Freedom System 2.0 Architecture

Philippe Boucher – philippe@zeroknowledge.com
Adam Shostack – adam@zeroknowledge.com
Ian Goldberg – ian@zeroknowledge.com
Zero-Knowledge Systems, Inc.

December 18, 2000

Abstract

This white paper, targeted at the technically savvy reader, offers a detailed look at the Freedom 2.0 System architecture. It is intended to give the reader a good understanding of the components that make up this system and the relationships between them, as well as to encourage analysis of the system.

Introduction

The Freedom product line is designed to be the most integrated, strongest and easiest-to-use privacy system available. This white paper gives the technical reader a good understanding of each component and of the system as a whole.

This paper focuses on the Freedom 2.0 system architecture. It will first list a set of key features of the system, followed by an overview of the network and a description of each major software component. It replaces the architectural information presented in the "Freedom Network 1.0 Architecture" document[1].

This paper concentrates on the Freedom System architecture, as it exists at the time of publication. It does not describe the protocols used by this system; these are discussed in the “Freedom 2.0 System Protocols”[2].

Key New Features

A number of key new features have been implemented in the 2.0 version of Freedom that set it apart from the previous version. These are:

A Single Network

All Freedom operations are done on a single network. No separate software or infrastructure is required to support different types of nyms and their security levels. There is a single nym namespace and a single domain name, freedom.net.

Kernel Space AIP

The AIP now operates inside the Linux kernel of a Freedom Server Node to route encrypted traffic. This reduces operational overhead, providing much improved performance over the previous version of the Freedom AIP. This allows us to increase the overall amount of data that can be transported through the Freedom System and the speed at which this traffic can be processed.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

Scalable Core Systems

The various servers that comprise the Freedom Core Systems have been made scalable. Multiple instances of each server type can now be run to support the needs of a greater number of Freedom clients.

A Clearly Defined Threat Model

A cleaner threat model[3] for the Freedom System has been defined. This allows the Freedom System design to better focus against attacks that we can defend against, and removes certain constraints that interfere with the Freedom System's network performance.

A New Mail System

The reply block mail system has been completely replaced by a POP box-based mail system hosted by Zero-Knowledge Systems, Inc. This dramatically increases the performance of the system. It can support a large number of users and provides them with simpler, more secure, and faster mail services.

Freedom System Overview

The Freedom Network is an overlay network that runs on top of the Internet. It uses layers of encryption to allow a Freedom user to engage in a wide variety of pseudonymous activities, hiding the user's real IP address, email address, and other identifying information from eavesdroppers and active attempts to violate the user's privacy.

Users are encouraged to create pseudonyms ("nyms") for each area of activity in which they want to preserve their privacy. The nyms that someone uses cannot be tied together. Thus, it is not possible to know if superman@freedom.net and clarkkent@freedom.net are the same or different people. Superman is happy with this situation because he doesn't want his super villain enemies to know about his life.

Similarly, when jobseeker@freedom.net browses a resume web site, his employer cannot see that Clark ("jobseeker") isn't happy with his job at the Daily Planet and wants to work elsewhere. Freedom protects Clark's privacy by proxying the various supported protocols, and sending those proxied packets through a private network before they are deposited on the Internet for normal service. That private network, as a system, is operated by Zero-Knowledge. Individual nodes in the network are operated by Zero-Knowledge or our partners, so that no single operator has comprehensive knowledge of what data is flowing through the network. Thus, the main components of the system are Freedom Clients and Freedom Servers. In the next section, we offer precise definitions of these and other entities.

Freedom Network Layout

The Freedom System consists of a set of Freedom Server Nodes that make up the Freedom Network, and the Freedom Core Servers that provide basic services. The network transports encrypted IP traffic from one node to the next. Nodes on the Freedom Network are called Anonymous Internet Proxies (AIPs). The number of nodes used in a route is chosen by the user by setting her security level in the Freedom client. The server nodes themselves are not linked by a fixed topology, instead, they can communicate with any other server node on the network, as requested by a client when creating a route. The Freedom client is given a network topology that identifies a set of reliable links between nodes in order to simplify route selection. This topology is defined solely on the basis of AIP-AIP performance characteristics.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

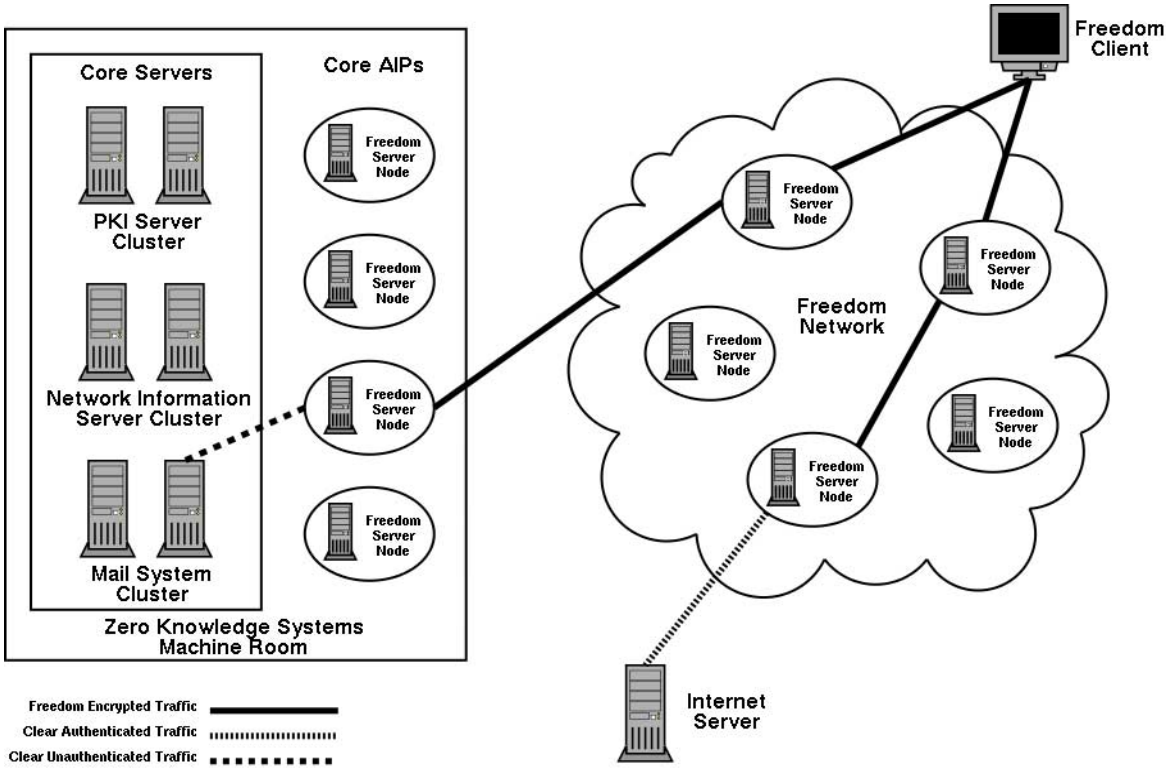

Figure 1: Freedom Network Layout

Authenticated routes to an exit AIP are granted through route creation requests signed by a nym. Upon receipt of such a request, an AIP will retrieve the requesting nym's public signature key and verify the request. Only nyms that have full access to the network can create such a route. Authenticated routes allow access to any host on the Internet.

