer type, operating system, web browser, previous web site visited *etc.* Mallory is gathering these information about Alice even if Alice does not accept cookies, which is primarily used for profiling browser behavior. Mallory analyzes the HTTP_USER_AGENT, REMOTE_HOST, and HTTP_REFERER variables, which almost all web browsers provide to each site visited as part of the HTTP protocol.

HTTP_USER_AGENT reveals the user's browser software, which the remote web site could use to generate web pages specifically tailored to the browser's capabilities. However, both Netscape and IE also includes the user's computer type and operating system as part of this variable.

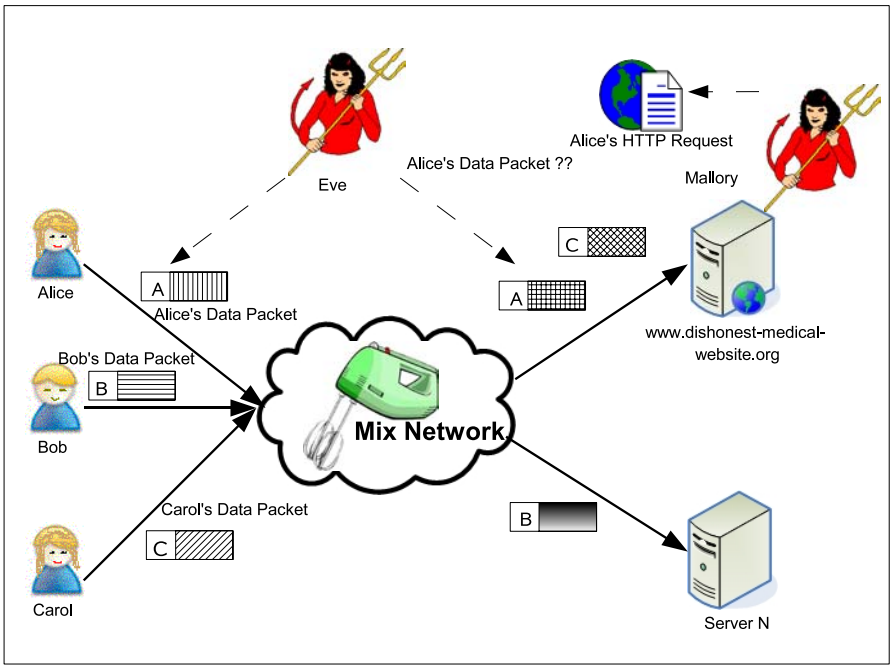

Figure 6: Compromised Anonymity by Header Matching

The REMOTE_HOST variable reveals the Internet address of the computer making the request for a web

page. Let us assume that Alice is running a single-user workstation. The computer's identity may be the key to an enormous source of personal information. Using the Unix 'finger' command, Mallory can identify the Alice's full name, email address, and even phone numbers. Even if Alice is accessing the web via a large commercial provider such as AOL, or from behind a corporate firewall, the REMOTE_HOST field reveals that the user is an AOL member or an employee of that particular company. People who access the web via local Internet service providers (ISPs) reveal the identity of that ISP, which in turn reveals their geographic location. Mallory then performs a 'whois' lookup from the InterNIC database to find and report the physical address associated with the user's Internet host.

The HTTP_REFERER variable reveals the previous page visited by Alice. All of these informations are provided without Alice's consent or knowledge and thus is a threat to Alice's privacy. Mallory also combines the data with another publicly-accessible database such as a phone directory, marketing data, voter registration list, *etc.* She gains a significant amount of personal data on every visitor to her pages. All of this information-gathering is accomplished without Alice's authorization or awareness.

Other than the application layer protocol attacks, the data packets carrying the request from Alice to the end host are also vulnerable to inspection by a global passive eavesdropper Eve. These packets contain the IP addresses for routing. Figure 6 shows that Alice is sending the packets through a Mix network along with other users Bob and Carol. The input data packets to the mix are encrypted. The mix decrypts the data, re-encrypts it and sends them out after adopting a mix strategy. However, the IP addresses associated with the data packets are needed for routing. Eve does not have access to the data content and because of the mix network's mixing and morphing mechanism she cannot correlate the input and output packets based on data content. But with the use of the headers, she can easily identify which message is coming from Alice.

