Electronic Money and Payment System (IMONEEZ)

A.A.Emagoulov eartur@asplinux.ru July, 2003

Contents

1 Goals

Main goal of this project is to provide standard mechanism for money transactions and internet payments and sure to create usefull and easy understandable environment for people who will use the payment software and application developers who will develop such software. And to make clearance and security of money transaction in internet. At last time the process of integrating internet and information technologies into business is growing and there are many people and companies has a look to make payments using the internet, but also at this time there are many standards and mechanisms are used for this task, and they differs from each other.

The standardized mechanism of money transactions will help to control and to make them secure and, may be the most important thing, will gives people confidence with them transactions.

2 Concepts

Concept of imoneez is based on anonymous coins and cheques used, which are the RSA ecnrypted messages. The coin is the one time generated random number with random public RSA key, stored in the database, the private encripted data is sent to peer. When the peer wants to make payments, it makes request to money server with 'TRANSFER' request (see MIME extensions for transactions) with coin attchament, server verifies the coins and ability of request and stores the request in its transaction database deleting coins accepted for request from peer. When this process finished successfully server returns 'OK reply' (see payment protocol docs) to peer with 2 cheques included, one for peer, one for recipient of transactions as keys to new moneys. Then peer must send recipients cheque to the recipient, which takes its own new generated coin. So both sides has cheques of transaction made, and the third side (server) stores both cheques for possible disagreements.

Let see the figure. The start of transaction is when the sender makes an invoice of transaction (1 phase), which includes the type of invoice, payment value and coins attached. Next the sender sends his invoice

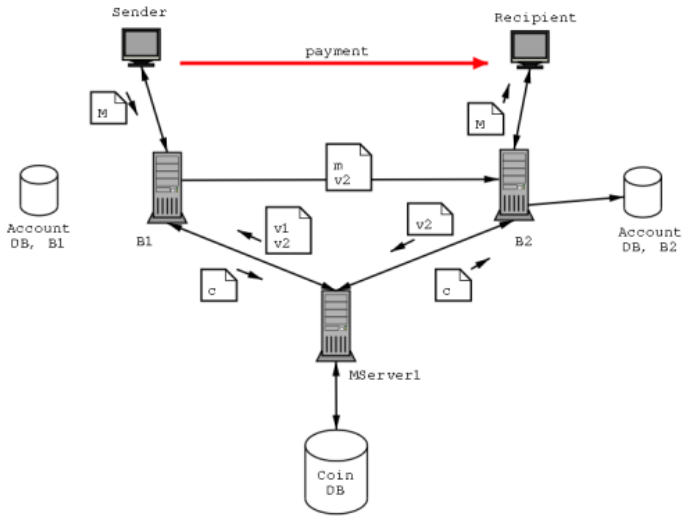

Figure 1: The diagram showing the process of transaction

throgh the special mail server, which has filter installed for invoice messages. Filter searches some fields in messages and if the message contains needed fields it filters body, if none, then messsge is sent as normal email. The body is searched for attachments with defined MIMEs. If the transaction is normal transfer transaction, body must contain attched coins, which are filtered and verified by the filter. Verification is a next level request to monetary server (MServer1 on figure) with coins sender's included and with the value of transaction. If coins are valid coins, then the filter replaces coins attached with the replied cheques for transactions made by monetary server during coins verification.

And then mail server (B1 on figure) sends the invoice as a normal email. Next phase of transaction is a message delivering by another mail server (B2 on figure), which has filter on delivering agent, which task is to place messages to approapriate folders. It filters the messages and searches invoice fields in it then it does back procedures on the invoice - replaces cheques attached with the coins, retrieved from monetary server using this cheques.

