Network Working Group
Request for Comments: 1898
Category: Informational

D. Eastlake 3rd
CyberCash
B. Boesch
CyberCash
S. Crocker
CyberCash
M. Yesil
CyberCash

February 1996

CyberCash Credit Card Protocol Version 0.8

Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Abstract

CyberCash is developing a general payments system for use over the Internet. The structure and communications protocols of version 0.8 are described. This version includes credit card payments only. Additional capabilities are planned for future versions.

This document covers only the current CyberCash system which is one of the few operational systems in the rapidly evolving area of Internet payments. CyberCash is committed to the further development of its system and to cooperation with the Internet Engineering Task Force and other standards organizations.

Acknowledgements

The significant contributions of the following persons (in alphabetic order) to this protocol are gratefully acknowledged:

Bruce Binder, Judith Grass, Alden Hart, Steve Kiser, Steve Klebe, Garry Knox, Tom Lee, Bob Lindenberg, Jim Lum, Bill Melton, Denise Paredes, Prasad Chintamaneni, Fred Silverman, Bruce Wilson, Garland Wong, Wei Wu, Mark Zalewski.

In addition, Jeff Stapleton and Peter Wagner made useful comments on the first version of this memo.

History

For historic purposes, it should be noted that this document was first posted as an Internet draft, and thus made publicly available, on 8 July 1995.

Table of Contents

1. Overall System

CyberCash, Inc. of Reston, Virginia was founded in August of 1994 to partner with financial institutions and providers of goods and services to deliver a safe, convenient and inexpensive system for making payments on the Internet. The CyberCash approach is based on establishing a trusted link between the new world of cyberspace and the traditional banking world. CyberCash serves as a conduit through which payments can be transported quickly, easily and safely between buyers, sellers and their banks. Significantly - much as it is the real world of commerce - the buyer and seller need not have any prior existing relationship.

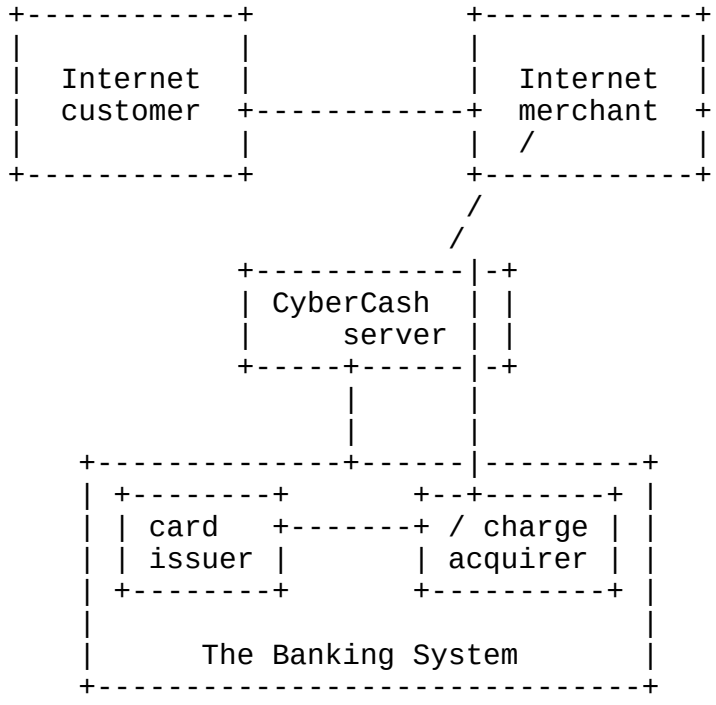
As a neutral third party whose sole concern is ensuring the delivery of payments from one party to another, CyberCash is the linchpin in delivering spontaneous consumer electronic commerce on the Internet.

1.1 System Overview

The CyberCash system will provide several separate payment services on the Internet including credit card and electronic cash. To gain access to CyberCash services, consumers need only a personal computer with a network connection. Similarly, merchants and banks need make only minimal changes to their current operating procedures in order to process CyberCash transactions, enabling them to more quickly integrate safe on-line payments into their existing service offerings. Communications with banks are over existing financial communications networks.

To get started, consumers download free software from CyberCash on the Internet. This software establishes the electronic link between consumers, merchants and their banks as well as between individuals. To make gaining access to the CyberCash system even easier, CyberCash "PAY" buttons may be incorporated into popular on-line service and software graphical user interfaces so that consumers using these products can easily enter the CyberCash system when they are ready to make payments for goods and services. Consumers need not have any prior relationship with CyberCash to use the CyberCash system. They can easily set up their CyberCash persona on-line.

Transactions are automated in that once the consumer enters appropriate information into his own computer, no manual steps are required to process authorization or clearance transactions through the entire system. The consumer need only initiate payment for each transaction by exercising the pay option on an electronic form. Transactions are safe in that they are cryptographicly protected from tampering and modification by eavesdroppers. And they are private in that information about the consumer not relevant to the transaction is not visible to the merchant.



SYSTEM OVERVIEW

1.2 Security Approach

The CyberCash system pays special attention to security issues. It uses encryption technology from the world's leading sources of security technology and is committed over time to employing new security technologies as they emerge.

1.2.1 Authentication and Persona Identity

Authentication of messages is based on Public Key encryption as developed by RSA. The CyberCash Server maintains records of the public key associated with every customer and merchant persona. It is thus able to authenticate any information digitally signed by a customer or merchant regardless of the path the data followed on its way to the server. The corresponding private key, which is needed to create such digital signatures, will be held by the customer or merchant and never revealed to other parties. In customer software, the private key is only stored in an encrypted form protected by a passphrase.

While the true CyberCash identity of a customer or merchant is recognized by their public/private key pair, such keys are too cumbersome (over 100 hex digits) to be remembered or typed by people. So, the user interface utilizes short alphanumeric ID's selected by the user or merchant for purposes of specifying a persona. CyberCash adds check digits to the requested ID to minimize the chance of accidental wrong persona selection. Persona IDUs are essentially public information. Possession of an persona ID without the corresponding private key is of no benefit in the current system.

Individuals or organizations may establish one or more CyberCash customer personas directly with CyberCash. Thus, an individual may have several unrelated CyberCash personas or share a CyberCash persona with other individuals. This approach provides a degree of privacy consistent with Internet presence generally and with cash transactions specifically. However, persona holders who wish to use a credit card for purchases in conjunction with their CyberCash persona must first meet such on-line identification criteria as the card issuing organization requires.

Control over a CyberCash persona is normally available only to an entity that possesses the private key for that persona. However, a special provision is made to associate an emergency close out passphrase with a CyberCash persona. On receipt of the emergency close out passphrase, even if received over insecure channels such as a telephone call or ordinary email, CyberCash will suspend activity for the CyberCash persona. This emergency close-out passphrase can be stored separately from and with somewhat less security than the

private key for the persona since the emergency passphrase can not be used to divert funds to others. This provides some protection against loss or misappropriation of the private key or the passphrase under which the private key in kept encrypted. In the cash system, the emergency close-out passpharase may also transfer the persona balance to a designated bank account.

1.2.2 Privacy

Encryption of messages use the Digital Encryption Standard (DES), commonly used in electronic payment systems today. It is planned to superencrypt (i.e., encrypted more than one level) particularly sensitive information, such as PIN numbers, and handle them so that the plain text readable version never exists in the CyberCash system except momentarily, within special purpose secure cryptographic hardware that is part of the server, before being re-encrypted under another key.

The processing of card charges through the CyberCash system is organized so that the merchant never learns the customerUs credit card number unless the merchantUs bank chooses to release this information to the merchant or it is required for dispute resolution. In addition, the server maintains no permanent storage of card numbers. They are only present while a transaction involving that card is in progress. These practices greatly reduce the chance of card number misappropriation.

1.3 Credit Card Operation

Using the CyberCash system for credit card transactions, once price has been negotiated and the consumer is ready to purchase, the consumer simply clicks on the CyberCash "PAY" button displayed on the merchant interface, which invokes the merchant CyberCash software. The merchant sends the consumer an on-line invoice that includes relevant purchase information which appears on the customerUs screen. (See PR1 message.) The consumer adds his credit card number and other information by simply selecting from a list of credit cards he has registered to his CyberCash persona. All this information is digitally signed by the customer’s CyberCash software, encrypted, and passed, along with a hash code of the invoice as seen by the customer, to the merchant. (See CH1 message.)

Upon receipt, the merchant adds additional authorization information which is then encrypted, electronically signed by the merchant, and sent to the CyberCash Server. (See CM1 & CM2 messages.) The CyberCash Server can authenticate all the signatures and be sure that the customer and merchant agree on the invoice and charge amount. The CyberCash Server then forwards the relevant information to the

acquiring bank along with a request on behalf of the merchant for a specific banking operation such as charge authorization. The bank decrypts and then processes the received data as it would normally process a credit card transaction. The bank's response is returned to the CyberCash Server which returns an electronic receipt to the merchant (see CM6 message) part of which the merchant is expected to forward to the customer (see CH2 message). The transaction is complete.

2. General Message Wrapper Format

Version 0.8 of the external format for the encoding of CyberCash messages is described below. CyberCash messages are stylized documents for the transmission of financial data over the Internet.

While there are numerous schemes for sending information over the Internet (HTTP, SMTP, and others), each is attached to a specific transmission mechanism. Because CyberCash messages will need to travel over each of these media (as well as others) a transmission independent mechanism is needed.

2.1 Message Format

CyberCash messages consist of the following components:

1. Header - defines the start of the CyberCash message and includes version information.

2. Transparent Part - contains information that is not private.

3. Opaque Part(s) - contains the financial information in the message and is both privacy protected as well as tamper protected. An opaque part is not present in some messages. When present, the opaque part usually provides tamper protection for the transparent part.

4. Trailer - defines the end of the CyberCash message and includes a check value to enable the receiver to determine that the message has arrived undamaged. Note: this check value is intended only to detect accidental damage to the message, not deliberate tampering.

   No null characters (zero value) or characters with the eighth bit on are permitted inside a CyberCash message. "Binary" quantities that might have such byte values in them are encoded in base64 as described in RFC 1521.

2.2 Details of Format

The header consists of a single line which looks approximately like this

```
$$-CyberCash-0.8-$$
```

or like this

```
$$-CyberCash-1.2.3-Extra-$$
```

It includes a number of fields separated with the minus character "-"

1. "$$" - the literal string with the initial $ in column 1.

2. "CyberCash" - the literal string (case insensitive)

3. x.y or x.y.z - the version number of the message format. x is the primary version number. y is a subversion number. z, if present, is a subsubversion number.

4. "Extra" - an optional additional alphanumeric string.

5. "$$" - the literal string

Version numbers start at 0.7 and count up. The ".z" is omitted when z is zero. 0.7 and 0.8 are the test and initial shipped version of the credit card system. 0.9 and 1.0 are expected to also incorporate the test and initial shipped versions of the cash facilities as well as improvements to the credit card system.

The "Extra" string is used within secure environments so that one subcomponent can scribble a note to another with minimum overhead. For example, a server firewall could put "HTTP" or "SMTP" here before forwarding the message to the core server within the firewall perimeter.

2.3 Body Parts

The body parts of the message (both transparent and opaque) consist of attribute value pairs in formats that are reminiscent of the standard electronic mail header (RFC822) format. However, there are some differences.

Attribute names start with and are composed predominantly of letters and internal hyphens except that they sometimes end with a hyphen followed by a number. Such a trailing number is used when there is logically an indexed vector of values. Attribute names are

frequently referred to as labels.

If the label ends with a ":", then RFC822 processing is done. While the existence of trailing white space is significant, all leading white space on continuation lines is stripped. Such lines are wrapped at 64 characters in length, excluding any line termination character(s).

However, if the label is terminated with a ";", this indicates a free-form field where new-line characters and any leading white space, after the initial space that indicates a continuation line, is significant. Such lines should not be wrapped except that, to avoid other processing problems, they are forcibly wrapped at 200 characters.

Blank lines are ignored and do not signify a change to a different mode of line handling.

Another way of looking at the above is as follows: after having found an initial $$ start line, you can treat any following lines according to the first character. If it is alphanumeric, it is a new label which should be terminated with a ":" or ";" and indicates a new label-value pair. If it is a white space character, it indicates the continuation of the value for the preceding new label line. (Exactly how the continuation is processed depends on the label termination character.) If it is "$", it should be the end line for the message. If it is #, it is a comment line and should be ignored. Other initial characters are undefined. (As of this date, no software sends CyberCash messages with # lines but they are convenient for commenting messages stored in files.)