Unauthenticated routes to a core AIP are granted to any Freedom client that requests one. These are limited in that only Freedom core servers can be accessed.

The core servers consist of PKI server clusters, reporting and network information server clusters, a token server, and the mail system. The core servers are accessed by creating completely anonymous routes that terminate at a core AIP. These are the only AIPs on the Freedom Network capable of accessing the core servers.

Freedom Network Server Nodes

A Freedom server node is a host running part of the Freedom Network. This node can be administered by Zero-Knowledge or by a third party server operator. A Freedom server node consists of an AIP, a local NIQS daemon client, some administration utilities and a set of key management utilities.

The node's operator generates the signature and encryption keys and submits the public components to Zero-Knowledge Systems for distribution to the rest of the network. Zero-Knowledge Systems never sees the node's private keys, ensuring that only that node is capable of decrypting its layer from the traffic that goes through it.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

The server node software needs to be able to communicate with the key query servers, NIQS, NISS, and other AIPs on the Freedom Network. It accepts connections from Freedom clients wanting to transport encrypted IP traffic on the Freedom Network.

A Freedom server node also runs a DNS server that is used to respond to DNS queries generated by a Freedom client. This is done through the same authenticated route as the client that issued the query, thus ensuring that any hostname lookup done by a Freedom user pseudonymous and private.

Freedom Core Systems

The Freedom Core Systems provide the base services that keep the Freedom Network running. This includes providing public keys, nym creation and update, network information and reporting, token creation, and the mail system. All of these services are currently provided by Zero-Knowledge and are located within our machine room. This new architecture introduces the concept of a core AIP that allows unauthenticated connections to any of the core servers in the network. The servers are no longer constrained by performance limitations or affected by the AIP running on the same host, as they were in the previous version of the system. Multiple instances of each core server type are also now supported. This allows the Freedom System core to load balance incoming requests across several machines as required.

Software Components

This section describes the Freedom System components that make up the architecture design, including the Freedom client, AIP, network information and reporting system, PKI servers, and the mail system.

Freedom Client

The Freedom client allows networking applications running on a computer to access the Internet through the Freedom System. It is designed to work seamlessly with these applications by trapping and redirecting the data streams that they generate. What follows is a simplified description of the client that applies to both the Windows and Linux versions. A more thorough description is beyond the needs and scope of this document.

The Freedom client consists of several components: a graphical user interface, a network access layer, a traffic filter, application filters, and a set of libraries that implement the functionality required for encryption, routing, nym management, route creation, etc.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

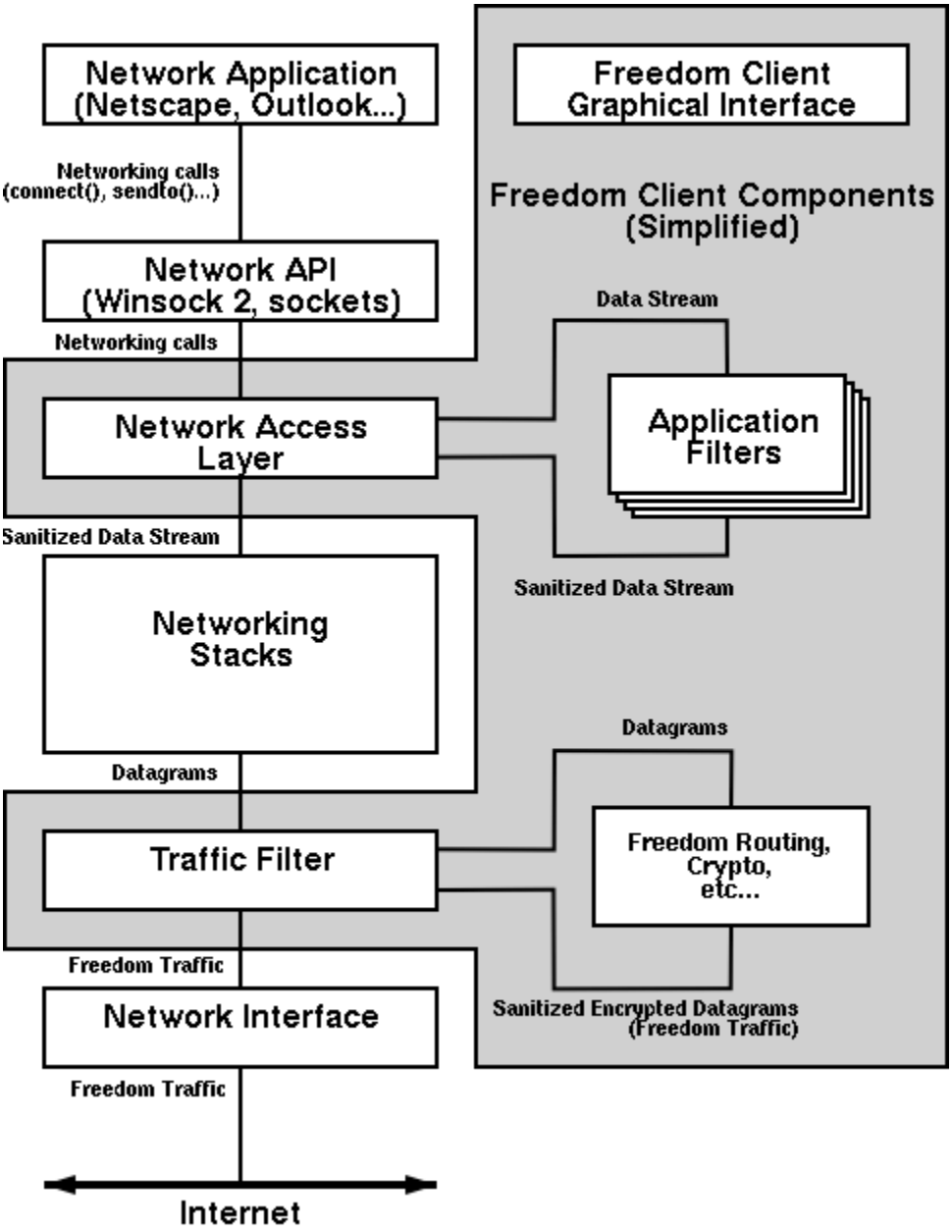

Figure 2: Freedom Client Internal Architecture

The interface allows a user to control the operation of the various components. It creates and destroys routes in the Freedom System, updates network information, manages pseudonymous identities and controls the application and traffic filters.