3 Design & Implementation

The design of imoneez is described below. Therms:

- Peer
- Recipient
- Coin
- Cheque

Imoneez consists of three sides, really of five sides and three layers. Layered design is shown on figure 1. Let see at the figure and will understand what happense on the first (server) layer. The communication process is based on MIME messages exchanging. The complete detailed format of messages are in the attachment. So let's look at the usuall situation, when the peer wants to make payment, and must do request to server first. This request called TRANSFER-request and must be the following format

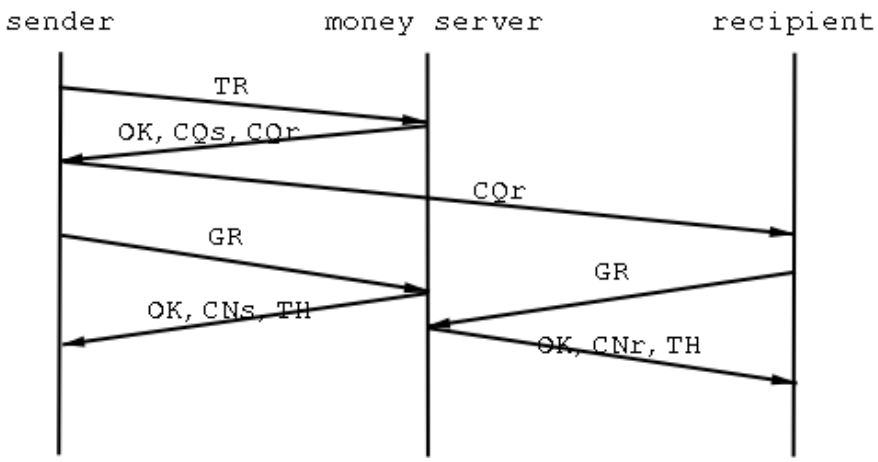

Figure 2: fig.1 The diagram of successfull TRANSFER transaction

```
Request-Type: TRANSFER          ; Request type field
Currency-Name: RU               ; Currency Code field
Cache-Value: 32.1200            ; Value to be transferred
Content-Type: Multipart/Mixed   ; Specifies that there are the attachments in the message
<CLCR>
<CLCR>
attachments
.
.
.
```

Brief description of reductions in figure.

- TR - TRANSFER Request
- CQr,CQs - recipient's and sender's cheques
- GR - GET request
- CNr,CNs - recipient's and sender's coins
- TH - Transaction Hash as a guarantee of transaction for possible disagreements

3.1 Client side transaction composition

Client side transaction composition is similar to email composition, really it is the email composition itself, but with additional fields included (see fig.2). In the example we used mozilla mail agent as a transaction source, but with some header fields preferences added. So, the sender to make the transaction must define additional fields in message, Invoice-Type, Invoice-Value and Invoice-Currency. Optional fields are From-Address, From-Name, From-Phones, To-Address, To-Name, To-Phones, which may be usefull for organizations.

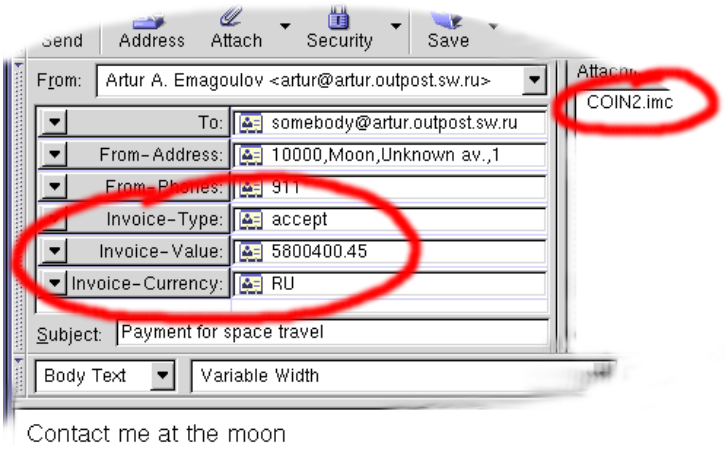

Figure 3: The diagram showing the process of transaction