AN ANALYSIS AND COMPARISON OF E-COMMERCE TRANSACTION PROTOCOLS - PURCHASING ORDER

A Survey Paper for the completion of

CMPE 298

by

Judy Nguyen

Summer 1999

SJSU

Abstract

One of the major part of E-Commerce that makes the communication between the buyer and the merchant possible is the transaction protocol that transmits data between two parties or among many parties. Some of the E-Commerce transaction protocols, such as EDI, follow certain message standard to put the data from the sender into standard format understood by the receiver, encrypt it for security, and send it across the internet via some transport technologies such as S-HTTP (Secure HyperText Transport Protocol), SSL (Secure Sockets Layer), SMTP (Simple Mail Transfer Protocol), and TCP/IP (Transmission Control Protocol/Internet protocol). Yet, some other E-Commerce protocols, such as SET, do not follow any specific message standard and just uses its own standard for each message in a transaction. In this case, more work might need to be done on the receiver's side in order to understand the data received.

This paper surveys some of the common E-Commerce transaction protocols used in transmitting purchase order information, payment information, and in some cases receipt and shipping information between the buyer and the merchant. These protocols include EDI, SET, OBI, and OTP. They are examined and compared based on the architecture they provide, the format of the message being sent between two parties, the sequence of the messages, their performance, scalability, reliability, security, flexibility, exception handling, acknowledgment, and their application. Benefits and limitations of each protocol will be analyzed and described. A conclusion will also be drawn to determine problems of these existing protocols and to propose a new transaction protocol which will help solve most, if not all, of the problems in the existing protocols by combining some of the key functionalities in some of these protocols, and by providing additional features which can be used in the E-Commerce purchasing server.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

In today's businesses, quality, efficiency and cost cutting are very important. To reduce overhead and speed up service delivery, electronic commerce (or e-commerce) has come into play. E-commerce involves buying and selling of information, products, and services via computer networks. One of the key building blocks of e-commerce is to streamline business processes by reducing paperwork and increasing automation. E-commerce is becoming increasingly popular among all kinds of businesses, from customer service to new product design because it enables formation of new types of information-based products such as interactive games, electronic books, and other information on-demand that can be very useful to consumers. Since e-commerce involves a lot of communication of all kinds of information among different types of trading partners (most commonly known are buyers and sellers), information processing is very important. Purchasing server which is one of the major parts of e-commerce infrastructure is where the information processing is done. The actual transmission of information and formatting of the data is done by the transaction protocol. Currently, there are numerous different transaction protocols that are being developed and used today. The four most well-known transaction protocols that this paper will discuss are EDI, SET, OBI, and OTP.

The purchasing server in e-commerce involves the process of receiving purchase orders from the buyer via graphical user interface (i.e. an applet application running on a web page), sending confirmation to the buyer, formatting and structuring the purchase orders, handling the purchase orders by communicating with appropriate trading partners, booking request to transport company, sending status and shipping notice to the buyer, and formatting information of an invoice to be sent to the buyer. Once the buyer receives the invoice, he can then proceed to send *Payment* to the seller. The payment is handled by the payment server which will not be discussed in detail in this paper but will be touched upon briefly as some parts of the purchasing server interact with it.

Since communication between the buyer and seller involve a lot of transmissions of structured data traveling back and forth, the transaction protocol plays a important role in making this communication possible. Most transaction systems read information passed into it from the purchasing server's buyer interface application. It then generates some kind of message format or structure. This formatted data stream is then transmitted to the seller. At the seller's site, the data is fed into the transaction system which maps the standard fields into the simple file needed by the receiving computer application, edits and verifies the incoming information, and then passes it to the receiving order entry application for processing. For some transaction protocols, security and acknowledgement are also part of this transaction process.

Transaction protocols which have fixed message standard, such as EDI, only work well with one-to-one communication. However, as more businesses are starting to get into e-commerce, a one-to-many or many-to-many communication is needed. These existing transaction protocols are therefore not powerful enough to support this kind of bursty transmissions of data among many different trading partners. Transaction protocols which support flexible message structure, such as SET and OTP, are probably more efficient since they can handle communication among multiple trading parties.

This paper describes the e-commerce infrastructure, discusses the architecture and features of each of the four transaction protocols, gives an analysis and comparison of the information found, and proposes possible solutions and design for a new transaction protocol. Part 1 of this paper gives an introduction. Part 2 discusses the motivation for e-commerce. Part 3 describes the infrastructure for e-commerce. Part 4 discusses the architecture, features, and standards used in the four transaction protocols. Part 5 gives analysis and comparison of the four transaction protocols. Part 6 presents a proposal for a new design of the transaction protocol. Part 7 provides information regarding the environment and technology used in implementing the new protocol. Part 8 gives the results of new design and implementation. Finally, a conclusion is drawn in part 9. Appendices and references are also included at the end of this paper.

1 OVERVIEW

In today's businesses, quality, efficiency and cost cutting are very important. To reduce overhead and speed up service delivery, electronic commerce (or e-commerce) has come into play. E-commerce involves buying and selling of information, products, and services via computer networks. One of the key building blocks of e-commerce is to streamline business processes by reducing paperwork and increasing automation. E-commerce is becoming increasingly popular among all kinds of businesses, from customer service to new product design because it enables formation of new types of information-based products such as interactive games, electronic books, and other information on-demand that can be very useful to consumers. Since e-commerce involves a lot of communication of all kinds of information among different types of trading partners (most commonly known are buyers and sellers), information processing is very important. Purchasing server which is one of the major parts of e-commerce infrastructure is where the information processing is done. The actual transmission of information and formatting of the data is done by the transaction protocol. Currently, there are numerous different transaction protocols that are being developed and used today. The four most well-known transaction protocols that this paper will discuss are EDI, SET, OBI, and OTP.

EDI stands for Electronic Data Interchange. It follows a fixed message format from the ANSI X12 standard. It supports direct transmission of information from one party to another which means both the sender and receiver must use the same network transport protocol in order to receive, send, and understand each other's messages. EDI can also be implemented with a VAN (Value Added Network) which helps deliver the message from one party to another. However, with VAN, both parties do not need to use the same transport protocol since they are not talking to each other directly. All messages sent will go the VAN. VAN will put the messages in the party's mailboxes which the receiving party can access at any time. Since EDI only concentrates on the transmission of the data between two parties, it does not provide security or any additional e-commerce features. For data confidentiality and authentication, a security package can be added to the EDI implementation. Section 4.1 will discuss details of the architecture and message structure of EDI and describe how it works.

SET (Secure Electronic Transaction) protocol was developed by VISA and MasterCard specifically for enabling secure credit card transactions on the Internet. This protocol is mainly used for payment. However, since it separates the purchase order information from the payment information to be sent to the Merchant and Payment Authority respectively, it can also be used to transmit order information. SET does not follow any fixed standard for its message format. Instead, it provides its own message format for each type of message sent between the merchant and consumer. SET uses public-key encryption and X.509 digital certificates for its security implementation. One of the nice additional features provided by SET is the error handling capability. SET handles errors such as duplicate, corrupted, or malformed messages. Depending on the type of the error found, SET either ignores the message or retransmits it.

OBI (Open Buying on the Internet) is a standard for business-to-business Internet commerce with an initial focus on automating high-volume, low-dollar transactions between trading partners. OBI was founded by the Internet Purchasing Rountable. The OBI standard includes not only contents of OBI Order Requests and OBI Orders, but also the customer profile, and the security and transport of the order requests. OBI involves four participants: Requisitioner, Buying Organization, Selling Organization, and Payment Authority. The Requisitioner browses the catalog of goods and services and selects items from it. The Buying Organization represents the Requisitioner in obtaining the catalog of gods and services. The Selling Organization represents the merchant in presenting the catalog of goods and services. Finally, the Payment Authority validates the payment, authorizes order fulfillment, and issues invoice. Even though OBI does not provide good error handling mechanism, it does provide very efficient customer profile which helps to uniquely identify the requisitioner and to construct a specialized catalog view from which the requisitioner can select items and check out. The OBI data part of the message follows the EDI-formatted order which conforms to the ANSI X12 standard. OBI follows SSL V3 API for cryptography and X.509 V3 certificates for public key certificates and Certificate Authorities.

OTP (Open Trading Protocol) is a protocol for the development of software products that will permit product interoperability for the electronic purchase that is independent of the chosen payment mechanism. The OTP protocol includes five trading roles: Consumer, Merchant, Deliverer, Value Acquirer, and Customer Care Provider. Most of the transactions are done between the Consumer and the other four roles. There are no transactions among the other four roles. The Consumer communicates with the Customer Care Provider to resolve Consumer disputes, with the Value Acquirer to accept value for Merchant, with the Merchant to make purchase request, and with the Deliverer to receive goods or services from the Merchant. The OTP message structure is broken down into blocks and components, each of which follows the XML standard. Because it uses the XML standard, OTP provides greater flexibility in terms of message format. Security implementation can also be included in the message as one of the message components. Like SET, OTP also provides error handling mechanism which not only checks for duplicates and corrupted messages but also monitors the length of time that the message was sent and automatically re-send the message if the timer expires.

The following sections will describe each protocol in more detail. This paper will discuss the architecture and features of each of the four transaction protocols, and gives an analysis and comparison of the information found. A conclusion is also drawn at the end.

2 EXISTING STANDARDS AND PROTOCOLS FOR E-COMMERCE TRANSACTION

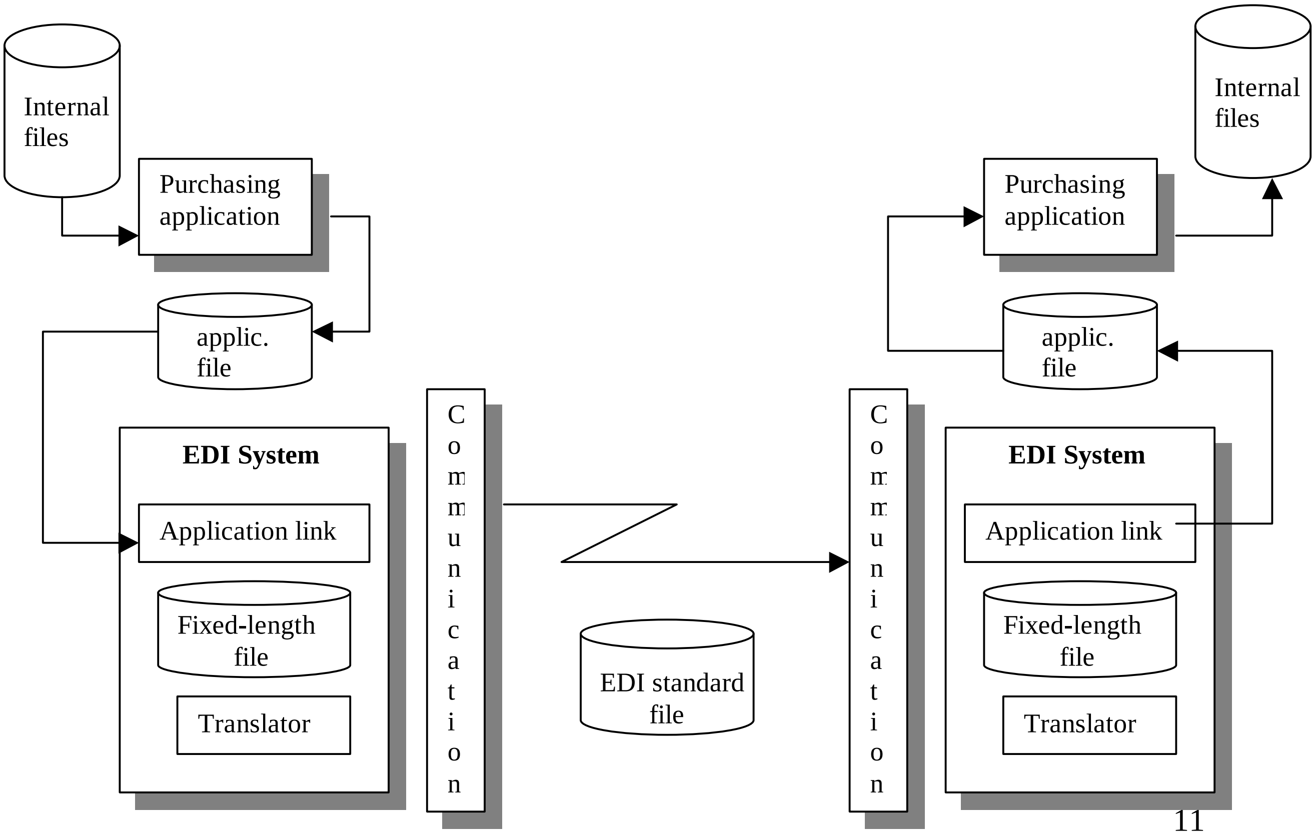
4.1 EDI

What is EDI?

EDI (Electronic Data Interchange) is an e-commerce transaction protocol for transacting business across a machine-to-machine interface with no form of human intervention. It is also sometimes considered as the application-to-application exchange of electronic data (in a structured, machine-readable format) across organizational boundaries, in such a way that no human intervention or interpretation is required.

Concepts

The EDI system first reads in a purchase order from the buyer. It then generates the machine-readable EDI standard file through an EDI translator. This file is then encrypted and transmitted to the seller via telephone lines, either directly from the sender to receiver or via a communications intermediary company called and EDI third-party service provider or VAN (Value Added Network). At the seller's site, this file is first decrypted. Then, it is fed into the EDI system which maps the standard fields, via EDI translator, into the simple file needed by the receiving computer application, edits and verifies the incoming information, and then passes it to the receiving order entry application for processing. This whole process is described in figure 4.1 below.



Two essential elements of EDI are [13]:

Figure 4.1 EDI process architecture (Source: "Opportunities for EDI")

- A standardized message that defines how data is to be represented. The format must be understood by both (or all) of the trading partners; and
- Software that interfaces the organization's in-house system(s) to the EDI communications system, as well as converting the messages to a standard format from that used by an in-house system and vice versa.

These requirements imply that for EDI to be widely used, parties must develop standards that facilitate EDI. This requirement leads to the problem of The Tower of Babel which is discussed in section 5.6.

4.1.1 Architecture

EDI architecture specifies four layers: the semantic (or application) layer, the standards translation layer, the packing (or transport) layer, and the physical network infrastructure layer [12].

EDI semantic layer	Application level services	
EDI standard layer	EDIFACT business form standards	
	ANSI X12 business form standards	
EDI transport layer	Electronic mail	X.435, MINE
	Point to point	FTP, TELNET
	World Wide Web	HTTP
Physical layer	Dial-up lines, Internet, I-way	

Figure 4.1.1a Layered architecture of EDI
(Source: "Frontiers of Electronic Commerce", Kalakota & Whinston)

Semantic Layer

The semantic layer describes the business application that is driving EDI. Examples of these data are quotes, price quotes, purchase orders, acknowledgments, and invoices. At the sender's site, this layer consists of the business application to convert the business proprietary information into EDI standard format which is agreed upon by all business partners involved. This standard formatted data is then sent to the trading partner. At the receiver's site, the EDI translator translates the EDI file into the receiver's local format. The EDI translator on the receiver's site sits in this semantic layer.

EDI Standards

The EDI standards follow the X12 and EDIFACT standards to specify business form structure and to some extent influence content seen at the application layer. For example, an address field on the purchase order might hold maximum of 20 characters in an X12 standard. An application using 50-character address field length will produce string truncation during the translation from the application layer to the standard layer [12].

