Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol 2, No.6, 2011

Heterogeneous Web Client

Zalake Chetan Ramchandra (Corresponding author)
D.Y.Patil College of Enginerring, Akurdi, Pune-44
University of Pune
Tel: 020-27653054 E-mail: che12tan12@gmail.com

Kalyan D.Bamane
Lecturer
D.Y.Patil College of Enginerring, Akurdi, Pune-44
University of Pune
Tel: 020-27653054 E-mail: kalyandbamane@gmail.com

Abstract

Many electronic mail systems are now available. Interconnecting these largely independent systems broadens the communities of users that can communicate electronically. When the systems are at different sites, the existing style of loose interconnection is sufficient. But when a local site has multiple mail systems, a user cannot easily perform tasks such as reading mail from all the systems during a single session. An integrated local mail environment that lets a user perform such tasks easily desirable.Such an integrated environment should have three properties. First, it should be inexpensive to implement the environment. Replacing the existing mail systems with a single, new system is too expensive. Second, it should be inexpensive to integrate a new mail system into the environment. As new systems are acquired, it should be easy to use their mail services effectively. Third, it should be possible to integrate diverse classes of mail systems into the environment. We have designed and implemented research on Heterogeneous Mail System as an environment with exactly these properties.

Keywords: Heterogeneous mail system server, Scalability, Security and authentication, Encryption, Decryption.

1. Introduction

The purpose of this document is to convict, analyze and define high-level needs and features of Composite Mail Server. It focuses on the capabilities needed by the stakeholders, and the target users, and why these needs exist. The details of how the Composite Mail Server fulfils these needs are detailed in the use-case and supplementary specifications. The design of the target system is provided. The various aspects of software like data, program, and interfaces are designed. The project sizing and scheduling, work breakdown structure is done. The test plan for Composite Mail Server is also provided through the same document.

1.1.1 **Purpose**

Problem:

With the advances in the Computer Technologies and the Internet, there has been extensive use of emails for communication. Generally, a user has different email accounts. User has to login separately into each account to view his emails. User has to remember all his passwords and it is time consuming to login into multiple sites. It is not possible to read large mails thoroughly, so user cannot get the actual meaning. So there is a need to develop such a mail server which can solve these problems.

Solution:

The aim of this project is to create an account for user so that he can access to different email accounts without login into multiple sites. User can view all unread and read mails of different accounts in single window. This server manages user's accounts with security. In this, user can perform different mail related functionalities like compose mail, send mail, folder, delete mail. This application provides the facility to create summary of large mails.

Comparison of Existing and Proposed Model:

Refer Note 2:

1.1.2 **Scope**

As clearly stated earlier, this project can be used by any user who wants to access different email accounts in a site. It will help user to view the unread and read mails from different accounts. User can compose and send mail from any email account as he wishes. Important mails can be saved by user in different folders. Address book facility is provided where user can store his friends email ids. This is used on LAN. This project can further increased globally to access Gmail, yahoo or rediff accounts. It will be more beneficial for a user who frequently receive mails and has to reply immediately. It provides simple and fast access to emails. This application provides the facility to create summary of large mails.

1.1.2 **Features**

1. It will be helpful to view unread and read mails at a time from various accounts.
2. Multiple users can access their accounts simultaneously.
3. For each user unique id provided.
4. Multiple profiles for single user to access multiple accounts simultaneously.
5. The security is provided through SQL prepared statements.
6. Various mail functions like compose, send, delete, save to folder, inbox, search, draft, etc.
7. Address book facility is provided to store email ids.
8. Perform actions such as "Mark as Read" or "Junk mails" from the email Inbox.
9. Facility to change password, security question, delete user account.
10. Provide summary of large mail.

2. Heterogeneous Mail System Server

The system server exports a simple interface to heterogeneous mal system clients. The system-server interface is nearly identical to the mail semantic manager interface and is implemented largely in terms of the underlying mail semantic managers. The primary difference is that each operation applies not to a particular subset of a user's mailboxes but to all the mailboxes maintained for a user throughout the local environment. Thus each client of the system server can simply invoke these operations regardless of which mail semantic managers are used to implement the interface. By providing the view of a single mail system to heterogeneous mail system clients, no further modifications are needed when new underlying mail systems are introduced into the local environment.