In and out. Alice and Bob are in a location tracking system where they want to anonymize the information of their whereabouts. The system has regions called the mix zone that work on the principle of mix network. Once multiple agents enter the region and then come out it should be impossible to identify the agents. The system has a handle for all the agents. The handle acts as a pseudonym for the agent. Suppose Alice has the pseudonym *agent12345* and Bob has the pseudonym *agent12346*. Now *agent12345* and *agent12346* enter the mix zone and come out. However, they still retain their pseudonym and therefore are trivially identified.

Context
You are designing a system to protect the privacy of the users. This system will maintain sender anonymity in a messaging scenario. Although this is an important application area, the context is not limited to messaging only. The context also entails other scenarios like anonymity in a location tracking system or anonymity preserving data sharing in a data publishing system.

Problem
Any metadata associated with the data traffic in the network reveals information about the originator of that data packet. The packet headers are used for information routing in forward and reverse direction. The information in the packet headers like IP addresses reveal private information about sender's identity and their location.

In the spatial mix zone based location anonymity system, the agents are identified by their pseudonyms. If the pseudonym of an agent entering the mix zone and the agent exiting the mix zone remains the same, a location monitoring system can successfully correlate the outgoing agent with the incoming agent.

When private databases are shared for public use, obfuscation mechanisms like k-Anonymity [32] are applied on the database to create anonymity set. This anonymity set is compromised if information can be revealed by using meta-information associated with the data. Fow example, the sort order of a table in the database can be used to compromise private information. If there is a patient in a medical database whose last name starts with Z, and the patient database is sorted by last name then that person's information should appear in the last portion of the database. If the sort order is retained, the data obfuscation may be unsuccessful.

How can the meta-data associated with the data content be hidden?

Forces
The forces that need to be considered when choosing to use this pattern are as follows.

1. *Anonymity Service.* Users might be sensitive about some of their Internet browsing behavior and do not want these behaviors to be associated with their profile. For some other browsing activities, users might be apathetic to the fact that they are being profiled. In some cases users might be willing to be profiled over a long time so that they can get customized service experience.

2. *Routing.* Packet headers are used for routing. Stripping off packet headers would make it difficult to route the request. Also the response from the recipient has to be sent back to the request originator. If the header is irretrievably tampered, then the response cannot be routed back to the originator. Encrypting packet headers do not work either because, the intermediate mixes have to access the packet headers for routing. If a mix is controlled by an adversary, the plaintext header information at that node would reveal sender identity.

3. *Performance.* The system should not have complex operations that add to the latency in a messaging system. The system should be applicable in a low-latency messaging domain like web browsing.

4. *Type of Adversary.* A global, passive adversary monitors the data traffic in the network and do not manipulate the data content. Active adversaries may control the mix nodes and manipulate the through traffic. Mix nodes can also be controlled by passive (or semi-honest) adversaries that adhere to the mix protocol but only monitor the data content passing through the node. The adversary has access to other third-party data sources that he can use to infer information.

5. *User Consent.* Most of the web browsing information are gathered without users' consent. The technology underlying web browsing makes it possible for web sites to collect varying amounts of personal information about each user. Although it is important that the consumers have the right to be informed about the privacy and security consequences of an online transaction before entering into one, current technology does not provide any mechanism to enforce this user right.

6. *Payment for Privileged Service.* Users are sometimes willing to pay for privileged anonymity service for their sensitive data. The privileged service would involve lower latency in data transmission and better anonymity.

7. *Law Enforcement.* Anonymity can be abused by malicious users and to thwart that law enforcement agencies might require that the anonymity solutions sometimes lift their anonymity cover to investigate on crime suspects.