2.4 Transparent Part

The transparent part includes any clear-text data associated with the financial transaction as well as information needed by CyberCash and others to decrypt the opaque part(s). It always includes a transaction field which is the transaction number generated by the requester and which is repeated in the response. It always includes a date field that is the local date and time at the requester and is repeated in the response. In all cases other than an initial registration to establish a persona ID, it includes the requester’s persona ID.

On messages bound for the server, there is a "cyberkey:" field that identifies which server public key was used to encrypt the session key.

2.5 Opaque Part

The opaque part consists of a single block of characters encoded using base64 encoding (see RFC 1521). The data in the opaque section is always encrypted before encoding.

The label "opaque" or "merchant-opaque" precedes the opaque part depending on whether the data was encrypted by the client or merchant software.

On messages inbound to the server, the data to be opaqued is DES CBC encrypted under a random transacton key and then that DES key is RSA encrypted under one of the server's public keys. The RSA encrypted DES key appears as the first part of the base64 encoded field and is not broken out as a separate value in the message. The corresponding outbound reply from the server can simply be DES encrypted under this transaction key as there is enough plain text information to identify the transaction and the customer or merchant will have remembered the transaction key from the inbound message.

A signature is not generally necessary in the opaque part of a reply message. Knowledge of the transaction key is adequate authentication. In order for someone to forge the response, they would have to know the server's private key to be able to get at the transaction key. It is assumed that if anyone tampered with the response opaque part, the probability that it would decrypt to something that would parse is insignificant. (Just the fact that the 8th bit has to be off means a chance of 1 in 2**n where there are n characters and that's ignoring the rest of the formatting.) While someone can tamper with the transparent part, this usually either has no effect or means that the client won't find the transaction key, in which case it's just a particular example of denial of service by damaging a message.

2.6 Trailer

The trailer is intended to close the message and provide a definitive and parseable end of the message.

The trailer consists of several fields separated by "-" as in header.

1. "$$" - literal string.

2. "CyberCash" - literal string (case insensitive).

3. "End" - literal string (case insensitive).

4. transmission checksum.

5. "$$" - literal string.

The transmission checksum is the MD5 has of all printable characters in the version number in the start line and those appearing after the second $$ of the start line and before the first $$ of the trailer line as transmitted. Note that all white space is left out of this hash, including any new-lines, spaces, tabs, carriage returns, etc. The exact label terminators actually used (: or ;) are included as would any # comment line. Note that the optional "Extra" string in the $ start line is not included. The idea is to check correct transmission while avoiding sensitivity to gateways or processing that might change the line terminator sequence, convert tabs to spaces, or the like.

2.7 Example Messages

Simple message from a client:

```
$$-CyberCash-0.8-$$
id: DONALD-69
transaction: 918273645
date: 199512250102
cyberkey:CC1001
opaque:
 GpOJvDpLH62z+eZlbVkhZJXtTneZH32Qj4T4IwJqv6kjAeMRZw6nR4f0OhvbTFfPm+GG
 aXmoxyUlwVnFkYcOyTbSOidqrwOjnAwLEVGJ/wa4ciKKI2PsNPA4sThpV2leFp2Vmkm4
 elmZdS0Qe350g6OPrkC7TKpqQKHjzczRRytWbFvE+zSi44wMF/ngzmiVsUCW01FXc8T9
 EB8KjHEzVSRfZDn+lP/c1nTLTwPrQ0DYiN1lGy9nwM1ImXifijHR19LZIHlRXy8=
$$-End-CyberCash-End-jkn38fD3+/DFDF3434mn10==-$$
```

Message from a merchant:

```
$$-CyberCash-a.b.c-extra-$$
merchant-ccid: acme-69
merchant-date: 19951231115959
merchant-transaction: 987654321
label: value

labelx: multiple line
   value...
# comment
# another comment line
label; text with a real
  multi-line
     format !
merchant-cyberkey: CC1001
merchant-opaque:
```

```
 C1Q96lU7n9snKN5nv+1SWpDZumJPJY+QNXGAm3SPgB/dlXlTDHwYJ4HDWKZMat+VIJ8y
 /iomz6/+LgX+Dn0smoAge7W+ESJ6d6Ge3kRAQKVCSpbOVLXF6E7mshlyXgQYmtwIVN2J
 66fJMQpo31ErrdPVdtq6MufynN8rJyJtu8xSNolXlqIYNQy5G2I3XCc6D3UnSErPx1VJ
 6cbwjLuIHHv58Nk+xxt/FyW7yAMwUb9YNcmOj//6Ru0NiOA9P/IiWczDe2mJRK1uqVpC
 sDrWehG/UbFTPD26trlYRnnY
$$-CyberCash-End-kchfiZ5WAUlpk1/v1ogwuQ==-$$
```

3. Signatures and Hashes

Inbound CyberCash request messages normally have a signature, as described below, of all of the messages fields outside of the signature. This signature is transmitted inside the opaque part of the message. It enables the server to authenticate the source of the message.

Messages from a merchant to a client initiating a purchase sequence have fields signed by the merchant. These fields and this signature are included by the client in the opaque part of their card purchase message to the merchant so that, when all is passed on to the server, it can verify that the client saw the information the merchant intended.

More information on CyberCash signatures and the hash codes they are based on, is given below.

3.1 Digital Signatures

Digital signatures are a means of authenticating information. In CyberCash messages, they are calculated by first taking the hash of the data to be authenticated, as described below, and then encoding the hash using an RSA private key.

Anyone possessing the corresponding public key can then decrypt the hash and compare it with the message hash. If they match, then you can be sure that the signature was generated by someone possessing the private key which corresponded with the public key you used and that the message was not tampered with.

In the CyberCash system, clients, merchants, and the server have public-private key pairs. By keeping the private key secret and registering their public key with the server (for a merchant or client) or publishing their public key or keys (for the server), they can provide high quality authentication by signing parts of messages.

An RSA digital signature is approximately the size of the modulus used. For example, if that is 768 bits long, then the binary digital signature would be 768 bits or 96 bytes long and its base 64 encoding would be 128 bytes.

3.2 Hash Codes

The hashes used in CyberCash messages are message digests. That is, a non-invertable fingerprint of a message such that it is computationally infeasible to find an alternate message with the same hash. Thus the relatively small hash can be used to secure a larger message. If you are confident in the authenticity of the hash and are presented with a message which matches the hash, you can be sure it is the original message, at least as regards all aspects that have been included in the hash.

The hash is calculated using the MD5 algorithm (see RFC 1321) on a synthetic message. The synthetic message is composed of the labels and values specified in a list for the particular hash. Since the hash is input order dependent, it is essential that the label-value pairs be assembled in the order specified. In some cases, a range of matching labels is specified. For example, "card*" to match card-number, card-expiration-date, and all other labels starting with "card". In such cases, all existing matching labels are used in ascending alphabetic order by ASCII character code.

If a label is specified in a signature list but is not present in the label-value data on which the hash is being calculated, it is not included in the hash at all. That is, even the label and label terminator are omitted from the synthetic message.

Before being hashed, the text of the synthetic message is processed to remove all "white space" characters. White space characters are defined as any with an ASCII value of 32 (space) or less or 127 (rubout) or greater. Thus all forms of new-line/carriage-return and distinctions such as blank lines, trailing spaces, replacement of a horizontal tab character by multiple spaces, etc., are ignored for hash purposes.

MD5 hashes are 16 bytes long. This means that the base 64 encoding of such a hash will be 24 characters (of which the last two will always be padding equal signs).

4. Specific Message Formats

This section describes the formats of the Verison 0.8 CyberCash messages by example with comments. The reader in assumed to be familiar with terms such as "acquirer", "PAN" (primary account number), etc., defined in ISO 8583, and currency designations as defined in ISO 4217. A few fields not relevant to current operations have been removed to simplify this exposition.

In the following example messages, signatures, hashes, and encrypted sections are fake nonsense text and ids are fictitious.

4.1 Persona Registration and Application Retrieval

The first step in customer use of CyberCash is registering a persona using the customer application. This is done with the R1 message defined below. The CyberCash server responds with the R2 message.

When the customer application learns that it is out of date, it can use the GA1 request message to the server and its GA2 response to download a new signed version of itself.

4.1.1 R1 - registration

Description: This is the initial message sent to create a new CyberCash persona.

```
##############################################################################
Sender: CyberApp
Receiver: CyberServer
##############################################################################
Sample Message:

$$-CyberCash-0.8-$$
transaction: 123123213
date: 19950121100505.nnn
cyberkey: CC1001
opaque:
 FrYOQrD16lEfrvkrqGWkajM1IZOsLbcouB43A4HzIpV3/EBQM5WzkRJGzYPM1r3noBUc
 MJ4zvpG0xlroY1de6DccwO9j/0aAZgDi9bcQWV4PFLjsN604j3qxWdYn9evIGQGbqGjF
 vn1qI1Ckrz/4/eT1oRkBBILbrWsuwTltFd84plvTy+bo5WE3WnhVKsCUJAlkKpXMaX73
 JRPoOEVQ3YEmhmD8itutafqvC90atX7ErkfUGDNqcB9iViRQ7HSvGDnKwaihRyfirkgN
 +lhOg6xSEw2AmYlNiFL5d/Us9eNG8cZM5peTow==
$$-CyberCash-End-kchfiZ5WAUlpk1/v1ogwuQ==-$$

##############################################################################
```

Opaque Key: Generated using CyberCash encrypting public key identified in CyberKey.

```
##############################################################################
Opaque Section Contents:

type: registration
swversion: 0.8win
content-language: en-us
requested-id: MyRequestedCCID
```

```
email: myemail@myemailhost.com
pubkey:
 0VdP1eAUZRrqt3Rlg460Go/TTs4gZYZ+mvI7OlS3l08BVeoms8nELqL1RG1pVYdDrTsX
 E5L+wcGCLEo5+XU5zTKkdRUnGRW4ratrqtcte7e94F+4gkCN06GlzM/Hux94
signature:
 v6JGmxIwRiB6iXUK7XAIiHZRQsZwkbLV0L0OpVEvan9l59hVJ3nia/cZc/r5arkLIYEU
 dw6Uj/R4Z7ZdqO/fZZHldpd9+XPaqNHw/y8Arih6VbwrO5pKerLQfuuPbIom

##############################################################################
signature is of the following fields: transaction, date, cyberkey,
    type, swversion, content-language, requested-id, email, pubkey

##############################################################################
Explanation:
content-language is taken from the MIME header field (see RFC1766)
    and is the language text messages should be generated in.  (only
    en-us implemented at this time.
swversion used to check if client application is old.
```

4.1.2 R2 - registration-response

```
Description: This message gives the success/failure response to R1.

##############################################################################
Sender: CyberServer
Receiver: CyberApp
##############################################################################
Sample Message:

$$-CyberCash-0.8-$$
transaction: 12312313
date: 19950121100505.nnn
opaque:
 r1XfjSQt+KJYUVOGU60r7voFrm55A8fP5DjJZuPzWdPQjGBIu3B6Geya8AlJfHsW11u8
 dIv1yQeeYj/+l9TD1dXW21/1cUDFFK++J2gUMVv8mX1Z6Mi4OU8AfsgoCliwSkWmjSOb
 kE62sAlZTnw998cKzMFp70TSlI3PEBtvIfpLq5lDCNbWidX8vFZV0ENUmMQ9DTP3du9w
 fsFGvz1mvtHLT/Gj8GNQRYKF4xiyx4HYzTkSMhgU5B/QDLPS/SawIJuR86b9X0mwsr0a
 gbGTzECPJTiKkrhxxMG/eymptsVQSLqNaTCx6w==
$$-CyberCash-End-kchfiZ5WAUlpk1/v1ogwuQ==-$$

##############################################################################
Opaque Key: Same as session key for R1 for same Transaction and
    connection (there may be no ID!).

##############################################################################
Opaque Section Contents:
```

```
type: registration-response
server-date: 19950121100506.nnn
requested-id: MyRequestedCCID
response-id: CyberCashHandle
email: myemail@myemailhost.com
response-code: success/failure/etc.
pubkey:
 0VdP1eAUZRrqt3Rlg460Go/TTs4gZYZ+mvI7OlS3l08BVeoms8nELqL1RG1pVYdDrTsX
 E5L+wcGCLEo5+XU5zTKkdRUnGRW4ratrqtcte7e94F+4gkCN06GlzM/Hux94
swseverity: fatal/warning  [absent if ok]
swmessage; Tells CyberApp that it is obsolete.  Display this
 text to the user.  [only present if SWSeverity present]
message;
       Free text of the error/success condition.
       This text is to be displayed to the person
       by the CyberCash application...

       In general this includes: duplicate-id, bad-signature,
       or ill-formed-registration

##########################################################################
Signature is of the following fields: no-signature

##########################################################################
Explanation:
responseid is used to suggest a unique ID if the failure was due
    to the requested ID being already in use... If the reason for
    failure was not due to duplicate ID then this field may be
    omitted.
responseid gives the actual ID with check characters appended if
    success.
swseverity can warn user of old client application or indicate
    failure due to old or known buggy version.
```