The network access layer is used to trap an application's networking calls such as connect() and sendto() and redirects them to the local application filters. This allows the application filter to see the original data stream generated by a network application and the response data stream generated by the destination host of the requested connection. After the local application filters have processed the data, the networking calls continue through the Freedom client host's networking stacks.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

The application filters are used to sanitize a network application's data stream. Many application protocols such as SMTP and HTTP will reveal information about a user; the filters edit or remove this information by parsing the stream according to the appropriate protocol and reconstructing a new sanitized stream. As responses arrive from a remote host back through the filter, the stream is again processed so that the local network application is not aware of the changes that have taken place. This allows the Freedom client to implement functionality such as the Cookie Manager and the Ad Manager without hindering the operation of the user's applications

After the data stream is returned from the application filter and the network access layer, it enters the client host's networking stack. This converts the data stream to IP, TCP, and UDP datagrams. Instead of allowing the host to place these datagrams directly on its network connection, the datagrams are picked up by the Freedom traffic filter. This filter sanitizes the datagrams by removing the source IP address, TCP and IP checksums, and zeroing out other fields. The datagram is then encrypted for the route that has been created for the active nym. The encrypted datagram is now sent to the first Freedom Server Node in the route. Returning traffic is processed in reverse. Datagrams are decrypted according to the route parameters, the headers are reconstructed and the traffic filter injects the datagram back into the client host networking stacks. These are sent through the application filters and back to the network application. This method allows the networking stack on the Freedom client host and the remote networking stack to negotiate a connection with each other without being aware that the Freedom System is transporting the traffic.

The Freedom Client currently supports the following protocols through its application filters:

- **HTTP**: All types of HTTP requests can be sanitized.
- **SMTP**: SMTP mail is trapped and sent to the Freedom Mail System.
- **POP**: The POP filter will connect to the user's regular POP server as well as the Freedom Mail System. It will automatically decrypt and authenticate Freedom messages.
- **SSL**: The SSL filter is only able to verify that the datastream follows the SSL protocol. It cannot decrypt it to verify the contents.
- **IRC**: Basic IRC is supported but DCC is not.
- **Telnet**: In order to speed up performance, telnet is run in line buffer mode in order to reduce the number of packets that must be transported. This filter is also able to support SSH.
- **NNTP**: NNTP is supported for post only. Usenet posts are redirected through the Freedom Mail System and sent to a commercial news server affiliated with Zero-Knowledge Systems, Inc. Currently, news must be read from the web.

The Freedom Mail System is closely linked to the SMTP and POP application filters. On top of the sanitization work done by these filters, they also encrypt, decrypt, reconstruct and scan messages, using the appropriate keys as required by the selected nym. This allows the Freedom client to seamlessly integrate a user's preferred mail application into the Freedom System.

Masquerading AIP

The Anonymous Internet Proxies (AIPs) are the core network privacy daemons that make up the Freedom Network. They pass encapsulated network packets between themselves until they reach an exit node. The masquerading AIP is incorporated into the Linux kernel in order to access the kernel's networking features. This allows the AIP to route large amounts of network traffic efficiently. A user-level daemon handles reliability-demanding operations such as key negotiations.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