Solution
Obfuscate the metadata associated with the data. For the web browsing domain, create a middleman between the request sender and the recipient that strips off identity-revealing meta-data from the packet headers. The sender submits the request to the middleman that acts as a proxy. It submits the request to the recipient on behalf of the sender, but removes the values from the tags like HTTP_USER_AGENT and HTTP_REFERRER from the header. For email messaging, use a remailer that strips identifying header information from outbound email messages. Hide the metadata that do not hamper message routing or the service provided by the end server upon receiving the message.

For the location anonymity domain, strip off the pseudonym associated with an agent when he enters the mix zone. Assign a different pseudonym to the agent when he comes out of the mix zone.

For privacy preserving data sharing, scramble the sorting order of the data in the table. Remove any other meta-information in the data that can compromise the privacy.

Design Issues
Storage of State Information. The anonymizer should keep track of the changes it has made to the sender's request header. This is needed to forward the response back to the sender. This can be stored in a table-based storage with hashed key, or a database if the anonymizer controls large amount of traffic. The anonymizer proxy should also keep the states to prevent replay attacks.

Traffic Analysis of Anonymyzing Proxy. The ingress and egress traffic of anonymizing proxy can be monitored, and privacy can be compromised if the anonymity set is small. The anonymizer can adopt mix technologies to prevent against these attacks. Also, the sender might submit traffic to the anonymizer through a mix network or onion routing portal to achieve stronger anonymity guarantee.

Trust Relationship with the Anonymizer. The users should establish a trust relationship with the anonymizing proxy. A malicious proxy could in principle track its users' browsing patterns and make unscrupulous use of that information. The trust establishment can be achieved with a legal contract, or can be done dynamically using explicit trust negotiation protocols.

Bandwidth Requirement. The anonymizing proxy has to handle hundreds of thousands of page requests. For each request, the anonymizing proxy has to fetch, process, and forward a web page from elsewhere on the net. A subscription mechanism can be introduced to create users of various privilege levels and the requests can be prioritized based on that.

Direct Sender-recipient Link. Many ActiveX controls require direct linkage between the sender and the recipient. For example, RealAudio goes around the proxy by establishing their own direct net connections. Recent Ajaxian applications also require direct connection between the browser and the server. The link-rewriting mechanism of anonymizer proxy cannot provide anonymity for browsing these pages with active controls.

Consequences
The pattern has the following benefits.

1. *Privacy.* The anonymizing proxy provides an alternative for privacy. It does not depend on costly cryptographic operations like a mix network. Also mix networks require wide-scale deployment, an issue that is

not relevant to the architecture of anonymizing proxy. The anonymity service is offered transparently by the anonymization proxy, and the users do not have to modify their normal behavior to use the service.

2. *Business Incentive for running an Anonymizer.* Business venture can be established to provide anonymity services like anonymizer, because it does not rely on wide-spread deployment of services. In a mix network, the mix node has to communicate with other nodes beyond the organizational boundary. The anonymizer can be deployed independently and the success of the it depends on the reputation of the authority running the anonymity service. Moreover, the anonymizer can provide privileged service based on subscription. A client using free service would experience higher latency than the paid service.

The pattern has the following liabilities.

1. *Performance Overhead.* Heavy traffic can throttle beyond the bandwidth limit of the anonymizer creating a DoS scenario. The storage and maintenance of meta-information of anonymized packets and packet processing cost has severe performance overhead.

2. *Single Point of Failure.* The anonymizing proxy is a single point of failure. For a mix network, a passive adversary has to monitor different parts of the network. On the other hand, for an anonymizing proxy monitoring the ingress and egress paths is sufficient for the attacker.

