Micro Payment Transfer Protocol (MPTP) Version 1.0

INTERNET DRAFT — Phillip M. Hallam-Baker
November 22nd 1995 — World Wide Web Consortium
Expires May 27th 1996 — email: <hallam@w3.org>

Micro Payment Transfer Protocol (MPTP) Version 0.1

<draft-hallam-micropayment-00.txt>

Status of this Memo

This draft, filename draft-hallam-baker-internet-micropayment-00.txt is intended to be become one or more Proposed Standard RFCs. It is also available as hypertext as a World Wide Web Consortium Working Draft as http://www.w3.org/pub/WWW/TR/WD-mptp951122.html. Distribution of this document is unlimited. Comments should be sent to the author <hallam@w3.org>.

Micro Payment Transfer Protocol (MPTP) Version 1.0

Abstract

A protocol for transfer of payments through the services of a common broker is described. The processing demands of the protocol make it practical for small payment amounts. The latency makes it practical for use in interactive applications. The scheme thus satisfies the two key criteria for a micropayments scheme. MPTP implements a variation of the Pay-Word proposal of Rivest and Shamir [RivestSh95]. It is also inspired by the Millicent proposal by Manasse [Manasse95] and the iKP proposal by Bellare et. al. [BellareEt95]. A proposal similar to the PayWord scheme by Torben Pedersen [Pedersen9?] was reported after this draft was begun.

For efficiency it is desirable to be able to combine transfer of payments instructions with those accomplishing the delivery of goods. For this reason MPTP may be layered on a variety of Internet protocols including HTTP and SMTP/MIME.

Although the protocol is optimized for use as a payment scheme it is suitable for the transfer of larger amounts. The protocol is also suitable for use as an access control or resource allocation mechanism. With modification the protocol could be made to provide anonymity guarantees.

Micro Payment Transfer Protocol (MPTP) Version 1.0

Contents

Introduction

Commerce on the Internet may be broadly divided into three categories, advertising, sale of tangible goods and sale of non-tangible goods. Advertising is not directly considered in this paper. Movement of tangible goods requires handling and shipping whose costs set a minimum value for which trading is economic and introduces a substantial delay into the process. Speed and cost of processing are thus not the critical factors in evaluating payment systems for this application. Where non-tangible goods are involved the contract may in many cases be fulfilled through the internet.

There is a large interest in payment systems which support charging relatively small amounts for a unit of information. Here the speed and cost of processing payments are critical factors in assessing a schemes usability. Fast user response is essential if the user is to be encouraged to make a large number of purchases. Processing and storage requirements placed on brokers and vendors must be economic for low value transactions. MPTP is optimized for use for low value transfers between parties who have a relationship over a period of time. It also provides a high degree of protection against fraud making it applicable in wider scenarios, including sale of tangible goods.

The following performance statistics were used as a guide in designing the protocol [RivestSh95]. Current workstation and high end personal computer class machines are capable of approximately two public key signature creation operations, two hundred public key verification operations and 20,000 one way hash functions per second. Latency introduced by round trip communications may be a second or more, the time taken for a signal to be relayed by a satellite in geostationary orbit. Communications also introduce potential unreliability.

A distinction is made between the cost of inline and offline processing. Inline processing takes place on the critical path for payments. Servers must thus have substantial excess capacity to deal with fluctuation in demand. Offline processing may be performed asynchronously when processing load permits. As a rough estimate inline processing was considered to incur two orders of magnitude more cost than offline processing.

In addition to processing costs, the cost of online storage must be borne in mind. If online storage must be accessed during a payment transfer the cost of storing the data in RAM or of disk head contention must be considered.

MPTP is an asynchronous protocol. Much of the processing required may be done offline. In particular payment does not require an online communication with the broker (unless the symmetric signature option is used). MPTP is also symmetric, there is no distinction between customer and vendor accounts except in relation to specific transactions where the flow of payment is generally in a single direction. The ability to make payments need not preclude the ability to accept payments unless this is a matter of broker policy. However in some cases it might be desirable to have different accounts for these functions, a vendor accepting large payments might wish to avoid the danger of a security compromise allowing unauthorized payments to be made via the Internet accept only account.

One of the most significant differences between the World Wide Web and print media is that the cost of publishing on the Web is commensurate with the cost of readership and does not involve a high capital outlay. Many of the initial users of the Web were primarily interested in publishing existing information, both within organizations and to the wider internet community. One important characteristic which a Web micropayment scheme must satisfy is that it permit access to commercial publication to both small and large publishers. MPTP provides for transfer of small payment amounts through vendor amortization. Use of public key signature screening as opposed to verification makes it economic for use by small publishers.

Principles and Parties

MPTP involves three parties, a customer _C_ who makes the payment, a vendor _V_ who receives the payment and a broker _B_ who keeps accounts for the parties concerned. At present only a single broker model is considered, this means that both customer and vendor must share the same broker. Note however that the protocol does not restrict the broker to use of a single server.