4.1.3 GA1 - get-application

Description: Used by CyberApp to get an updated version.

```
##########################################################################
Sender: CyberApp
Receiver: CyberServer
##########################################################################
Sample Message:

$$-CyberCash-0.8-$$
transaction: 123123213
date: 19950121100505.nnn
```

```
cyberkey: CC1001
opaque:
 VHMS611wGkUmR6bKoI+ODoSbl7L5PKtEo6aM88LCidqN+H/8B4xM3LxdwUiLn7rMPkZi
 xOGb+5d1lRV7WeTp21QYlqJr8emc6FAnGd5c0csPmcnEpTFh9xZDJaStarxxmSEwm2mw
 l2VjEUODH6321vjoMAOFQWn7ER0o
$$-CyberCash-End-0QXqLlNxrn4GNQPPk9AO1Q==-$$

#########################################################################
Opaque Key: Generated using CyberCash encrypting public key identified
   in CyberKey.

#########################################################################
Opaque Section Contents:

type: get-application
swversion: 0.8win

#########################################################################
Signature is of the following fields: no signature

#########################################################################
Explanation:
There may not be a customer persona so there is no ID.  There
    may not be a customer public/private key pair so there is
    no signature.  The swversion is mandatory so the server can
    tell what to return.
```

4.1.4 GA2 - get-application-response

```
Description: Return success and URL of up to date copy of CyberApp
    or failure.

#########################################################################
Sender: CyberServer
Receiver: CyberApp
#########################################################################
Sample Message:

$$-CyberCash-0.8-$$
transaction: 12312313
date: 19950110102333.nnn
opaque:
 EDD+b9wAfje5f7vscnNTJPkn1Wdi7uG3mHi8MrzLyFC0dj7e0JRjZ2PmjDHuR81kbhqb
 nX/w4uvsoPgwM5UJEW0Rb9pbB39mUFBDLPVgsNwALySeQGso0KyOjMxNs1mSukHdOmDV
 4uZR4HLRRfEhMdX4WmG/2+sbewTYaCMx4tn/+MNDZlJ89Letbz5kupr0ZekQlPix+pJs
 rHzP5YqaMnk5iRBHvwKb5MaxKXGOOef5ms8M5W8lI2d0XPecH4xNBn8BMAJ6iSkZmszo
 QfDeWgga48g2tqlA6ifZGp7daDR81lumtGMCvg==
```

```
$$-CyberCash-End-0QXqLlNxrn4GNQPPk9AO1Q==-$$


##############################################################################
Opaque Key: session key from GA1

##############################################################################
Opaque Section Contents:

type: get-application-response
server-date: 19950110102334.nnn
response-code: success/failure/etc.
message; Text message to be displayed to the user providing more
    information on the success/failure.
swversion: 0.8win
application-url: http://foo.cybercash.com/server/0.8WIN.EXE
application-hash: lSLzs/vFQ0BXfU98LZNWhQ==

##############################################################################
Signature: none.

##############################################################################
Explanation:
application-hash is the MD5 of the binary of the application.
application-url & application-hash omitted on failure.
swversion is the version being transmitted to the customer.
```

4.2 Binding Credit Cards

The CyberCash system is design to give the card issuing organization control over whether a card may be used via the CyberCash system. The customer, after having registered a persona with CyberCash as described above, can then bind each credit card they wish to use to their CyberCash persona. This is done via the BC1 message from the customer to their CyberCash server and the BC4 response from the server.

4.2.1 BC1 - bind-credit-card

Description: This is the initial message in the process of binding a credit card to a CyberCash persona.

```
##############################################################################
Sender: CyberApp
Receiver: CyberServer
##############################################################################
Sample Message:
```

```
$$-CyberCash-0.8-$$
id: MyCyberCashID
date: 19950121100505.nnn
transaction: 12312314
cyberkey: CC1001
opaque:
 EDD+b9wAfje5f7vscnNTJPkn1Wdi7uG3mHi8MrzLyFC0dj7e0JRjZ2PmjDHuR81kbhqb
 nX/w4uvsoPgwM5UJEW0Rb9pbB39mUFBDLPVgsNwALySeQGso0KyOjMxNs1mSukHdOmDV
 4uZR4HLRRfEhMdX4WmG/2+sbewTYaCMx4tn/+MNDZlJ89Letbz5kupr0ZekQlPix+pJs
 rHzP5YqaMnk5iRBHvwKb5MaxKXGOOef5ms8M5W8lI2d0XPecH4xNBn8BMAJ6iSkZmszo
 QfDeWgga48g2tqlA6ifZGp7daDR81lumtGMCvg==
$$-CyberCash-End-kchfiZ5WAUlpk1/v1ogwuQ==-$$

#########################################################################
Opaque Key: generated from CyberCash encryption key identified in
    CyberKey

#########################################################################
Opaque Section Contents:

type: bind-credit-card
swversion: 0.8win
card-number: 1234567887654321
card-type: mastercard
card-salt: 46735210
card-expiration-date: 05/99
card-name: John Q. Public
card-street:
card-city:
card-state:
card-postal-code:
card-country:
signature:
 tX3odBF2xPHqvhN4KVQZZBIXDveNi0eWA7717DNfcyqh2TpXqgCxlDjcKqdJXgsNLkY7
 GkyuDyTF/m3SZif64giCLjJRKg0I6mqI1k/Dcm58D9hKCUttz4rFWRqhlFaj

#########################################################################
signature is of the following fields: id, date, transaction,
    cyberkey, type, swversion, card-number, card-salt,
    card-expiration-date, card-name, card-street, card-city,
    card-state, card-postal-code, card-country

#########################################################################
Explanation:
salt is needed so that the hash stored at the server is less
    informative.  Server just remembers the "prefix" of the card
    number and the hash of the combined card number and salt. If it
    just hashed the card number, it would be recoverable with modest
```

effort by trying to hash all plausible numbers. We don't want to store the card numbers on the server because it would make the server files too valuable to bad guys.

4.2.2 BC4 - bind-credit-card-response

Description: Indicates that the process of binding a credit card terminated. Returns success or failure.

```
##########################################################################
Sender: CyberServer
Receiver: CyberApp
##########################################################################
Sample Message:

$$-CyberCash-0.8-$$
id: mycybercashid
transaction: 12312314
date: 19950121100505.nnn
opaque:
 EDD+b9wAfje5f7vscnNTJPkn1Wdi7uG3mHi8MrzLyFC0dj7e0JRjZ2PmjDHuR81kbhqb
 nX/w4uvsoPgwM5UJEW0Rb9pbB39mUFBDLPVgsNwALySeQGso0KyOjMxNs1mSukHdOmDV
 4uZR4HLRRfEhMdX4WmG/2+sbewTYaCMx4tn/+MNDZlJ89Letbz5kupr0ZekQlPix+pJs
 rHzP5YqaMnk5iRBHvwKb5MaxKXGOOef5ms8M5W8lI2d0XPecH4xNBn8BMAJ6iSkZmszo
 QfDeWgga48g2tqlA6ifZGp7daDR81lumtGMCvg==
$$-CyberCash-End-kchfiZ5WAUlpk1/v1ogwuQ==-$$

##########################################################################
Opaque Key: Session key from BC1 with same Transaction and ID

##########################################################################
Opaque Section Contents:

type: bind-credit-card-response
server-date: 19950121100506.nnn
swseverity: fatal/warning  [absent if ok]
swmessage; message about obsoleteness of customer software
    to be shown to the customer.  [only present if SWSeverity present]
response-code: success/failure/etc.
card-number: 1234567887654321
card-type: visa
card-salt: 47562310
card-expiration-date: 01/99
card*: [other card* lines to also be given in CH.1 message]
message; Plain text for the user
    can be multiple lines
```

```
   ##########################################################################
   Signature is of the following fields: no-signature

   ##########################################################################
   Explanation: All the card* lines can be saved as a blob to be
       submitted in CH.1.  card-expiration-date, card-number, card-salt,
       and card-type should always be present.
   Depending on reason for failure, not all fields may be present.
```

4.3 Customer Credit Card Purchasing Messages

In general, CyberCash involvement in the credit card purchasing cycle starts after the user has determined what they are buying. When they click on the CyberCash payment button, a PR1 message is sent by the merchant to the customer as the body of a message of MIME type application/cybercash.

If the customer wishes to proceed, they respond to the merchant with a CH1. The merchant responds with a CH2 but between the receipt of the CH1 and issuance of the CH2, the merchant usually communicates with the CyberCash server via the CM* messages.

4.3.1 PR1 - payment-request

Description: This message is the first message that is defined by CyberCash in the purchase-from-a-merchant process. The shopping has completed. Now we are at the point of paying for the purchases.

```
   ##########################################################################
   Sender: MerchantApp
   Receiver: CyberApp
   ##########################################################################
   Sample Message:

   $$-CyberCash-0.8-$$
   type: payment-request
   merchant-ccid: ACME-012
   merchant-order-id: 1231-3424-234242
   merchant-date: 19950121100505.nnn
   note;
     ACME Products

     Purchase of 4 pairs "Rocket Shoes" at $39.95 ea.
     Shipping and handling $5.00
```

```
     Total Price: 164.80

     Ship to:
          Wily Coyote
          1234 South St.
          Somewhere, VA 12345
   merchant-amount: usd 164.80
   accepts: visa:CC001, master:CC001,amex:CC001,JCPenny:VK005,macy:VK006
   url-pay-to: http://www.ACME.com/CybercashPayment
   url-success: http://www.ACME.com/ordersuccess
   url-fail: http://www.ACME.com/orderfail
   merchant-signed-hash:
    a/0meaMHRinNVd8nq/fKsYg5AfTZZUCX0S3gkjAhZTmcrkp6RZvppmDd/P7lboFLFDBh
    Ec0oIyxWeHfArb3OtkgXxJ7qe0Gmm/87jG5ClGnpBnw0dY7qcJ6XoGB6WGnD
   $$-CyberCash-End-lSLzs/vFQ0BXfU98LZNWhQ==-$$

   ##########################################################################
   Opaque Key: no opaque section

   ##########################################################################
   Opaque Section Contents: no opaque section

   ##########################################################################
   merchant-signed-hash is the signature under the merchant's
       private key of the hash of the following fields: type,
       merchant-ccid, merchant-order-id, date, note, merchant-amount,
       accepts, url-pay-to, url-success, url-fail

   ##########################################################################
   Explanation:
   This message is signed by the merchant but the customer cannot
       directly verify this signature. When the payment is made, the
       Customer includes the signature with the hash (derived by the
       customer directly) in the payment. If these do not match, the
       CyberCash will not perform the payment function.
   accepts: The client software will only recognized single word card
   name in the accepts field of PR1. For example,
     MasterCard
     AmericanExpress
   are recognized where as
     Master card
     American express
   are not recognized. MasterCard and masterCard are both
   recognized as master card.
   Card type followed by key designator.  For main line credit cards,
       this will be a CC*.  Client can use or ignore the * number as
       it chooses.  For proprietary card, this will be VK* where * is
       the CheckFree key to use (1 based).  Cards separated by comma,
```

key designator follows card type and colon.
url-pay-to is where the CH1 should be sent. url-fail and url-success are where the browser should look after failure or success.