3. *Forced Compromise of Privacy.* An anonymizing proxy may keep track of the obfuscations it is making on incoming data to generate outgoing data traffic. This is especially necessary because the response traffic has to be routed in the reverse direction. Maintainer of an anonymizing proxy can be forced by law enforcement authorities to divulge this information, thereby undermining the anonymity of the proxy users. This can also lead to extortion from influential organizations. One of the first remailers built on this concept (the Penet remailer, developed in 1993) came under attack several times from different organizations. In 1995, the Church of Scientology filed a lawsuit against Johan Helsingius (the creator and maintainer of the Penet remailer) to disclose the identity of an anonymous user, who posted a stolen file anonymously in the `alt.religion.scientology` newsgroup. The file was stolen from the Church's internal server. The Church's initial claim was to reveal the identity of all the users of the remailer (about 300,000 in that time), but in the end they settled with the disclosure of the person responsible for the post. The identity of the anonymous user, who was posting under the pseudonym "-AB-" and the anonymous ID `an144108@anon.penet.fi`, was revealed to be Tom Rummelhart, a system administrator of the Church of Scientology's INCOMM computer system.

Johan Helsingius was also contacted by the government of Singapore as part of an effort to discover who was posting messages criticizing the government in the newsgroup `soc.culture.singapore`. This time Johan did not have to compromise the identity of the user because the Finnish law did not rule the posting as a crime.

Then in September 1996, Church of Scientology sued Grady Ward [22] under the suspicion that he posted secret files under the posting title "Scamizdat" in the Penet remailer and forced Johan Helsingius to disclose the identity of two users, `an498608@anon.penet.fi` and `an545430@anon.penet.fi`, posting under the handle "DarkDemonStalker". Johan decided to close the remailer in September, 1996. The stories of these attacks on the Penet remailer has been written in many newspaper and online articles [25, 23, 11, 17]. Ironically, the Church extorted the information of the anonymous post, but it turned out to be anonymized by another anonymous remailer, the `alpha.c2.org` nymserver. `alpha.c2.org` was a more advanced and

more secure remailer that obfuscated the mapping of the input and the output, and hence the Church could not get the conviction they were after.

Known Uses
The Penet remailer (anon.penet.fi) [18] was a pseudonymous remailer operated by Johan Helsingius of Finland from 1993 to 1996. The concept of this remailer is to provide a portal that stores pseudonyms for users. The users send messages hiding behind the pseudonym. By stripping the user's name and assigning a pseudonym, the system provides sender anonymity through pseudonymity. Moreover, recipient anonymity can be achieved if the recipient of the mail is also a user behind a pseudonym. Because the users always use one pseudonym, it has the advantage that the users can create a reputation by using the pseudonym for a long time. But repeated use of the pseudonym means that privacy can be weakened by long term salvage of context information from a user's correspondence.

The Anonymizer [1] provides a technological means for preserving a user's privacy when surfing the web. A third-party web site (`http://www.anonymizer.com`) is set up to act as a middleman between the sender and the recipient. When the client wants to visit a web site, say the Google web site (`www.google.com`), he does not send the request directly to the Google server. Instead it directs the request through the anonymizer proxy by using the URL `http://www.anonymizer.com:8080/www.google.com`. The Anonymizer then connects to google.com without revealing any information about the user who requested the information, and forwards the information received from Google to the user.

The first version of the Anonymizer was based on the public-domain CERN proxy server, but with several modifications to preserve anonymity:

- It does not forward the source IP address of the end-user.
- It eliminates revealing information about the user's machine configuration from the "User-Agent" MIME header, user's name from the "From" MIME header, and previously visited site name from the "Referer" MIME header.
- It does not forward the user's email address to serve as a password for FTP transactions.
- It filters out Java applets and JavaScript scripts which may compromise anonymity.
- It filters out all "magic cookies" which may compromise anonymity.
- It gives positive feedback to the user by displaying an Anonymizer header on the page and adding the word "[Anonymized]" to the page's title.

The Anonymizer provides an easy-to-use interface which allows users to bypass the configuration procedure normally associated with using a proxy. Users access the service simply with extended URLs, such as `http://www.anonymizer.com:8080/www.google.com/`. The interface is flexible and the users can freely switch to their regular browsing behavior when anonymity is not required.

