第三章 即時傳送架構下Mail的防匿方法

3.1 Mail的運作方式

目前已被應用在TCP/IP網路上的Mail系統如sendmail、zmailer等，其傳輸方式皆遵循SMTP(Simple Mail Transfer Protocol)[1]，而傳輸文件之格式亦以RFC822(Standard for the format of ARPA Internet text messages)爲依歸。下圖即爲在SMTP架構下Mail的運作方式：

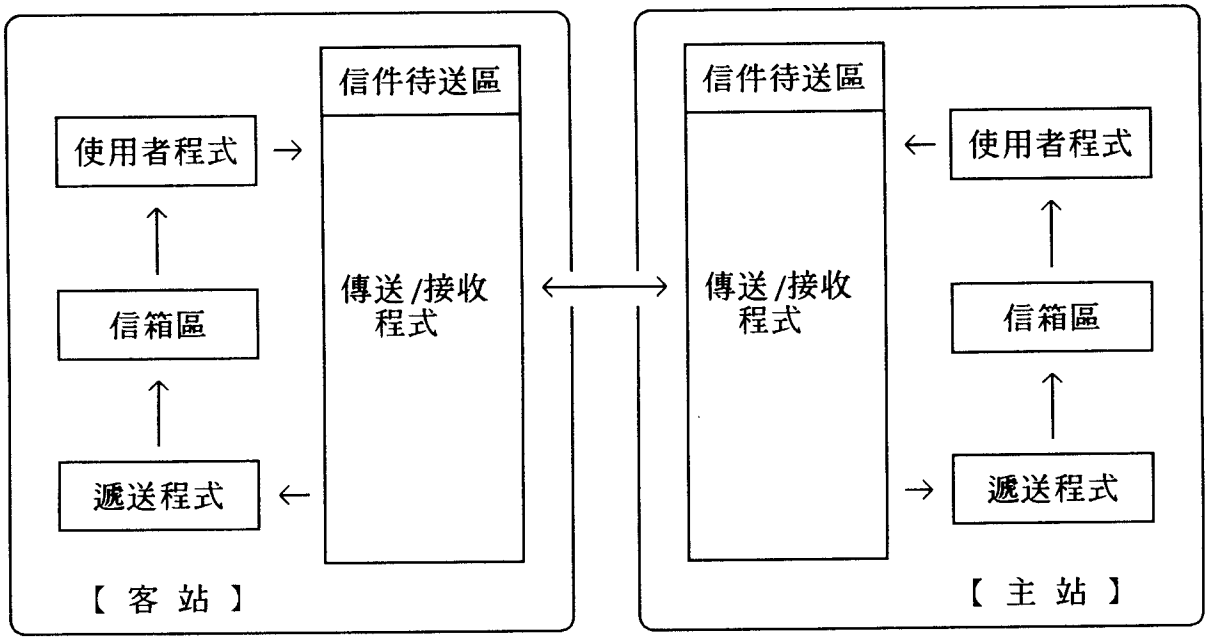


圖3.1 SMTP架構下Mail的運作方式

傳送/接收程式：如sendmail、smail、zmailer等。

遞送程式：如mail、rmail、deliver等。

使用者程式：如mail、mailx、elm等。

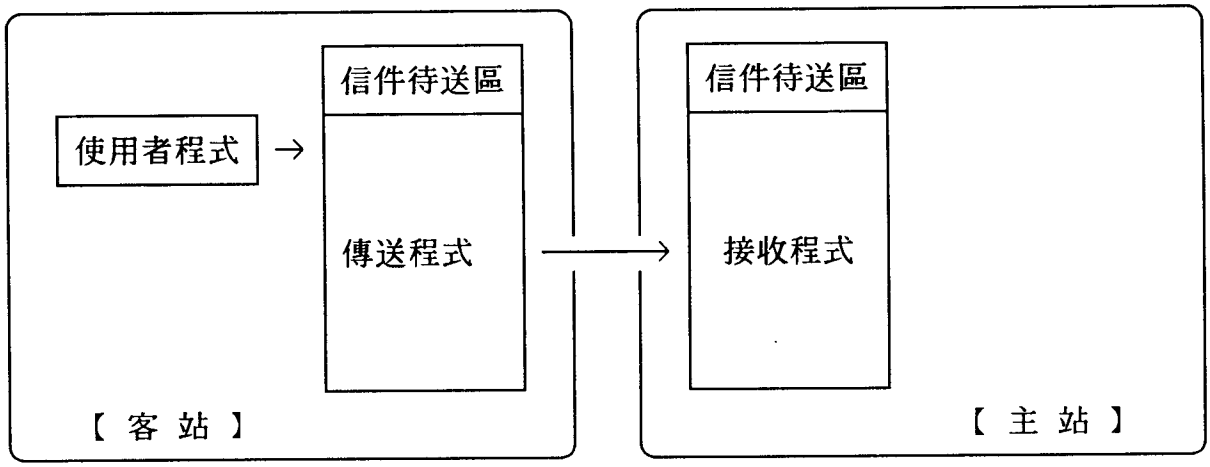
從上圖我們可以很清楚地看出Mail的防匿關鍵：在客站爲使用者程式與傳送程式；在主站爲接收程式。

3.2 在Mail客站上的防匿方法

一篇使用者所寫的信函從編寫完成到被傳送至Mail主站前，Mail系統

[1]詳見RFC821, Simple Mail Transfer Protocol。

會先作如下的處理：



使用者程式：任意編寫稿頭與文稿。

```
To: ruser@rhost.edu.tw
Subject: test

Here comes the mail body
```

傳送程式：檢查稿頭，確定各欄位如From等是否齊全，若有缺則予補齊，並加蓋郵戳(Received)，再將信函傳送給主站。

```
To: ruser@rhost.edu.tw
Subject: test
Received: by shost.edu.tw
Message-Id: <datenumber@shost>
From: suser@shost.edu.tw
Date: April 15, 1994


Here comes the mail body
```