4.3.2 CH1 - credit-card-payment

Description: This message represents the presentation of a "credit card for payment".

```
##########################################################################
Sender: CyberApp
Receiver: MerchantApp
##########################################################################
Sample Message:

$$-CyberCash-0.8-$$
type: card-payment
id: myCyberCashID
order-id: 1231-3424-234242
merchant-ccid: ACME-012
transaction: 78784567
date: 19950121100505.nnn
pr-hash: c77VU/1umPKH2kpMR2QVKg==
pr-signed-hash:
 a/0meaMHRinNVd8nq/fKsYg5AfTZZUCX0S3gkjAhZTmcrkp6RZvppmDd/P7lboFLFDBh
 Ec0oIyxWeHfArb3OtkgXxJ7qe0Gmm/87jG5ClGnpBnw0dY7qcJ6XoGB6WGnD
cyberkey: CC1001
opaque:
 iff/tPf99+Tm5P7s3d61jOWK94nq9/+1jOWK9+vr9+b+94n3tYzmiveJ9/+09/334ubg
 3rWM5Ir3ier3/7WM5Ir36+v35v73ife1jOWK94n3/7T3/ffm5uD+7N339/f39/eq3ff3
 9/eFiJK5tLizsoeSmpW7uLS8/7iio7Wisfv38biio7uyufv3tfv35uH+7N3d9/exuKX3
 5+z3vuu4oqO7srnsvvz8/venoqO0v7al/7iio7WisYy+iv7s3ff3p6KjtL+2pf/wi7nw
 3ard3Q==
$$-CyberCash-End-7Tm/djB05pLIw3JAyy5E7A==-$$

##########################################################################
Opaque Key: Created using CyberCash encrypting public key in CyberKey.

##########################################################################
Opaque Section Contents:
swversion: 0.8win
amount: usd 10.00
card*: [from successful BC4 (includes card-expiration-date,
    card-number, card-type, and card-salt)]
signature:
 meO38aULnoP09VhTS2E56tnuZBRRlGfbwqaleZ9zNnv7YjExJKBFxuaqYTUDEj427HHh
```

```
 mm9BVmHRwCq6+8ylZXixGHI1I9A/ufAMrpqMIi6DS3PRlc8WC3CCWoAHyAqr

##########################################################################
signature is under client private key of the following fields:
    type, id, order-id, merchant-ccid, transaction, date,
    pr-hash, pr-signed-hash, cyberkey, swversion, amount,
    card*

##########################################################################
Explanation:
The pr-signed-hash field is the same as the merchant-signed-hash in
    the PR1 message but has a different name for historic reasons.
```

4.3.3 CH2 - charge-card-response

```
Description: Return to customer from a CH1 attempt to pay via credit
    card.  Indicates success/failure.

##########################################################################
Sender: MerchantApp
Receiver: CyberApp
##########################################################################
Sample Message:

$$-CyberCash-0.8-$$
type: charge-card-response
merchant-ccid: ACME-012
id: myCyberCashID
transaction: 78784567
date: 1995121100500.nnn
merchant-date: 19950121100505.nnn
merchant-response-code: failure/success/etc.
pr-hash: 7Tm/djB05pLIw3JAyy5E7A==
pr-signed-hash:
 a/0meaMHRinNVd8nq/fKsYg5AfTZZUCX0S3gkjAhZTmcrkp6RZvppmDd/P7lboFLFDBh
 Ec0oIyxWeHfArb3OtkgXxJ7qe0Gmm/87jG5ClGnpBnw0dY7qcJ6XoGB6WGnD
merchant-message; This is a message to display to the user from the
    merchant. Can be multiple lines...  Is not secure.
opaque:  [might not be present, see explanation]
 EDD+b9wAfje5f7vscnNTJPkn1Wdi7uG3mHi8MrzLyFC0dj7e0JRjZ2PmjDHuR81kbhqb
 nX/w4uvsoPgwM5UJEW0Rb9pbB39mUFBDLPVgsNwALySeQGso0KyOjMxNs1mSukHdOmDV
 4uZR4HLRRfEhMdX4WmG/2+sbewTYaCMx4tn/+MNDZlJ89Letbz5kupr0ZekQlPix+pJs
 rHzP5YqaMnk5iRBHvwKb5MaxKXGOOef5ms8M5W8lI2d0XPecH4xNBn8BMAJ6iSkZmszo
 QfDeWgga48g2tqlA6ifZGp7daDR81lumtGMCvg==
$$-CyberCash-End-7Tm/djB05pLIw3JAyy5E7A==-$$

##########################################################################
```

Opaque Key: Same customer session key from CH1 passed through CM1 for ID and Transaction

```
##########################################################################
Opaque Section Contents (from CM.6):

server-date: 19950121100706.nnn
amount: usd 10.00
order-id: 1231-3424-234242
card*:  [from successful BC4]
response-code: failure/success/etc.
swseverity: fatal/warning
swmessage; Tells CyberApp that it is obsolete.  Display this
 text to the user.  [only present if SWSeverity present]
message;
       Free text of the error/success condition.
       This text is to be displayed to the customer
       by the CyberCash application...

##########################################################################
Signature is of the following fields: no signature

##########################################################################
```

Explanation:

Opaque section optional because the CH1 to the merchant can fail due to bad order-id, date, wrong merchant-ccid, etc., etc. So the server may not be involved at all in which case there is no mechanism for generating a secure opaque section. (It could even be that merchant attempt to contact the server times out.)

If transaction makes it through server (via CM*) then Response-Code at top level should mirror response-code to merchant from server. (Hopefully the same as the response-code to customer from server but the merchant can't tell that.)

Note that there can be two messages, one from merchant and one from the server.

4.4 Merchant Credit Card Purchasing Messages

The merchant presents credit card purchases, makes adjustments, and the like via the CM* series. In general, the credit card cycle is one of getting authorization for a purchase, then capturing the purchase in a batch for clearance, then performing the clearance. It is also possible to void a capture (i.e., remove an item from a batch), and process credits (returns). (See section 5.1.)

Authorizations always come from an acquirer via the response to a CM1 or CM2 message. If capture is being performed by the acquirer or some entity between the CyberCash server and the acquirer, this is done via a CM3 or CM2 message depending on the arrangement between the merchant and the entity doing the capture. Returns (credits) are handled via message CM5. Message CM4 is provided for voiding a capture or return before the batch is cleared. CM6 is the message format used for responses to all the other CM* messages.

An MM* series has also been implemented for purely merchant originated CyberCash charges as described in section 3.4.7

Current credit card dispute resolution systems assume that the merchant knows the card number. Thus, to work with these systems, special bypass messages have been set up that allow the merchant to obtain, for a particular transaction, the information that CyberCash otherwise goes to lengths to hide from the merchant. See sections 3.4.8 and 3.4.9. This makes the obtaining os such information by the merchant an auditable event.

Many present day merchants operate in a "terminal capture" mode where the authorizations are captured by the merchant and the merchant later submits the settlement batch. Messages have been defined and are being implemented so that such merchant captured batches can be submitted via CyberCash.

4.4.1 CM1 - auth-only

Description: This message is used by the merchant to perform an authorization operation on the credit card sent in by the customer.

```
##############################################################################
Sender: MerchantApp
Receiver: CyberServer
##############################################################################
Sample Message:

$$-CyberCash-0.8-$$
merchant-ccid: ACME-69
merchant-transaction: 123123
merchant-date: 19950121100705.nnn
merchant-cyberkey: CC1001
cyberkey: CC1001
opaque:
 EDD+b9wAfje5f7vscnNTJPkn1Wdi7uG3mHi8MrzLyFC0dj7e0JRjZ2PmjDHuR81kbhqb
 nX/w4uvsoPgwM5UJEW0Rb9pbB39mUFBDLPVgsNwALySeQGso0KyOjMxNs1mSukHdOmDV
```

```
    4uZR4HLRRfEhMdX4WmG/2+sbewTYaCMx4tn/+MNDZlJ89Letbz5kupr0ZekQlPix+pJs
    rHzP5YqaMnk5iRBHvwKb5MaxKXGOOef5ms8M5W8lI2d0XPecH4xNBn8BMAJ6iSkZmszo
    QfDeWgga48g2tqlA6ifZGp7daDR81lumtGMCvg==
   merchant-opaque:
    6BVEfSlgVCoGh1/0R+g1C143MaA6QLvKpEgde86WWGJWx45bMUZvaAu4LVeqWoYCqSGf
    aWKUF7awol0h1i1jtgieyAcXB8ikvRJIsupSAwsRMyoNlekR6tucvfv/622JY7+n7nGO
    dGbMzP0GJImh2DmdPaceAxyOB/xOftf6ko0nndnvB+/y2mFjdUGLtFQP/+3bTpZttZXj
    j7RO1khe1UrAIk2TGQJmNw+ltsu0f42MgsxB8Q31vjPtoiPi5LEmD0Y4jlpJ7Jg2Ub84
    F9vJhYpmzNkdiJUe83Hvo/xfJRbhafJpXFEsUZwQK0jU1ksU6CQd2+CPBB+6MxtsHoxJ
    mjD6ickhd+SQZhbRCNerlTiQGhuL4wUAxzGh8aHk2oXjoMpVzWw2EImPu5QaPEc36xgr
    mNz8vCovDiuy3tZ42IGArxBweasLPLCbm0Y=
   $$-CyberCash-End-7Tm/djB05pLIw3JAyy5E7A==-$$

   ##########################################################################
   Merchant-Opaque Section Contents:

   type: auth-only
   order-id: 12313424234242
   merchant-amount: usd 10.00
   pr-hash: 7Tm/djB05pLIw3JAyy5E7A==
   pr-signed-hash:
    a/0meaMHRinNVd8nq/fKsYg5AfTZZUCX0S3gkjAhZTmcrkp6RZvppmDd/P7lboFLFDBh
    Ec0oIyxWeHfArb3OtkgXxJ7qe0Gmm/87jG5ClGnpBnw0dY7qcJ6XoGB6WGnD
   id: myCyberCashID
   transaction: 78784567
   date: 19950121100505.nnn
   merchant-signature:
    v4qZMe2d7mUXztVdC3ZPMmMgYHlBA7bhR96LSehKP15ylqR/1KwwbBAX8CEqns55UIYY
    GGMwPMGoF+GDPM7GlC6fReQ5wyvV1PnETSVO9/LAyRz0zzRYuyVueOjWDlr5

   ##########################################################################
   merchant-opaque key is generated from the CyberCash encrypting public
        key identified in merchant-cyberkey.

   Customer opaque section (Opaque) - see CH1.

   ##########################################################################
   Opaque Section Contents & Signature:  (exactly as in CH1)

   swversion: 0.8win
   amount: usd 10.00
   card*: [from successful BC4 (includes card-expiration-date,
       card-number, and card-salt)]
   signature:
    48SBKUfojyC9FDKCwdCYNvucgiDxYO9erZW4QndIXZRyheTHXH8OeIhwUkyLmgQSD/UK
    +IX9035/jUkdNPOxUQq9y/beHS1HU9Fe0wlzfXYRtnjlqvQX+yUfQ4T7eNEs

   ##########################################################################
```

merchant-signature is on the following fields: merchant-ccid, merchant-transaction, merchant-date, merchant-cyberkey, type, order-id, merchant-amount, pr-hash, pr-signed-hash, id, transaction, date, cyberkey

Customer Signature: see CH1

##
Explanation:
The merchant signature ensures integrity of the majority of the message. validation of the customer signature ensures that the customer opaque part was not tampered or replaced.

4.4.2 CM2 - auth-capture

Description: Do authorization and actually enters charge for clearance. Message just like CM1 except for different type.

##
Sender: MerchantApp
Receiver: CyberServer
##
Sample Message:

[exactly the same as CM1 except

type: auth-capture

]

4.4.3 CM3 - post-auth-capture

Description: Captures a charge previously authorized. Message is the same as CM1 except that it also has an authorization-code field (which is also included in the signature) and the type is different.