EDI Transport Layer

This layer involves the network mechanism to transport data between two parties. When EDI was first introduced, direct communication between two parties was used which means that both parties must communicate through the same transport protocol such as SMTP (Simple Mail Transport Protocol). However, as e-commerce becomes more popular and EDI being more widely used, a third-party service provider was introduced to eliminate problems with maintaining many different transport protocols for communicating with different trading partners. This third-party service provider is commonly known as VAN (Value Added Network). VAN acts as a postal system for EDI. These third-party service providers furnish clients with electronic mailboxes, storage and forwarding services, tracking capabilities, and translation services. VANs are an integral part of the EDI picture and without them, a large scale EDI program is virtually impossible. When using a VAN, there is no need to worry about different communication protocols between partners, a partner's system downtime, or system security risks. VANs provide 24-hour access to the merchant and all the business partners' mailboxes and can ensure data integrity. Also, VAN interconnects allow different VANs to pass information back and forth [9].

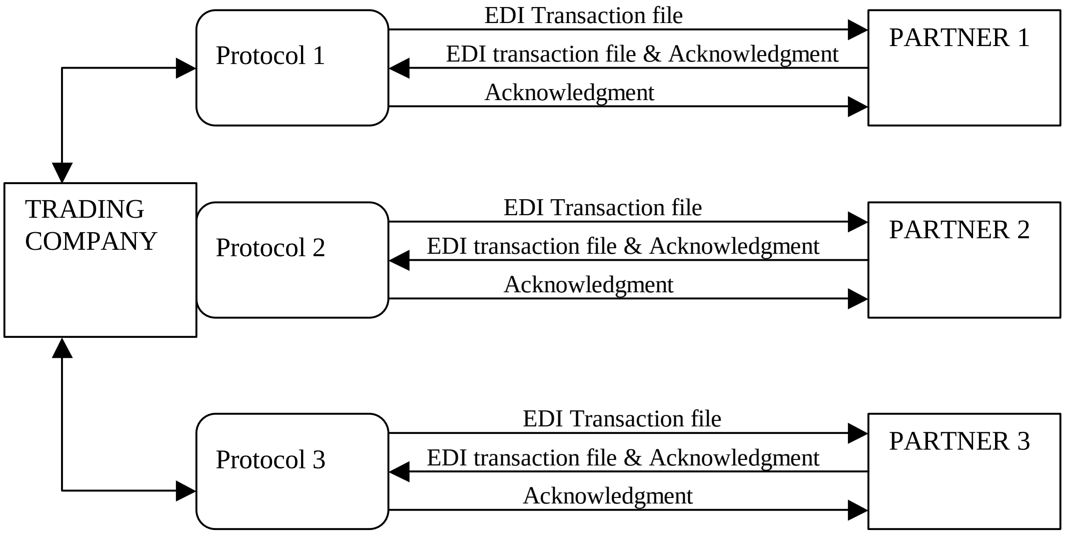


Figure 4.1.1b Direct EDI communication between trading partners
(Source: "Business Issues of EDI")

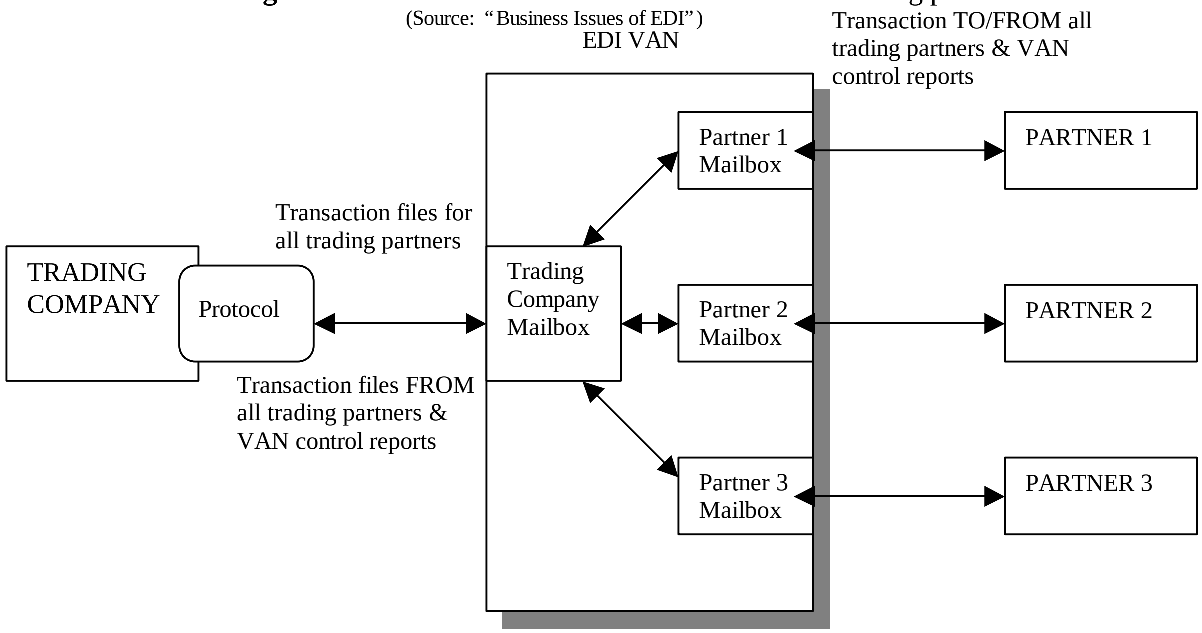

EDI VAN
Transaction TO/FROM all trading partners & VAN control reports
Partner 1 Mailbox
PARTNER 1
Transaction files for all trading partners
TRADING COMPANY
Protocol
Trading Company Mailbox
Partner 2 Mailbox
PARTNER 2
Transaction files FROM all trading partners & VAN control reports
Partner 3 Mailbox
PARTNER 3

Figure 4.1.1c EDI through a VAN
(Source: "Business Issues of EDI")

EDI Document Transport

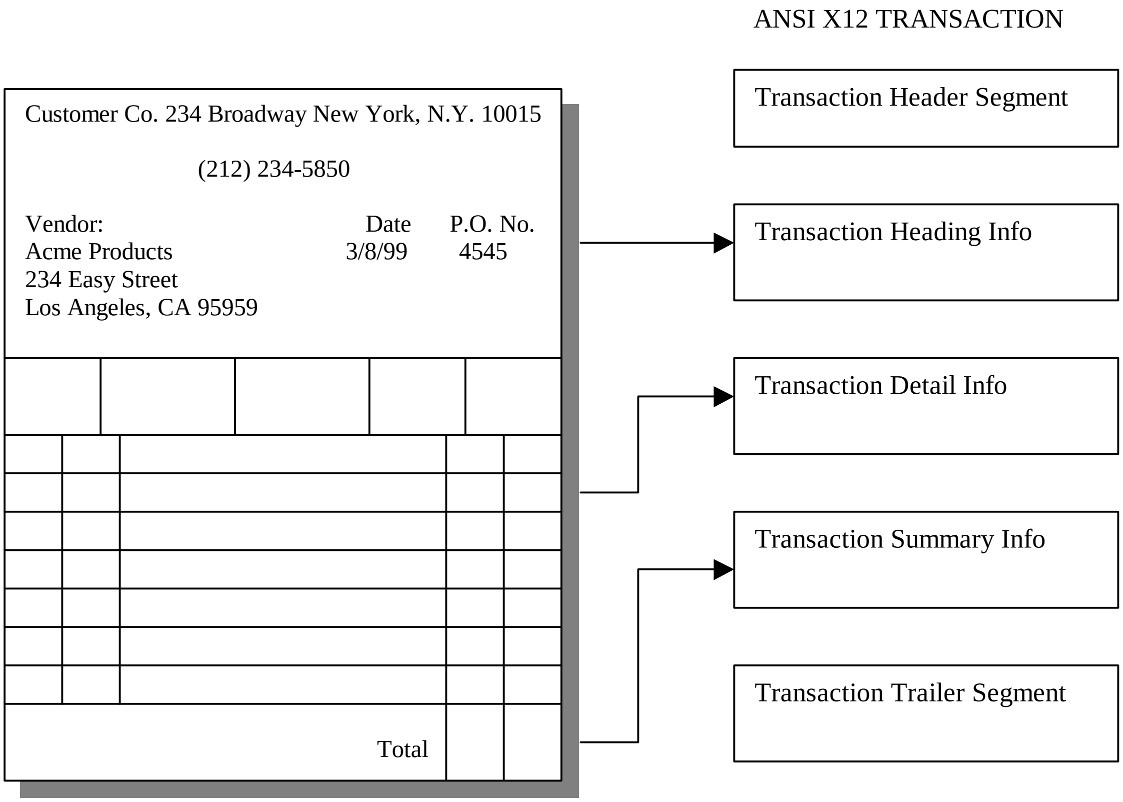
This document transport layer is far more complicated than simply sending email messages between two parties. EDI documents are structured and contain header information that is used during transport and authentication process.

4.1.2 Standards

EDI relies on the use of standards for the structure and interpretation of electronic business transactions. All trading partners must agree on and use a particular standard to reduce errors and ensure accurate translation of data, regardless of the computer systems involved.

ANSI X12

The ANSI chartered the Accredited Standards Committee in 1979 to research and develop standards for business documents. The X12 committee developed standards to facilitate EDI in business transactions such as order processing, shipping and receiving, and invoicing. The X12 transaction sets generally map traditional paper document to electronic format that can move easily over telecommunication networks. Each transaction includes several data segments which define the business functions and instructive information for correct network routing. The X12 transactions can be transmitted to the trading partner through either the X400 e-mail protocol, or the multipurpose Internet mail extensions (MIME) protocols. [12]



Customer Co. 234 Broadway New York, N.Y. 10015
(212) 234-5850

Vendor:
Acme Products
234 Easy Street
Los Angeles, CA 95959

Date	P.O. No.
3/8/99	4545

Total

Figure 4.1.2 The electronic form of an X12 transaction set
(Source: "Business Issues of EDI")

UN/EDIFACT

EDIFACT was developed by the United Nations based on TRADECOMS which was developed by the U.K. Department of Customs and Excise. It was further developed by the United Nations Economic Commission for Europe. EDIFACT is becoming widely accepted as the foremost international EDI standard. Today, EDIFACT and ANSI are working towards compatibility. [12]

Because most EDI information exchanges are domestic, X12 is more widely used than EDIFACT in the United States. However, as global trade expands, most U.S. companies will require implementation of both standards. This will make EDI harder to implement and understand since it has to abide to the standard that is derived from so many government and business representatives around the world. Migration to the new international standard will be a challenge for businesses.

Other than message format, EDI does not follow other standards for security or network transport since these features are not part of EDI's architecture. Below is a table listing standards used in EDI.

Purpose	**Standard**	**Examples**
Content Display	Current standards for Web browser (e.g. HTTP and HTML)	Netscape or Internet Explorer
Order Request	X12/EDIFACT	Purchase order in EDI standard files
Order Transmission	Depends on the transport protocol used by the trading party	VAN (Value Added Network) or SMTP with S/MINE
Secure Internet Communication	Depends on the secure communication protocol used by trading party	SSL
Cryptography	Depends on the cryptography mechanism used by the trading party	Public-Key encryption

Table 4.1.2 Standards used in EDI

4.1.3 Structure of EDI Transactions

EDI messages share a common structure:

1. *Transaction set* is equivalent to a business document, such as a purchase order. Each transaction set is made up of data segments.

2. *Data segments* are logical groups of data elements that together convey information, such as invoice terms, shipping information, or purchase order line.

3. *Data elements* are individual fields, such as purchase order number, quantity on order, unit price.

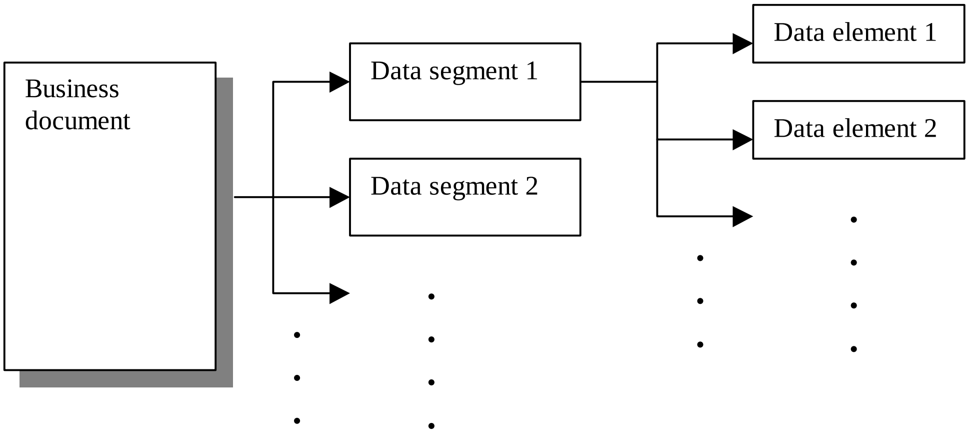


Figure 4.1.3 EDI transaction structure

4.1.4 Security

EDI is mainly just a transaction protocol and therefore does not include security in its implementation. Thus, security control must be added to the EDI process to ensure confidentiality of the data transmitted. Figure 4.1.4a shows the security modules that are needed in EDI [4].

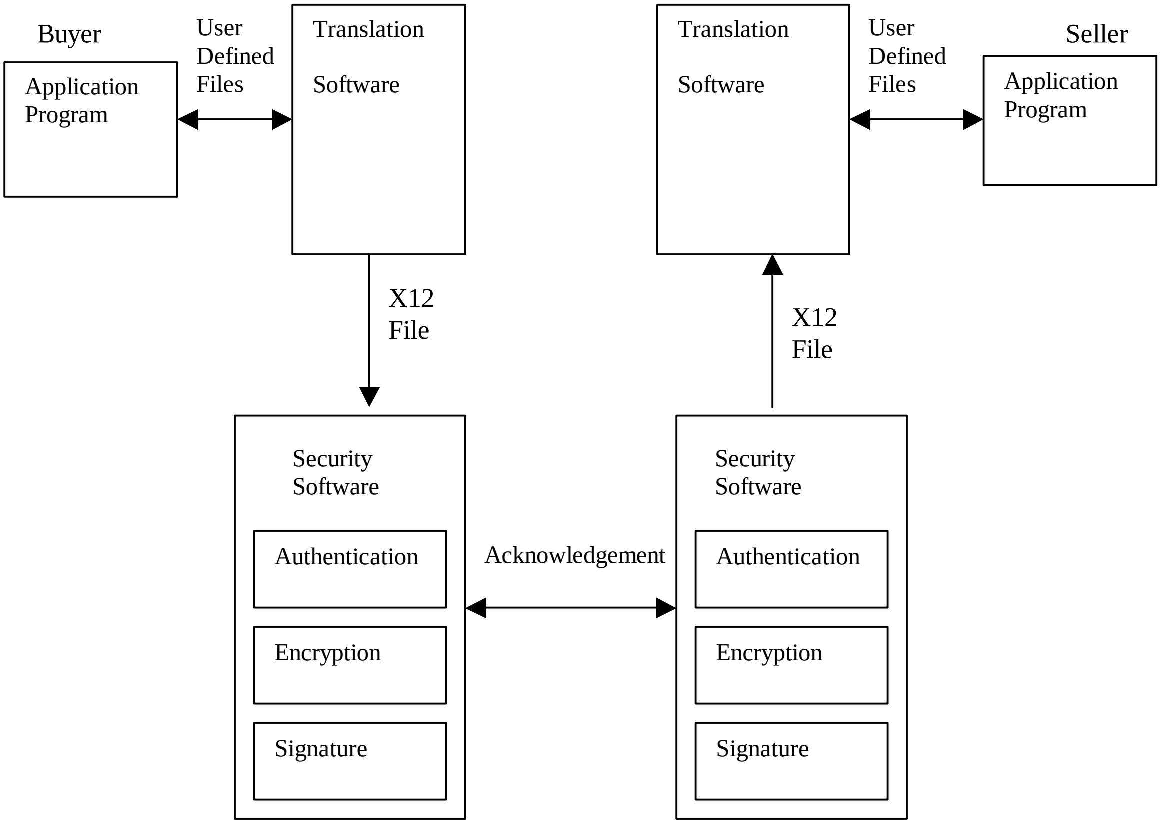


Figure 4.1.4a Data Communications Security

Authentication

Authentication is a way of identifying the parties involved in the communication. A normal means of providing authentication is through the use of passwords. The latest technology provides authentication through the use of digital certificates which function like ID cards. The digital certificate has multiple functions, including browser authentication [4].