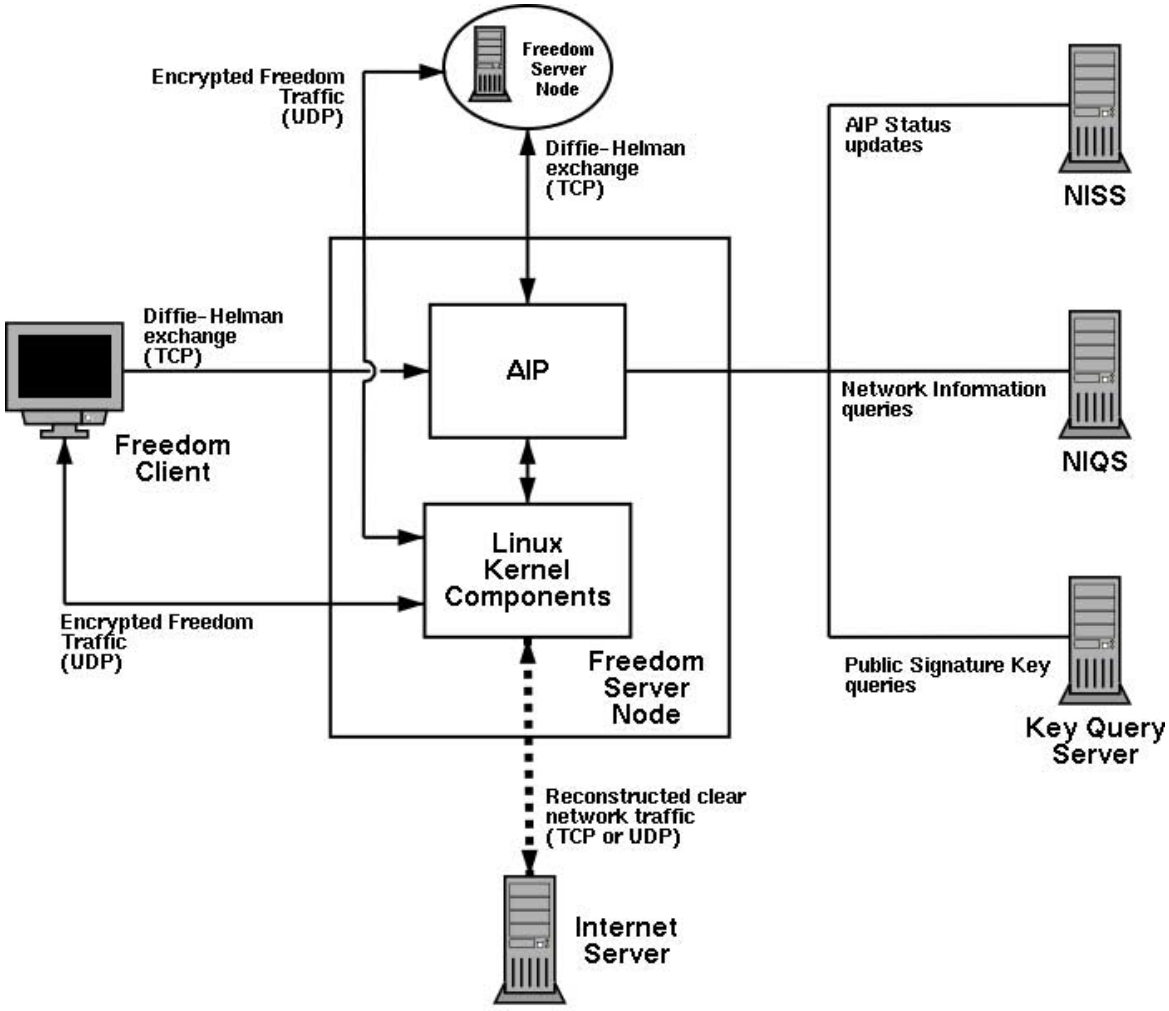

Figure 3: Freedom 2.0 Masquerading AIP

The Masquerading AIP for release 2.0 has a number of features that set it apart from the previous version of the Freedom AIP:

- The Masquerading AIP does not use fixed packet sizes--it allows variable packet sizes. This greatly lowers the bandwidth transport overhead since only relevant information is transported.
- The Masquerading AIP is able to transport some ICMP traffic through the Freedom Network. This allows entities communicating through the Freedom Network to have better control of their TCP streams.
- The Masquerading AIPs have no knowledge of the Freedom Network topology. Although a list of Freedom server nodes is provided to each AIP, the AIPs are free to connect to any other AIP on the network. Although the client is still provided with a topology, it is now only used by the client as a guide.
- The kernel's own networking stack is used to reconstruct traffic going out to the Internet. This provides a high level of performance and functionality
- Bandwidth limiting is available.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

- There is no cover traffic. In order to meet the current security threat model[3], cover traffic is not required. This improves performance by reducing the amount of "garbage" data on the network and reducing the "throw away" overhead that is required by an AIP to process a packet.
- An unauthenticated route allows a client to connect to a number of predetermined hosts. This allows the core AIPs to grant access to any of the core servers, thus allowing the Freedom core to better load balance incoming connections.

The Masquerading AIP obtains network information describing nodes and servers within the Freedom System from an NIQS server. It uses the key query server in order to lookup nym public signature keys that are used to sign authenticated route creation requests. The AIP will periodically report its status to a Freedom NISS server.

The capacity of the Freedom Network can be scaled up simply by adding new AIPs. Since there is no topology, each new AIP can be fully used to support new users.

Network Information and Reporting

The Freedom System's network information and reporting capability is made up of nodes that contain three different components. There is an NISS server to receive status updates from server entities in the Freedom System, a network information database (NIDB) to store the information, and an NIQS server that mines this database to answer network information queries for all entities in the Freedom System. As status updates arrive at the Freedom Core, they are replicated and a copy is sent to each network information node.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

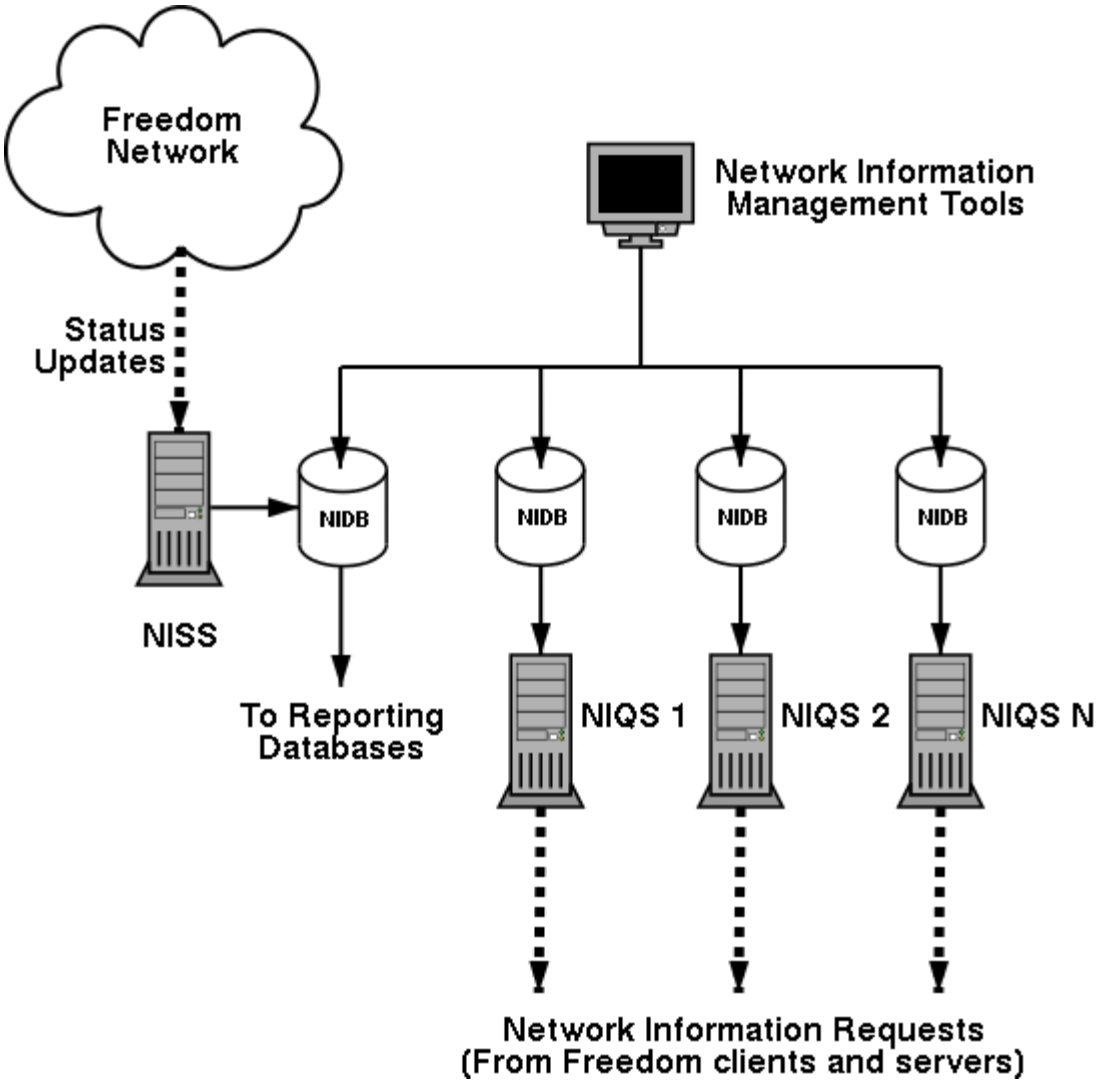

Figure 4: Network Information and Reporting System

The Freedom 2.0 NIQS server adds a number of features that significantly improve the speed of a Freedom client's network update and the overall number of Freedom clients that can simultaneously query Freedom Network information. These improvements include:

- Compressed network description. A new NIQS command returns the entire network description in a single response with a single signature. This response is compressed and fully downloaded only when it differs from the client's local version of the network representation. Although individual entity records can be queried, the Freedom 2.0 client only downloads this compressed information.
- The NIQS does not provide real time network status. The Freedom 2.0 client maintains its own statistics on network performance and availability. This allows it to provide a more meaningful view of the network to the user. As a result, the NIQS server no longer needs to provide this information. Instead, the NIQS only notifies Freedom clients when a Freedom server is marked as unavailable for system administration purposes. This reduces the amount of recomputation that must be done by the NISS and NIQS since Freedom Network entity records do not need to be rebuilt as a result of Freedom server status updates.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

- Reduced client update frequency. Because the Freedom client maintains its own statistics on the network, the client's update frequency for Freedom Network information is greatly reduced. It is currently set to once every 4 hours of operation.

The Freedom client does not receive real-time status information on the Freedom Network. Instead, it maintains its own statistics regarding the state of the network based on its accesses to it.

Freedom System Entities

Each distinct component type in the Freedom System is known as an entity. These entities can be software components, such as an AIP or a nym server, or a data object, such as a nym, or a special purpose keyset. The entities are organized into a hierarchy that is used to delegate authentication authority. This is further explained in the *Signature Key Hierarchy* section. The NIQS serves description information concerning all entities except ZkMaster, FrYear, FrMonth, and Nym.

Most entities have both a signature key pair and an encryption key pair. In some cases however, only one type of key pair is used by the entity.

ZkMaster	This entity is used by the Zero-Knowledge master key for the Freedom System. The public component of this key is hardwired into every piece of Freedom software.
FrYear	This is the Zero-Knowledge Freedom yearly key. Once per year, this key is used to sign all keys below it in the hierarchy.
FrMonth	This is the Zero-Knowledge Freedom monthly key. This key is used to sign all keys below it in the hierarchy.
TokSrv	This is the token server entity.
NymSrv	This is the nym server entity.
Niqs	This is the NIQS server entity.
Niss	This is the NISS server entity.
Nmta	The NMTA is the mail system server that processes mail sent from a nym and arriving at the Freedom System core.
Imep	The IMEP is the mail system server that processes mail sent from an Internet user to a nym and arriving at the Freedom System core.
Pop	Pop server entities can be accessed by nyms in order to download their mail.
KeyQry	This is the key query server entity.
Nym	This is the entity used for all nyms in the Freedom System regardless of the actual type of the nym.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

PKI Systems

The PKI servers consist of three types of server, a key query server, a nym server, and token server. A nym server is used to create and manage a Freedom client user's pseudonymous identities. A key query server is used to provide public keys on request to Freedom servers and clients. The token server provides nym creation tokens, which allow a user to anonymously create nyms without identifying their real identity.

The basic architecture is shown in the diagram below. The Key Query Server provides access to public signature and encryption keys for all entities on the Freedom Network. It is used by all entities in order to validate signatures and keys that they receive in the course of their activities.

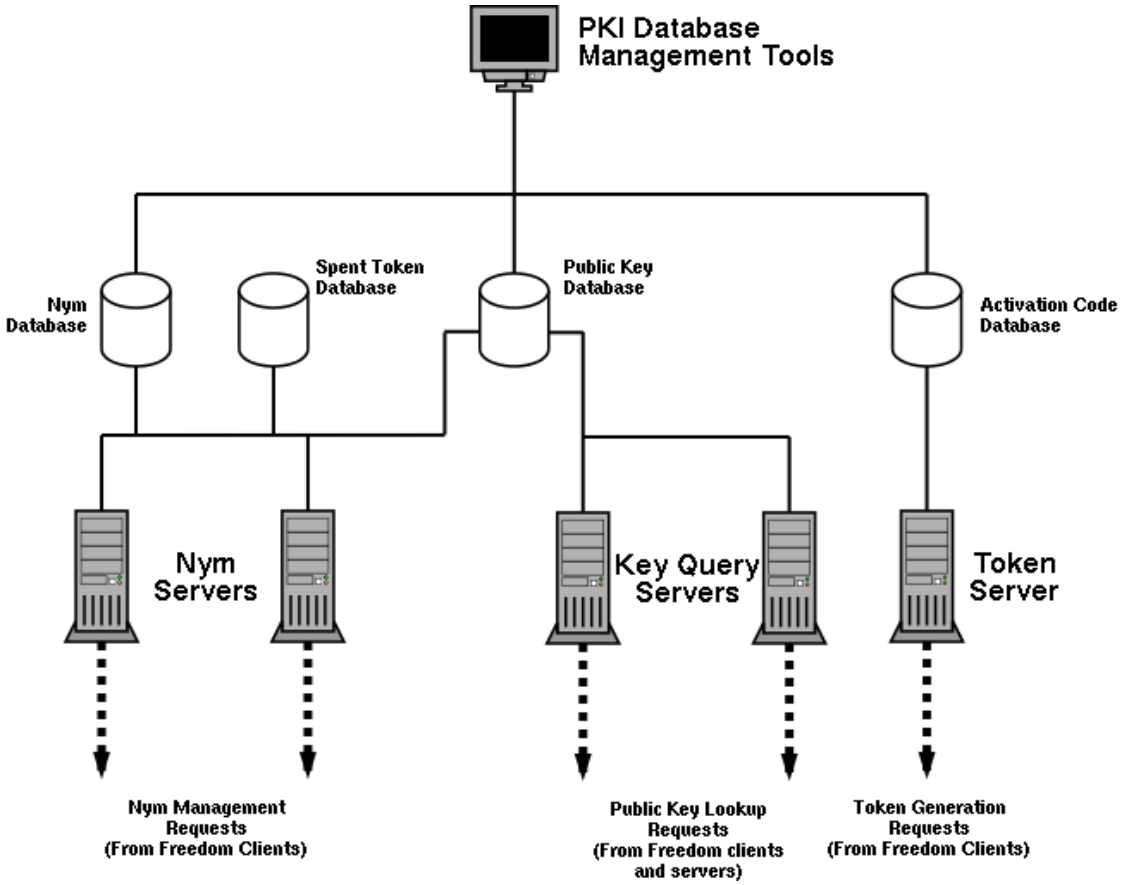

Figure 5: Freedom 2.0 PKI Subsystems

Signature Key Hierarchy

The key query server provides signed public keys in response to requests from various Freedom entities. These responses are signed following a hierarchy that ultimately allows a Freedom client to authenticate them.