Such a capability is essential if customers are to be able to surf the Web using a single payment account and vendors are to be able to accept payments from any source through a single account. Although accounts are in principle being used by both customers and vendors

it is likely that brokers will wish to specialize in particular types of account. ISPs for example have an existing client base which is billed on a regular basis. Issuing bills requires a very different type of procedure to issuing payments however and thus vendor support may require the services of a specialist broker. Support for Inter-Broker transfers will be required in the long term to permit the system to be scaled effectively. Some notes on the technical and political difficulties involved are made in the commentary section at the end.

It is important that a payment protocol does not interfere with the established trust relationships between the parties. Where a protocol allows collection of data on another parties activity this should be made clear in advance. It is not a requirement that the protocol duplicate the trust models of a particular financial instrument precisely. It is more important that the protocol provide flexibility in the establishment of trust relationships than attempt to define which party accepts what risk.

The term Broker is used to refer to a financial intermediary. In the context of this proposal a broker might be any organization with the ability to bill a significant number of customers at a small marginal cost. This means that in addition to use by financial institutions MPTP might be suitable for use by Internet Service Providers (ISPs), telephone companies or any other organization sending out large numbers of invoices.

Risk Model

[The risk model will be developed in depth in a companion paper. It is intended that the risk model be comprehensive, permitting cross comparisons between different types of schemes to be made.]

The main risks faced by the customer are liability for unauthorized payment and either not receiving the goods at all or receiving goods different from those advertised. The vendor risks not being paid. The broker risks the cost of customer service due to malfunction or incompatibilities and liability for payments in cases where the customer and vendor are in dispute.

The following risks were considered:

Credit Abuse
: An account is used to make repeated payments without intention to pay.

Counterfeiting
: A fake payment order is constructed.

Unauthorized Withdrawal
: The broker (or an employee) makes an unauthorized withdrawal

from an account.

Purchase order modification.
: A customer issues a payment order intending to purchase one set of goods but the order is intercepted and modified by a third party.

Failure to Credit Payment
: The broker debits the customer account but does not credit the vendor account.

Double Spending
: A payment instruction is used twice, either by a customer, a vendor or a third party.

Denial of Service
A customer or vendor is denied use of their account.

Repudiation
A party may deny making a payment.

Credit liability
Where a customer is extended credit liability must be controlled.

Failure to Deliver
A vendor may accept payment but fail to supply the advertised goods.

Framing
A party is able to convince another that a third party acted in bad faith.

In many cases it is in the interests of a party to accept risk. A broker may offer to guarantee payments to vendors irrespective of whether the customer pays and require a payment of a higher commission in return. The high cost of customer service enquiries must be borne in mind. Protocols which cannot clearly identify which party was at fault are likely to incur substantial customer service costs.

Policy

The interests of the parties may conflict. In such cases the choice of which parties interest will be prioritized is a matter of policy. MPTP is designed to be policy neutral, permitting a broker to offer a wide number of policy options. Individual vendors may thus choose the precise terms on which they offer goods. Customers may choose the terms they are willing to accept on a vendor by vendor basis

One of the areas in which conflict of interest occurs is whether goods are to be delivered before or after payment. It is in the customers interest to withhold payment until the goods are delivered

to ensure that they are satisfactory. The Vendor may wish to ensuring payment is made by insisting of payment in advance however.

As previously noted it may be in the interest of the vendor to encourage customers to purchase goods by accepting a degree of risk. This is particularly important when the vendor has no established reputation with the customer. This risk may be made acceptable to the vendor provided there is a high enough probability that payment will be made. In the green-commerce model [SteinStBoRo94] the broker acts to encourage this by monitoring the number of refusals made by a customer and excluding customers with a bad track record.

MPTP permits a considerable degree of flexibility in establishing a payments policy. A vendor may permit a customer a certain amount of trade before requiring a firm payment commitment or require all purchases to be paid for in advance. The first policy may be applicable where the goods offered cannot be evaluated by the customer in advance. A Vendor who has established a reputation with the customer may be in a position to insist on prior payment.

As an example consider the vendor of a software program who wishes to charge for its use on an hourly basis. The vendor may be willing to offer a customer a free trial provided there is a guarantee that the customer cannot simply request a fresh retrial each time a trial expires.

Mechanism

In the Pay Word scheme a payment order consists of two parts, a digitally signed payment authority and a separate payment token which determines the amount. A chained hash function,is used to authenticate the token. These are described by Lamport [Lamport81] and employed in the S/Key [Haler94] authorization mechanism. To create the payment authority the customer first chooses a value _w_n _at random. The customer then calculates a chain of payment tokens (or _paychain_) _w_0, w_1, ... w_n _ by computing

_w_i = h (w_i+1)_

Where h is a cryptographically secure one way has function such as MD5 [Rivest92c] or SHA [AccreditedSC93a].