Encryption

Encryption involves the process of encoding the data so that it is indecipherable to anyone except the intended recipient. Encryption prevents snoopers, hackers, and other parties from viewing data that is transmitted over telecommunications channels. Two basic encryption schemes are public-key and private-key encryption.

Signature

Electronic signatures are the computerized version of the signature function. Signatures are needed in some business applications for authorization purposes. A digital signature algorithm can be used to generate digital signatures. The digital signature itself is used to detect unauthorized modification to data and to authenticate the identity of the signature.

Security is inserted into the EDI message as follows [23]:

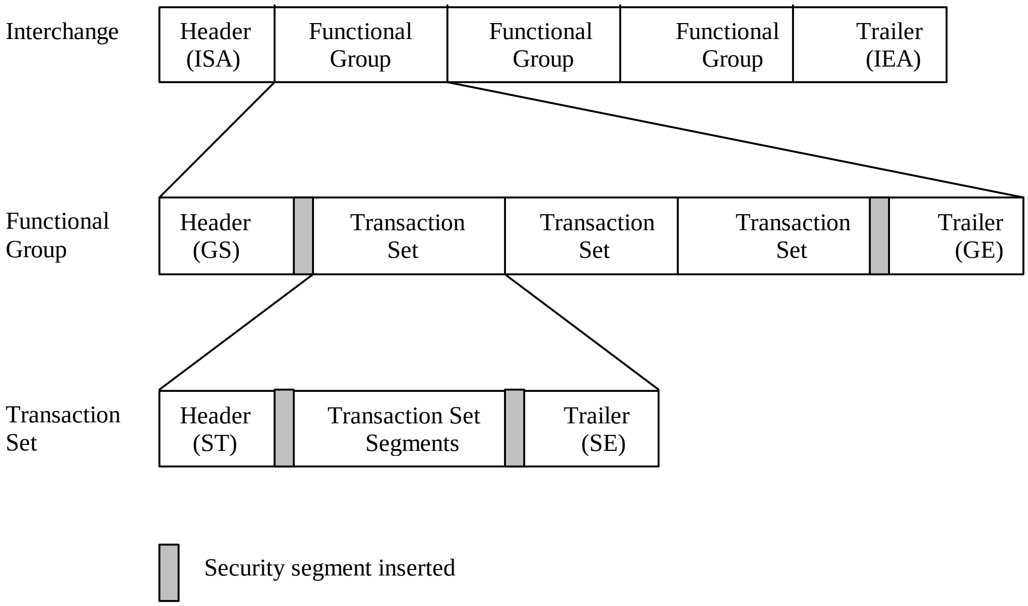


Figure 4.1.4b Security in ANSI X12 Interchange Structure
(Source: "Secure Electronic Commerce", Ford and Baum)

4.2 SET

What it is

SET is the Secure Electronic Transaction protocol developed by Visa and MasterCard specifically for enabling secure credit card transactions on the Internet. It uses digital certificates to ensure the identities of all parties involved in a purchase and encrypts credit card information before sending it across the Internet [15]. When a consumer initiates a transaction at their computer, they can use a piece of software called electronic wallet containing a digital certificate that authenticates that the consumer is who he or she says, while the merchant’s certificate provides the consumer with the same authenticating information about the merchant [2].

SET is basically a process that operates under a fixed set of rules. It has characteristics that are different than those associated with digital money such as DigiCash. Digital money by itself can cause a transaction to occur. SET requires a functional meeting (or at least an exchange of messages) of all the interested parties before a transaction can occur. A SET transaction must be authorized by all the parties concerned in order to complete, while merchants in the digital money realm may change the resultant value of a digital cash token. SET merchants cannot directly touch cardholder information, which is available only to the Issuer (and by extension the Payment Gateway), while digital money merchants can derive transaction/recipient information [13].

SET allows for the merchant's identity to be authenticated via digital certificates. However, SET also allows for the merchant to request users to authenticate themselves through digital certificates. This makes it much more difficult for someone to use a stolen credit card.

Concepts

One of the critical design objectives of SET was that messages could be resent without penalty to the overall transaction, since some messages will always be lost or misdirected in transit during operation [13].

According to SET business plan, SET should [13]:

1. Provide confidentiality of payment information and enable confidentiality of the order (merchandise) information that is transmitted along with the order’s payment information

2. Ensure integrity for all transmitted data

3. Provide authentication that a cardholder is a legitimate user of a branded payment card account

4. Provide authentication that a merchant can accept branded payment card transactions through its relationship with an acquiring financial institution

5. Ensure the use of the best security practices and system design techniques to protect all legitimate parties of an electronic commerce transaction

6. Ensure the creation of a protocol that is neither dependent on transport security mechanism nor prevents their use

7. Facilitate and encourage interoperability across software and network providers

Before starting the purchase request phase by invoking SET, all ordering information or functions, including adding on any applicable shipping charges and state tax, must be completed before SET transactions are begun. Below is a brief description of the SET process [14].

- When the Cardholder selects the form of payment to use for the order, SET is initiated at this point.

- Cardholder sends completed order form and payment instructions to the Merchant. SET is used to sign these order forms and payment instructions digitally using the Cardholder's digital certificate to prove they came from the Cardholder and no one else.

- Merchant requests payment authorization from the Issuer of the payment card using its Merchant account through its Acquirer's payment system. SET wraps these messages in cryptography to assure their privacy and confidentiality.

- Merchant ships goods or performs requested services based on the order.

- Merchant requests to capture the payment that was previously approved for processing. SET wraps these messages in cryptography, to ensure their privacy and confidentiality. Those steps not included under SET are considered out-of-band (or out of scope) activities, and their implementation is left up to the involved parties. SET provides open and robust data structures and corresponding security to handle virtually any type of order processing.

4.2.1 Architecture

The basic operations that constitute a routine SET transaction usually involves four main participants:

1. *Certificate Authority* (CA) helps to assure all the parties in a transaction that their partners are, indeed, who they purport to be. It has several primary functions in the certificate chain, including to receive registration requests, process and approve (or reject) them, and issue certificates. Certificates contain information that, after being processed, can establish their own validity and origin. Digital certificates are authentication certificates which use public-private key cryptography to identify people, privileges, and relationships. Before any transaction can take place, all parties involved needs to own a SET digital certificate.

2. *Cardholder* is the person that does the web browsing and shopping and is the authorized holder of a bank card who is registered with the corresponding issuer to perform electronic commerce. The Cardholder must register herself with a Cardholder Certificate Authority (CCA) so that she can participate in SET transactions with merchants. Unless she has certificates that can be traversed to the Root CA, merchants will refuse any SET attempts she makes.

3. *Merchant* is the seller of goods, services, or information, who accepts payment electronically. The Merchant also needs to register for the same reason as cardholder's registration, since a merchant needs a properly signed certificate in order to process SET orders.

4. *Acquirer Payment Gateway* is a system that provides online electronic commerce services to merchants. It is needed to validate SET digital certificates and preprocess authorization, capture, and settlement work. Payment Gateways are operated by whatever company is serving the Charge Processor duties for Merchants and banks.

Figure 4.2.1a shows the connections between these four participants.

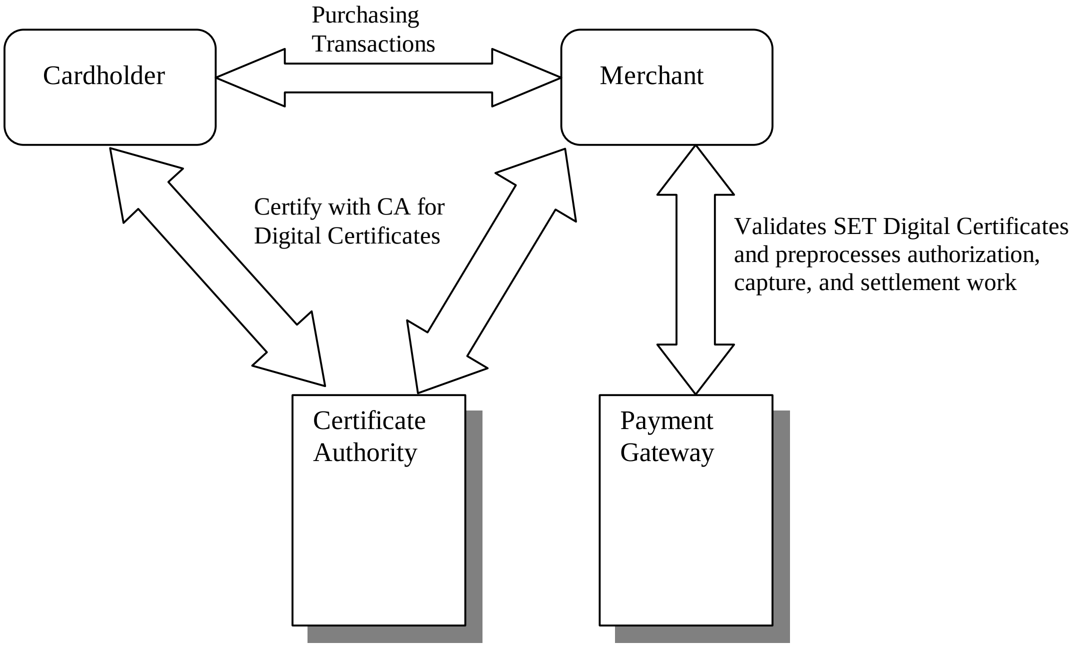


Figure 4.2.1a SET Architecture

Protocol Transaction

A list steps below show how these participants interact with each other in a SET transaction.

1. ***Obtaining digital certificates***. All parties involved need to obtain digital certificates from Certificate Authority *(CertReq/CertRes)*.

2. ***Payment initiation processing***. When payment option has been selected, Cardholder's E-wallet (which contains consumer's digital signature or certificate)

initiates the first SET payment processing message pair, called the Payment Initialization Request (*PInitReq)*, which is generated and sent back to the Merchant SET POS software. With a successful Payment Initialization Response (*PinitRes*), the E-wallet then creates a SET Purchase Request message (*Preq)*. This message has two components: Purchase Order information (*OI)* and Payment information (*PI)*. The Merchant POS software can only read the Purchase Order. The Payment Instructions, containing information about credit card account, can only be read and processed by the Payment Gateway software.

3. ***Payment Authorization Request***. Merchant's SET POS prepares a SET Authorization Request (*AuthReq)* message and Payment Instructions to be sent to the Cardholder's Payment Gateway. The message contains details about the amount of the sale, the Merchant account requesting it, and the previously created Payment Instructions component that the Cardholder E-wallet software generated in step 2. With successful translation, Cardholder's Payment Gateway creates a standard authorization request and places it on the bank's Interchange Network that locates the Cardholder's account at bank A (for instance). With an approval code from bank A to proceed with the sale, the Payment Gateway responds with a SET Authorization Response (*AuthRes)* that tells the Merchant to complete the sale. It then creates a Purchase Response message and send it to the Cardholder's E-wallet that confirms the sale and produces an electronic version of a receipt or record of charge.

4. ***Delivery of Goods***. Merchant will take care of delivering the goods to the Cardholder.

5. ***Capture and Settlement***. With the successful authorization code from step 3, Merchant's SET POS software received a capture record (SET calls these tokens). With the sale completed and the goods delivered, the Merchant can initiate a Capture Request (*CapReq)* to finalize the sale with Cardholder's Payment Gateway system. With each Capture Response (*CapRes)*, the Settlement File builds up, awaiting Merchant's decision to deposit these receipts into the Merchant's account at bank B (for instance) in exchange for funds transfer. Settlement File or Batch Processing is also carried out via the Internet using SET's Batch Administration message pairs, designed specifically for those purposes.

Below are diagrams which illustrate the SET process and the involvement of the four participants [13][14].

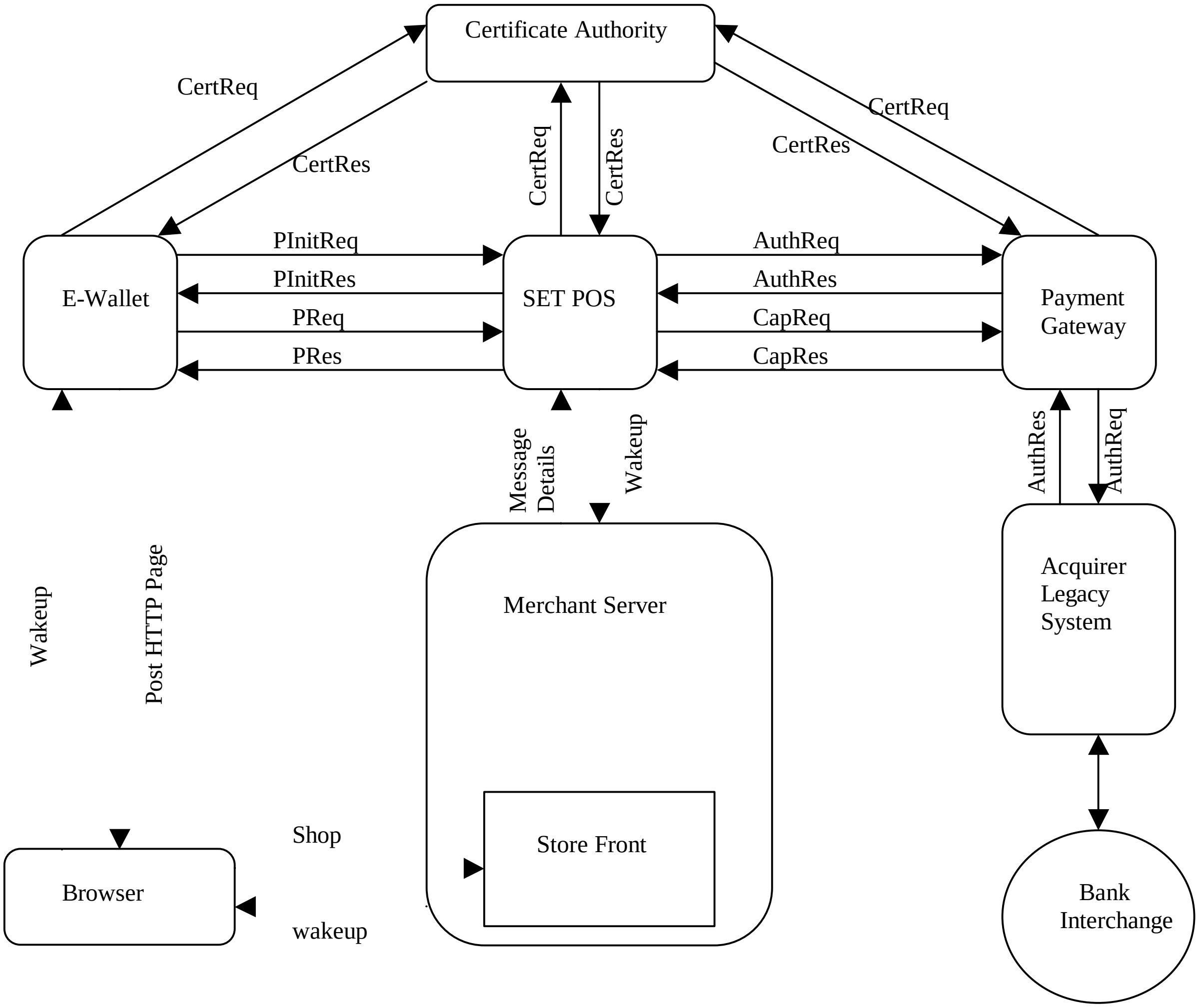


Figure 4.2.1b Interactions among all SET entities
(Source: "Building SET Applications for Secure Transactions", Merkow, Breithaupt, Wheeler)