接收程式：接收客站送來的信函，並加蓋郵戳(Received)。

```
To: ruser@rhost.edu.tw
Subject: test
Received: by shost.edu.tw
Received: by rhost.edu.tw
          from shost.edu.tw
Message-Id: <datenumber@shost>
From: suser@shost.edu.tw
Date: April 15, 1994




Here comes the mail body
```

圖3.2 一般Mail主、客程式處理Mail的過程

從上圖我們可以看出一個給匿名者有機可乘的漏洞：由於傳送程式完全接受使用者所編寫的文稿(包括稿頭中的From欄位)，所以From欄位

的眞實性就非常值得懷疑。因此我們似乎必須對傳送程式或使用者程式加以修正，但是做這樣的修正工作並不恰當，原因有三：

- 爲了配合發函人可能會用別名(alias name)與隱域名稱(hidden domain)來組合通訊地址的習慣，許多使用者程式均提供了修改From欄位的功能。
- 使用者程式的種類太多(如mail、mailx、elm等)，不值得我們逐一修改，而且使用者程式處理過的文稿還要再送傳送程式處理。
- From欄位的重寫牽涉到各傳送站的重寫規則(rewriting rules)。我們千萬不能只考慮單純的

  起點→終點

  的傳送，應該要顧及在

  起點→中點→終點

  的傳送過程中，各站處理From欄位的方式是否一致。在第二種情況下，若是From欄位在每一站上都先被剔除然後再重寫，那麼當信件還未被送達終點前，From欄位所記載的發函人資料早就失眞了。

幸好絕大部份的Mail傳送程式都會在傳送信件前，檢查信件的第一行是否爲

From 發函人@發函人地址 [1]

若無，則補；若有，則不補。這項功能乃是因循傳統UNIX系統對信件的處理方式[2]。這項功能在某些傳送程式中是可以選用的(optional)，但在某些傳送程式中是屬於強制性的。因此，只要我們能讓這項功能表現出來，就不須再修改傳送程式，並達到防匿的效果。