The signed payment authority contains _w_0_, the root of the payment chain and defines a value for each link in the chain. Payments are made by revealing successive paychain tokens. Once the vendor or broker has authenticated a payment authority an arbitrary payment token may be authenticated by performing successive hash functions and comparing against the root value. It should be noted however that the broker is only presented with the final payment order. It is therefore unnecessary for the broker to maintain large online databases.

MPTP permits use of double payment chains. This allows

implementation of a broker mediated satisfaction guarantee scheme. The pair of payment chains represent the high and low watermarks for the payment order. The low watermark chain represents the amount that the customer has fully committed to pay. The high watermark chain represents partial commitments. The vendor exposure is the difference between the counter values.

MPTP also supports use of multiple payment counters denoting different units of currency. This allows some optimization of processing time through shortening of the payment chains.

MPTP provides protection against double spending through vendor and broker checking of authority identifiers. The size of required Vendor authority identifier matching tables (th _double spending buffer_may be controlled by checking that the authority timestamp is within bounds. An alternative approach would incorporate challenge/response sequence into the session establishment protocol. This could be used to simplify broker double spending prevention measures if constraints were placed on the challenge identifiers. The reduction in vendor resource requirements do not appear to justify an additional round trip delay however.

The mechanism could be modified to use a collection of payment tokens as opposed to a chain. Each token would consist of a the hash of a shared secret which would be revealed to make a payment. This might provide a solution to possible patent difficulties concerning the use of the Lamport hash chain mechanism. It would also permit payments to take place in parallel.

Signature

MPTP permits use of both shared secret and public key based signature schemes. Schneier [Schneier96] describes a wide variety of public key signatures schemes and one way hash functions suitable for constructing Message Authentication Codes (_MACs_). Choice of algorithm, key length etc. is left to the parties involved. It is desirable to minimize the latency introduced in the signing of the initial payment order and also to minimize computational needs of the vendor and broker.

A number of digital signature techniques permit some calculations to be performed offline, i.e. in advance of the message being known [EvenGoMi90]. Such schemes include the Digital Signature Standard [NIST91] and El-Gamal [ElGamal95]. Pre-calculation permits the time taken to generate a signature to be performed outside the human interaction loop and thus appear transparent to the user. Note however that many MPTP applications may chose to perform speculative calculation of an authority in advance of user instructions, unused calculations would simply be discarded. Note however that use of this technique might negatively interact with techniques intended to prevent double spending since a user might delay sending the authority for a significant time.

A recent proposal by Shamir [Shamir95] permits signatures to be validated very rapidly, although at the cost of introducing a small risk that an invalid signature will be accepted. Shamir describes a variation of the Rabin [Rabin79] signature scheme which permits signatures to be "screened" using a test which will detect a fraudulent signature half the time and never reject a valid one. The scheme may be applied repeatedly to provide the desired degree of assurance as to the authenticity of the result.

The signature screening approach has a number of characteristics which are particularly suitable for micropayment schemes. The degree of assurance may be adjusted to reflect the level of risk. Small payments might be accepted with only minimal checking and additional checks performed as the risk increased. another useful property is that the level of protection may be adjusted to reflect server load. This is especially important since it substantially reduces the excess server capacity required to cope with peaks in demand and allows unexpected increases in demand to be dealt with in an orderly manner.

A final signature option is to used a shared secret and keyed digest. This requires the customer and broker to establish a shared secret and for the broker to provide an online verification service. Use of shared secret authentication permits rapid generation of payment authorities. Validation of such authorities requires communication between vendor and broker which may incur a delay.

Certificates and establishment of trust.

Certificates bind a public key to an account number under the public key of the broker. It is assumed that the broker public key is known to all parties. Implementations might require broker public keys to be verified through some additional means.

Each party generates their own public-private key pair locally. The public key certificate is communicated to the broker at account establishment.

The certificate issuance policy may require frequent re-issuance of certificates to enable close control of credit risks or permit certificates to be valid for longer periods of time. It is not necessary for a re-issued certificate to establish a new public key.

Account revocation lists are supported to enable credit risks to be prevented from engaging in further abuse. Separate certificate revocation lists are not supported since compromise of public key certificates may be dealt with through the same mechanism.

Where an account has special attributes concerned with risk management these attributes should be included in and authenticated by the certificate. For example an account might be limited to making payment orders for no more than a certain amount. A broker might chose to guarantee payments up to a certain amount but require

an authorization to guarantee payments for larger amounts. Alternatively a broker might require a vendor to accept the risk regardless of amount and authorization. The following certificate attributes are supported:

IP-Address _mask_, _value_
: Specifies a set of internet addresses for which the certificate is valid. Only payment requests originating from IP addresses which equal the specified value after being logically ANDed with the mask. If more than one IP-Address attribute is specified a single match is sufficient.