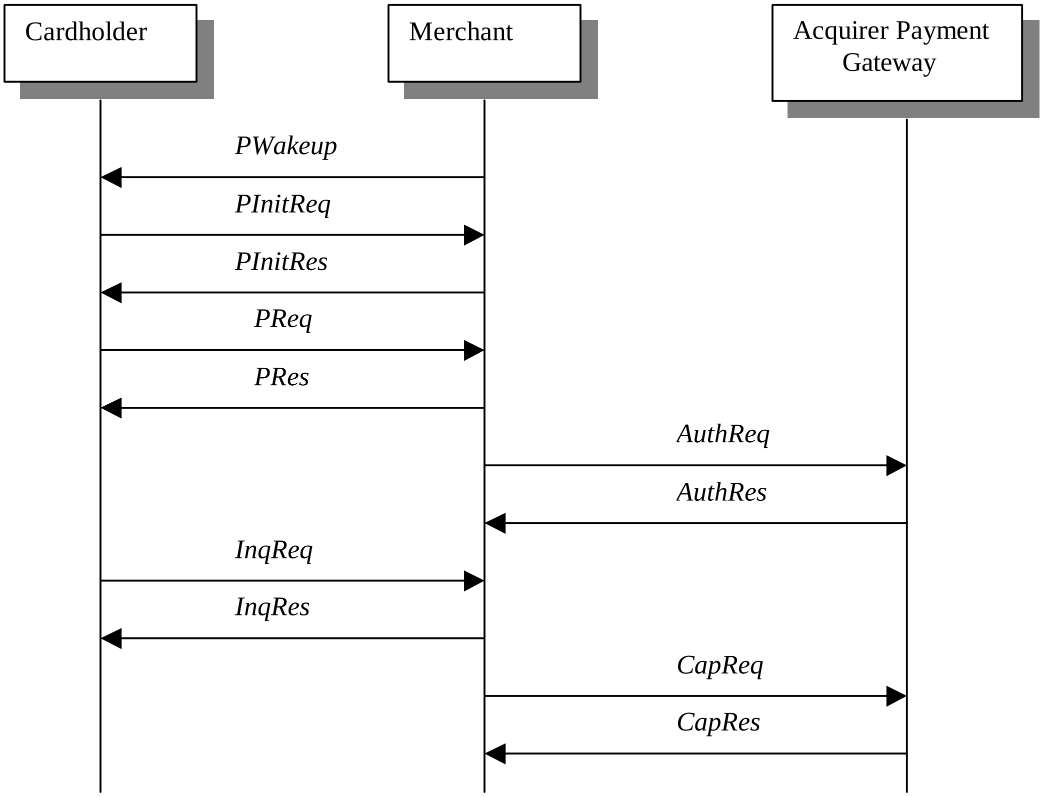


Figure 4.2.1c Sequence of SET message pairs
(Source: "Secure Electronic Transactions", Larry Loeb)

Note that status of the order can also be obtained with *InqReq* and *InqRes* messages.

Transport Mechanism

SET uses either HTTP (HyperText Transport Protocol) or SMTP (Simple Mail Transport Protocol) as network transport protocol to transmit data from one party to another.

In an HTTP session, MIME headers are transmitted along with any other headers following the initial request/response. HTTP supports binary data as the default transfer encoding. This means that SET messages do not have to invoke any MIME transfer encoding, since it's taken care of by the defaults inherent in HTTP [13].

SET has some functional areas of conflict with HTTP.

1. Because of security issues, SET does not allow payment transactions to be initiated by the merchant.
2. MINE support in a web browser provides a way for the merchant to communicate with the SET payment application, but similar way of passing information back from the SET application to the merchant using a browser is not possible.

3. For smoothest transition, the contents of the URLs should replace the merchant page that triggered the payment request. But there is no simple way via HTTP to allow a browser to retrieve any specific page.
4. There is no way for SET application to maintain state by staying “live” through the entire course of a SET transaction.
5. SET software operating over HTTP connections may not receive a timely response. Also, there needs to be limits on the number and frequency of retries for SET software so that it does not spend unlimited time retrying an interrupted connection over a dead communication path.

For now, the workaround to these problems is to use POST instead of GET to collect data from browsers as well as to set page expiration to “immediate”.

SET messages passed through e-mail must use the Simple Mail Transport Protocol supported by the Internet. SMTP-based electronic mail has two significant limitations in its ability to support SET transactions:

1. Not all SMTP servers support 8-bit and binary data. In fact, SMTP must generally assume the lowest common denominator of 7-bit transmissions.
2. SMTP mail delivery does not always occur in a direct source-to-destination fashion. Mail messages usually must pass through several intermediate “relay” hosts and therefore may encounter significant delays. This is a problem for supporting timely interactive communication using e-mail as a transport mechanism.

Currently, the existing workarounds for both of these problems is to use Base64 encoding to get around the lack of direct binary transmission of files. Remove two initial messages (PInitReq and PInitRes) and put the transaction-specific merchant values normally provided in the PinitRes message in the main message to speed up the email-based SET transaction.

4.2.2 Message Structure

Each of SET’s message pair (e.g. PInitReq/PInitRes) has its own message format with header, date, ID, SET extensions, etc. An example of the contents of one message pair, PInitReq/PInitRes, is shown in table 4.2.2a and table 4.2.2b. For details of the exact format of other message pairs, refer to *SET Secure Electronic Transaction Specification; Book 3: Formal Protocol Definition, Version 1.0.*

SET Extensions

An extension field has been added to the end of certain SET protocol data units to allow for changes within SET as business conditions warrant. Some of the protocol data units that have extension added are MessageWrapper, PInitReq, PInitResData, SaleDetail, and Transaction Detail [13].

An extension consists of three components:

- *An object identifier.* This identifier uniquely identifies the extension and helps permit SET applications to recognize an extension and process the data contained within that extension.
- *A criticality flag.* This flag indicates whether or not the recipient must understand the extension to process the message containing the extension
- *Data component.* This component provides additional information that necessitate the definition of the extension. The layout of the data will be defined by the organization creating the extension.

One important step that must be followed is that organizations defining an extension must register the object identifier and the data content of the extension with the payment card companies (or their designate) prior to deploying software that transmits messages (including test messages) over an open network [13].

PinitReq	**{RRPID, Language, LID-C, [LID-M], Chall-C, BrandID, BIN, [Thumbs], [PIRqExtensions]}**
RRPID	Request/response pair ID
Language	Cardholder’s natural language
LID-C	Local ID; convenience label generated by and for the Cardholder system
LID-M	Copied from SET initiation messages (if present) described in the External Interface Guide
Chall-C	Cardholder’s challenge to Merchant’s signature freshness
BrandID	Cardholder’s chosen payment card brand
BIN	Bank Identification Number from the cardholder’s account number (first six digits)
Thumbs	Lists of Certificate, CRL, and BrandCRLIdentifier thumbprints in Cardholder’s cache
PIRqExtensions	Note: The purchase initialization request is not encrypted, so this extension must not contain confidential information

{ } A grouping of zero or more data elements
[ ] Optional

Table 4.2.2a PinitReq
(Source: “SET Secure Electronic Transaction Specification, Book 3”, VISA/MasterCard)

PinitRes	**S(M, PinitResData)**
PinitResData	**{TransIDs, RRPID, Chall-C, Chall-M, [BrandCRLIdentifier, PEThumb, [Thumbs], [PIRsExtensions]}**
TransIDs	**{LID-C, [LID-M], XID, PreqDate, [PaySysID], Language}** *(see SET Secure* Electronic Transaction Specification for more details)
RRRPID	Request/Response pair ID
Chall-C	Copied from **PinitReq**
Chall-M	Merchant's challenge to Cardholder's signature freshness
BrandCRLIdentifier	List of current CRLs for all CAs under a Brand CA.
PEThumb	Thumbprint of Payment Gateway key-exchange certificate
Thumbs	Copied from **PinitReq**
PIRsExtentions	Note: The purchase initialization response is not encrypted, so this extension must not contain confidential information

{ } A grouping of zero or more data elements
[ ] Optional

Table 4.2.2b PinitRes
(Source: "SET Secure Electronic Transaction Specification, Book 3", VISA/MasterCard)

4.2.3 Security

In a typical SET transaction, there are two kinds of information. One set of information such as items being ordered is private between the customer and the merchant. The other set of information such as the customer's credit card number is private between the customer and the bank. The information for the bank is encrypted using the bank's public key while information for the merchant is encrypted using the merchant's public key. Both of these sets of information are digitally signed. SET, however, also allows for both of these sets of private information to be included in a single, digitally signed transaction by combining the two signatures [15]. SET relies on cryptography technologies such as public-key encryption and x.509 digital certificates to ensure message privacy, integrity, and authentication.

Many of the low-level functions, especially the cryptographic ones, in SET are usually handled by programmers with toolkits. However, before using the toolkits, programmers are also required to understand the details of SET in order to take full advantage of the toolkits and correctly implement SET. Since SET is not dependent on any transport security plan (like S-HTTP or SSL), SET is forced to generate its own security when needed even though it does not require this in order to assure confidentiality [13].

4.2.4 Error Handling

While SET incorporates transaction identifiers to evaluate necessary responses, it's still the programmer's responsibility to build reality checks into the code and adhere to them.

the idempotency of SET is an essential characteristic of the rational design of any program that uses SET.

SET handles errors by first trying to classify them. Duplicate messages are detected by examining plaintext information found in the SET MessageWrapper (the highest level of information transmitted in SET) and are ignored [13].

Corrupted (non-parsable) messages are conceptually easy to handle. For example, if enough data have been received to determine the RRPID (a SET term that stands for Request/Response pair ID, an individual transaction's statistically unique ID) of the transmitted SET message, the message is simply retransmitted to clean up the situation. If the corruption is such that nothing can be parsed, the message is ignored [13].

If a message is malformed, or differs after parsing from what is expected, SET will return an error message to the originator, who will deal with the situation and retransmit if indicated [13].

Lastly, if authentication tests fail, an error message is returned to the message originator. The message should be both delayed and generic so that details about the failure will not feed a possible attack.

4.2.5 Standards

SET follows a set of standards listed below.

Purpose	**Standard**	**Examples**
Content Display	Current standards for Web browser (e.g. HTTP and HTML)	Netscape or Internet Explorer
Order Request	Does not follow any specific standard. SET has its own fixed set of rules	Purchase Request (PReq) / Purchase Response (PRes) message pair
Order Transmission	HTTP 1.0 using SSL or SMTP	HTTP servers available from many vendors including Netscape, Microsoft
Secure Internet Communication	Has its own security mechanism	Digital signatures
Cryptography	Public-Key encryption and X.509 digital certificates	Hashing and digital certificates

Table 4.2.5 Standards used in SET

4.3 OBI

What it is

OBI (Open Buying on the Internet) is a standard for business-to-business Internet commerce with an initial focus on automating high-volume, low-dollar transactions between trading partners. It is a freely available standard that any organization can obtain and use. The OBI standard is the result of the work conducted by the Internet Purchasing Roundtable to provide access to easy-to-use, open, standards-based Internet purchasing solutions [17]. The Roundtable tries to produce a balanced and fair standard for all involved buying organizations, selling organizations, technology companies, financial institutions and service providers. The standard follows the principles below:

- *Common Business Vision.* The standard is based on the participants' common issues and expectations for business-to-business Internet commerce solutions.

- *Vendor Neutrality.* The standard is intended to be vendor-neutral to encourage competitive product offerings from a variety of technology companies and service providers.

- *Interoperability.* The standard encourages the implementation of interoperable Internet purchasing solutions.

- *Value-Added Services.* The standard encourages competitive products and services through value-added services and features that help make all parties to differentiate themselves.

- *Cost Effectiveness.* OBI solutions should be easily and inexpensively integrated into existing IT infrastructures. Maintenance costs must be reasonable.

- *Robust Infrastructure.* The standard should be able to support large populations of requisitioners reliably and securely.

- *Flexibility.* The standard should be flexible enough to accommodate the variations that will arise with specific OBI implementations and the evolving business needs of buying and selling organizations.

Concepts

OBI standard includes precise technical specifications for the security, transport, and contents of OBI Order Requests and OBI Orders. For security, OBI uses SSL (Secure Sockets Layer) protocol to secure communications on the Internet. Digital certificates are based on ITU-T's X.509v3 standard and used for authenticating requisitioners and servers. Exchange of OBI Order Requests and OBI Orders is accomplished via HTTP over SSL. Contents of OBI Order Requests and OBI Orders are based on the ANSI ASC X.12's 850, a standard for an EDI purchase order.

Once a requisitioner selects their required products, the supplier's server formats the order as an OBI Order Request, digitally signs it and sends it back to the purchaser's server for verification. The purchaser's server formats the Order Request in order to integrate it with the purchasing company's workflow system. The formatted Order Request is served up to the requisitioner for final approval. Once they complete it, it is formatted again as an OBI Order and sent back via HTTP/SSL to the supplier's server. From there, the order is integrated into the supplier's normal fulfillment process, including handling, packing, and shipping [3].

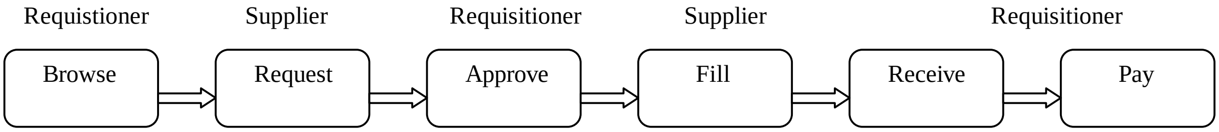


Figure 4.3 OBI Purchasing Process

4.3.1 Architecture

Figure 4.3.1 below describes the OBI architecture at an abstract level [22]:

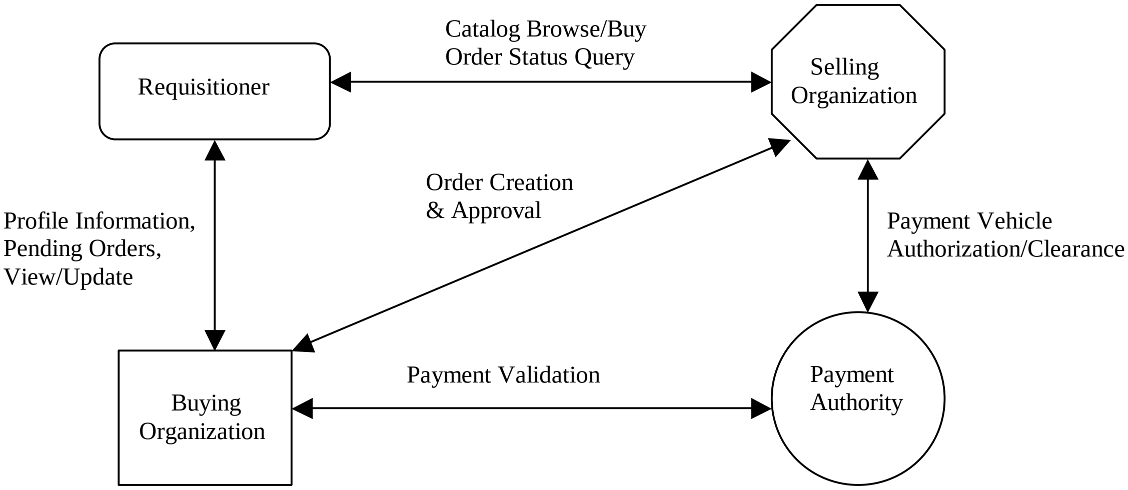


Figure 4.3.1 The OBI Architecture
(Source: SupplyWorks, Inc.)