A user or message-transfer agent becomes a client of the heterogeneous mail system by replacing calls to its system specific submission, retrieval, and delivery operations with functionally equivalent calls to the

system server. In many cases especially with local message transfer agents these changes can be made by altering configuration files. In any case, the required modifications are simple.

Refer Note 3 : Heterogeneous Mail system approach and Note 4: Simple Mail System approach.

3. Security and Authentication

Authentication in a mail system prevents recipients from accessing other user's mail and prevents an originator from posing as another user. Because the Heterogeneous Mail System uses various existing mail systems, it cannot provide greater security than those systems do. But the heterogeneous mail system model does not weaken the security provided by any constituent mail system, since the interfaces exported by each component of the heterogeneous mail system model include a parameter for authentication data. This information, along the authentication mechanisms for the mail system comprising the heterogeneous mail system and to prevent breaches in mail security when an operation requires crossing mail system boundaries.

Before executing a request, each mail semantic manager uses the supplied authentication data to authenticate the caller in the system managed by the mail semantic manager. If successful the mail semantic manager carries out the request; otherwise the mail semantic manager does nothing and notifies the caller that there is an authentication problem. Similarly the system server authenticates its callers in the originating system. The system server facilitates authentication across different mail systems by including authentication data for each system in the caller's master mailbox list. All data added by the heterogeneous mail system to local name services including the master mailbox lists are stored in an encrypted form.

Because the heterogeneous mail system performs a request only if the caller is authenticated, recipients are prevented from accessing other users mail messages and local originators are prevented from posing as others. However the delivery agent of a local mail system may make a request on behalf of a user outside the local environment. In this case, we partly rely on the authentication mechanisms of the individual systems to authenticate the originator. The call to the system server, however, will require authentication of the delivery agent before the delivery request is performed.

4. Scalability

The issue of scale in distributed system has three dimensions: the number of users and components, the distance between components, and the number of separate yet cooperative administrative domains. The general techniques of distribution, replication, and caching are typically used to address these three dimensions. Our model uses the well-known techniques to resolve the first and third dimensions. Because our work focuses on the local mail environment, we do not directly address the second dimension; we believe the issue of component distance should be resolved by the individual mail systems.

The first dimension requires that the work to deliver, retrieve, or delete a message be bounded as the numbers of local mail systems and users grow. Also a single component should not become a bottle neck. In our model most of the required processing resides with the individual mail systems; additional processing is distributed across the heterogeneous mail system server and the mail semantic managers. These components are replicated, the any replica can handle requests. We use information catching to further to reduce the load on these components. Furthermore, the number of operations required by each heterogeneous mail system component to perform a user request is fixed to the number of mail systems associated with the request.Most of the data required for each user resides with the individual mail systems. Additional data about the users and about individual mail systems are stored in a distributed and replicated name service. Although data is maintained for each mail system and for each user, the amount of information is relatively small.

For administration, the heterogeneous mail system model scales naturally because, their operation, individual mail systems continue to be administered independently. The additional data required by the heterogeneous mail system is small and easy to administer locally.

5. System Features :

5.1Access to different email accounts

5.1.1 Description

User has many email accounts. To access all accounts simultaneously, it is time consuming process. Using this application, user can view all mails from his mentioned accounts.

5.1.2 Functional Requirements

REQ-1: It checks the validity of user id and password entered by user.

REQ-2: If any new mail arrives in user's email account, it displays in his account window.

REQ-3: When user creates new account, he has to enter ids along with passwords which system checks with mail server.

5.2 Mail functionalities

5.2.1 Description

Like other mail servers, this mail server provides various mail functions like compose, send, delete, save to folder, inbox, search, draft. Address book facility is provided to store email ids.

5.2.2 Functional Requirements

REQ-1: After sending mail to others, mail is stored in sent items.

REQ-2: Entries in address book are stored in user's database.

6. Encryption and Decryption:

Encryption is the act of encoding text so that others not privy to the decryption mechanism (the "key") cannot understand the content of the text. Encryption has long been the domain of spies and diplomats, but recently it has moved into the public eye with the concern of the protection of electronic transmissions and digitally stored data. Standard encryption methods usually have two basic flaws: (1) A secure channel must be established at some point so that the sender may exchange the decoding key with the receiver; and (2) There is no guarantee who sent a given message. Public key encryption has rapidly grown in popularity (and controversy, see, for example, discussions of the Clipper chip on the archives given below) because it offers a very secure encryption method that addresses these concerns.