##
Sender: MerchantApp
Receiver: CyberServer
##
Sample Message:

$$-CyberCash-0.8-$$
merchant-ccid: ACME-012

```
   merchant-transaction: 123123
   merchant-date: 19950121100705.nnn
   merchant-cyberkey: CC1001
   cyberkey: CC1001
   opaque:
    EDD+b9wAfje5f7vscnNTJPkn1Wdi7uG3mHi8MrzLyFC0dj7e0JRjZ2PmjDHuR81kbhqb
    nX/w4uvsoPgwM5UJEW0Rb9pbB39mUFBDLPVgsNwALySeQGso0KyOjMxNs1mSukHdOmDV
    4uZR4HLRRfEhMdX4WmG/2+sbewTYaCMx4tn/+MNDZlJ89Letbz5kupr0ZekQlPix+pJs
    rHzP5YqaMnk5iRBHvwKb5MaxKXGOOef5ms8M5W8lI2d0XPecH4xNBn8BMAJ6iSkZmszo
    QfDeWgga48g2tqlA6ifZGp7daDR81lumtGMCvg==
   merchant-opaque:
    6BVEfSlgVCoGh1/0R+g1C143MaA6QLvKpEgde86WWGJWx45bMUZvaAu4LVeqWoYCqSGf
    aWKUF7awol0h1i1jtgieyAcXB8ikvRJIsupSAwsRMyoNlekR6tucvfv/622JY7+n7nGO
    dGbMzP0GJImh2DmdPaceAxyOB/xOftf6ko0nndnvB+/y2mFjdUGLtFQP/+3bTpZttZXj
    j7RO1khe1UrAIk2TGQJmNw+ltsu0f42MgsxB8Q31vjPtoiPi5LEmD0Y4jlpJ7Jg2Ub84
    F9vJhYpmzNkdiJUe83Hvo/xfJRbhafJpXFEsUZwQK0jU1ksU6CQd2+CPBB+6MxtsHoxJ
    mjD6ickhd+SQZhbRCNerlTiQGhuL4wUAxzGh8aHk2oXjoMpVzWw2EImPu5QaPEc36xgr
    mNz8vCovDiuy3tZ42IGArxBweasLPLCbm0Y=
   $$-CyberCash-End-7Tm/djB05pLIw3JAyy5E7A==-$$

   ##########################################################################
   Merchant-Opaque Section Contents:

   type: post-auth-capture
   authorization-code: a12323
   order-id: 1231-3424-234242
   merchant-amount: usd 10.00
   pr-hash: 7Tm/djB05pLIw3JAyy5E7A==
   pr-signed-hash:
    a/0meaMHRinNVd8nq/fKsYg5AfTZZUCX0S3gkjAhZTmcrkp6RZvppmDd/P7lboFLFDBh
    Ec0oIyxWeHfArb3OtkgXxJ7qe0Gmm/87jG5ClGnpBnw0dY7qcJ6XoGB6WGnD
   id: myCyberCashID
   transaction: 78784567
   date: 19950121100505.nnn
   merchant-signature:
    vxyEF1ZHn5Rgmtms3H3t/+UB6RAvZQA1AdddjvlS0H75N1x83FyJuh8V9Ok6t4EUQQZ6
    Mnptzc6phJi3Ar0s0oumELsdc8upJdXpNpJV021PGJXfDKfHP0heJIWLodXr

   ##########################################################################
   merchant-opaque key is generated from the CyberCash encrypting public
        key identified in merchant-cyberkey.

   Customer opaque section (Opaque) - see CH1.

   ##########################################################################
   Opaque Section Contents & Signature:  (exactly as in CH1)

   swversion: 0.8win
```

```
amount: usd 10.00
card*: [from successful BC4 (includes card-salt, card-number,
    and card-expiration)]
signature:
 48SBKUfojyC9FDKCwdCYNvucgiDxYO9erZW4QndIXZRyheTHXH8OeIhwUkyLmgQSD/UK
 +IX9035/jUkdNPOxUQq9y/beHS1HU9Fe0wlzfXYRtnjlqvQX+yUfQ4T7eNEs

######################################################################
merchant-signature is on the following fields: merchant-ccid,
    merchant-transaction, merchant-date, merchant-cyberkey, type,
    authorization-code, order-id, merchant-amount, pr-hash,
    pr-signed-hash, id, transaction, date, cyberkey

######################################################################
Explanation:
The merchant signature ensures integrity of the majority of the
    message validation of the customer signature ensures that the
    customer opaque part was not tampered or replaced.
```

4.4.4 CM4 - void

```
Description: Voids out a charge/return if received before
    clearance.  Message is the same as CM1 except that it also has
    a retrieval-reference-number field (which is also included in the
    signature) and the type is different.

######################################################################
Sender: MerchantApp
Receiver: CyberServer
######################################################################
Sample Message:

$$-CyberCash-0.8-$$
merchant-ccid: ACME-012
merchant-transaction: 123123
merchant-date: 19950121100705.nnn
merchant-cyberkey: CC1001
cyberkey: CC1001
opaque:
 EDD+b9wAfje5f7vscnNTJPkn1Wdi7uG3mHi8MrzLyFC0dj7e0JRjZ2PmjDHuR81kbhqb
 nX/w4uvsoPgwM5UJEW0Rb9pbB39mUFBDLPVgsNwALySeQGso0KyOjMxNs1mSukHdOmDV
 4uZR4HLRRfEhMdX4WmG/2+sbewTYaCMx4tn/+MNDZlJ89Letbz5kupr0ZekQlPix+pJs
 rHzP5YqaMnk5iRBHvwKb5MaxKXGOOef5ms8M5W8lI2d0XPecH4xNBn8BMAJ6iSkZmszo
 QfDeWgga48g2tqlA6ifZGp7daDR81lumtGMCvg==
merchant-opaque:
 6BVEfSlgVCoGh1/0R+g1C143MaA6QLvKpEgde86WWGJWx45bMUZvaAu4LVeqWoYCqSGf
 aWKUF7awol0h1i1jtgieyAcXB8ikvRJIsupSAwsRMyoNlekR6tucvfv/622JY7+n7nGO
```

```
    dGbMzP0GJImh2DmdPaceAxyOB/xOftf6ko0nndnvB+/y2mFjdUGLtFQP/+3bTpZttZXj
    j7RO1khe1UrAIk2TGQJmNw+ltsu0f42MgsxB8Q31vjPtoiPi5LEmD0Y4jlpJ7Jg2Ub84
    F9vJhYpmzNkdiJUe83Hvo/xfJRbhafJpXFEsUZwQK0jU1ksU6CQd2+CPBB+6MxtsHoxJ
    mjD6ickhd+SQZhbRCNerlTiQGhuL4wUAxzGh8aHk2oXjoMpVzWw2EImPu5QaPEc36xgr
    mNz8vCovDiuy3tZ42IGArxBweasLPLCbm0Y=
  $$-CyberCash-End-7Tm/djB05pLIw3JAyy5E7A==-$$

  ##########################################################################
  Merchant-Opaque Section Contents:

  type: void
  retrieval-reference-number: 432112344321
  order-id: 1231-3424-234242
  merchant-amount: usd 10.00
  pr-hash: WATCQuH2q17lRuoxD78YBg==
  pr-signed-hash:
    8zqw0ipqtLtte0tBz5/5VPNJPPonfTwkfZPbtuk5lqMykKDvThhO0ycrfT7eXrn/hLUC
    kXoSctahEVdw1KBJbp0EVr1zVzcN9Aa7m2fJgxNfiisTgIRW+PMaa78rn+Ov
  id: myCyberCashID
  transaction: 78784567
  date: 19950121100505.nnn
  Merchant-Signature: lkjladjslkjflsakjflkjsdljflsakjflkjsdljflsakjflkj
      flsakjflkjsdljflsakjflkjsdljflsajflksdjflksdjflsdjssf=

  ##########################################################################
  Merchant-Opaque key is generated from the CyberCash encrypting public
       key identified in Merchant-CyberKey.

  Customer opaque section (Opaque) - see CH1.

  ##########################################################################
  Opaque Section Contents & Signature:  (exactly as in CH1)

  swversion: 0.8win
  amount: usd 10.00
  card*: [from successful bc4 (includes card-salt, card-number,
      and card-expiration)]
  signature:
    48SBKUfojyC9FDKCwdCYNvucgiDxYO9erZW4QndIXZRyheTHXH8OeIhwUkyLmgQSD/UK
    +IX9035/jUkdNPOxUQq9y/beHS1HU9Fe0wlzfXYRtnjlqvQX+yUfQ4T7eNEs

  ##########################################################################
  merchant-signature is on the following fields: merchant-ccid,
      merchant-transaction, merchant-date, merchant-cyberkey, type,
      retrieval-reference-number, order-id, merchant-amount, pr-hash,
      pr-signed-hash, id, transaction, date, cyberkey

  ##########################################################################
```

Explanation:
The merchant signature ensures integrity of the majority of the message. Validation of the customer signature ensures that the customer opaque part was not tampered or replaced.

4.4.5 CM5 - return

Description: Reverse a previous charge. Really sort of a negative charge. Message just like CM1 except for different type.

```
##########################################################################
Sender: MerchantApp
Receiver: CyberServer
##########################################################################
Sample Message:

[exactly the same as CM1 except

type: return

]
```

4.4.6 CM6 - charge-action-response

Description: This receipt is given to the merchant as a receipt for a completed charge action. Indicates success/failure/etc.

```
##########################################################################
Sender: CyberServer
Receiver: MerchantApp
##########################################################################
Sample Message:

$$-CyberCash-0.8-$$
merchant-ccid: ACME-012
merchant-transaction: 123123
merchant-date: 19950121100705.nnn
opaque:
 EDD+b9wAfje5f7vscnNTJPkn1Wdi7uG3mHi8MrzLyFC0dj7e0JRjZ2PmjDHuR81kbhqb
 nX/w4uvsoPgwM5UJEW0Rb9pbB39mUFBDLPVgsNwALySeQGso0KyOjMxNs1mSukHdOmDV
 4uZR4HLRRfEhMdX4WmG/2+sbewTYaCMx4tn/+MNDZlJ89Letbz5kupr0ZekQlPix+pJs
 rHzP5YqaMnk5iRBHvwKb5MaxKXGOOef5ms8M5W8lI2d0XPecH4xNBn8BMAJ6iSkZmszo
 QfDeWgga48g2tqlA6ifZGp7daDR81lumtGMCvg==
merchant-opaque:
 6BVEfSlgVCoGh1/0R+g1C143MaA6QLvKpEgde86WWGJWx45bMUZvaAu4LVeqWoYCqSGf
 aWKUF7awol0h1i1jtgieyAcXB8ikvRJIsupSAwsRMyoNlekR6tucvfv/622JY7+n7nGO
```

```
    dGbMzP0GJImh2DmdPaceAxyOB/xOftf6ko0nndnvB+/y2mFjdUGLtFQP/+3bTpZttZXj
    j7RO1khe1UrAIk2TGQJmNw+ltsu0f42MgsxB8Q31vjPtoiPi5LEmD0Y4jlpJ7Jg2Ub84
    F9vJhYpmzNkdiJUe83Hvo/xfJRbhafJpXFEsUZwQK0jU1ksU6CQd2+CPBB+6MxtsHoxJ
    mjD6ickhd+SQZhbRCNerlTiQGhuL4wUAxzGh8aHk2oXjoMpVzWw2EImPu5QaPEc36xgr
    mNz8vCovDiuy3tZ42IGArxBweasLPLCbm0Y=
  $$-CyberCash-End-7Tm/djB05pLIw3JAyy5E7A==-$$

  ##########################################################################
  Merchant-Opaque Key: Session key same as that of CM1/2/3/4/5 for
      same Merchant-Transaction and Merchant-CCID.

  Opaque Key:  Same customer session key from CH1 passed through CM*
      for ID and Transaction

  ##########################################################################
  Merchant-Opaque Section Contents:

  type: charge-action-response
  server-date: 19950121100706.nnn
  action-code: XXX  [per ISO 8583]
  response-code: failure/success/etc.
  order-id: 1231-3424-234242
  pr-hash: 7Tm/djB05pLIw3JAyy5E7A==
  pr-signed-hash:
   8zqw0ipqtLtte0tBz5/5VPNJPPonfTwkfZPbtuk5lqMykKDvThhO0ycrfT7eXrn/hLUC
   kXoSctahEVdw1KBJbp0EVr1zVzcN9Aa7m2fJgxNfiisTgIRW+PMaa78rn+Ov
  retrieval-reference-number: 432112344321
  authorization-code: a12323
  card-hash: 7Tm/djB05pLIw3JAyy5E7A==
  {
  card-prefix: nnxxxx  [Returned if merchant is not full-PAN]
  }
      or
  {
  card-number: 1234567890123456  [Returned if merchant is full-PAN]
  }
  expiration-date: 12/34  [always present]
  merchant-swseverity: fatal/warning
  merchant-swmessage; Message for merchant about out of date
      protocol number in $$ start line of merchant message.
  merchant-message;
         Free text of the error/success condition.
         This text is for the merchant from the server...
  id: myCyberCashID
  transaction: 78784567
  date: 19950121100505.nnn

  Opaque (Customer) contents:
```

```
server-date: 19950121100706.nnn
amount: usd 10.00
order-id: 1231-3424-234242
card*: [from successful BC4]
response-code: failure/success/etc.
swseverity: fatal/warning
swmessage; Tells CyberApp that it is obsolete display this
 text to the user.  [only present if SWSeverity present]
message;
       Free text of the error/success condition.
       This text is to be displayed to the customer
       by the CyberCash application...

##########################################################################
Signature is of the following fields: no signature

##########################################################################
Explanation:
retrieval-reference-number is needed for voids. authorization-code
    is needed for post-auth-capture.  These fields are each only
    present in the CM6 if they were returned by the bank which
    depends on what operation was being done.
card-prefix is first two and last four digits of card-number.
At merchant's bank's discretion the card-number or card-prefix is
    returned.
card-hash is really the hash of the full card number and the salt
    provided by the customer.  card-hash is needed so the merchant
    can, if they wish, sort customer transactions by card without
    knowing the card number.
card* is the card* fields delivered in the CM* messages being
    responded to.  They appear in alphabetic order.
server-date duplicated in customer opaque area for security.
{}'s in column one just for clarity of alternatives and do not
    actually appear in the message.
[]ed comments appear after some fields.
```

4.4.7 The MM* Message Series

The CM* message series above is the primary CyberCash credit card purchase system for securely handling charges from CyberCash customers. However, merchants, who are authorized by their acquiring bank to accept such charges, may also receive telephone, mail, and over-the-counter sales. To avoid any necessity for the merchant to have a second parallel system to handle these charges, an MM1 through MM6 message series is defined and has been implemented for these less secure transactions.