The OBI architecture above follows the model below:

- *Requisitioner* is the individual that does the browsing of the catalog of goods and services and selecting catalog items.
- *Buying Organization* represents the Requisitioner in obtaining the catalog of goods and services and interacting with the Selling Organization and Payment Authority to complete a purchase order.

- *Selling Organization* represents the merchant in presenting the catalog of goods and services and dealing with the Buying Organization and Payment Authority in a purchase transaction.
- *Payment Authority* validates the payment, authorizes order fulfillment, and issues invoice.

4.3.2 Sequence of Protocol Transaction

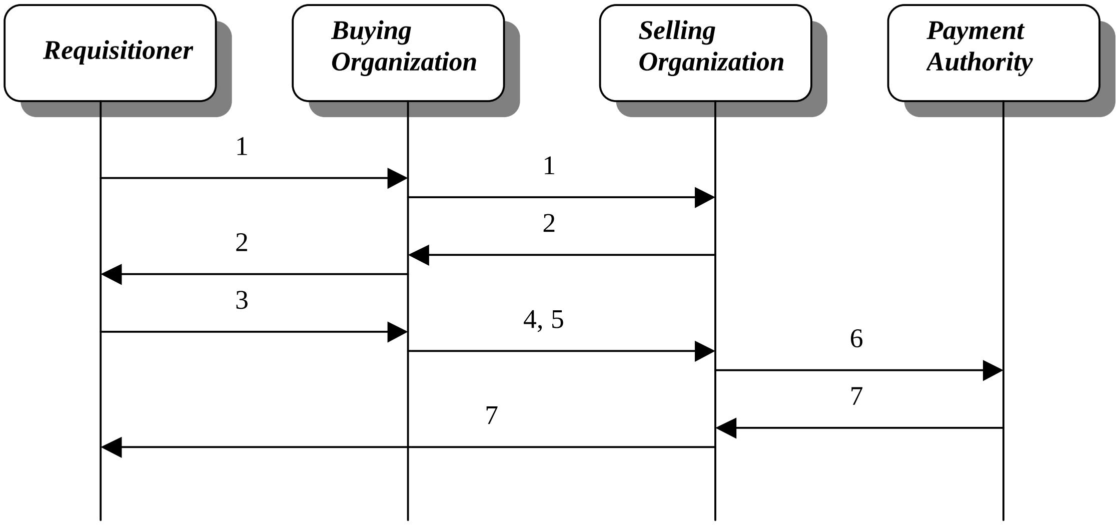


Figure 4.3.2 Sequence of OBI Transaction

1. A *requisitioner,* using a Web browser, connects to a local purchasing server located at the *Buying Organization* and selects a hyperlink to a *Selling Organization's* merchant server containing an online catalog of goods and services.

2. The Selling Organization's server authenticates the requisitioner's identity and organizational affiliation. Authentication information and optional profile information help to uniquely identify the requisitioner and to construct a specialized catalog view from which the requisitioner can select items and check out.

3. The content of the requisitioner's shopping cart is mapped into an order request (EDI-compatible). A digital signature is calculated (optionally); the order request (and digital signature if used) is encapsulated in an OBI object which is encoded and transmitted securely to the Buying Organization over the Internet using HTTP and SSL. The Buying Organization server receives the encoded OBI object, decodes it, extracts the order request, verifies the signature (if appropriate) and translates the order request into an internal format for processing.

4. Administrative information, including payment type, is added to the order request at the Buying Organization, and the order is processed internally either automatically or through a workflow-based process.

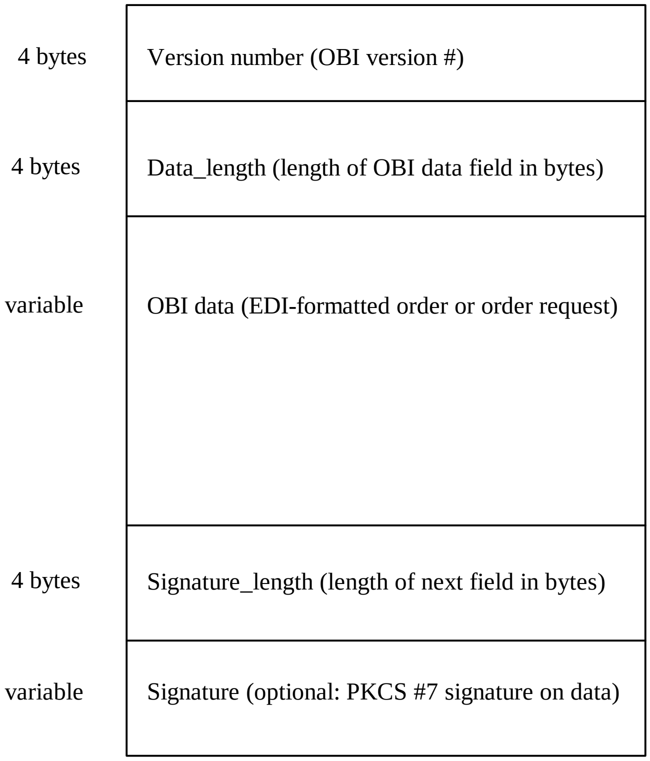
5. The completed and approved order is formatted as an OBI order (EDI-compatible) and a digital signature is calculated if desired. The order (and digital signature if used) is encapsulated in an OBI object which is encoded for transport and transmitted securely from Buying Organization server to Selling Organization server via the Internet using HTTP over SSL. The Selling Organization receives the encoded OBI object, decodes it, extracts the order, verifies the signature (if appropriate), and translates the order into its internal format.

6. The Selling Organization obtains authorization, if necessary, and begins order fulfillment.

7. The payment authority issues an invoice and receives payment.

4.3.3 Structure of an OBI object

Since OBI data must be received by the recipient in the exact form it was produced by the sender, data should be in a format which can be processed by any platform. An OBI object is the standard data structure used to exchange order-related data between trading partners. The structure of the OBI object is shown in figure 4.3.3a [16].



4 bytes	Version number (OBI version #)
4 bytes	Data_length (length of OBI data field in bytes)
variable	OBI data (EDI-formatted order or order request)
4 bytes	Signature_length (length of next field in bytes)
variable	Signature (optional: PKCS #7 signature on data)

Figure 4.3.3a Structure of an OBI object
(Source: "OBI Technical Specification",
OBI Consortium)

version

This is a four-byte integer field which uses a <major><minor> numbering scheme to indicate a version of the OBI object.

data_length

This is a 32-bit integer field which represents the number of bytes in the OBI data field.

OBI data

This is a variable-length string which contains an OBI Order or Order Request in an OBI-specified EDI format.

signature_length

This is a 32-bit integer field which represents the length in bytes of the signature field. This field must always be present and should be 0 if no signature is included.

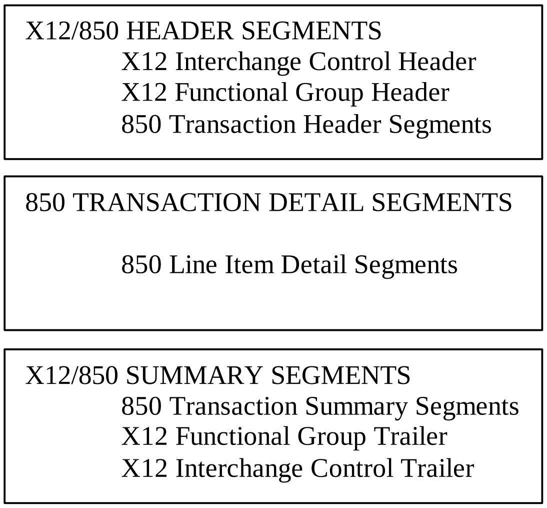
signature

This is a variable-length field which contains a signature on the contents of the OBI data field. The content of the signature field is a BER-encoded PKCS #7 data object. If no signature is included, this field is empty.

Data Structure

All OBI 850 transactions follow the structure shown in figure 4.3.3b [16]:

Within this structure, actual order data is carried in *data segments* which are made up of one or more *data elements*. Data elements contain actual data, identifiers, or codes. Specific data segments have assigned locations within the header, detail or summary areas [16].



X12/850 HEADER SEGMENTS
X12 Interchange Control Header
X12 Functional Group Header
850 Transaction Header Segments

850 TRANSACTION DETAIL SEGMENTS
850 Line Item Detail Segments

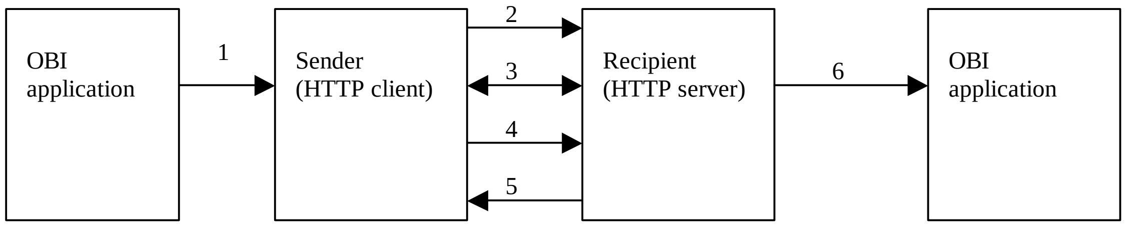
X12/850 SUMMARY SEGMENTS
850 Transaction Summary Segments
X12 Functional Group Trailer
X12 Interchange Control Trailer

Figure 4.3.3b OBI 850 EDI Data Structure
(Source: "OBI Technical Specification", OBI Consortium)

Transmission of OBI object

OBI defines two methods for transporting OBI objects between trading partners over the Internet: server-to-server method and server-browser-server method.

Figure 4.3.3c describes the transport of OBI object from server to server. To ensure portability, all OBI objects must be encoded in base64, as specified in Internet RFC 1521, prior to any HTTP transmission [16].



1 Base64 encoded OBI object arrives in sender's queue
2 Sender initiates TCP/SSL connection
3 SSL parameters negotiated
4 Sender transmits OBI object within HTTPS POST
5 Recipient commits OBI object to stable storage and acknowledges receipt with 200 response; sender removes object from queue
6 Recipient hands off OBI object for processing

Figure 4.3.3c HTTP Transport of an OBI object from server to server
(Source: "OBI Technical Specification", OBI Consortium)

The transmission of OBJ objects using HTTP makes use of the Secure Sockets Layer (SSL) Protocol which provides data encryption, server authentication, message integrity, and optional client authentication for a TCP/IP connection [16].

The server-browser-server method for transmission of order requests relies on the use of hidden fields within HTML forms and uses the requisitioner's browser during the transmission. Hidden values in an HTML form allowa CGI application to pass name-value pairs to a browser without the knowledge of the end user. Using the POST method, a browser can send name-value pairs from an HTML form to a CGI application via a Web server. The server then starts the designated CGI application and passes the browser-supplied data to the application [16].

4.3.4 Standards

The OBI architecture is built on existing standards in order to maximize interoperability and decrease implementation costs. Below is a list of standards that are used in OBI [16].

Purpose	Standard	Existing Examples
Content Display	Evolving standards for Web browsers (currently based on HTTP and HTML) as specified by the W3C	Netscape Navigator V3.0 or later; Microsoft Internet Explorer V3.0 or later
Order Requests and OBI Orders	X12 850 EDI standard	OBI/2.0 order format specification (defined by the OBI Consortium)
Order Transmission	HTTP 1.0 using SSL	HTTP servers available from many vendors including Netscape, Microsoft, Oracle
Secure Internet Communication	SSL V3	SSL supported by many vendors including Netscape, Microsoft, Oracle
Cryptography	SSL V3 API Public Key Cryptography Standards (PKCS)	Netscape SSL API RSA BSAFE Microsoft CryptoAPI
Public Key Certificates & Certificate Authorities	X.509 V3 certificates	GTE CyberTrust Verisign

Table 4.3.4 Standards relevant to OBI
(Source: "OBI Technical Specification",
OBI Consortium)

Content Display

This display involves a software which will be the standard for displaying information on requisitioner desktops and interfacing to most corporate applications including electronic ordering.

Order Formats

A common format for electronic orders is needed to provide communication between different platforms and implementation software and support interoperability among systems managed by different organizations and simpler interfaces to external systems. The ANSI X12 EDI syntax was used in OBI order format because it provides a proven encapsulation for the common purchasing documents and ease of connectivity with existing EDI systems in the U.S.

Order Transmission

A common protocol for transmission of standard order documents will provide interoperability among systems managed by different organizations. OBI uses HyperText Transport Protocol (HTTP) because it is a proven and widely adopted Internet protocol.

Security

Security for doing business over the Internet involves:

- *Secure Internet Communications* between World Wide Web servers and browsers
- Authentication with *public key certificates* for identification of individuals, organizations, and machines
- Data Confidentiality with *cryptography* for encrypting data to be transmitted over the internet

4.3.5 Error Handling

With server-to-server method of transmission, OBI does not define error messages for anything other than initial HTTP POST. Trading partners will need to agree on how they will handle other kinds of errors, for example, problem with the OBI object, problem with the EDI format, problem with the order itself, unable to verify a digital signature, no signature when one expected, etc. Trading partners should also discuss and agree on the time-out lengths for retramissions [16].

With server-browser-server method, if the workstation on either side crashes before the transaction completes, there is no way to automatically retransmit the message since there is no acknowledgement. Duplicates are also hard to track since the Selling Organization creates unique transaction ID for recreated order request.

4.4 OTP

What it is

OTP (Open Trading Protocol) is a protocol for the development of software products that will permit product interoperability for the electronic purchase that is independent of the chosen payment mechanism. It encapsulates the payment with the offers, invoice, receipts for payment and delivery. OTP allows two unfamiliar parties using electronic commerce capabilities to buy and sell that conform to the OTP specifications to be able to complete the business safely and successfully. In general, OTP supports familiar trading models, new trading models, and global interoperability [21].

Concepts

The Open Trading Protocols (OTP) define a number of different types of OTP Transactions [21]:

- *Purchase.* This supports a purchase involving an offer, a payment and optionally a delivery.

- *Refund.* This supports the refund of value as a result of, typically, an earlier purchase.

- *Value Exchange.* This involves two payments which result in the exchange of value from one combination of currency and payment method to another.

- *Authentication.* This supports the authentication of one party to make sure that another party is who they appear to be using a challenging-response mechanism.

- *Withdrawal.* This supports the withdrawal of electronic cash from a financial institution.

- *Deposit.* This supports the deposit of electronic cash at a financial institution.

These OTP Transactions are "Baseline" transactions since they have been identified as a minimum useful set of transactions. Later versions of OTP may include additional types of transactions [21].

Each of the OTP Transactions above involve a number of organizations playing a *Trading Role*, and a set of *Trading Exchanges*. Each Trading Exchange involves the exchange of data, between Trading Roles, in the form of a set of *Trading Components.*

Trading Roles

The Trading Roles identify the different parts which organizations can take in a trade. The five Trading Roles used within OTP are illustrated in the diagram below [21].