Not Guaranteed _amount_
: The broker will not guarantee that payment will be made for amounts exceeding the specified amount.

Guaranteed _amount_
: The broker guarantees payment up to the specified amount without separate authorization.

Authorization-Required _amount_
: Payments above the specified amount require separate authorization to be guaranteed.

Distributed Implementation

The offline nature of PayWord lends itself well to a fully distributed solution. It is not necessary for the Broker to use a single server for all customers.

Online verification of credit-worthiness (e.g. in the symmetric signature scheme) requires a vendor to have access to a server which holds the authentication information. This information may be encoded in the account certificate.

Risk Control

Having described the risk model and mechanism of MPTP we describe the manner in which risk may be controlled.

Credit Liability

If a broker chooses to act as guarantor for a payment a credit liability risk may be incurred. Note that MPTP supports an option for the broker to transfer this risk to the vendor by refusing a guarantee of payment.

In either case a credit liability is incurred. Such liabilities are a familiar consideration in the financial industry. A similar risk is accepted in parts of the publishing industry where newspapers are sold from unattended vending machines which cannot control the number of copies taken by each customer. In certain countries no precautions are taken to prevent a copy being taken without any

payment at all.

Credit Abuse

The problem of credit abuse is linked to but distinct from that of credit liability risks. For example an account might be created in a false name and its authentication information widely published with the intention of permitting general access to charged material for free.

Credit abuse might be discovered through broker tracing of payment patterns to detect sudden increases in payment activity and then terminated through the revocation list mechanism. The case of widespread use of a single connection may be controlled through checking of the certificate IP-Address attribute if specified. If no IP address attribute is specified a vendor might employ code to detect accesses from multiple IP addresses within a suspiciously short interval.

Counterfeiting

MPTP payment orders are vendor specific and digitally signed. Provided the signature scheme is secure it is not possible for a party to construct a payment order without having access to the secret information corresponding to the key.

Unauthorized Withdrawal

Unauthorized withdrawal is not possible without detection by the account holder who may require an audit trail from the broker for each transaction. Note that this requires the broker to maintain a substantial quantity of online logging information.

Purchase Order Modification

In a purchase order modification attack an external party modifies a purchase request in order to cause different goods to be delivered. This risk is not directly addressed in the MPTP scheme although the satisfaction guaranteed policy might be used to protect the customer.

Without authentication of the purchase order there is no method of avoiding this attack. The cost of this authentication might be reduced by establishing a shared key between vendor and customer during the session establishment protocol. Such shared keys might have a lifetime spanning several payment orders.

Double Spending

Payment orders are specific to a particular vendor and carry a unique authority identifier. A broker is required to detect an attempt to deposit the same payment order more than once and act accordingly. In some cases this may mean increasing the amount of

payment authorized.

Failure to Credit Payment

Currently MPTP does not address this risk. A Broker may deliberately deduct a payment amount from the account of one party without making a corresponding credit to another party.

One approach to this problem is to make information concerning bad debts available for scrutiny. A broker might be required to issue a frequent list of bad debts signed under the broker's public key. Such debts might be rendered unlinkable through a use of a one way hash function on the authority identifier. The proportion of bad debts might be concealed through addition of padding. In this way both customer and vendor could ensure that the broker acted in good faith.

Denial of Service

Denial of service is a significant risk, unfortunately it is one that the underlying infrastructure of the Internet does not protect against. Consequently any application protocol level protection against a denial of service attack can at best provide limited protection against this risk.

Use of Shamir's signature screening algorithm substantially reduces the risk of a denial of service attack against a vendor or broker through construction of bogus payment orders.

Repudiation

MPTP payment orders are non-repudiable in the sense that the customer cannot deny having made a payment authorization. This is distinct from the option for a vendor or broker to permit a customer the right to refuse payment after receiving the goods.

Failure to deliver

Failure to deliver may occur for many reasons including vendor fraud. The Internet is an unreliable transport medium and a customer may in good faith offer to buy an article and a vendor in good faith may intend to supply but delivery fail nevertheless. The HTTP protocol in particular does not currently provide for customer acknowledgment of receipt.

One solution to the failure to deliver risk is to permit the customer to refuse payment through the "satisfaction guaranteed" policy described earlier.

Framing

The vendor has the opportunity to frame a customer, albeit at a direct monetary loss to himself. In this scenario a vendor receives

a valid payment chain from a customer but chooses not to deliver the authorization paychain token, instead delivering only the promissory paychain token. The vendor is thus able to frame the customer, albeit at the cost of the payment.

This risk is not currently addressed in MPTP. One approach to addressing this risk would be for the customer to opt to make the payment during account reconciliation.