The MM* messages look very similar to the CM* series but the "customer opaque" section is actually signed by the merchant and no separate customer CyberCash ID or prior card binding is required. The MM* message examples are omitted here in the interests of brevity.

4.4.8 CD1 - card-data-request

Description: Used by merchant to get card-number, etc., if information needed by merchant to resolve a dispute.

```
##########################################################################
Sender: MerchantApp
Receiver: CyberServer
##########################################################################
Sample Message:

$$-CyberCash-0.8-$$
merchant-ccid: ACME-69
merchant-transaction: 123123
merchant-date: 19950121100705.nnn
merchant-cyberkey: CC1001
cyberkey: CC1001
opaque:
 EDD+b9wAfje5f7vscnNTJPkn1Wdi7uG3mHi8MrzLyFC0dj7e0JRjZ2PmjDHuR81kbhqb
 nX/w4uvsoPgwM5UJEW0Rb9pbB39mUFBDLPVgsNwALySeQGso0KyOjMxNs1mSukHdOmDV
 4uZR4HLRRfEhMdX4WmG/2+sbewTYaCMx4tn/+MNDZlJ89Letbz5kupr0ZekQlPix+pJs
 rHzP5YqaMnk5iRBHvwKb5MaxKXGOOef5ms8M5W8lI2d0XPecH4xNBn8BMAJ6iSkZmszo
 QfDeWgga48g2tqlA6ifZGp7daDR81lumtGMCvg==
merchant-opaque:
 6BVEfSlgVCoGh1/0R+g1C143MaA6QLvKpEgde86WWGJWx45bMUZvaAu4LVeqWoYCqSGf
 aWKUF7awol0h1i1jtgieyAcXB8ikvRJIsupSAwsRMyoNlekR6tucvfv/622JY7+n7nGO
 dGbMzP0GJImh2DmdPaceAxyOB/xOftf6ko0nndnvB+/y2mFjdUGLtFQP/+3bTpZttZXj
 j7RO1khe1UrAIk2TGQJmNw+ltsu0f42MgsxB8Q31vjPtoiPi5LEmD0Y4jlpJ7Jg2Ub84
 F9vJhYpmzNkdiJUe83Hvo/xfJRbhafJpXFEsUZwQK0jU1ksU6CQd2+CPBB+6MxtsHoxJ
 mjD6ickhd+SQZhbRCNerlTiQGhuL4wUAxzGh8aHk2oXjoMpVzWw2EImPu5QaPEc36xgr
 mNz8vCovDiuy3tZ42IGArxBweasLPLCbm0Y=
$$-CyberCash-End-7Tm/djB05pLIw3JAyy5E7A==-$$

##########################################################################
Merchant-Opaque Section Contents:

type: card-data-request
password: xyzzy
server-date: 19950121100505.nnn  [optional]
order-id: 12313424234242
merchant-amount: usd 10.00
pr-hash: 7Tm/djB05pLIw3JAyy5E7A==
```

```
pr-signed-hash:
 IV8gWHx1f8eCkWsCsMOE3M8mnTbQ7IBBcEmyGDAwjdbaLu5Qm/bh06OX1npe2d3Hijxy
 +X8vKcVE6l6To27u7A7UmGm+po9lCUSLxgtyqyn3jWhHZpc5NZpwoTCf2pAK
id: myCyberCashID
transaction: 78784567
date: 19950121100505.nnn
merchant-signature:
 8zqw0ipqtLtte0tBz5/5VPNJPPonfTwkfZPbtuk5lqMykKDvThhO0ycrfT7eXrn/hLUC
 kXoSctahEVdw1KBJbp0EVr1zVzcN9Aa7m2fJgxNfiisTgIRW+PMaa78rn+Ov

##########################################################################
merchant-opaque key is generated from the CyberCash encrypting public
     key identified in merchant-cyberkey.

Customer opaque section (Opaque) - see CH1.

##########################################################################
Opaque Section Contents & Signature:  (exactly as in CH1)

swversion: 0.8win
amount: usd 10.00
card*: [from successful BC4 (includes card-expiration-date,
    card-number, and card-salt)]
signature:
 48SBKUfojyC9FDKCwdCYNvucgiDxYO9erZW4QndIXZRyheTHXH8OeIhwUkyLmgQSD/UK
 +IX9035/jUkdNPOxUQq9y/beHS1HU9Fe0wlzfXYRtnjlqvQX+yUfQ4T7eNEs

##########################################################################
merchant-signature is on the following fields: merchant-ccid,
    merchant-transaction, merchant-date, merchant-cyberkey, type,
    password, server-date, order-id, merchant-amount, pr-hash,
    pr-signed-hash, id, transaction, date, cyberkey

Customer Signature: see CH1

##########################################################################
Explanation:
[see also CM1 explanation]
The merchant may need to know the card involved and other
    information in order to resolve a disputed transaction.  This
    information is all contained in the original CH1 embedded in the
    CM1 for the transaction.  If the merchant saves the CM1 and other
    transaction information, they can send this CD1 message to the
    server.  While this reduces the pass through confidentiality of
    the system, the merchant is then on record as asking for this
    particular credit card number and excessive CD1's from a merchant
    can be flagged.
password is an extra level of security intended to be manually entered
```

at the merchant to authorize the unusual action. Server stores a hash of the merchant-ccid and the password.

4.4.9 CD2 - card-data-response

Description: Respond to CD1 with failure or with success and card data.

```
##############################################################################
Sender: CyberServer
Receiver: MerchantApp
##############################################################################
Sample Message:

$$-CyberCash-0.8-$$
merchant-ccid: ACME-012
merchant-transaction: 123123
merchant-date: 19950121100705.nnn
merchant-opaque:
 t731/86R72ZLrqHLIf0VG6m3ybvs+dG6K705L8LFKEXgCti0NGjK83CwDsUdiso7U1JP
 2Z0BClVHLmhIBY7+QXx5iCEGHy8JKC9IWyNNi2O/OOIDgLeJAkMSZYbNQrSKViY34imS
 0s7Q6uDk9wV0fixjvRBuNO2B7urWWsqfkLOYDnHy0RvhyUzYxLrMaTX+/6IkyU5Z0lH3
 BXYBUNV8DgitEjgLXmyWuXRDlEBN02yeZgsFRm9GmuBHfCTySm2XqnifizpmKMUa9UiH
 onNx9W86fuBdcJF7CJgH5Gct2M/dx/f2VpoRkmeSmWxFrYi8wgtvddSXF9my40NZ8WZz
 CEUEvQhcmruopwEeehv+bejc3fDDZ23JKrbhlZ17lSvFR14PKFsi32pXFqTO0ej9GTc5
 L6c8nM3tI1qdHNCe0N5f7ASdKS0tYSxAYJLIR6MqPrXjNJEaRx7Vu1odMlkgrzGOV1fo
 5w33BQHK3U2h+1e5zYBeHY3ZYG4nmylYYXIye4xpuPN4QU0dGrWZoImYE44QOwjd5ozl
 xulPBjj6cpEI/9wTwR3tpkBb4ZfYirxxnoj9JUkPK9Srv9iJ
$$-CyberCash-End-7Tm/djB05pLIw3JAyy5E7A==-$$

##############################################################################
Opaque Key: session key from CD1.

##############################################################################
Opaque Section Contents:

type: card-data-response
server-date: 19950121100706.nnn
response-code: failure/success/etc.
order-id: 1231-3424-234242
pr-hash: 7Tm/djB05pLIw3JAyy5E7A==
pr-signed-hash:
 IV8gWHx1f8eCkWsCsMOE3M8mnTbQ7IBBcEmyGDAwjdbaLu5Qm/bh06OX1npe2d3Hijxy
 +X8vKcVE6l6To27u7A7UmGm+po9lCUSLxgtyqyn3jWhHZpc5NZpwoTCf2pAK
card-hash: 7Tm/djB05pLIw3JAyy5E7A==
card-number: 4811123456781234
card-type: visa
```

```
card-name: John Q. Public
expiration-date: 01/99
merchant-swseverity: fatal/warning
merchant-swmessage; Message for merchant about out of date
    protocol number in $$ start line of merchant message.
merchant-message;
       Free text of the error/success condition.
       This text is for the merchant from the server...
id: myCyberCashID
transaction: 78784567
date: 19950121100505.nnn

##########################################################################
Signature is of the following fields: no signature.

##########################################################################
Explanation:
This normally returns selected fields from the decoding of the
    opaque part of a CH1 as sent to the server in a CD1.
```

4.5 Utility and Error Messges

A number of utility, status query, and special error reporting messages have also been found necessary in implementing the CyberCash system.

It is desirable to be able to test connectivity, roughly synchronize clocks, and get an initial determination of what client protocol and software versions are accepted. This is done via the P1 client to server message and its P2 server to client response.

Clients need to be able to determine the status of earlier transactions when the client or merchant has crashed during or has suffered data loss since the transaction. Two transaction query messages are defined, TQ1 and TQ2. One just queries and the other also cancels the transaction, if it has not yet completed. The response to both of these messages is a TQ3 response from the server.

Since the system operates in a query response mode, there are two cases where special error messages are needed. If a query seems to be of an undeterminable or unknown type, the UNK1 response error message is sent. If a response seems to be of an undeterminable or unknown type or other serious error conditions occur at the client or merchant which should be logged at the CyberCash server, the DL1 or DL2 diagnostic log message is submitted by the client or merchant in question respectively.