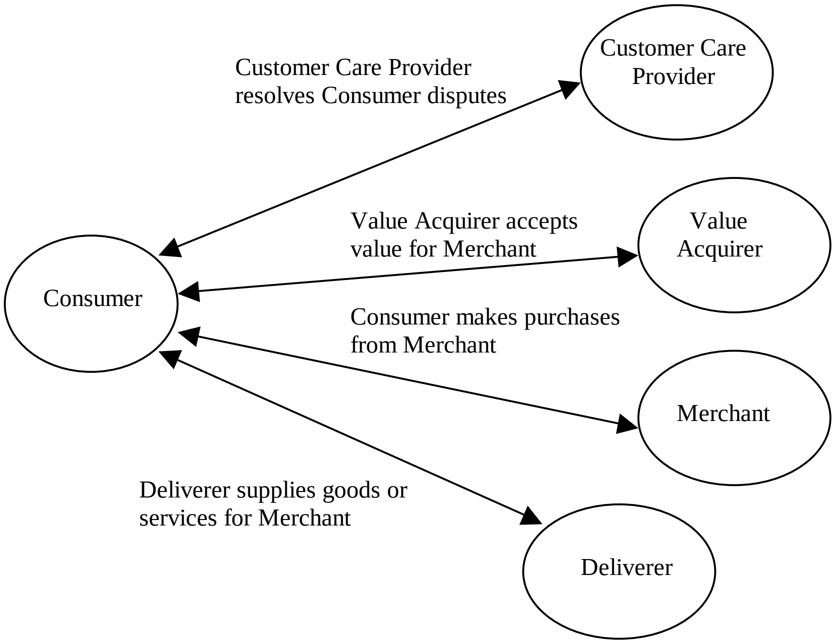

Customer Care Provider resolves Consumer disputes
Customer Care Provider
Value Acquirer accepts value for Merchant
Value Acquirer
Consumer
Consumer makes purchases from Merchant
Merchant
Deliverer supplies goods or services for Merchant
Deliverer

Figure 4.4 OTP Trading Roles
(Source: "Internet Open Trading Protocol Specification", OTP Consortium)

The *consumer* receives and pays for the goods or services. The *merchant*, with whom the purchase is being made, is responsible for providing the goods and services and receives the benefit of the payment made. The *value acquirer* is the entity that actually receives the payment from the consumer on behalf of the merchant. The *deliverer* delivers the goods or services to the consumer on behalf of the merchant. The *customer care provider* handles customer dispute negotiation and resolution on behalf of the merchant. These roles can be carried out by the same organization or different organizations.

Trading Exchanges

OTP supports four Trading Exchanges which involve the exchange of data between the Trading Roles. The four Trading Exchanges are:

- The *Offer Exchange* which provides the consumer with the reason why the trade is taking place. It is called an Offer since the consumer must accept the Offer if a trade is to continue. The Offer generally includes Order, Pay Amount, Delivery Information, and necessary signatures.
- The *Payment Exchange* which is some value that is being transferred between the consumer and the value acquirer. However, the list of payment types are offered from the Merchant. After a payment type is selected, communication between the consumer and Merchant may proceed.
- The *Delivery Exchange* which transmits either the online goods or delivery information about physical goods from the deliverer to the consumer. The merchant must first provide consumer with Delivery Information which includes order, shipping tracking number, and its signature before the consumer can request delivery from the Deliverer.
- The *Authentication Exchange* which is used by any Trading Role to authenticate another Trading Role.

OTP Transactions are composed of various combinations of these Trading Exchanges. For example, an OTP *Purchase* transaction includes *Offer, Payment,* and *Delivery* Trading Exchanges. An OTP *Value Exchange* transaction, for example, can be composed of an *Offer* Trading Exchange and two *Payment* Trading Exchanges. Trading Exchanges consist of Trading Components that are transmitted between the various Trading Roles.

4.4.1 Architecture

The OTP design involves the five Trading Roles and four Trading Exchanges (described earlier). The interactions among these components are shown in figure 4.4.1 below.

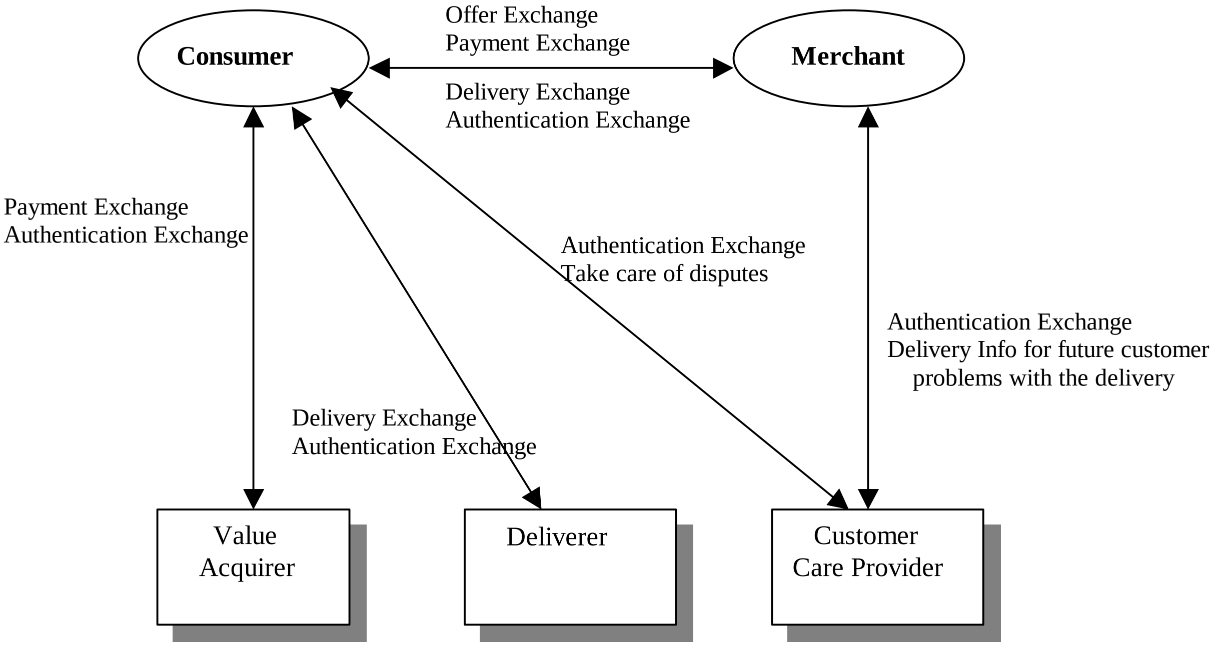


Figure 4.4.1 OTP Architecture

The transactions that can take place with OTP protocol are Authentication, Deposit, Purchase, Refund, Withdrawal, Value Exchange, and Restart Transaction.

4.4.2 Sequence of OTP Transactions

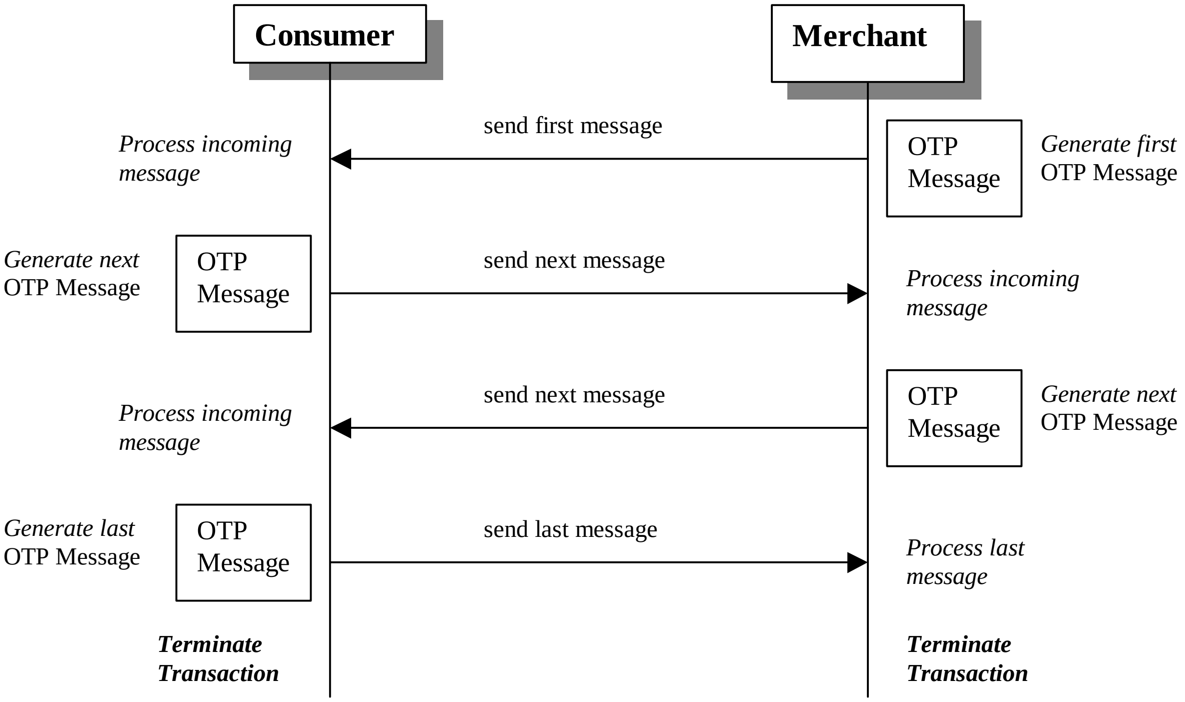


Figure 4.4.2 An OTP Transaction

Transactions can be terminated by any of the following:

- the Consumer
- one of the other Trading Roles in the trade
- an error in the OTP Message
- a cancellation by one of the Trading Roles because it could not accept the transaction

4.4.3 OTP Message Structure

Figure 4.4.3 shows the structure of the OTP message.

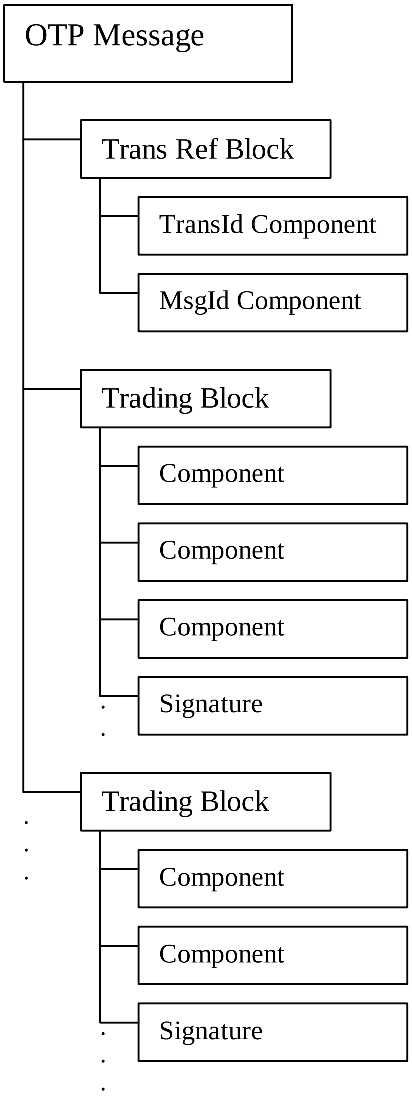


Figure 4.4.3 OTP Message Structure
(Source: "Internet Open Trading Protocol Specification", OTP Consortium)

OTP Message – an XML Document which is transported between the Trading Roles.

Trans Ref Block – contains information which describes the OTP Transaction and the OTP Message.

Trans Id Component – uniquely identifies the OTP Transaction. The Trans Id Components are the same across all OTP messages that comprise a single OTP transaction.

Msg Id Component – identifies and describes an OTP Message within an OTP Transaction.

Trading Block - an XML Element within an OTP Message that contains a predefined set of Trading Components.

Trading Components – XML Elements within a Trading Block that contain a predefined set of XML elements and attributes containing information required to support a Trading Exchange.

Signature Component – an optional XML element that contains a digital signature. The signature may sign hashes of the Trans Ref Block and any Trading Component in any OTP Message in the same OTP Transaction. Other components may hold a certificate for use with the signature.

As part of the message, OTP also provides for *Opaque Embedded Data* to be embedded within OTP messages, such as Order Component, Brand Element, Pay Protocol Element, and Payment Scheme Component. The embedded data is not processed by OTP but is instead passed to or from order or payment processing software.

One or more Trading Blocks may be included in each OTP Message. Each of these Trading Blocks may consist of one or more Trading Components and optionally one or more Signature Components. The OTP Messages are physically sent in the form of an XML documents between different Trading Roles.

The Trading Blocks available in OTP are TPO (Trading Protocol Options) Block, TPO Selection Block, Offer Response Block, Authentication Request Block, Authentication Response Block, Payment Request Block, Payment Exchange Block, Payment Response Block, Delivery Request Block, Delivery Response Block, and Failure Block.

Some of the Trading Components that OTP supports are Protocol Options Component, Authentication Component, Authentication Component, Authentication Response Component, Order Component, Organization Component, Brand List Component, Brand Selection Component, Pay Amount Component, Payment Scheme Component, Pay Receipt Component, Delivery Component, Delivery Component, Signature Component, and Certificate Component.

Each of these Trading Components has its own DTD (Document Type Definition) which is a set of rules for a document format. The DTD is explained further in the XML Specification [8]. This separation of DTD for each Trading Component allows for easier addition of elements or changes to the format of the Trading Component and the OTP Messsage since the Trading Component is part of the message.

Since this paper only describes briefly what OTP protocol is and how it works, an explanation of XML language will not be discussed in detail here. To see what OTP messages in XML format looks like and to understand its contents, one should refer to the XML specification [8].

4.4.4 Error Handling

The OTP protocol also has support for handling some types of errors during the transmission of a message.

When a message is sent, a timer is started. If a reply is not received after a period of time, the OTP Message is re-sent. The *Status* attribute of the message is changed, in this case, from *Original* to *Resend.*

If there are failures with security checks or verification of signatures, a log will be created. However, this will be considered as an incorrect response and therefore will not cause the timer to reset. The number of retransmissions allowed for repeated time-outs depends on the retransmission limit set before the start of the original transmission.

OTP Messages received are usually being checked for duplicates. If duplicate is found, the message is ignored. In order to handle duplicate messages, a local cache or database of OTP Messages is kept.

4.4.5 Security

OTP uses XML elements to support data confidentiality and authentication. Signatures for each OTP transaction are calculated and encoded as XML elements based on some hashing rules and guidelines developed by OTP Consortium. OTP signatures support both asymmetric (public-key) signatures and symmetric (secret key) message authentication codes (MACs) [21]. The diagram below gives an overview of how signatures are generated.