There is a set of keys at the top of the hierarchy that are used only for signing. These keys include the Master, Yearly, and Monthly keys. The Monthly key signs the signature keys for all of the Freedom System's servers. There is a Master key, whose public parameters are encoded into all Freedom software for reference. The Master key is a 2048 bit DSA key, using the same prime parameters as are used in

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

PGP. The Master Key signs a Yearly key. The Yearly key is 1024 bit DSA; it is used to sign the Monthly key and client software updates.

Each Freedom server entity signs its own public encryption key using its private signature key. The NymSrv key also signs the Nym signature keys. The Monthly key, and the keys it signs, is stored online.

The nym server provides a way for clients to create and manage pseudonymous identities. It functions as a nym directory service for the Freedom System. The token server allows the Freedom client to redeem an activation code for a set of nym creation and upgrade tokens. This allows pseudonymous identity creation to occur without being able to track a nym back to the originally purchased activation code. For details on this process, please refer to the paper "Untraceable Nym Creation on the Freedom 2.0 Network"[4]. A nym can be revoked by deleting its keys from the public key databases. This makes a nym unable to use the system within a few hours as cached copies of its keys expire in the AIP key caches. These caches set a TTL of a few hours on each public nym signature key.

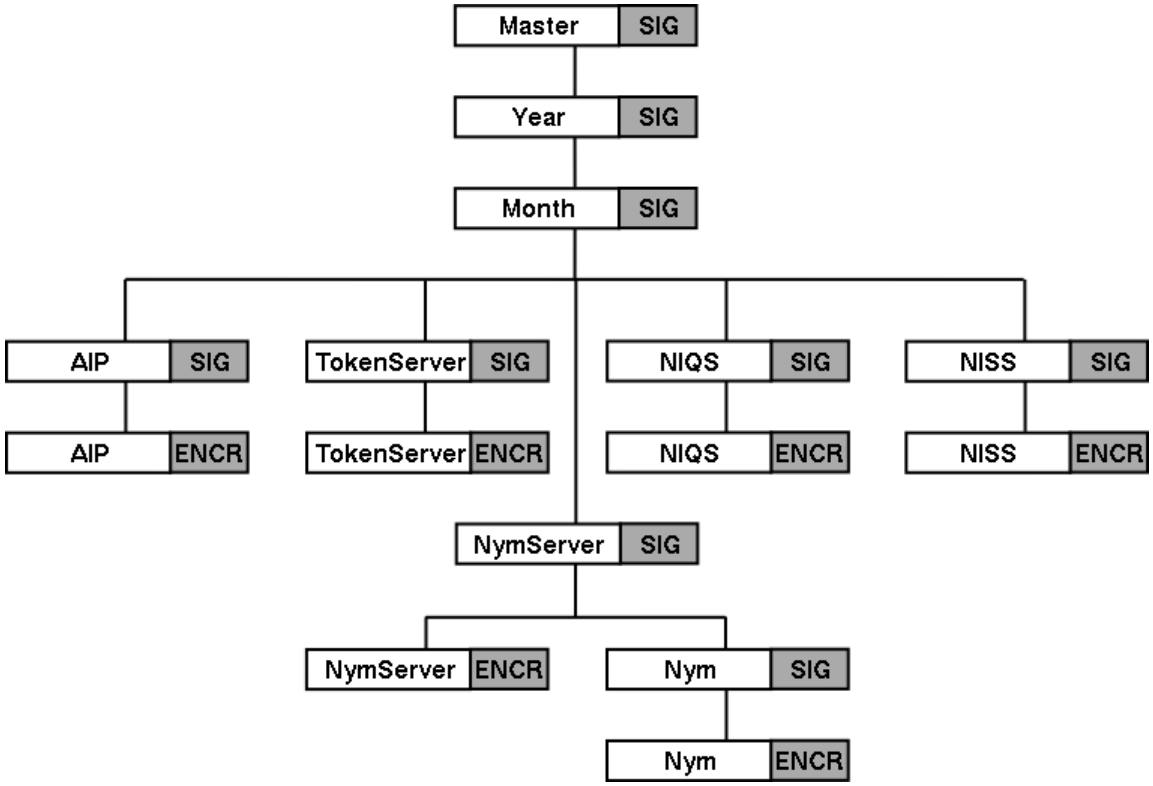

Figure 6: Key Hierarchy

The Freedom System supports multiple types of nyms, each with different capabilities. These nyms are identified by a type number that implies a certain set of capabilities common to all nyms of that type. These capabilities can be arbitrarily changed. The ability to upgrade from one nym type to another is currently disabled.

Premium These full access nyms are offered by the Freedom System. These currently have access to all of the system's capabilities, including the mail system, authenticated and unauthenticated route creation.

Standard Standard nyms are created in the Freedom System. However, they can do nothing other than nym management operations on the nym server and create unauthenticated routes to the Freedom System core. These nyms exist to allow

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

a user to reserve an entry in the nym namespace and to facilitate the use of certain Freedom client features such as cookie management, the form filler, ad manager, and the personal firewall. These nyms are not able to create authenticated routes on the Freedom Network.

The token server is used to redeem activation codes for nym creation and upgrade tokens. This transaction is done anonymously through the Freedom Network. This layer of indirection in the nym creation process ensures that a nym cannot be tracked back to its owner when it is created.

The PKI systems interact with most other components of the Freedom System. AIPs lookup public signature keys for nyms that sign route creation requests. The nym server creates mail access certificates that are returned to the client and are in turn passed to the mail system to create nym mail accounts. The Freedom client uses the nym server to create and modify new nyms. The PKI servers themselves obtain network information from the network information system's NIQS server and report their status to the NISS

Database Systems

The Freedom databases are accessed through the PKI and Network information servers. Any response generated by one of these servers will always be signed by the server's signature key.