In a classic cryptosystem in order to make sure that nobody, except the intended recipient, deciphers the message, the people involved had to strive to keep the key secret. In a public-key cryptosystem. The public key cryptography solves one of the most vexing problems of all prior cryptography: the necessity of establishing a secure channel for the exchange of the key.
The RSA algorithm, named for its creators Ron Rivest, Adi Shamir, and Leonard Adleman, is currently one of the favorite public key encryption methods. Here is the algorithm:

1. Choose two (in practice, large 100 digit) prime numbers **p** and **q** and let **n** = **pq**.

2. Let $\mathbf{P}_i$ be the block of (plain) text to be encrypted. Actually $\mathbf{P}_i$ is the numerical equivalent of the text which may either be single letters or blocks of letters, just as long as $\mathbf{P_i}< (p-1)(q-1)= Ø(n)$

3. Choose a random value **E** (usually small) such that **E** is relatively prime to Ø(n). Then the encrypted text is calculated from

$$\mathbf{C_i = P_i^E \ mod(n).}$$

The pair of values (**n**,**E**) act as the public key.

4. To decode the ciphertext, we need to find an exponent **D**, which is known only to the person decoding the message, such that

$$\mathbf{DE = 1mod((p-1)(q-1)).}$$

Note that **Ø(n)= Ø(pq) = (p-1)(q-1)**. Then we may calculate

$$\mathbf{C_i^D = (P_i^E)^D = P_i^{DE} = P_i mod(n)}$$

This step is based on the following result:

$$\mathbf{(a^x)^y = a^{xy} = a^x mod(n),}$$

Where **z=xymod(Ø (n))**. Show that this result is true.

(Since **xy = zmod(n),then xy=m Ø (n)+z** for some integer m. Thus, applying Euler's theorem we have

$$\mathbf{a^{xy} = a^{mØ\,(n)+z} = (a^{Ø\,(n)})^m a^x = (1)^m a^z = a^z mod(n).}$$

By Euler's theorem

$$\mathbf{E^{Ø\,(Ø\,(n))} = 1mod\ (Ø\ (n))}$$

Provided **E** and Ø (n) are relatively prime, which is true by the choice of **E**. So we obtain

$$\mathbf{DE = 1mod\ (Ø\ (n)),}$$

$$\mathbf{DE = E^{Ø\,(Ø\,(n))}\ mod\ (Ø\ (n)),}$$

$$D = E^{Ø(Ø(n))-1} \bmod (Ø(n)),$$

Therefore, we have an equation that can be used to find the "key" exponent **D**. The central result of the RSA algorithm is that this equation is computationally solvable in polynomial time (actually using the Euclidean Algorithm) whereas solving by factoring **n** requires exponential computational time. [Note however that this last statement has never actually been proven but only demonstrated given today's algorithms. Should someone discover an algorithm that factors integers in polynomial time, the RSA and similar algorithms could be rendered useless for secure communications? Central to these calculations knows the factorization of **n**, since we will need to calculate both (Ø (n)) and Ø (Ø (n)).

A Simple explanation of RSA Algorithm in view to computer:

Let p and q be distinct large primes and let n be their product. Assume that we also computed two integers, d (for decryption) and e (for encryption) such that

$$d * e \equiv 1 \pmod{ø(n)}$$

Where ø(n) is the number of positive integers smaller than n that have no factor except 1 in common with n

The integer's n and e are made public, while p, q, and d are kept secret.

Let m be the message to be sent, where m is a positive integer less than and relatively prime to n. A plaintext message is easily converted to a number by using either the alphabet position of each letter (a=01, b=02... z=26) or using the standard ASCII table. If necessary (so that m<n), the message can be broken into several blocks. Prime numbers.

The security of RSA depends on the fact that it takes an impractical amount of time to factor large numbers.

Refer Note 1: **Example of encryption and decryption.**

Note 1: **Example of encryption and decryption:**

Suppose we wish to encode the plaintext message $\mathbf{P}_i = 3$ (that is, under our encoding some letter has been assigned the numerical value 3) subject to our choices of **p**=11, **q**=17 (thus, **n**=187) and **E**=7 (note that 7 is relatively prime to 187.) Then the cipher text $\mathbf{C}_i$ is given by

$$\mathbf{C_i = 3^7 = 2187130 \bmod (187).}$$

Thus the receiver must decode the message $\mathbf{C}_i = 130$. To decode this "message" the receiver must calculate the exponent **D**. [Note that in this example the factorization of **n** is relatively easy, so someone could break

the code by factoring **n** and calculating **D**. However, in practice, we could choose **n** large so that only we would (theoretically) know the factorization.]