Message Formats

In this draft we describe the message content without entering into consideration of the corresponding byte streams. We assume that a systematic encoding of these message formats is employed such as the Basic Encoding Rules ASN.1. Choice of encoding rules is left to the working group. It is assumed that the encoding permits payment messages to be transport via standard Internet protocols through simple processing (e.g. BASE-64 encoding).

We define the following data types which are of general use:

```
Identifier : Array [Octet]

Amount : Struct
    value                    Integer
```

```
    currency                 Identifier

Signed [Any] : Struct
    data                     Any
    certificate              AccountCertificate
    signature_algorithm      Identifier
    signature                Identifier

Wrapper [Any] : Struct
    version                  Identifier
    message-id                  Identifier
    data                     Any
```

All messages are transported enveloped using the Wrapper structure which states the protocol version and message identifier.

Account Establishment and Maintenance

The following account options are supported:

Accept Payments Only
: The account will only accept payment.

Refuse Payments
: The account does not accept payments of any type.

Refuse Micropayment
: The account permits payments to be made to it but cannot initiate micropayments. Such accounts may be useful for customers who may wish to be able to transfer money into their account but do not require the ability to accept large numbers of arbitrarily small payments.

Create Account

Account creation requires a binding to be established between an account identifier supplied by the broker, a public key supplied by the potential client and billing information. The exact nature of the billing information is left to implementations.

```
CreateAccountRequest : Struct
    public_key               PublicKey
    identity_binding         ???

AccountFlag : Choice
```

```
    AcceptPaymentsOnly
    RefusePayments
    RefuseMicroPayments
```

It might be appropriate to encrypt the identity binding information.

Reissue Certificate

Depending on broker policy certificates may require frequent reissue. This process may or may not require the establishment of a new public key. Note that this is not a suitable mechanism for dealing with certificate compromise situations.

```
ReissueCertificateRequest : Signed [ReissueCertificateData]

ReissueCertificateData : Struct
    account_id               Identifier
    public_key               PublicKey
```

Delete Account

The delete account message is used to terminate an account.

```
DeleteAccountRequest : Signed [DeleteAccount]

DeleteAccount : Struct
    account_id               Identifier
```


Account Certificate

The account certificate is returned by the broker in response to an account creation request.

```
AccountCertificate : Signed [AccountData]

AccountData : Struct
    account_id               Identifier
    flags                    Set[AccountFlags]
    credit_limit             Amount
    broker_servers           List [BrokerServer]
    attributes                    List [Attribute]
    not_valid_before         Date
```

```
    not_valid_after          Date

Attribute : Choice
    IPAddress
     mask                 Address
        value                Address
    NotGuaranteed
        amount               Amount
    Guaranteed
        amount               Amount
    AuthorizationRequired
        amount               Amount
```

Payment Dataflow

The payment dataflow consists of three phases. First an account authority is created, next a sequence of paywords is transferred, finally a termination message closes the payment session. Payment sessions may span multiple transport sessions.

A payment authority may optionally incorporate a direct payment instruction which does not require confirmation using a pay-chain.

Authority

```
Authority : Signed [AuthorityData]

AuthorityData : Struct
    version                  Identifier
    authority_id             Identifier
    payer_id                 Identifier
    recipient_id             Identifier
    date                     Date
    hash_algorithm           Identifier
    chains                   List [PayChain]
```



```
PayChain : Struct
    flags                    Set[ChainFlag]
    amount                   Amount

ChainFlag : Choice
    Accepted
    Pending

BrokerServer : Struct
```

```
    address                  Array [Octet]
    port                     Array [Octet]
    quality                  Integer
```

It is assumed for the sake of convenience that all pay-chains under a given authority will employ the same hash function.

Charge

```
Charge : Struct
    authority_id             Identifier
    paywords                 List [PayWords]
    terminate                Boolean

PayWord : Struct
    pay_word                 Array [Octet]
    increment                Integer
```

The terminate flag terminates a payments session and informs the vendor that no further payments are to be made.

Vendor-Broker Communications

The collection process permits the vendor to collect payment on paychains

Collection Request

A collection request consists simply of a list of authority, charge pairs.

```
CollectionRequest : Struct
    account_id               Identifier
    collections              List [Collection]

Collection : Struct
    authority                Authority
    charge                   Charge
```


Collection Response

There is no need to respond with an affirmation for every payment. Simply provide a list of the duds.

```
CollectionResponse : Struct
    status                  BrokerStatus
    refusals                List [ChainResponse]

ChainResponse : Struct
    authority_id            Identifier
    reason                  RefusalReason
```

Validation Request

Validation requests are required whenever symmetric key signatures are employed. Validation requests might also be required as a matter of broker policy in certain circumstances, such as purchases for large amounts or to guarantee payments.

```
ValidationRequest : Struct
    vendor_id               Identifier
    authority               Authority
```

Validation Response

```
ValidationResponse : Struct
    broker_id                Identifier
    vendor_id                Identifier
    authority_id             Identifier
    response                 ValidationResponseCode
ValidationResponseCode : Choice
    Authorized
    NotAuthorized
```

Revocation List

```
RevocationList : List [Revocation]

Revocation : Struct
    account_id              Identifier
    reason                  RevocationReason
```

Processing of Data Flows

[This is a placeholder for detailed descriptions of the required processing steps.]