There are various other anonymity providing services that are built on the principle of anonymizer, *e.g.* iProxy [19] and the Lucent Personalized Web Assistant (LPWA) [2]. LPWA does not offer the Anonymizer's page-rewriting mechanism which enables users to easily change between anonymized and non-anonymized browsing. However, it does provide an additional feature, support for anonymous authentication and registration at web sites which provide personalized services.

Mix zone [3, 4] is the same concept of mix networks taken into the domain of location anonymity. Agents are identified by pseudonyms in the mix zone. When the pseudonymous agent enters into the mix zone, his pseudonym is changed, so that once he comes out of the zone, his identity cannot be correlated with that of the entering agent. The mix zone concept was added with the Active Bat [35] system to provide location anonymity.

The principle of k-Anonymity [32] was introduced by Latanya Sweeney for publishing of secret data. The sensitive information in a dataset is obfuscated by replacing them with a more general information. The sort order is also altered to obfuscate meta-information.

Layered Encryption

Sender Anonymity

Intent

Use a sender-initiated packet routing scheme and encrypt the data packets in multiple layers so that the intermediaries only have access to a particular layer and use that information to route the packet to the next hop.

Also Known As

Onion Routing.

Motivation

Ghost in the Machine. Alice, Bob and Carol are using a mix based system to communicate over the Internet. Alice sends her data through a node in the network where it gets mixed with the data coming from other sources (*e.g.* Bob, Carol, *etc.*). Figure 7 shows that Mallory is an active semi-honest adversary who controls the mix node, *i.e.* Mallory obeys the mix protocol to appear as an honest mix, but she tries to learn information by looking at the packets that are routed through the mix.

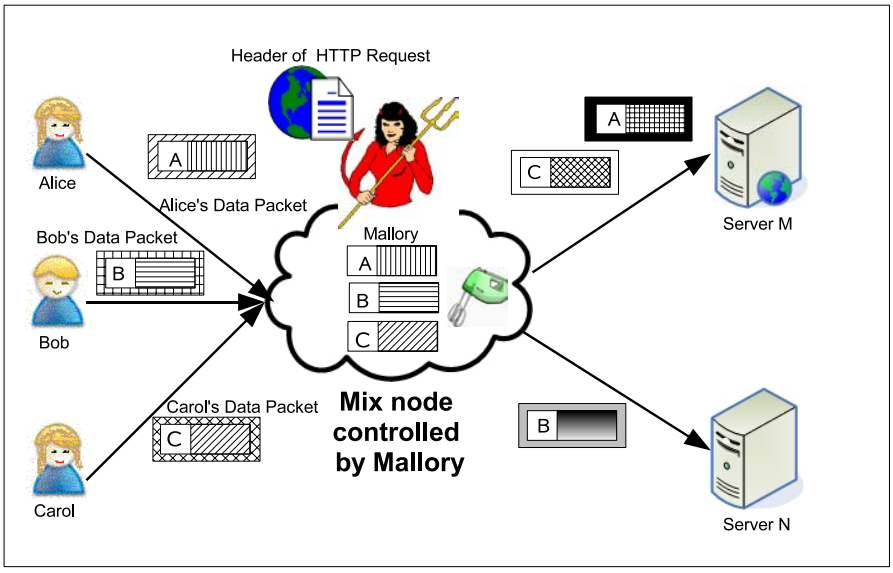

Figure 7: An active attacker controlling a mix node

The body of Alice's packet is encrypted with a symmetric key between Alice and the recipient, so that the intermediaries cannot access the content. According to the mix protocol, when Alice's packet is in transit between Alice and the mix node, it is encrypted with a shared symmetric key between her and the mix. Passive observers monitoring the link between Alice and the mix node cannot access the header of the packet. The mix network decrypts the packet, reads the header, finds the next hop and routes the packet to the next hop after encrypting it with a shared key between the mix node and the next hop. Again passive adversaries monitoring the egress packets of the mix node cannot access the packet header. Moreover, an adversary monitoring the ingress and egress links of the mix network cannot correlate the incoming and outgoing packets because of the mix protocol.