Since **n** = 11 ·17, then **ø(n) = 10.16 = 160,** and **ø(ø(n)) = ø(160) = 64**

[WARNING! WARNING! Will Robinson.] Thus we obtain

$$\mathbf{D = E^{\,\phi(\phi(n))-1} = 7^{63} \bmod (160)}$$

Example: Calculate 7^{63} mod (160)

Why was there a warning in the previous example? If you have been closely examining what has taken place in the RSA algorithm you may have noticed that although we know the factorization of **n** (since we choose the prime factors **p** and **q**) and hence **ø(n) = (p-1)(q-1)** , we may not have an easy time determining **ø(ø(n)) = ø((p-1)(q-1))**, which requires us to know all the factors of what could be a very large number. This seems to contradict the polynomial time needed to solve for the key. The solution is (and is a key -- unintentional pun -- element of the RSA algorithm) that the formula for **D**, although concise, is not the way the solution is found in practice. The actual method of solution (which does require polynomial time computation) is based on the Euclidean algorithm.

Returning to our previous example, recall that we want to solve

$$\mathbf{DE = 1 \bmod (160),}$$

$$\mathbf{7D = 1 \bmod (160).}$$

By our choice of **E**, 7 and 160 are relatively prime, and thus

$$\mathbf{160 = 7(22) + 6}$$

$$\mathbf{7 = 1(6) + 1}$$

using the Euclidean algorithm. Working in reverse gives

$$\mathbf{1 = 7 - 1(6)}$$

$$\mathbf{1 = 7 - 1(160 - 7(22))}$$

$$\mathbf{1 = 23(7) - 1(160)}$$

$$\mathbf{1 = 23(7) + (-1)(160).}$$

In algebraic terms, we say we have written 1 as a linear combination of 7 and 160. Since 160 is the modulus, we have

$$\mathbf{(23)(7) = 1 \bmod (160)}$$

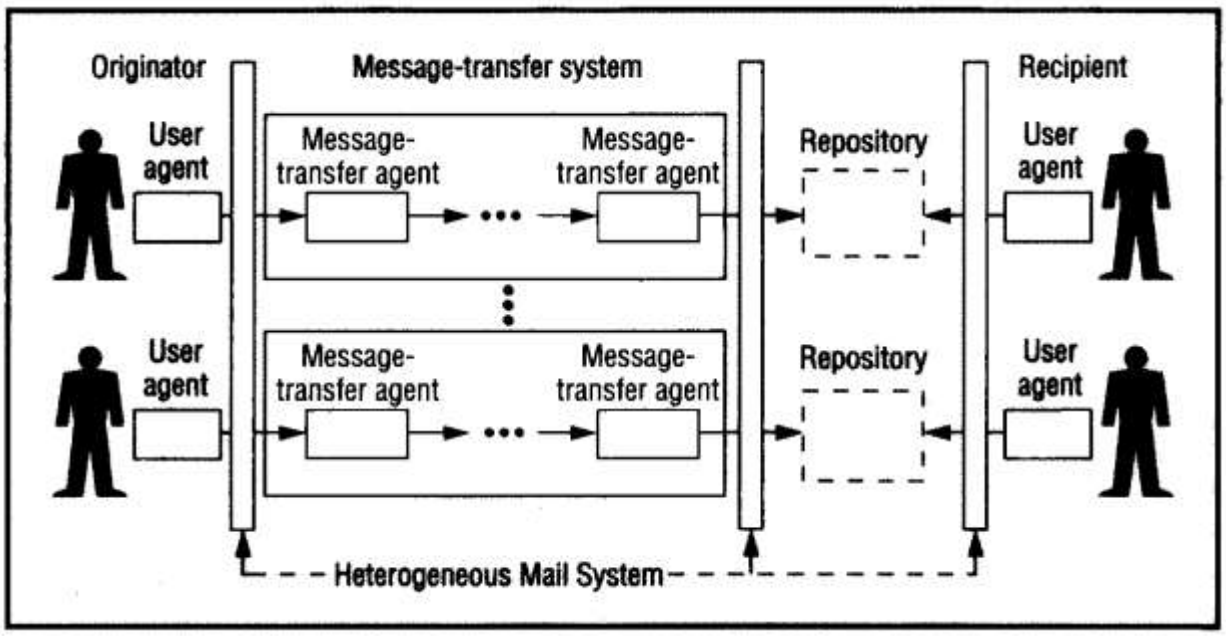
Hence **D**=23! Thus the real key to the solution of **D** is knowing **ø(n)** which requires the knowledge of the factorization of **n** since **ø(n) = (p-1)(q-1)**.