Account Creation

[To Be Specified]

Account Modification, Enquiry, Deletion

[To Be Specified]

Payment Flow

Session Establishment [Customer]

The customer performs the following steps to create an Authority:

1. Calculates PayChains, stores head, may additionally store all or part of PayChain.

2. Creates unique authority identifier. Alternatively the paychain root might be used for this purpose.

3. Fills remaining slots in Authority structure.

4. The authority is sent to the vendor.

Session Establishment [Vendor]

On receipt of an Authority the vendor performs the following steps:

1. The date of the authority is checked to ensure that it is within the vendor determined permitted timeframe.

2. The authority identifier is checked against those in the double spending check buffer.

3. The authority identifier is added to the double spending check buffer. [The vendor may opt to remove expired entries from the double spending check buffer at this time].

4. If public key signatures are used
   the signature of the customer certificate validated.

5. If public key signatures are used:
   The signature of the authority is validated.

6. If the account certificate does not offer the required payment guarantees or symmetric signatures are used:
   A validation request is performed.

7. The Authority is appended to the online file.

Session Establishment with Validation Request [Vendor]
If the vendor determines that an account enquiry is required an account enquiry is created:

1. The account enquiry packet is created

2. The account enquiry is authenticated using a MAC and a shared secret established between vendor and broker.

3. The account enquiry is sent to the broker.

Session Establishment with Validation Request [Broker]

1. A Validation Request is received.

2. The validation request is authenticated

3. The account information corresponding to the customer id is retrieved.

4. A decision is made to accept or reject the authorization.

5. The Validation Response is sent to the Vendor

Session Establishment with Validation Request [Vendor]
On reciept of an account enquiry response a vendor:

1. Checks to see that the response is genuine.

2. Checks to see that the account is authenticated and the required payments guarantees provided.

The Customer may then send a sequence of Paywords which are processed as follows:

Payment Transfer [Customer]
The Customer prepares a Charge message as follows:

1. The Authority information corresponding to the vendor id is retrieved.

2. The Payword(s) corresponding to the desired payment amount is determined.

3. The Charge message is sent to the vendor.

Payment Transfer [Vendor]
The Vendor processes the Charge message as follows:

1. The vendor receives the charge message.

2. The session record is retrieved using the authority-id.

3. The payword is validated using the paychain root.

4. The payword information and increment are updated in the session record.

Collection Flow

[To Be Specified]

Resource and Performance Analysis

The critical features distinguishing a micropayments protocol from a payments protocol are low latency, low processing requirements and low storage requirements.

Latency

Processing of MPTP micropayments should not introduce noticable delay into the user interaction of a well written application program during normal browsing patterns.

Establishment of Payments Session

Establishment of a payment session requires one digital signature to be generated and two signatures to be checked plus the generation of one or more paychains. Paychain generation and part of the signature generation may be performed offline as a background task, reducing the latency of the interation.

Note that in many cases a sophisticated customer application program may perform the entire process of creating an authority on a speculative basis before the user requests a session to be established. This carries no risk since an authority does not allow payment unless accompanied by a valid payword and in any case the authority would not be sent to the vendor unless a session was to be established.

Subsequent Payments

Subsequent payments require only the generation of the next payment token in the chain and its verification. The generation process may be accelerated or avoided entirely through partial or complete caching of the original paychain.

Processing

The most common processing operation are those connected with the payments dataflow itself.

Customer

The customer bears the most substantial processing costs. Establishment requires the creation of a paychain and digital signature. Offline signature techniques and pre-calculation of pay-word chains may often be performed as background tasks while the

processor is idle.

Vendor

The vendor must process two signature verifications per establishment of a payment session and one hash operation per payword transferred, two if double chains are employed.

Hash chain and signature calculations would normally be calculated inline. Use of signature screening might be combined with signature verification to control the inline/offline calculation load.

Broker

The broker must perform one signature verification per collection, plus one hash calculation per payword transferred. It may be possible for a broker to perform probabilistic checking of collection operations, checking only ten percent of a vendors collection request.

All broker calculations may be performed offline.

Storage

Many proposed micropayment schemes offer low processing overhead but require large quantities of data to be kept online for rapid access. Where the frequency of incomming requests is high online access cannot be satisfactorily provided by secondary storage such as disks since head contention becomes the limiting factor. Online storage requirements are thus effectively RAM storage requirements.