The problem arises because the mix node is controlled by an active adversary Mallory. Mallory shares the encryption key with Alice and the next hop, and accesses Alice's packet header to determine the routing option. This compromises the sender anonymity of Alice. Also from the header of Alice's packet, Mallory can determine the ultimate recipient, and therefore can compromise sender and recipient unlinkability.

Context
You are designing a mix based system to protect the privacy of the users. You want to have sender anonymity and sender and receiver unlinkability for the communicating parties. The system can be a mix based filter in the Internet that is used for email messaging or web browsing.

Problem
In the mix protocol, the mix nodes share symmetric keys between themselves. The mix decrypts and then re-encrypts the packets flowing through the node. This protects against a passive adversary observing the network traffic, but is insufficient against an active adversary controlling a mix node.

The mix node accesses the packet headers in order to identify the next hop. The header contains the ultimate destination, and the choice of next hop is determined by that. A malicious attacker controlling the mix node can follow the mix protocol, and yet profile the behavior of a message sender, because of the header in plaintext available to him.

How can the mix network be made secure against an active adversary?

Forces
The forces that need to be considered when choosing to use this pattern are as follows.

1. *Type of Adversary.* Privacy can be compromised by different types of adversaries. A passive adversary only observes the network traffic, but does not manipulate the data packets. An active adversary manipulates the data packets or compromises and controls a mix node. After controlling the mix network, an adversary can act semi-honestly, *i.e.* he continues to act like an honest node by following the mix protocol but at the same time tries to get information from the packets flowing through the mix node. Adversaries can also collude to undermine the anonymity of the message sender.

2. *Routing Mechanism.* The packet header contains information that is essential for the routing decision. A distributed routing mechanism would delegate this decision to the intermediary nodes. Contrarily, in a centralized routing mechanism, the sender determines the route that the packet will take and adds that information in the packet header.

3. *Cost of Encryption.* The cost of decryption and encryption can become an overhead. The mix network should be usable for a low latency messaging requirement like web browsing.

4. *Key Establishment.* The network follows protocols for dynamic negotiation and establishment of keys. A symmetric key share can be established by using a PKI scheme, but it assumes the presence of a global PKI framework. Key sharing schemes with low infrastructure requirements can be used, *e.g.* Diffie-Hellman key exchange [10].

5. *Application Independence.* The mechanism for achieving sender anonymity should be application indepen-

dent. It should be applicable for low latency messaging domain like web browsing, and latency independent messaging domain like email messaging.

Solution
The sending client is responsible for establishing the path between the sender and the recipient. The neighboring nodes in the circuit share symmetric keys between themselves. The packet is then encrypted in multiple layers (like the onion skin). The innermost layer is encrypted with the symmetric key used in the last hop before the server, the next layer is encrypted with the symmetric key used in the preceding hop and so on.

Thus the sending client has to construct a chain of nodes, and when the message is in transit through these nodes, each node strips off a layer using its key share, finds the identity of the next hop within the decrypted bundle, and forwards it to that node. For example, for a remailer E_i, with R_i as its public key, A_i as its address, and B as the destination address, a three link route between Alice and Bob looks like

Alice –[E_1(A_2, E_2(A_3, E_3(B, M)))]– >
R_1 –[E_2(A_3, E_3(B, M)))]– >
R_2 –[E_3(B, M)]– >
R_3 –[M]– > Bob

Each remailer is able to decrypt the bundle it receives, but it cannot itself look more than one link ahead, let alone determine the final destination. Moreover, after the first link, the sender's identity has been removed. The first remailer R_1 is connected with the sender but when it receives the message, it has no way to determine whether its previous node is the sender or just another mix node in the remailer chain.

Design Issues
The design issues described in the MORPHED RESPRESENTATION pattern also applies here. Additional issues are as follows.

Service Composition. Layered encryption can be used in conjunction with other services. Layered encryption can be used for the path between a request sender and the anonymizing proxy (*e.g.* Anonymizer, LPWA *etc.*) and the proxy then submits the request on the sender's behalf.