Note 2:

	Existing	Proposed
1.	User has multiple accounts and has to login every time.	User has to login once to access every account.
2.	Do not give summary of large mails.	Gives summary of large mails.
3.	Multiple profiles cannot be viewed at a time.	Multiple profiles can be viewed at a time.
4.	Address book facility is not provided.	Address book facility is provided.
5.	FTP uploader and downloader are not provided.	FTP uploader and downloader are provided.

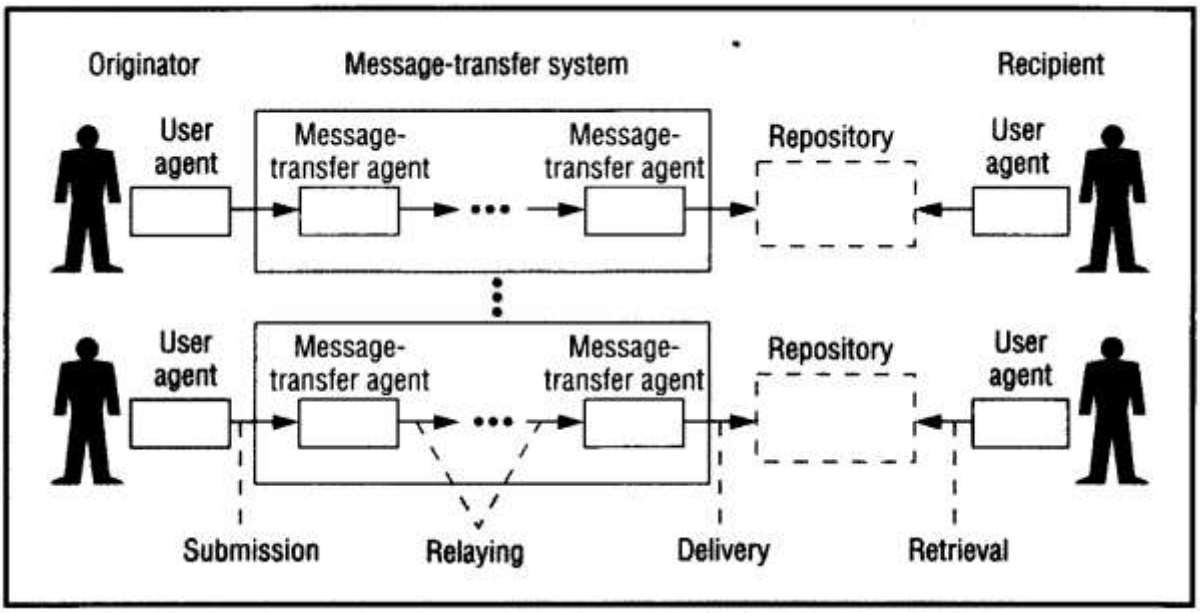
Table of Comparison between Existing and Proposed model.

Note 3:Heterogeneous Mail System approach.



Heterogeneous Mail System Approach

Note 4 : Simple Mail system approach



Simple Mail System Approach

7. References :

1. E. Allman, " Sendmail : An Internetwork Mail Router, " Unix Programmers Manual 4.2BSD,2C, Computer Science Research Group, University of California, Berkeley, California,June 1985

2. A.D.Birrelletal., " Grapevine : An Exercise in Distributed Computing, " Comm. AGM, April 1982, pp 260-274

3. DD.Redell and J.E.White, " Interconnecting Electronic Mail Systems, " Computer, Sept.1983, pp 55-63

4.B.N.Bershad et al., "A Remote Procedure Call Facility For Interconnecting Heterogeneous Computer Systems,"IEEE Trans. Software Eng.,Aug 1987, pp.880-894.