Nym Database

The nym database holds all publicly available information that describes each nym. None of the stored information reveals the true identity of a nym owner. This database can be queried by anyone, but only the owner of a nym is allowed to modify a nym record. When a nym is created, that entry in the nym name space becomes permanent. If the owner of a nym decides to destroy their nym, or the nym expires, that nym name is not returned to the name space and it cannot be used by another Freedom user.

Public Keys

The public key database holds the public signature and encryption keys for all entities in the Freedom System, including nyms and servers. All keys are stored signed, according to the Freedom key hierarchy. When a public key lookup is made by a Freedom entity, it needs to lookup the keys used to sign the request key in order to verify the validity of the response. These extra keys are cached by the requesting entity in order to reduce extra key lookups.

Spent Tokens

The spent token database is used to prevent a nym creation or upgrade token from being reused. When an attempt to spend a token at a nym server is made, the database is searched for a hash of the token. If it is found, the token has already been used and is therefore refused. If it is not found, the hash is stored in the database and the token is accepted. The hash of the token is not linked to any other piece of information.

Activation Code Database

The activation code database is used by the token server to keep track of activation codes that can currently be redeemed by users. Each activation code has an associated capability code that is used to determine the type and number of tokens that are generated when the activation code is redeemed. There is no token database. These are generated and signed as required by the token server.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

Network Information Database (NIDB)

This database maintains information regarding all Freedom server entities. An entity record can be queried independently, or a complete compressed description of the entire network can be queried.

Reporting Database

This database is maintained by the NISS using the status updates that are sent to it from the various Freedom servers. It keeps track of things such as the uptime of each server, the number of bytes transferred, the number of messages processed, etc. This database is used to manage the Freedom System.

Mail System

The Freedom mail system handles all mail sent to and by nyms. It uses the Freedom Network to anonymously deliver mail messages and uses a POP box in the Freedom Core to store mail that is to be read by nyms.

The POP mail system completely replaces Freedom's original Reply Block based mail system. Where possible, the design reuses proven and existing technology, generated both internally and externally of Zero-Knowledge Systems, rather than designing and implementing our own custom solution set.

The design for the POP mail solution is based on qmail[5], an open and freely available software suite that has gained popularity as the basis for reliable, high performance systems. qmail is a highly modular software solution. The new mail system design and implementation embraces this philosophy; each of the subcomponents, which are needed in order to modify a stock qmail installation to the purposes of the Freedom Network, is a small, loosely coupled, and crisply defined unit of functionality. The assembly and integration of these units form the functionally complete mail system.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

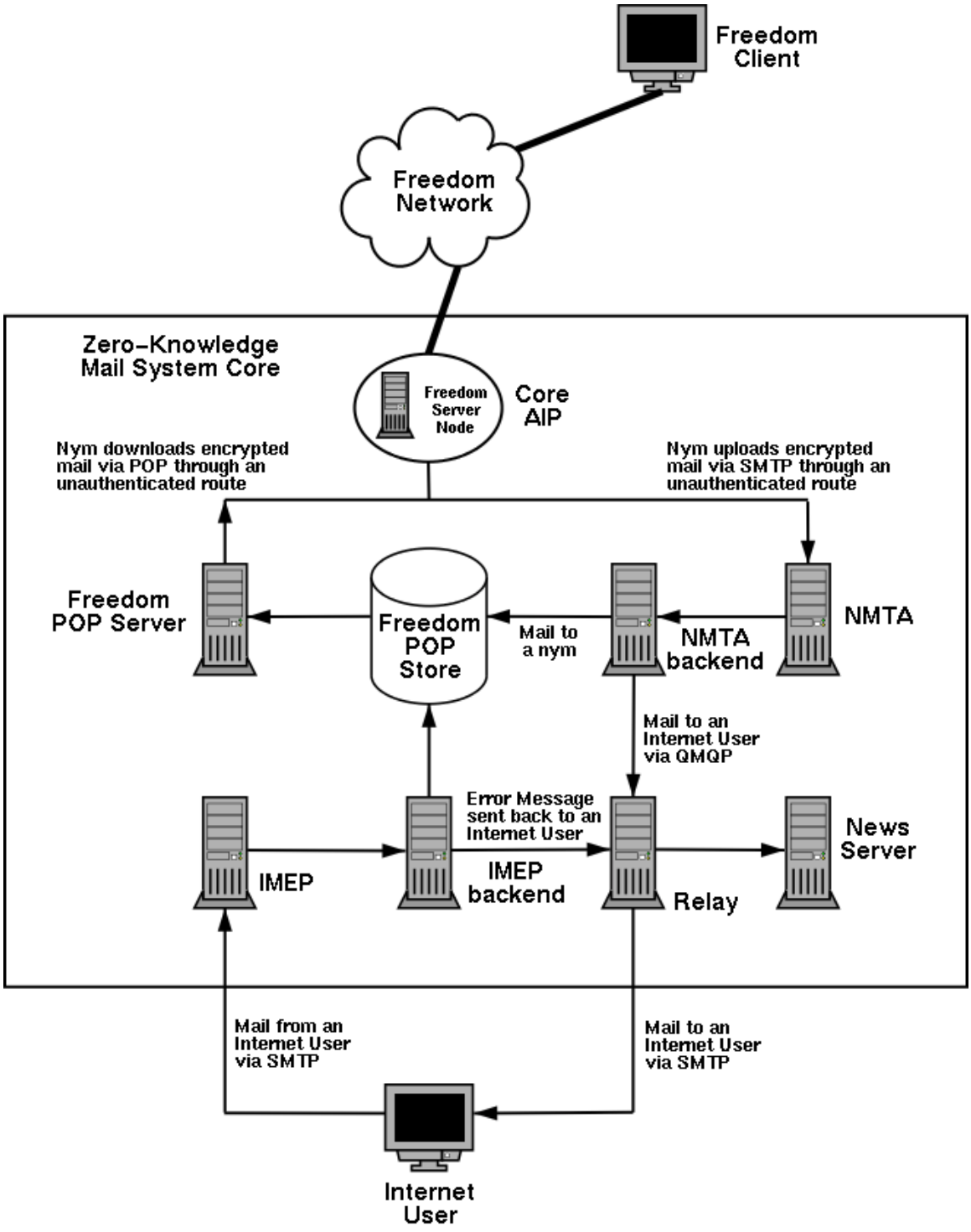

Figure 7: Freedom 2.0 Mail System

There are three new entity types for the mail system: POP, IMEP, and NMTA. The POP entity is the Freedom POP server that is accessed via the local pop proxy on a Freedom client when it wishes to download a nym's mail. The NMTA (Nym Mail Transfer Agent) entity is used to process mail from a nym. The IMEP (Internet Mail Encryption Proxy) is used to process mail from an Internet user to a nym. The

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

IMEP and NMTA are essentially queuing mechanisms used to grab mail messages. The IMEP and NMTA backends consist of a series of filters used to process the mail. Some of these filters are:

Authentication

Programs and libraries to provide authentication services to the mail system such that the PKI system can provide nyms with information that they can use to authenticate themselves to the system. Included in this is proper handling of nyms that have had their privileges revoked.