Offline storage requirements are unlikely to be a significant factor in the economics of a payments scheme. Many existing servers handle a heavy load of incoming requests while keeping comprehensive log files.

Customer

The customer must track each open session. It may in addition be desirable to store the computed paychains in complete or partial form.

Vendor

The vendor must maintain an online record for each open session. This record is fixed in length consisting of the authority identifier and payer identifier from the authority, and the paychain root or most recent valid pay-word plus the currency unit.

Prevention of double spending requires the maintenance by the vendor of an online record of all payment sessions established within the timestamp validity window. Note however that it may be desirable to place loose limits on validity windows to permit use of speculative

calculation of authorities.

A server satisfying 100,000 micropayment operations per hour of which 10% are session establishment requests would require only 8Mb of online storage for both recording of current sessions and maintenance of the double spending prevention window. Such a server would generate $1000 per hour at a cent per transaction which would be more than enough at present prices to meet the cost of the memory.

Broker

If the symmetric signature option is not provided the broker may perform almost all operations offline in batch. Incoming collection requests from vendors may be pre-processed to optimize access to secondary storage such as disk. Detection of double spending requires a record of all transactions to be available at the time when a record is added. This need not involve the expense of online memory however.

One way round the double spending problem is to give each vendor a counter which must be incremented at each step. It is then only necessary to keep one online storage location per account. Note however that it is undesirable that this token be advertised to the customer since it would reveal the number of purchase requests made to that vendor. Another problem would be enforcing the serialization of the tokens, what would happen if one customer terminated a session much later than one started after it? This would seem to imply that the serialization option would require rapid redemption of the tokens which is itself undesirable.

If the symmetric signature option is provided the registry of shared secrets must be available in primary storage. In most practical schemes this will require the data to be stored in RAM.

Commentary and Further Work

This proposal is considered incomplete and comments are invited. A number of additional considerations which might be explored are noted below.

Benchmarks and statistics

It would be useful to have timings for the various processes involved and more comprehensive estimates of relative costs. Detailed statistics concerning customer browsing patterns would be an advantage. How frequently does a customer change site, what proportion of one off purchases does a customer make?

Risk factors

Statements concerning the relative importance of various risk factors to potential customers, vendors and brokers would be of

assistance.

Inter-Broker Settlement Model

The protocol described requires perfect trust between servers acting as brokers. Compromise of one server compromises all others. The provisions allowing for multiple servers are not designed to permit multiple brokers competing brokers to participate within a single payment system.

In a large scale use more than one organization would offer broker services but payment transfer should be possible nevertheless even if the parties did not have a common broker. Each transaction may thus involve two brokers, in credit card terminology an acquiring (vendor appointed) bank and issuing (customer appointed) bank. In addition the services of a clearing house may also be required. Such a provision is probably essential to the long term acceptability of a scheme.

The chief difficulty in extending the scheme concerns the establishing to what trust relationships may be assumed between brokers and to what degree enforcement mechanisms must be provided for. If brokers are able to establish a high degree of trust the impact upon the protocol is small. If brokers are unable to establish such trust the impact might be large.

As an example of a nave Inter-Broker settlement scheme let us consider extending the certification hierarchy to include one or more broker certification authorities. We assume that inter-broker settlement takes place either through direct exchange of payment orders or employs the services of a clearing house. The only direct impact of these changes as far as the customer and vendor are concerned is the need to authenticate the broker certificate.

The impact on the broker trust relationship is more complex. In particular a customer's broker has the opportunity to commit fraud with negligible probability of detection. On receipt of a payment instruction a customer broker might deduct the amount from the customer's account but report it as bad credit to the vendor's broker.

A more subtle problem concerns the trust relationships between the vendor and the vendor's broker. In the credit card system this relationship is transparent. The identity of the vendor's broker is not revealed either to the customer or even the customer's broker. While this is a significant disadvantage for a scheme intended to use the credit card charging infrastructure as previously noted this need not be the case.

Many of the trust and information sharing issues connected with the introduction of inter-broker settlement may be rendered moot by the nature of the initial acceptance community. In particular the question of which party bears what risk is central to this issue. It

is therefore premature to make a detailed proposal concerning this issue.

Wider Application

MPTP is a general resource management protocol. It might also be used for control of resources such as printer pages [Hallam-Baker 95b], CPU time and similar low unit cost items within an institution. MPTP might also be applied to provide resource constraint enforcement in applications such as interactive multi-player games (e.g. MUDs, MOOs). [Hallam-Baker95a]

Another potential application area is authorization. MPTP might be used to establish generalised and decentralized authorization in a distributed environment in a similar manner to Kerberos.