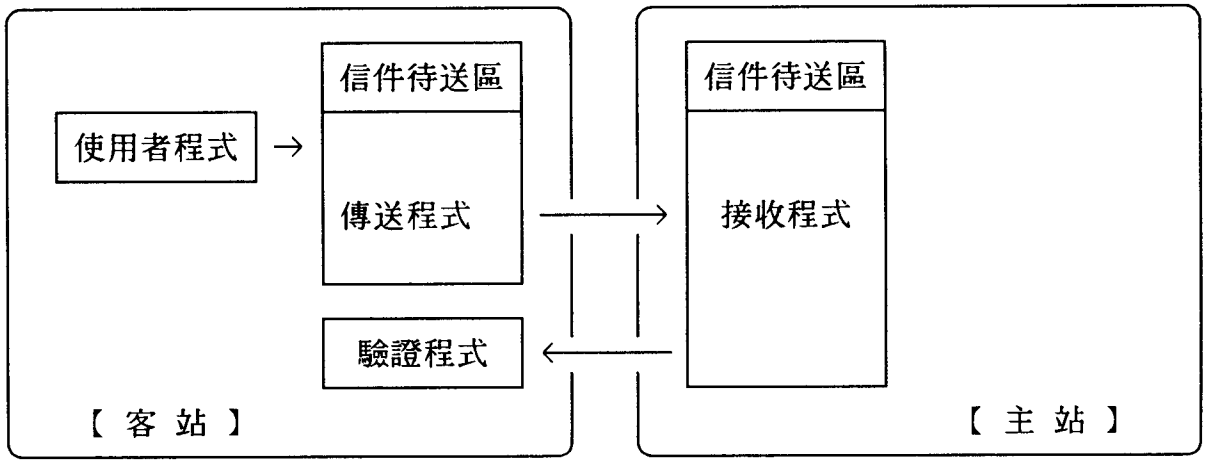
---

[1]請注意，是"From 發函人@發函人地址"，而非"From: 發函人@發函人地址"。

[2]在mail folder中，"From 發函人@發函人地址"代表著一封信件的開始。

3.3 在Mail主站上的防匿方法

我們在第二章第三節曾提到有些匿名者知道如何與主站上的接收程式直接溝通，根本就不理會我們所修正的傳送程式。因此，我們在此亦將驗證協定的技術引用在Mail的防匿機制上。其運作方式如下圖：



使用者程式：任意編寫稿頭與文稿。

```
From: fake@shost.edu.tw
To: ruser@rhost.edu.tw
Subject: test

Here comes the mail body
```

傳送程式：檢查稿頭，確定各欄位如From是否齊全，若有缺則予補齊，並加蓋郵戳(Received)，再將信函傳送給主站。

```
To: ruser@rhost.edu.tw
Subject: test
Received: by shost.edu.tw
Message-Id: <datenumber@shost>
From: fake@shost.edu.tw
Date: April 15, 1994

Here comes the mail body
```

接收程式：接收客站送來的信函，並向客站上的驗證程式查詢傳送者的資料，且將查詢結果寫入郵戳(Received)中。

```
To: ruser@rhost.edu.tw
Subject: test
Received: by shost.edu.tw
Received: by rhost.edu.tw
          from fake@shost.edu.tw
Message-Id: <datenumber@shost>
From: fake@shost.edu.tw
Date: April 15, 1994


Here comes the mail body
```

圖3.4 具防匿功能的主、客端程式處理Mail的過程

由於在Mail的稿頭中並無類似於News中的Nntp-Posting-Host欄位，而且Mail中的From欄位牽涉到複雜的重寫規則，並不適合加入驗證程式所得到的傳送者資料，因此我們選定郵戳Received作爲我們填入驗證資料的欄位。

3.4 其它的防匿方法

我們在第二章第四節曾提到兩個在News上的防匿構想，但是很顯然的這些構想並不適用於Mail上。這是因爲News的投送通常是發生在某些特定的主、客站上，而Mail的傳送卻可以在任意的兩站之間產生。因此，若是改變了SMTP原有的協定，將會導致信件無法傳送。