4.5.1 P1 - ping

Description: Very light weight check that we have connectivity from the customer to the server. Does no crypto to minimize overhead.

```
##########################################################################
Sender: CyberApp
Receiver: CyberServer
##########################################################################
Sample Message:
$$-CyberCash-0.8-$$
type: ping
id: myCyberCashID  [optional]
transaction: 123123213
date: 19950121100505.nnn
$$-CyberCash-End-7Tm/djB05pLIw3JAyy5E7A==-$$

##########################################################################
Explanation:
id optional as persona may not have been set up yet.
```

4.5.2 P2 - ping-response

Description: Response to the P1 light weight ping. Does no crypto to minimize overhead.

```
##########################################################################
Sender: CyberServer
Receiver: CyberApp
##########################################################################
Sample Message:

$$-CyberCash-0.8-$$
type: ping-response
id: myCyberCashID  [if present in P1]
transaction: 12312313
date: 19950121100505.nnn
server-date: 19950121100506.nnn
swseverity: fatal/warning  [absent if ok]
swmessage; Tells CyberApp that it is using an obsolete protocol.
     Display this text to the user.  [only present if SWSeverity
    present]
response-code: success/failure/etc.
message;
       Free text of the error/success condition.
       This text is to be displayed to the sender
```

```
       by their CyberCash application...
   supported-versions: 08.win, 0.81win, 0.8mac
   $$-CyberCash-End-7Tm/djB05pLIw3JAyy5E7A==-$$

   ########################################################################
   Explanation:
   swversion does not appear in P1 for security reasons so
       swseverity and swmessage appear only if the server can tell
       that things are old from the $$ header protocol version.
   supported-versions lets client know as soon as possible what
       versions are supported and, by implication, which are not. Does
       not compromise security by having client say what version it
       is.
```

4.5.3 TQ1 - transaction-query

```
   Description: Client query to server for Transaction status.

   ########################################################################
   Sender: CyberApp
   Receiver: CyberServer
   ########################################################################
   Sample Message:

   $$-CyberCash-0.8-$$
   id: MyCyberCashID
   date: 19950121100505.nnn
   transaction: 12312314
   cyberkey: CC1001
   opaque:
    VFaztHuj757Jrv+JxZFsHORy/zgkrxhBCu9cPdE04c1NnXzVlGOHygpSl+UGbUvnhkYl
    21QQaHkaE3geccRk03cqFYoLNRCclImcsyeIZCgVt+2dJTj1V+E7R7ePQtCj+0gY42+V
    L5BWhVtmDQFyg1DdJ6n3S/er6ZuObAjpcAogG+T1Na5dJmrTA1wRMiYVkqhXi2KMYdur
    3U47P8ZGUza7W0MST3DgvviN0kVhtmHEnm515mo6NTQdfdxw9WZpy6vMqrBGk2nTgi2c
    bnf+muO0+kiNPXVvEzRrO8o=
   $$-CyberCash-End-kchfiZ5WAUlpk1/v1ogwuQ==-$$

   ########################################################################
   Opaque Key: generated from CyberCash encryption key identified in
       CyberKey

   ########################################################################
   Opaque Section Contents:

   type: transaction-query
   swversion: 0.8win
   begin-transaction: 1234
```

```
   end-transaction: 4321
   signature:
    jJfFsKvOxLaV87gxu7lIPet3wIDwh1H2F61reYC9jmUrS6WAtUVFG9aCNuTEBoMixF0X
    vD5oPfyheJRIlnL6i0c4o/bfyO3edKAacmWjTmKt6/4y9p3qgvKkSX8r9aym

   ##########################################################################
   signature is of the following fields: id, date, transaction,
       cyberkey, type, swversion, begin-transaction,
       end-transaction

   ##########################################################################
   Explanation:
   This is a client status query of a previous transaction or
       transactions.
   begin-transaction and end-transaction can be the same.
```

4.5.4 TQ2 - transaction-cancel

```
   Description: Client query to server for Transaction
       cancellation/status.

   ##########################################################################
   Sender: CyberApp
   Receiver: CyberServer
   ##########################################################################
   Sample Message:

   $$-CyberCash-0.8-$$
   id: MyCyberCashID
   date: 19950121100505.nnn
   transaction: 12312314
   cyberkey: CC1001
   opaque:
    VFaztHuj757Jrv+JxZFsHORy/zgkrxhBCu9cPdE04c1NnXzVlGOHygpSl+UGbUvnhkYl
    21QQaHkaE3geccRk03cqFYoLNRCclImcsyeIZCgVt+2dJTj1V+E7R7ePQtCj+0gY42+V
    L5BWhVtmDQFyg1DdJ6n3S/er6ZuObAjpcAogG+T1Na5dJmrTA1wRMiYVkqhXi2KMYdur
    3U47P8ZGUza7W0MST3DgvviN0kVhtmHEnm515mo6NTQdfdxw9WZpy6vMqrBGk2nTgi2c
    bnf+muO0+kiNPXVvEzRrO8o=
   $$-CyberCash-End-kchfiZ5WAUlpk1/v1ogwuQ==-$$

   ##########################################################################
   Opaque Key: generated from CyberCash encryption key identified in
       CyberKey

   ##########################################################################
   Opaque Section Contents:
```

```
type: transaction-cancel
swversion: 0.8win
begin-transaction: 1234
end-transaction: 4321
signature:
 kD7DEav2uLQIYMtP9gbhYaBUpB2a5whNwnK2eXbbyTCf56F6dl3DIVf7D8Z4WxbY2YZn
 ByRIKeqlhmss7fbdnBiDYmKfOuc+I4bi/Oslml5riaciQhTd2JdHG+PCcHwZ

######################################################################
signature is of the following fields: id, date, transaction,
    cyberkey, type, swversion, begin-transaction, end-transaction

######################################################################
Explanation:
This is a client attempt to cancel a previous transaction or
    transactions.
begin-transaction and end-transaction can be the same.
```

The transaction-cancel transaction (TQ.2) is defined between the client and the server. This transaction permits the client to query the status of an operation and to stop the operation from occurring if it has not already occurred.

4.5.5 TQ3 - transaction-response

Description: Reports generated by a TQ1 or TQ2

```
######################################################################
Sender: CyberServer
Receiver: CyberApp
######################################################################
Sample Message:

$$-CyberCash-0.8-$$
id: mycybercashid
date: 19950121100505.nnn
transaction: 12312314
server-date: 19950121100505.nnn
opaque:
 eFXRL+H0J5q318M21wRdtcbhu9WCyLysQkeF9oIcjtbstymx343bbt0EAtU1gcJaUKJZ
 3skgvwrhcxU4bFcE68OPlUXAvLq10I3MczPYPsiGrsU0K4bZtQvDZmn727QQAfONBm5s
 s1yjIha+Fj481BJQs0CTYc3ju90lAjCYgirXtnnR6yJXoDO75b7UjthvHSnrTWVZvktX
 PvTuUCYzbXSFoYvwFM3Y+yHqSHlmWutYKQpYze8zbUSDQfmwTCJyw3aY2JasZ+xMP/CD
 JWbCA+gCLBYCnvzM/ExKTZTFD3xr5JBfNbV4p6CiK6lsfRFD7maAK6TSVnWjwCEJNpOv
 fyllfWD04fT7LINQcjJiQK1Pk/912Tk6Q35eRaQZorwv2hnY/7By2OkPyFdAqFL+D0H6
 TqzxmdEjEFKxi/PPT1+Cs/Nszy8wZzaGg8iWATfARY6stl+02dDhwOoFXSBNvchlVrcI
 IlvhumSIQs29Pntj3DbkYo4IEmmN/qi1vnzld22q7lA1q/CQakyc7jlQUFISx76buqwy
```

```
 35XiC9Yn8flE4Va14UxMf2RCR1B/XoV6AEd64KwPeCYyOYvwbRcYpRMBXFLyYgWM+ME1
 +yp7c66SrCBhW4Q8AJYQ+5j5uyO7uKyyq7OhrV0IMpRDPjiQXZMooLZOifJPmpvJ66hC
 VZuWMuA6LR+TJzWUm4sUP9Zb6zMQShedUyOPrtw1vkJXU1vZ5aI8OJAgUcLEitcD+dsY
 Df4CzA00fC10POkJ58HZB/pSBfUrHAa+IqMHyZkV/HBi9TjTwmktJi+8T9orXS0jSvor
 dMTGWn0ifETy2VXt
$$-CyberCash-End-0QXqLlNxrn4GNQPPk9AO1Q==-$$


###########################################################################
Opaque Key: Session key from TQ1/TQ2 with same Transaction and ID.

###########################################################################
Opaque Section Contents:

type: transaction-response
response-code: success/failure/etc.
message; general free form text message from server to
    customer....
swseverity: fatal/warning
swmessage; Message indicating that CyberApp software is obsolete.
    May be multiple lines.
report-fee: usd 0.15  [if non-zero]

transaction-1: old-transaction-number
transaction-status-1: success/failure/pending/cancelled/etc.
server-date-1: 19951212125959.nnn
date-1: 19950121100505.nnn
type-1: auth-only/etc.

###########################################################################
Signature is of the following fields:  no signature

###########################################################################
Explanation:
Report-fee is the notification that this report cost a fee and is
    only present if there is a fee.
There can be multiple transaction for the same transaction number as
    there could have been a auth, post-auth-capture, void, etc.

Terms
    "original transaction" refers to the payment or other transaction
that is being queried or canceled.
       Note: this transaction may not actually reside at the server.
    "request" refers to the requesting TQ.2 or TQ.1 message

id: id from the request message
date: date from the request message
transaction: transaction from the request message
```

server-date: current date/time
type: transaction-response
response-code: response code for request message, can be one of:
"success" means the request message was processed. Does not imply query or cancellation status of the request.
"failure-hard" means that the request message was not processed due to being ill-formed or otherwise inoperable.
"failure-swversion" means that the request message was not processed due to software revision problems.
message: the message applies only to the TQ transaction, not to the status of the transactions being queried or canceled. The message is provided according to the response-code as: "success" - message is omitted. "failure-hard" - use standard hard failure message. "failure-swversion" - use standard swversion message for fatal
swseverity: applies to request message
swmessage: applies to request message
-- per query/cancel fields ('N' is a series from 1 to N) --
transaction-N: transaction number of original transaction, or if the original transaction is not present in server the transaction number that the query / cancel request refers to
transaction-status-N: status of original transaction, may be one of:
"success" the original transaction was successfully processed. If request was TQ.2, cancellation is not performed.
"failure" the original transaction was not successfully processed. If request was TQ.2, cancellation is not performed (however, there is nothing to cancel, so it's all the same to the customer app).
"pending" the original transaction is still being processed and final disposition is not known.
"canceled" the original transaction has been canceled by the server. Later arrival of the original transaction will not be processed, but will be returned with a "failure-canceled" returned.
server-date-1: server-date field from original transaction or omitted if original transaction is not present in the server"
date-1: date field from original transaction or omitted if original transaction is not present in the server"
type-1: type field from original transaction or omitted if original transaction is not present in the server"

4.5.6 UNK1 - unknown-error

Description: This is the response sent when the request is so bad off you can't determine what type it is or the type is unknown to you. Sent from Merchant to Client or from Server to Merchant or from Server to Client.

```
   ########################################################################
   Sender: MerchantApp or CyberServer
   Receiver: CyberApp or MerchantApp
   ########################################################################
   Sample Message:

   $$-CyberCash-0.8-$$
   type: unknown-error
   unknown-error-message:
       Text message of error condition to display to user.  (CyberCash
       wrapper not found, wrapper integrity check fails, unknown protocol
       version specified, unknown type specified, etc.)
   {
   server-date: 19950121100506.nnn  [if sent by server]
   }
       or
   {
   merchant-date: 19950121100506.nnn  [if sent by merchant]
   }
   x-id: mycybercashID
   x-transaction: 123123213
   x-date: 19950121100505.nnn
   x-cyberkey: CC1001
   x-opaque:
    2DqiOQfGRZjzddWpEZwGsJnoTsp9Yiri8DE9cPUMPsJ7lTFuE4XHi4QfN2cAipDB2G/G
    9hr7Hj4u4xfMky7nPvJurClZejkI8eNp8iXLtrfS4DhR4yCFQjCiKk0dh83p+DDsFVV7
    TI3Du2B15sQS+SdaoPwkfVDnJv4Y+b7vu2cN7bG7exCkBapBcJZbReNaWX5sf+U8ypfw
    5V6QdMOzNXpef3z+cTTWfGOtmn9T1Pwo1Yi9ObyIf/wiK+IPb+bBZ9UwLZSB+qVMfJmX
    GnHXO3AnA/PD+jKYCtsm2Gxv2WB3CuezOyzPtORuqLp5ubgnLBF9aBBjxwLdbn+cp5sm
    lw51IHbmo1Jj7H6wyNnRpEjy4tM73jcosBfGeQDHxgyH1uaiFNr2D+WvmuYo7eun2dsy
    Wve2O/FwicWHvkg5aDPsgOjzetsn1JCNZzbW
   $$-CyberCash-End-7Tm/djB05pLIw3JAyy5E7A==-$$

   ########################################################################
   Opaque Key: see explanation

   ########################################################################
   Opaque Section Contents: see explanation

   ########################################################################
   Signature is of the following fields: see explanation

   ########################################################################
   Explanation:
   This message is sent as a response when you can’t find or understand
       even the type of a message to you.  It will always have type and
       unknown-error-message fields at the beginning.  Any fields from
       the request that are parseable are simply echoed back in the UNK1
```

message with "x-" prefixed to it. Thus, if an x-opaque appears, it was whatever the opaque was in the original request, etc. If you can decrypt the opaque section, you don't want to put the results here in the clear!

{}'s in the first column are to group alternatives only and do not appear in the message.