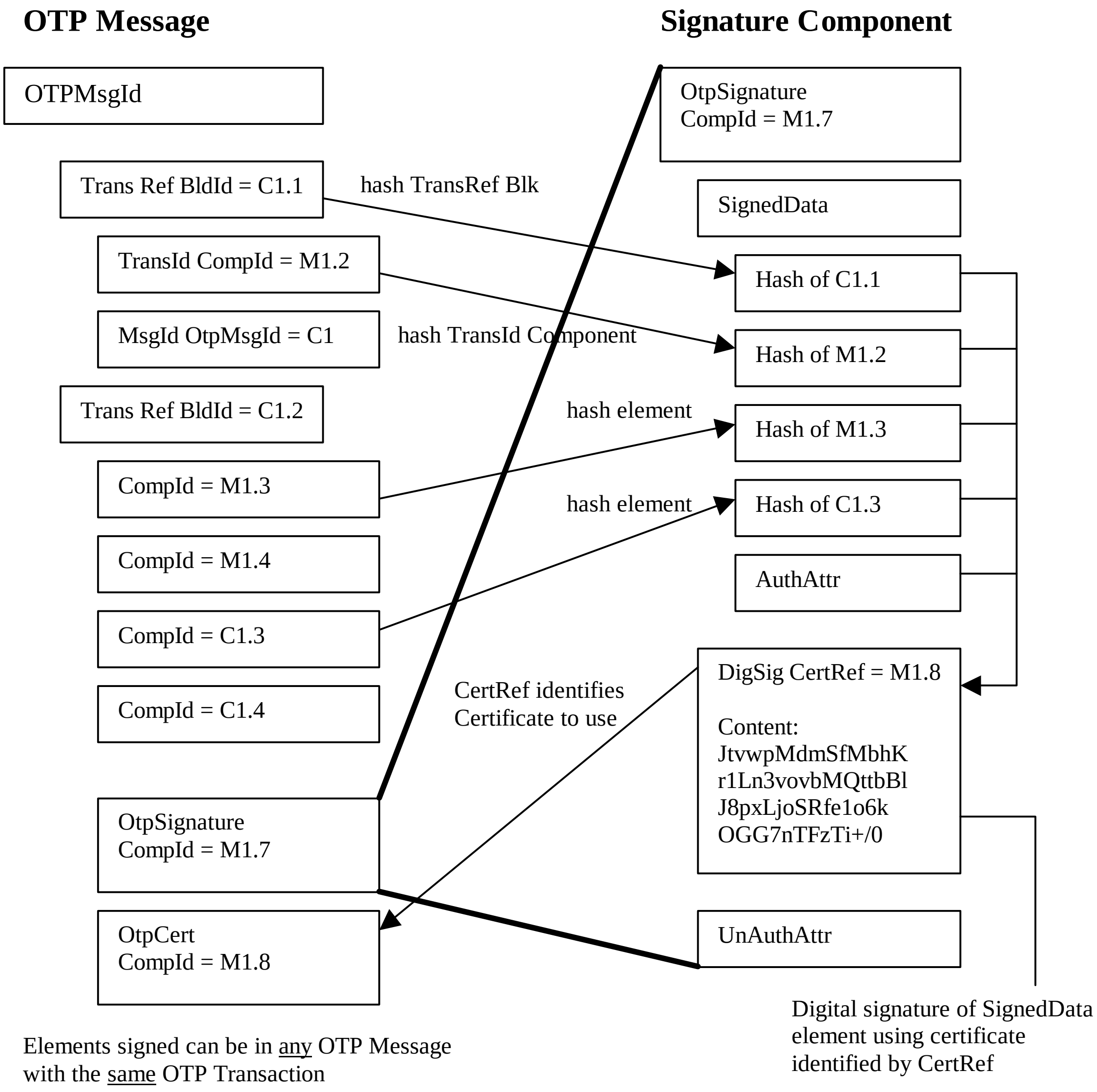


Figure 4.4.5 Signature Overview
(Source: "Internet Open Trading Protocol Specification", OTP Consortium)

Both the Transaction Id Component and the Trans Ref Block are required because the OTP protocol structure requires verification of the Transaction Id Component but not the Trans Ref Block in some situations [21].

Note that the Components and Blocks being signed may be in any OTP Message in the OTP Transaction.

4.4.6 Standards

A list of standards used in OTP is shown in the table below.

Purpose	**Standard**	**Examples**
Content Display	Current standards for Web browser (e.g. HTTP and HTML)	Netscape or Internet Explorer
Order Request	W3C XML 1.0	OTP Message Structure with different Trading blocks and components, each being specified by DTD
Order Transmission	Depends on the transport protocol used by the trading party	HTTP 1.0
Secure Internet Communication	Depends on the secure communication protocol used by the trading party	SSL
Cryptography	Depends on the trading party; but the encryption used must follows existing standard	Public Key encryption

Table 4.4.6 Standards used in OTP

5 ANALYSIS AND COMPARISON OF EXISTING TRANSACTION PROTOCOLS

5.1 Architecture

Among the four protocols being discussed, EDI has the simplest and most straightforward architecture with just one EDI translator at the sender and receiver sites.

OBI architecture is very modular with four main entities (Buying Organization, Selling Organization, Requisitioner, and Payment Authority), each having its own set of responsibilities and functionalities. Even though there needs to be interactions among these four entities, the communication among them are very straightforward. The complication that OBI might have is with the structure of the OBI object. Since part of the OBI object which is the data part is in EDI format, an EDI translator which conforms to X.12 standard must also be installed at the Buying, Selling, and Requisitioner in order to translate and process the order.

In SET, the architecture involves four participants: Cardholders, Merchants, Acquirer Payment Gateways, and Certificate Authorities. Each of these participants has its own responsibilities and must interact with each other in order to complete a transaction. The complication with the SET protocol is that there are many message pairs (Request/Response), and they must be used in the correct sequence in order for the transaction to succeed. Also, there is a concept of electronic wallet and certificate authority that must be understood thoroughly before implementing or using SET since they are the main factors of SET's processing and authentication.

Similar to SET, OTP also involves some Trading Roles: Consumer, Customer Care Provider, Value Acquirer, Merchant, and Deliverer. Since OTP supports different kinds of transactions, it provides different types of exchanges that can be included in each transaction. The difference between SET and OTP is that OTP provides more flexibility in transacting any types information between the merchant and the consumer while SET only offers specific messages which must be used in correct sequence.

5.2 Message Format

There are several different kinds of heterogeneity in network information systems. We need to resolve this by coming up with a structured mapping system that can translate the information from different system formats to a standardized one that everyone can understand, and vice versa. To accomplish this involves some machine learning techniques [1].

- *Lexical differences*

Synonyms problem occur when objects with different names represent the same concepts, and homonyms occur when the names are the same but different concepts are represented. Synonyms are resolved by consulting a synonym lexicon while a domain-specific ontology are necessary to identify homonyms.

- *Structural differences*

  Structural heterogeneity occurs since the participating information sources are designed individually. A data item may be designed as an entity on some systems and as an attribute on others.

- *Semantic differences*

  It occurs when there is a disagreement about the meaning, interpretation, or intended use of the same or related data. Identifying semantically related objects and resolving the schematic differences is the main approach to information system interoperability.

EDI (Electronic Data Interchange) attempts to handle these differences by requiring that trading parties wanting to communicate their data to each other must first agree on a standard machine-readable message format. Then, an EDI message translator software must be installed at all the sites of the parties involved. This EDI message translator will be used to translate data received in EDI standard format into a human-readable format which is fed into the local system application for processing. However, this strict requirement limits EDI protocol to support only one-to-many communication as we shall see in section 5.6.

SET does not follow any standard format for the message. Each of the message sent between the trading partners has its own format (see section 4.4.2). SET is mainly responsible for the transmission and verification of the message which includes two parts: PI (Payment Information) and OI (Order Information). One advantage of separating the message into PI and OI is that the merchant will be able to see or modify the customer's personal financial information since the payment information part of the message is sent directly to the Payment Authority. The disadvantage with this is that there will be one more message to process and protect for each transmission. Also, the burden of having to keep track of different formats for different types of messages is enormous.

Like EDI and SET, OTP converts the data into certain format before sending it to the trading partner. However, OTP doesn't follow a standardized message format like EDI nor does it have its own format like SET. OTP uses the general XML technology to format the data and send it as an XML document to the trading partner. Since XML has been accepted by the W3C Consortium as the standard markup language for the web, it is the best tool to use to format the data to be sent to any trading partner because any party that has an XML parser installed at their site will be able to understand it. Thus, as far as message formatting is concerned, OTP beats EDI and SET by far.

Because OBI order format is designed to support high-volume, low-dollar transactions involving non-production goods and services, its specification restricts the use of 850 data segments and data elements to those required for these kinds of transactions in order to simplify implementation and ensure interoperability. As a result, the OBI format will not support all types of purchasing transaction, especially the complex, high-dollar transactions or the acquisition of production good and services. For this reason, OBI also loses to OTP in message formatting.

5.3 Sequence

Sequence of EDI purchase orders are only guaranteed if VAN (Value Added Network) is used since it provides automatic acknowledgment of receipt of a transaction.

Since each SET transaction is sent with the party's certificate and transaction ID which provides information that uniquely defines the transaction and the transaction characteristics the particular message possesses, SET technology ensures proper transaction sequencing and ordering.

All OTP transactions and messages have Ids attached to them. This implies that sequence of messages sent is preserved. This is also true with OBI protocol.

5.4 Reliability

The reliability of any transaction protocol depends on the underlying communication protocol being used. For EDI, the most common transport protocol currently being used is SMTP (Simple Mail Transport Protocol) with S/MIME (Secure/Multipurpose Internet Mail Extensions). S/MIME specifies formats and procedures when the cryptographic security services of authentication, message integrity, non-repudiation of origin, and confidentiality are applied to Internet MIME messages [5]. S/MIME protocol's ability to enable e-mail applications to verify transmission and receipt of EDI messages contribute to the reliability of EDI [11].

SET uses either HTTP (HyperText Transport Protocol) or SMTP (Simple Mail Transport Protocol) as network transport protocol to transmit data from one party to another. Because of some problems that SET has with HTTP and SMTP, SET cannot guarantee reliability of data transport without workarounds.

Since OTP protocol supports retransmission of messages in case of errors or time-outs and handles duplicates, it is considered to have high reliability compared to EDI and SET.

Yet the protocol that has the highest reliability among these four protocols should be OBI. OBI uses HTTP/SSL to transport OBI objects which are encoded in Base64. Since HTTP provides reliable transport of data and SSL provides the security needed to protect the data being transmitted over the network, OBI is guaranteed to deliver the most reliable transport of data, and even more so with the OBI object being encoded using Base64.

5.5 Security

Since the main goal of EDI is to provide a standardized message format which can be used by all trading partners to communicate transactions such as purchase orders, shipping information, and payment, security was not included in the EDI message translating and transmitting process and was left to the application or the underlying transport protocol to handle. Since EDI mostly uses S/MIME as transport mechanism, cryptographic security services of authentication are guaranteed .

5.5.1 Data Confidentiality

Confidentiality requires that all communications between parties are restricted to the parties involved in the transaction. Data confidentiality is usually achieved through the use of encryption.

EDI protocol itself does not handle the confidentiality of the data being sent between trading partners. The data communication software used to transport EDI files usually includes some security software that will ensure the confidentiality of the data.

Confidentiality should be ensured by SET's use of message encryption, and payment information integrity can be protected by the use of cryptographic "hashing" and "digital signatures." The linking of cardholders and merchants to specific accounts is addressed by the use of digital signatures and certificates (the specifics of certificate use will be described later) [13].

SET is not dependent on any transport security plan (like S-HTTP or SSL). It generates its own security when needed. Additional security can be wrapped around a SET session but is not required to assure confidentiality.

Since SET design involves more than two parties – a consumer, a merchant, and one or more banks, it is considered a multiparty, rather than a point-to-point, protocol. Thus, it has the ability to guarantee integrity or confidentiality of messages in a multihop environment. In a SET transaction, the order and credit-card components are encrypted separately, so a cardholder's account information can be channeled to the bank acquiring the transaction without being exposed to the merchant.

OTP protocol provides an option of encrypting messages. Its encryption follows an industry standard form of encryption. Thus, transactions carried out over the Internet are hidden from unauthorized individuals [20].

Because OBI uses HTTP and SSL as transport mechanism to transmit messages from server to server, it guarantees data confidentiality.

5.5.2 Authentication

Authentication is a way of verifying that one party is communicating with the party with whom they think they are doing business.

EDI does not provide this feature but relies on other authentication technology like digital certificates or the use of passwords to verify the identities of the parties involved in the communication of a business transaction.

In SET, for a purchase to proceed, banks must first authenticate their relationships with the cardholder and the merchant by providing digital certificates. Then they must certify to the merchant that it will be paid. Therefore, authentication is also guaranteed with SET.

Unlike SET, OTP's authentication is embedded inside the OTP Message as an Signature Component. Since each OTP Component has its own format or set of rules of how its content look, OTP provides for even greater flexibility in adding or changing the format of the signature at any time. Thus, this makes OTP more powerful than EDI and SET as far as authentication is concerned.

OBI uses SSL V3 protocol and X.509 digital certificates to provide the Selling Organization with cryptographic assurance of the identity of the requisitioner and the requisitioner with cryptographic assurance of the identity of the catalog server. The combination of these two standards help make security of OBI protocol as strong as that of OTP. However, OBI probably cannot provide the flexibility that OTP has.

5.5.3 Data Integrity

Data sent as part of a transaction should not be modifiable in transit. Data integrity is a guarantee that what was sent by the sender is actually what is received by the receiver. The normal mechanism for acquiring data integrity is for the sender to run an algorithm against the data that is being transmitted and to transmit the result of the algorithm separately from the transmission. Upon receipt of the transmission, the receiver runs the identical algorithm and then compares the results. If the results are identical, then data has been modified [4]. Because of the complexity of this process, data integrity is rarely provided by most trading partners. For this reason, EDI does not support this feature.

Since SET transactions also include payment information, SET also needs to protect this payment information through the use of cryptographic "hashing" and "digital signatures".

Because OTP provides for strong data confidentiality and authentication, data integrity is preserved. Hashing is also used in OTP to protect the data being sent between trading partners. Therefore, to determine if SET provides better data integrity than OTP depends on the hashing algorithm used in each protocol.

Because OBI uses HTTP and SSL as transport mechanism to transmit messages from server to server, it also guarantees data integrity.

5.6 Flexibility

EDI

Due to its fixed form of standard transaction, EDI protocol does not allow for flexibility in data transaction and requires human intervention in translating the data to and from the EDI standard form. In fact, it has been reported that over 70% of all EDI actually takes place by emailed or FAXed EDI forms which are both written and read by human beings, with computers having no role in the transaction other than in transmission. [1]

Also, because EDI follows a particular standard, it only provides a one-to-one communication in that both sides must agree on an EDI standard. Thus, communicating to a third party would require another agreement of the EDI standard. For example, if X can “talk” to Y because they have agreed on data-exchange standards, and W can “talk” to Z with a different set of standards, how can X talk to Z? Such generalized communication can only come with widely accepted EDI standards. Government efforts may be necessary to unify interest groups around one standard. Thus, this involves more complicated issues and politics. [13]

SET

As far as communication issue is concerned, SET does allow for communication among multiple trading partners as long as they all follow SET’s rules and use the appropriate message format for certain Request/Response message.

As far as being flexible in terms of changes as business warrant, SET allows for changes within itself by the use of extensions. An extension field has been added to the end of certain SET protocol data units to support this flexibility [13]. When a message is sent, the recipient of the message checks for this extension field in the data to decide how to process the message. Therefore, modified or new features can be easily applied to SET without too much delay since the way the message is processed is dependent on this extension field. If an application has support for certain message extension, it will process the message accordingly. On the other hand, if the application does not support certain message extension, it can either reply with an error or ignore the message based on the criticality of the message. Yet, if an application wants to add support for certain message extension, it can do so easily.

OBI

Since OBI uses EDI X.12 standard for OBI data formats, OBI faces the same problem as EDI with being able to communicate messages among multiple trading partners. However, since OBI provides flexibility in payment, it can be used with any type of reimbursement model. OBI also supports a variety of purchase authorization and approval processing methods which help streamline transaction processing.

OTP

Since OTP uses XML technology to format and parse the data, any trading partner with an XML parser installed will be able to understand the document. For this reason, OTP is considered to have support for one-to-many or many-to-many communication. Also, because of the use of XML language, OTP is able to provide flexibility in adding new types of OTP Transactions to the OTP Baseline through extra XML elements and new user-defined values for existing OTP codes.