Security Considerations

This whole document is about security

Patent Rights

A number of companies sell patent rights to public key technology. The legal status of a number of these patents is currently disputed. Until then the standard IETF spiel with a revision to the PKP bit will appear here. Reports have also surfaced that the CAFE consortium may have a patent covering certain uses of S/Key technology with respect to payments applications.

References.

[AccreditedSC93a] _Working Draft: American National Standard X9.30-1993: Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry: Part 2: The Secure Hash Algorithm (SHA)_ Accredited Standards Committee X9 1993,

[BellareEt95]
Internet Keyed Payment Protocols (iKP) Mihir Bellare, Juan A. Garay, Ralf Hauser, Amir Herzberg, Hugo Krawczyk, Michael Steiner, Gene Tsudik, Michael Waidner: In Proceedings of the First USENIX Workshop on Electronic Commerce, New York, July 1995

[ElGamal85]
A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. T. El Gamal. IEEE Trans. Inform. Theory, Vol.31, 1985, pages 469--472

[EvenGoMi90]
Online/offline digital signatures. S. Even, O. Goldreich and Silvio Micali. CRYPTO 89, pages 263-277. Springer-Verlag, 1990, Lecture Notes in Computer Science No. 435.

[Haler94]

The S/Key One-Time password system.. N. M. Haller, In ISOC 1994

[Hallam-Baker95]
Electronic Payment Schemes. P. M. Hallam-Baker.
http://www.w3.org/pub/WWW/Payments/roadmap.html

[Hallam-Baker95a]
Multi User Dungeons P. M. Hallam-Baker

[Hallam-Baker95b]
Why Printing Should be via The Web P. M. Hallam-Baker

[Lamport81]
Password Authentication with Insecure Communication L. Lamport, Communications of the ACM Nov 1981 pages 770-771.

[Manasse95]
_Millicent (electronic microcommerce) _ Mark S. Manasse. 1995.

[NIST91]
Digital Signature Standard (DSS) National Institute for Standards and Technology Federal Register, vol 56 No. 169 August 1991.

[Pedersen9?]Technical report? Torben Pedersen, DAIMI, Aarhus

[Rabin79]
Digital Signatures and Public Key Functions as Intractable as Factorization". M. O. Rabin. MIT Laboratory for Computer Science Technocal Report, MIT/LCS/TR-212 Jan 1979.

[Rivest92c]
RFC 1321, The {MD5} Message-Digest Algorithm R. L. Rivest, Internet Request for Comments, April 1992

[RivestSh95]
_PayWord and MicroMint--Two Simple Micropayment Schemes, R.L. Rivest and A. Shamir. (To Appear.) _

[RivestShAd78]
A Method for Obtaining Digital Signatures and Public-Key Cryptosystems R. L. Rivest, A. Shamir and L. M. Adleman. CACM, Feb 1978 p 120-126.

[Schneier96]
Applied Cryptography (Second Edition). Bruce Schneier, John Wiley & Sons 1996.

[Shamir95]
Fast Signature screening. CRYPTO'95 rump session talk. to appear in RSA Laboratories _Cryptobytes_.

[SteinStBoRo94]
The Green Commerce Model L. H. Stein, E. A. Stefferud, N. S. Borenstein and M. T. Rose, Internet Draft. October, 1994, Work in Progress.

Acknowledgements

Thanks are due to Ron Rivest and Adi Shamir for their suggestion to use a chained hash function. The help and advice of Rohit Khare, Dave Ragget, Jim Gettys and Mark Manasse were of great assistance in producing this proposal.

Contact

The author may be contacted via email at hallam@w3.org

To Do List

* Check over symmetric key mode

* Firm up language, inline/offline poorly explained, establishment of payment session etc. Label each term introduced and check for definition. [would not a tool for this be nice]

* Message id business is poorly explained.

* Should firm up the collection loop semantics. It is not absolutely essential that payments be collected only once. If the server can check against double spending could allow partial and incremental collection

* Consider leakage of data to various parties.

* Protection against double spending by customer currently requires each vendor to maintain an online check of all previous payments by the customer. This is weak protection and could be firmed up substantially.

* Include a challenge response loop to initiate the establishment instruction, thus ensuring that double spending cannot take place?

* Response messages by the broker should be considered somewhat.

* Consider the specific case of payment for software on an hourly basis in detail.

* The issue of maintenance of blacklists by individual merchants requires special attention.

* Should there be a reputation mechanism built into the system so that poor payers suffer a declining credit rating which is advertised in their certificate?

* Need to consider the issue of account enquiries. There should be a mechanism whereby a client can rapidly ascertain that the broker account is correct, establish which funds are cleared etc.

* Should there be a validity interval for payments built into each payment order (cash by date) built into each authority?

* Should there be a do not pay before date in a payment authority?

* Additional references: Kerberos, HTTP, SMTP, MIME.

Expires May 27th 1996