Since the customer originates exchanges with merchant and server and merchant originates exchanges with server, this message will only be emitted from the merchant to the customer or the server to the customer or merchant. It should generally just be logged for debugging purposes.

You may need to watch out for denial of service via forged or replayed UNK1 messages.

4.5.7 DL1 - diagnostic-log

Description: Client diagnostic log of bad message from either merchant or server.

```
##########################################################################
Sender: CyberApp
Receiver: CyberServer
##########################################################################
Sample Message:

$$-CyberCash-0.8-$$
id: MyCyberCashID
date: 19950121100505.nnn
transaction: 1234
cyberkey: CC1001
opaque:
 2DqiOQfGRZjzddWpEZwGsJnoTsp9Yiri8DE9cPUMPsJ7lTFuE4XHi4QfN2cAipDB2G/G
 9hr7Hj4u4xfMky7nPvJurClZejkI8eNp8iXLtrfS4DhR4yCFQjCiKk0dh83p+DDsFVV7
 TI3Du2B15sQS+SdaoPwkfVDnJv4Y+b7vu2cN7bG7exCkBapBcJZbReNaWX5sf+U8ypfw
 5V6QdMOzNXpef3z+cTTWfGOtmn9T1Pwo1Yi9ObyIf/wiK+IPb+bBZ9UwLZSB+qVMfJmX
 GnHXO3AnA/PD+jKYCtsm2Gxv2WB3CuezOyzPtORuqLp5ubgnLBF9aBBjxwLdbn+cp5sm
 lw51IHbmo1Jj7H6wyNnRpEjy4tM73jcosBfGeQDHxgyH1uaiFNr2D+WvmuYo7eun2dsy
 Wve2O/FwicWHvkg5aDPsgOjzetsn1JCNZzbW
$$-CyberCash-End-kchfiZ5WAUlpk1/v1ogwuQ==-$$

##########################################################################
Opaque Key: generated from CyberCash encryption key identified in
    CyberKey

##########################################################################
Opaque Section Contents:
```

```
type: diagnostic-log
message: incorrect order-id
swversion: 0.8win

x-type: original-message-type
x-transaction: original-transaction-number
x-opaque: [if can't decrypt]
 9/eFiJK5tLizsoeSmpW7uLS8/7iio7Wisfv38biio7uyufv3tfv35uH+7N3d9/exuKX3
 5+z3vuu4oqO7srnsvvz8/venoqO0v7al/7iio7WisYy+iv7s3ff3p6KjtL+2pf/wi7nw

#########################################################################
Explanation:
Client application does not expect a response for this message. The
    decrypted original message will be in the opaque section unless
    decryption fails. If decryption fails then un-decrypted opaque
    in the original will be sent.
This message will be sent to a different script or socket or host
    than normal messages so that it will just be absorbed and never
    generate an UNK1 response or anything, even if this message
    itself is screwed up.
```

4.5.8 DL2 - merchant-diagnostic-log

```
Description: Merchant diagnostic log of bad message from  server.

#########################################################################
Sender: CyberMerchant
Receiver: CyberServer
#########################################################################
Sample Message:

$$-CyberCash-0.8-$$
merchant-ccid: MyCyberCashID
merchant-transaction: 1234
merchant-date: 19950121100505.nnn
merchant-cyberkey: CC1001
merchant-opaque:
 2DqiOQfGRZjzddWpEZwGsJnoTsp9Yiri8DE9cPUMPsJ7lTFuE4XHi4QfN2cAipDB2G/G
 9hr7Hj4u4xfMky7nPvJurClZejkI8eNp8iXLtrfS4DhR4yCFQjCiKk0dh83p+DDsFVV7
 TI3Du2B15sQS+SdaoPwkfVDnJv4Y+b7vu2cN7bG7exCkBapBcJZbReNaWX5sf+U8ypfw
 5V6QdMOzNXpef3z+cTTWfGOtmn9T1Pwo1Yi9ObyIf/wiK+IPb+bBZ9UwLZSB+qVMfJmX
 GnHXO3AnA/PD+jKYCtsm2Gxv2WB3CuezOyzPtORuqLp5ubgnLBF9aBBjxwLdbn+cp5sm
 lw51IHbmo1Jj7H6wyNnRpEjy4tM73jcosBfGeQDHxgyH1uaiFNr2D+WvmuYo7eun2dsy
 Wve2O/FwicWHvkg5aDPsgOjzetsn1JCNZzbW
$$-CyberCash-End-kchfiZ5WAUlpk1/v1ogwuQ==-$$

#########################################################################
```

```
   Opaque Key: generated from CyberCash encryption key identified in
       CyberKey

   ##########################################################################
   Opaque Section Contents:

   type: merchant-diagnostic-log
   server-date:  19950121100505.nnn  [optional]
   message: incorrect order-id

   x-type: original-message-type
   x-transaction: original-transaction-number
   x-opaque: [if can't decrypt]
    9/eFiJK5tLizsoeSmpW7uLS8/7iio7Wisfv38biio7uyufv3tfv35uH+7N3d9/exuKX3
    5+z3vuu4oqO7srnsvvz8/venoqO0v7al/7iio7WisYy+iv7s3ff3p6KjtL+2pf/wi7nw

   ##########################################################################
   Explanation:
   Merchant application does not expect a response for this message. The
       decrypted original message will be in the opaque section unless
       decryption fails. If decryption fails then un-decrypted message
       will be sent.
   This message will be sent to a different script or socket or host
       than normal messages so that it will just be absorbed and never
       generate an UNK1 response or anything even if this message
       itself is screwed up.
```

4.6 Table of Messages Described

The following 31 messages are described in this document.

C = Customer App, M = Merchant App, S = CyberCash Server

FLOW	SECTION	NAME
C->S	4.2.1	BC.1 bind-credit-card
S->C	4.2.2	BC.4 bind-credit-card-response
C->M	4.3.2	CH.1 credit-card-payment
M->C	4.3.3	CH.2 credit-card-response
M->S	4.4.8	CD.1 card-data-request
S->M	4.4.9	CD.2 card-data-response
M->S	4.4.1	CM.1 auth-only
M->S	4.4.2	CM.2 auth-capture
M->S	4.4.3	CM.3 post-auth-capture

```
M->S   4.4.4   CM.4 void
M->S   4.4.5   CM.5 return
S->M   4.4.6   CM.6 charge-action-response

C->S   4.5.7   DL.1 diagnostic-log
M->S   4.5.7   DL.2 merchant-diagnostic-log

C->S   4.1.3   GA.1 get-application
S->C   4.1.4   GA.2 get-application-response

M->S   4.4.7   MM.1 merchant-auth-only
M->S   4.4.7   MM.2 merchant-auth-capture
M->S   4.4.7   MM.3 merchant-post-auth-capture
M->S   4.4.7   MM.4 merchant-void
M->S   4.4.7   MM.5 merchant-return
S->M   4.4.7   MM.6 merchant-charge-action-response

C->S   4.5.1   P.1 ping
S->C   4.5.2   P.2 ping-response

M->C   4.3.1   PR.1 payment-request

C->S   4.1.1   R.1 registration
S->C   4.1.2   R.2 registration-response
C->S   4.5.3   TQ.1 transaction-query
C->S   4.5.4   TQ.2 transaction-cancel
S->C   4.5.5   TQ.3 transaction-response

S->C, S->M, M->C
       4.5.6   UNK.1 unknown-error
```

5. Future Development

CyberCash is extending the facilities available through the CyberCash system. We are committed to implementing a full cash system, including efficient transfer of small amounts of money, the extension of the credit card system to handle terminal capture and clearances, and other improvements.

5.1 The Credit Card Authorization/Clearance Process

There are six steps in credit card processing as listed below. The first four are always involved if a transacation is completed. The fifth and sixth are optional.

(1) authorization: merchant contacts their acquiring back which normally contacts the card issung bank and returns to the merchant an approval/guarantee or a disapproval. This

temporarily decreases the available credit on the card.
(2) capture: the charge information for a purchase is entered by the merchant into a batch.
(3) clearance: a batch of items is processed. This actually causes the items in the batch to appear on credit card statements as sent by the issuing bank to its carholders.
(4) settlement: the actual interbank transfer of net funds.
(5) void: the merchant undoes step 2 (or 6) and causes a charge (or credit) to be removed from a batch. Must be done before the batch is processed.
(6) credit: the merchant causes a "negative charge" or credit to be entered into a batch. This will appear on the cardholders statement.

The fourth step, settlement, is entirely within the banking community and does not concern us here. CyberCash 0.8 provides messages to do 1, 1&2, 2, 5, and 6. This is adequate for credit card processor systems where the batch is accumulated at the bank or between the bank and the merchant. CyberCash 0.8 supports such "host capture" systems. Other credit card processor systems require the merchant to accumulate the batch. Such systems are frequently referred to as "terminal capture". This makes actions 2, 5, and 6 internal to the merchant but requires messages to perform action 3. Such batch clearance messages will be included in future versions of the CyberCash merchant and server software.

5.2 Lessons Learned

The continuing rapid development of the CyberCash system is an interesting experience. The system must deal with many existing browsers and legacy banking systems. Existing credit card processors that convey transactions to acquiring banks have complex and varied interfaces. The sophistication of security attacks on the Internet is growing rapidly.

In the face of such a rapidly changing environment, it was essential to adopt a general message framework so that messages and fields could be added as they were found necessary. Any attempt to reduce the system to a small number of perfectly opimized messages in advance would have doomed the system to failure. (As of mid-October 1995, the total number of CyberCash messages defined, including those planned for cash and microcash, enhancements to the credit card system, and some old messages being phased out in favor of improved replacements, is just over a hundred.)

Flexible operational and error handing facilities are also, as usual, the bulk of the system. Version numbering and tracking has proved to be quite important and merchant versioning is being added.

Use of text for messages has proven very beneficial. This makes it possible to easily deal with messages using common everyday tools such as text editors and spead sheets. Use of binary TLV (type, length, value) encoding or the like is certainly possible but imposes a significantly higher level of complexity on every tool that has to deal with the messages.

Encryption and decryption impose some difficulties in development. Any confusion about decryption keys or algorithms will render encrypted material meaningless and tools are needed to provide decyrption for debugging outside of normal program operation. But this pales compared with the stringencies imposed by signatures. All parts of the system must have absolutely identical ideas as to the exact bit patterns to be hashed or signed and their exact order. Seemingly trivial differences in capitalization, punctuation, framing, order, or the like, in addition to any disagreement about keys or algorithms, will lead to frustrating failures of signatures to match. Passing signatures through an intermediate system and checking them at a third system, as is done when a customer's signature is passed through a merchant and checked at the CyberCash server, compounds the problem.

6. Security Considerations

The CyberCash Version 0.8 Credit Card system provides substantial protection to payment messages as described above in sections 1.2, 2.2.4, and 2.2.5. However, it provides no privacy to the shopping interaction which is essentially outside of its purview. It also provides no protection against dishonest merchants other than those normally available with credit card purchases. Care must be taken to avoid loss of control of the machines on which parts of this system runs or security may be compromised.

Current credit card dispute resolution systems require deliberate bypasses be implemented for some of the security normally established by CyberCash as described in section 3.4.

References

[ISO 4217] - Codes for the representation of currencies and funds

[ISO 8583] - Financial transaction card originated messages - Interchange message specifications, 1993-12-15.

[RFC 822] - Crocker, D., "Standard for the format of ARPA Internet text messages", STD 11, RFC 822, UDEL, August 1982.

[RFC 1521] - Borenstein, N., and N. Freed, "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, Bellcore, Innosoft, September 1993.

[RFC 1766] - Alvestrand, H., "Tags for the Identification of Languages", UNINETT, March 1995.

Authors' Addresses

Donald E. Eastlake 3rd
CyberCash, Inc.
318 Acton Street
Carlisle, MA 01741 USA

Phone: +1 508 287 4877
EMail: dee@cybercash.com

Brian Boesch
CyberCash, Inc.
2100 Reston Parkway, Suite 430
Reston, VA 22091 USA

Phone: +1 703-620-4200
EMail: boesch@cybercash.com

Steve Crocker
CyberCash, Inc.
2100 Reston Parkway, Suite 430
Reston, VA 22091 USA

Phone: +1 703-620-4200
EMail: crocker@cybercash.com

Magdalena Yesil
CyberCash, Inc.
555 Twin Dolphin Drive, Suite 570
Redwood City, CA 94065 USA

Phone: +1 415-594-0800
EMail: magdalen@cybercash.com