Layered Encryption Overhead. The main overhead of layered encryption is the path setup cost. Typically, it is much less than one second, and it appears to be no more noticeable than other delays associated with normal web connection setup on the Internet. Computationally expensive public key operation is only used during the connection setup phase. By using dedicated hardware accelerators on the routers, the burden of public key operations can be relaxed.

Consequences
The pattern has the following benefits.

1. *Sender-determined Routing and Privacy.* In a distributed routing protocol, the intermediaries determine the path of the packet on its route, but for this the intermediaries need to have access of sender and recipient information. Sender and recipient anonymity is achieved by using this pattern because here the routing decision is taken by the sender only. The sender initiates a path setup protocol to create the route. This can be done in offline (*i.e.* when the system is idle) to reduce performance overhead.

2. *Application Independence.* Layered Encryption can be used with proxy-aware applications, as well as several non-proxy-aware applications. Layered encryption supports various protocols, *e.g.* HTTP, FTP, SMTP, rlogin, telnet, finger, whois and raw sockets. Proxies can be used with NNTP, Socks 5, DNS, NFS, IRC, HTTPS, SSH and Virtual Private Networks (VPN).

The pattern has the following liabilities.

1. *Data Integrity.* The layered encryption technology does not perform integrity checking on the data. Any node in the path of layered encryption can change the content of data cells. However, if the adversary controlling a mix node alters the data content of the packet, the subsequent node will figure out the discrepancy. This mix node sends the information about the mal-formed data packet in the backward path, and eventually the sender finds out about the integrity violation. The sender can then initiate a new path and send the message along that path.

2. *Path Setup Overhead.* The sender has to create the complete route from the sender to the recipient and this setup cost is significant. The layered encryption systems balance this by creating the paths offline.

Known Uses
Onion Routing [15, 33] systems are based on mixes but they leverage the idea of mixes and add layered encryption. Onion Routing systems have better latency values than mix networks and therefore are more applicable in a web browsing scenario. Private web browsing systems for peer-to-peer communication like Morphmix [27], Tarzan [12] *etc.* follow layered encryption mechanism.

Related Patterns
LAYERED ENCRYPTION follows MORPHED REPRESENTATION by performing cryptographic operations at each nodes in the path.

Acknowledgement
I was inspired to write this paper after taking a course with Professor Nikita Borisov at University of Illinois on privacy preserving systems. Professor Eduardo B. Fernandez, as the shepherd, helped me take the paper in the right direction. I thank Professor Ralph Johnson for his encouragement throughout the writing process. Finally, I thank Paul Adamczyk and Brian Foote for their helpful comments.

References

[1] Anonymizer.com. Online privacy services.

[2] Bell Labs Technology Demonstration. The Lucent personalized Web assistant.

[3] A. Beresford and F. Stajano. Mix zones: User privacy in location-aware services. In *PerCom Workshops*, pages 127–131, 2004.

[4] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[5] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[6] D. Chaum. E-voting: Secret-ballot receipts: True voter-verifiable elections. *IEEE Security and Privacy*, 2(1):38–47, Jan./Feb. 2004.

[7] Computer Cryptology. *APAS Anonymous remailer use*, 2 Dec 2001. `http://www.faqs.org/faqs/privacy/anon-server/faq/use/part3/`.

[8] L. Cotrell. Mixmaster & remailer attacks, 1995. `http://riot.eu.org/anon/doc/remailer-essay.html`.

[9] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a type III anonymous remailer protocol. In *Proceedings of the 2003 Symposium on Security and Privacy*, pages 2–15, Los Alamitos, CA, May 11–14 2003. IEEE Computer Society.

[10] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[11] EFF. Johan Helsingius gets injunction in Scientology case: Privacy protection of anonymous messages still unclear, 23 Sept 1996. Press Release. `http://www.eff.org/Privacy/Anonymity/960923_penet_injunction.announce`.

[12] M. J. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, DC, November 2002.

[13] I. Goldberg. Privacy-enhancing technologies for the Internet, II: Five Years Later. In *International Workshop on Privacy Enhancing Technologies (PET), LNCS*, volume 2, 2002.