Bounce Handler

Bounce notifications resulting from failed deliveries have to be delivered to the sender such that the notice is meaningful.

Internet Block List Enforcer

Do not deliver mail to an Internet address that has specifically requested not to receive mail from a particular nym.

Mail2News Gateway

Route mail destined for newsgroups to the appropriate place, respecting our cross-posting constraints and sending limits.

Mail Encryption "Server"

A distributable service to perform encryption and processing of Internet to nym mail.

Mail Sending Limit Enforcer

Make sure that a nym does not exceed the daily allowance of mail that it is allowed to send.

Mail Store Quota Enforcer

Make sure that a nym's POP storage usage does not exceed its maximum allocated disk space.

Nym Block List Enforcer

Do not deliver mail to a nym that has specifically requested not to receive mail from a particular sender.

Spam Control

Insert spam control software into the delivery pipeline so that nyms which turn on our spam control features have several spam filters applied to their incoming mail.

Statistics Engine

Gather and process various statistics on mail system usage and load in order to facilitate system tuning and analysis.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

Tagline Inserter

Insert tagline text into messages to Internet addresses (that is, nym to non-nym email). This is not currently implemented.

The mail system needs to query nym public encryption and signature keys for encrypting and authenticating mail messages. Management of nym account parameters is done through special messages to the mail system itself so there is no dependency on the nym server. Client access to the mail system is done through unauthenticated routes through core AIPs. The mail system, however, is not dependent on this.

Script Server

The Freedom Script Server is an external HTTP server that is run by Zero-Knowledge. It provides the Freedom client with the scripts that allow the Form Filler feature to automatically complete online forms what can be found on various websites. It also provides the rules for the Ad Manager feature, which are used when the Freedom client's ad management capability is enabled.

These scripts and rules are regularly updated by Zero-Knowledge systems and signed by NIQS entity key called NIQS/ScriptServer. The client queries the server on startup and downloads the latest update. If this server is not available, the client will continue to function but will not be able to automatically adapt to new advertising strategies and to new forms. It will merely use the scripts and rules that it currently has in its database.

This server is used only by the Freedom 2.0 client, it is independent of the Freedom Network. The core servers and AIPs are not aware of its presence, nor does it have an associated entity type.

Conclusion

Every effort has been made to make Freedom the most integrated, strongest and easiest-to-use privacy system available, and we believe we have achieved this goal. No system is completely infallible, however, but this white paper, in conjunction with "Freedom 2.0 Security Issues and Analysis"[6], will show the reader the extent to which the system is secure under ordinary, and even extraordinary circumstances. We maintain a policy of full disclosure of the system's workings and weaknesses in an effort to be up-front and honest to the community of Freedom users and interested parties.

Glossary

Freedom	Zero-Knowledge's online privacy suite. Using client-server architecture, Freedom provides total privacy to Internet users. Freedom works transparently with current Internet applications. When using Freedom's premium services, Internet traffic is encrypted before leaving the user's computer and routed through private connections.
Nym	A pseudonymous digital identity. Nyms exist in cyberspace exactly the same way as regular online identities, without revealing the identity of the real-world person. Nyms are active, inactive, or expired. Inactive nyms can be reactivated or deleted. Expired nyms are removed from the Freedom Nym Server and are not available to any other Freedom user.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

Private Key	One of two keys used in private/public (asymmetric) key cryptography. Users receive messages encrypted with their public key and decrypt them using their secret private key. The private key can also be used to digitally "sign" a message, ensuring the authenticity of the sender. When someone uses their private key to sign a message, they encrypt a portion of the original message that the recipient decrypts using the sender's public key. If the public key decrypts the scrambled portion of the message, the recipient knows that the sender is who they claim to be. Private/public key cryptography is the more flexible of the two main encryption types (the other being symmetric key) because the key used to encrypt a message is available to all, but only one person holds the private key (key management is easier than with symmetric encryption). Ensuring the secrecy of the private key is easier when it does not have to be distributed, as is the case for symmetric encryption.
Public Key	One of two keys used in private/public (asymmetric) key cryptography. Users release their public key, which is used to encrypt messages that can only be decrypted using the user's private key. Private/public key cryptography is the more flexible of the two main encryption types (the other being symmetric key) because the key used to encrypt a message is available to all. Ensuring the secrecy of the private key is easier when it does not have to be distributed, as is the case for symmetric encryption.
Symmetric Encryption	An encryption system, where the person encrypting the message uses the same key as the person decrypting the message. Key management and secrecy is paramount to maintaining the integrity of the encrypted data. If the key is compromised or shared, anyone could encrypt and decrypt messages. Sender and receiver must be properly coordinated to ensure that the correct key is used in order to have secure communications. Also called secret-key, single-key, and one-key encryption.

Acknowledgements

The authors of this document would like to thank everyone in the Zero-Knowledge Engineering and Development teams for their valuable contributions. Their input and feedback helped to make this document a reality.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.

Endnotes

[1] "Freedom Network 1.0 Architecture", Ian Goldberg, Adam Shostack, Zero-Knowledge Systems, Inc. http://www.freedom.net/info/freedompapers/Freedom-Architecture.pdf

[2] "Freedom 2.0 System Protocols", Philippe Boucher, Ian Goldberg, Adam Shostack, Zero-Knowledge Systems, Inc. *Not yet published as of this publication date.*

[3] "Freedom 2.0 Threat Model" *Not yet published as of this publication date.*

[4] "Untraceable Nym Creation on the Freedom 2.0 Network", Russell Samuels, Ed Hawco, Zero-Knowledge Systems, Inc.

[5] qmail D. J. Bernstein http://www.qmail.org/

[6] "Freedom 2.0 Security Issues and Analysis", Adam Back, Ian Goldberg, Adam Shostack, Zero-Knowledge Systems, Inc.

© 2000 Zero-Knowledge Systems, Inc. All rights reserved.