[14] I. Goldberg, D. Wagner, and E. Brewer. Privacy-enhancing technologies for the Internet. In *Proc. of 42nd IEEE Spring COMPCON*. IEEE Computer Society Press, Feb. 1997.

[15] D. M. Goldschlag, M. G. Reed, and P. F. Syverson. Hiding routing information. In R. Anderson, editor, *Information hiding: first international workshop, Cambridge, U.K., May 30–June 1, 1996: proceedings*, volume 1174 of *ser-LNCS*, pages 137–150, pub-SV:adr, 1996. pub-SV.

[16] C. Gülcü and G. Tsudik. Mixing e-mail with BABEL. In *Proceedings of the Symposium on Network and Distributed Systems Security (NDSS '96)*, San Diego, California, Feb. 1996. Internet Society.

[17] J. Helsingius. Johan Helsingius closes his Internet remailer, 30 Aug 1996. `http://www.cyberpass.net/security/penet.press-release.html`.

[18] J. J. Helsingius. The `anon.penet.fi` anonymous server. help file, 1995.

[19] iProxy.net. iProxy anonymizer service. `http://iproxy.net/`.

[20] A. Meyerson and R. Williams. General k-anonymization is hard. Technical Report CMU-CS-03-113, CMU, 2003.

[21] C. A. Neff. A verifiable secret shuffle and its application to e-voting. In P. Samarati, editor, *Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS 2001)*, pages 116–125. ACM Press, November 2001.

[22] R. Newman. The Church of Scientology vs. Grady Ward, 24 Jul 1996. `http://www.xs4all.nl/~kspaink/cos/rnewman/grady/home.html`.

[23] R. Newman. The Church of Scientology vs. `anon.penet.fi` - Julf Helsingius voluntarily closes his remailer after Finnish court orders him to turn over a user name to Scientology, 23 Mar 1997. `http://www.xs4all.nl/~kspaink/cos/rnewman/anon/penet.html`.

[24] A. Pfitzmann and M. Waidner. Networks without user observability - Design options. In F. Pichler, editor, *Advances in cryptology: Proceedings of a Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT '85)*, volume 219 of *LNCS*, pages 245–253, Linz, Austria, Apr. 1985. Springer.

[25] D. G. Post. The first Internet war - The state of nature and the first Internet war: Scientology, its critics, anarchy, and law in Cyberspace. *Reason Magazine*, April 1996.

[26] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[27] M. Rennhard and B. Plattner. Introducing MorphMix: Peer-to-peer based anonymous Internet usage with collusion detection. In *WPES '02: Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*, pages 91–102, New York, NY, USA, 2002. ACM Press.

[28] M. Sadicoff, M. M. Larrando-Petrie, and E. B. Fernandez. Privacy aware network-client pattern. In Proceedings of the 12th Conference on Patterns Language of Programming (PLoP'05), 2005. `http://hillside.net/plop/2005/proceedings/PLoP2005_msadicoff0_0.pdf`.

[29] M. Schumacher. Security patterns and security standards - with selected security patterns for anonymity and privacy. In Proceedings of the European Conference on Patterns Language of Programming (EuroPLoP'02), 2002. `http://citeseer.ist.psu.edu/schumacher03security.html`.

[30] T. Schummer. The public privacy - patterns for filtering personal information in collaborative systems. In *CHI*, 2004.

[31] C. Shields and B. N. Levine. A protocol for anonymous communication over the Internet. In *CCS '00: Proceedings of the 7th ACM conference on Computer and communications security*, pages 33–42, New York, NY, USA, 2000. ACM Press.

[32] L. Sweeney. k-Anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[33] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. In *IEEE Symposium on Security and Privacy*, pages 44–54, Oakland, California, 4–7 May 1997.

[34] US Department of Homeland Health Services Office for Civil Rights. Summary of the HIPAA privacy rule, May 2003.

[35] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office, 1997.

[36] Wikipedia. Privacy — Wikipedia, the free encyclopedia, 2006.