5.7 Performance

With the existing EDI design, before a transaction is sent to the trading partner, it is first converted into EDI standard format. When the trading partner receives the EDI file, it has to first translate this file into a human-readable format. Then, this data is fed into the receiver's application for processing. This whole process is done automatically if both trading partners have already agreed to a standard EDI format. In this case, EDI's performance is very good. However, if a party wants to communicate with another party for the first time without having an agreeable message format, the EDI data that is received at the other party must first be looked at by an application developer in order to understand the data and modify the application accordingly before feeding the data into the application for processing. In this case, it can take a lot of time and overhead. Therefore, the performance of EDI protocol might not be very good.

With SET, on the other hand, because it can use different levels of encryption for different components, messages can be smaller, allowing faster throughput, up to twice as fast as SSL (Secure Sockets Layer) in IBM tests.

Because of the complicated message structure that OTP has with multiple blocks and multiple components having their own formatting rules, OTP's performance might be much slower than that of SET or even EDI.

Since OBI message structure is very simple, OBI messages can be transmitted very fast. However, because OBI data is in EDI format, OBI will have some slow down similar to EDI at the receiver's side for the translation and processing of the data.

5.8 Scalability

Currently, EDI follows X.12 standard format for messaging which is only being used domestically within the United States. Therefore, only a small number of businesses which have adopted the X.12 standard for EDI can communicate with each other. With direct communications among trading partners, the overhead will be even higher because of the different implementation for different transport protocols. Thus, this would reduce the scalability of EDI. If a third-party service provider, such as VAN, was used, EDI would scale better with one party being able to send hundreds and even thousands of transactions with less overhead since VAN will take care of the issues with implementing and maintaining different transport protocols.

SET involves basically three servers: Cardholder Internet browser, Acquirer Payment Gateway, and the Merchant network which includes Web server, Application server, Database server, and security server. Each of these servers provides different set of functionalities that help manage and process messages at different phases of a SET transaction. Since each of servers is concentrated on specific parts of the transaction and is not overloaded with too much work, they can provide very high scalability compared to EDI where each of the servers on both the sender's and receiver's side has to handle most of the issues with the whole EDI transaction.

OTP uses abbreviated versions of XML Element and Attribute names in version 1.0 of the protocol in order to reduce the physical length of OTP Messages. Thus, this reduces bandwidth and allows for more messages to be sent, giving OTP higher scalability.

Since OBI involves transacting businesses between two organizations instead of between individuals to an organization, it provides for a way of making multiple big purchases simultaneously and of processing and delivering the goods quickly because of the need from the Buying organization to continue with its business without delay of waiting for the goods on order. Thus, OBI's overall system must be capable of scaling to thousands of Buying Organizations and Selling Organizations. Its design and architecture must not limit the potential for growth in number of users, transactions, or organizations. OBI's ability to carry through its desired goal must scale very well in order to be successful with the transaction. For this reason, OBI is probably the most scalable protocol among the four discussed here.

5.9 Exception Handling

Since EDI relies on the underlying transport protocol for reliability of message transmission, EDI provides no exception handling when a problem like delayed or lost transmissions and duplicate transmissions occur.

SET, on the other hand, provides for error handling based on the classification of the error. If a message is a duplicate, it will be ignored. Corrupted (non-parsable) messages will be retransmitted. Malformed messages will be retransmitted if desired. If authentication fails, the message will be delayed and generic to prevent possible attack.

OTP also handles errors with security failures, time-outs, and duplicates. However, OTP doesn't check for malformed messages or messages which do not follow their own formatting rules. This will put more work on the application to handle the errors with the message itself. With SET, the malformed messages are also taken care of and are retransmitted if desired.

OBI does not provide any error handling mechanism or error message definitions. Management of any kinds of errors must be discussed and agreed upon by trading partners involved.

5.10 Customer Classification

Customer classification is way in which the merchant classifies the customers based on the items they purchased. These customers' profile are then stored in the database. When there are changes in the items, products, or brands that the customers are interested in, a notification will be sent to the customers.

EDI handles business transactions between trading partners and therefore does not maintain customer profile. Thus, it doesn't support the feature of classifying customers based on the catalog information in order to send updated changes to them. Like EDI, both SET and OTP do not have support for customer profile.

OBI is probably the only protocol that keeps track of customer profile and provides a feature very similar to the customer classification described above. When the requisitioner browses the supplier's catalog, OBI can specify that the catalog seen by the company has been tailored just for them, possibly having been generated dynamically from the database of the supplier. The catalog must have search facilities and also a list of items which are ordered frequently. Information about the buyer is held on a user profile database, including authorization limits, shipping preferences and billing [7]. This customer profile is usually used to uniquely identify requisitioner and establish appropriate catalog view for requisitioners [16].

5.11 Shipping and Receipt Confirmation

As part of EDI design and implementation, order confirmation/notification is automatically sent to from the seller to the buyer acknowledging the receipt of purchase order delivery if a VAN was used. Shipping confirmation, however, is not handled by EDI but is left to the application to implement. However, EDI can be used to handle the transmission of shipping information from the seller to the buyer.

Unlike EDI, SET does support acknowledgement of payment to the buyer. It also provides a way in which the buy can request (via InqReq message) the status of the order. SET, however, does not send the status to the buyer automatically. The shipment of goods or the arrangement of shipment is left up to the merchant, not in SET.

OTP does support the delivery and receipt confirmation. This is embedded inside an OTP Message as one of the Trading Components. This makes OTP a very complete protocol for Internet business transactions.

With OBI, Selling organization always provides requisitioner with receipt-of-order acknowledgements and the capability of turning this feature on or off [14].

5.12 Application

Typical applications of EDI include purchase orders, acknowledgements, pricing schedules, order status inquiries, shipping and receiving scheduling and confirmation, invoices, and payments. EDI is, in essence, electronic paperless contracting between two commercial parties [10]. One important point to note is that all of the applications of EDI mentioned here are *not* automatic, meaning the business application has to initiate the action using EDI.

Unlike EDI, SET protocol is mainly used for payment transaction. But due to its separation of the message into PI (Payment Information) to be sent to Payment Gateway Authorization and OI (Order Information) to be sent to the merchant, it can also be used for purchase order or other transactions.

The OTP protocol is designed to be applicable to any electronic payment scheme since it targets complete purchase process where the movement of electronic value from the payer to the payee is only one, but important, step of many that may be involved to complete the trade [21].

OBI is also trying to handle all kinds of purchases from all kinds of different organizations. It is applicable to any e-commerce transactions. However, since it follows SET payment cards, OBI only supports the payment types that are included in SET.

5.13 Summary

The four e-commerce transaction protocols, EDI, SET, OBI, and OTP, have been described, analyzed, and compared in this paper. Their features and architecture are summarized in the table below.

Protocol	**Architecture**	**Message Format**	**Sequence**	**Exception Handling**	**Confirmation**	**Customer Profile**
EDI	Simple	Complicated	Maybe*	No	Maybe*	No
SET	Complicated	Complicated	Yes	Yes	Yes	No
OBI	Efficient	Manageable	Yes	Yes	Yes	No
OTP	Complicated	Manageable	No	No	Yes	Yes

***** Depends on other factors

Table 5.13a Summary of Transaction Protocol Features

The history of these protocols are also described in the table below with the protocols' application, original designer/architect, date established, and existing clients using the protocols.

Protocol	Application	Originator	Date Established	Some existing clients
EDI	Purchase orders, acknowledgements, invoices. Cannot be used for payment transaction.	Accredited Standards Committee	Late 1970's	U.S. Government agencies and commercial businesses like Lotus Notes, HP, and Oracle
SET	Payment Transaction and Order Transaction	VISA/MasterCard	Feb. 1996	Banks like American Express, Novus/Discover and technology companies like Microsoft, Netscape
OBI	All kinds of e-commerce transactions	Internet Purchasing Roundtable (Fortune 500 Companies)	Oct. 1996	Motorola, Office Depot, Microsoft, Oracle, Commerce One
OTP	Any electronic payment scheme.	OTP Consortium which includes leaders in Internet Commerce like HP, IBM, Oracle, Microsoft, SUN, MasterCard	1998	No clients yet

Table 5.13b Profile of Transaction Protocols

The following two sections will discuss the benefits and limitations of these four protocols.

5.13.1 Benefits

One of the major benefits of EDI is the structured and organized format of the messages being sent between two parties. This structure in message format allows for modularity in implementation of the system and better maintenance of the system. Although the fixed message standard limits EDI's flexibility, this fixed standard, if approved world-wide, can make business transactions on the Internet faster and much simpler to maintain since there is only one standard to worry about.

Also, as more businesses begin to use EDI, EDI has expanded from handling transmission of only purchase orders and invoices to handling payment (with FEDI) and other valuable trading information among business partners.

SET also offers some benefits that make it a little bit more powerful than EDI. SET supports interoperability across a variety of hardware and software platforms without preferential treatment for any combinations. It is built using specific protocols and message formats that provide the degree of interoperability desired [14]. Distributed SET also supports centralization which permits the programmer to control and secure programs and servers under a mainframe-like environment that’s scalable, predictable, and easily monitored. Flexible and well-defined software layers permit the highest degrees of responsiveness to changing business needs, allowing for fast adoption of improvements in technology.

OTP provides a standard framework for encapsulating payment protocols which makes it easy to incorporate payment products into OTP solutions. Thus, this makes payment brands more widely distributed and available on a wider variety of platforms.

New merchants will be able to enter the new Internet market-place with new products and services using the new trading opportunities which OTP presents. Banks and financial institutions can use OTP related services to provide customer care for merchants and collect fees from processing new payments and deposits. They will also have an opportunity to build relationships with new types of merchants.

With OTP, customers have a larger selection of merchants with whom they can trade. There is a more consistent interface when making the purchase. Customer problems can be fixed through the merchant, rather than the bank. A record of the customer transaction is also saved in the accounting systems which can potentially be presented to the tax authorities [20].

Also, with the way OTP is structured, any kind of transactions such as purchase order, payment, withdrawal, and refund can be done with OTP whereas EDI only supports purchase order transactions and SET only supports purchase order and payment transactions, with concentration more in payment.

In addition to providing various different types of transactions, OTP can also be easily integrated with EDI and SET because of its common functionalities with these two protocols.

Since OBI is designed for business-to-business commerce, it supports the three-way relationship that exists in organizational buying. Consumer payment cards can tie a person to not only a Brand, but also to a company which in turn is tied to a payment card.

Order forms are also autofilled with standard user profile information regarding the authorized Requisitioner to minimize the data entry errors and speed the ordering process takes. There is a default tax status code in the User Profile that can be overridden at the order level and/or the line item at the time of order. The tax status code can also be passed through to the Selling Organization. The Requisitioner can specify special instructions at the time of order and request order status and order history. Another benefit of OBI is that Selling Organization can determine whether an order they have received has been “authorized” by the Buying Organization [14].

OBI provides easy-to-use, efficient process for end-user. It integrates suppliers from sourcing through payment, enables suppliers to meet the unique requirements of each customer through one standardized method, provides a vehicle for differentiation in the marketplace, and allows customers to order from multiple suppliers who do not have compatible catalogs or ordering format [3][16].

5.13.2 Limitations

Proposal has been made to provide support for interoperability among different Electronic Data Interchange (EDI) standards and other electronic forms. However, if we have millions of different EDI standards for millions of different trading parties, we would need a very complicated software to translate between different message format standards in order to have one-to-many or many-to-many communication among all trading parties. Also, the more complicated the translation software is, the slower it would take to complete a message translation and transaction. Thus, this will degrade performance tremendously which is not at all acceptable in the electronic business.

Also, since the EDI standard is derived from the agreement of a particular version of a selected standard from all business parties involved, the whole EDI standardization process involves a lot of bureaucracy and costs a lot of long hours to complete. The bureaucracy is involved because many senior personnel have to work together to come up with a compromising standard that can be used by everyone. Cost is also a factor of the standardization process because it involves taking a lot of time for negotiation of a standard and sometimes re-negotiation of the existing standard when new businesses join the community. Businesses are both unwilling and unable to be put on hold for this length of time [1].

Another limitation that EDI is facing is with the development and usage of the EDI system. If EDI message translator was made so smart as to resolve many standardization issues among many different businesses, it would be very complex and difficult to use. It would cost more time and money to train the staff to learn and maintain the system.

In short, EDI is being held back from becoming widely used by businesses because of the following reasons:

1. inflexibility and complexity for users partly due to the antiquated X12 and EDIFACT standards.
2. no automatic translation of semantics between two disparate databases on different sytems.
3. no real-world “common ontology” defined in EDIFACT and X12 standards which is need for any computational use.
4. lack of a single good standard for EDI transactions. US ANSI X12 contains only industry specific variants. The international EDIFACT standard is only partially defined and accepted.

For SET, the limitation is that there are too many message formats to learn and to keep track of since for each type of messages sent from one party to another, a different message format is used. The application processing these messages may be large and complex in order to support different types of message formats for different types of messages.

Even though OTP promises great flexibility with the use of XML and better reliability and security, it might not be very popular with the Internet businesses because of the complicated structure of OTP Message. In order to use OTP, an organization must hire a lot of people to understand the whole structure of OTP. Then, it must also train these employees to become somewhat experts on XML technology in order to be able to apply its full technology efficiently and benefit from all the features that it provides. For this reason, many businesses are shy away from using this protocol.

OBI is a new protocol that tries to provide the best solution that can possibly meet the all the needs of all businesses in complicated commercial purchasing. It supports additional features that other existing protocols do not have, such as customer profiling, shipping and receipt confirmation, and flexibility in adding or changing digital signatures which is embedded inside the OBI message. However, with all these nice and powerful features, we should expect, or at least be prepared, to encounter some types of problems during the transaction of a message. Error handling of some kind is needed to recover and handle any type of errors. But OBI do not provide any error handling mechanism.

To summarize the results of the analysis, the table below lists all the properties of the protocols' architectures and provides the values for these properties.

Protocol	**Reliability**	**Security**	**Flexibility**	**Performance**	**Scalability**	**Vendor Neutrality**	**Extensibility**	**Interoperability**
EDI	Maybe*	Maybe*	No	Average	Low	No	No	No
SET	Maybe*	Yes	Maybe*	Good	Average	Yes	Yes	No
OBI	Yes	Yes	No	Bad	Average	Yes	No	Yes
OTP	Yes	Yes	Yes	Good	High	Yes	Yes	Yes

* Depends on other factors

Table 5.13c Analysis of Transaction Protocol Architecture

6 PROPOSED TRANSACTION PROTOCOL

6.1 Description of New Protocol

There is no avoiding of the semantic translation because of the lexical, structural, and semantic differences (discussed in section 5.2) of different applications on different systems. We need a semantic basis for translating data content from one system to another.

6.1.1 Existing solution

Ken Steel solution

In current EDI practice, human beings in two different organizations must negotiate to agree in advance on the specific protocols for Purchase Orders, etc. Programmers then create translators from the local database format of the source into standard EDI format, and from EDI to the local database format of the target system. This implementation limits EDI to have only one-to-one communication and slows down the transaction handling process by involving human intervention. Ken Steel of University of melbourne has proposed some major modifications to the existing design of EDI to allow for flexible message standards which helps reduce the size of the data being transmitted, eliminates human intervention, provides simple mechanism for adding more business partners to the standards, and allows for fast and dynamic migration from the old EDI system to the new one. [1]

Steel proposed "ICSDEF message" (Interchange Structure Definition) for EDIFACT. An ICSDEF message specifies in advance which data fields will be needed in an EDI transaction series between two partners, and it can be tailored to the situation and formatted as the parties see fit. In effect it obviates the need to follow the standard formats for transactions in X12 and EDIFACT. There is one translation, from source file to target file, enabled by an ICSDEF "wrapper" that describes the format of the source file; often it will be a flat file. [19]

Kalakota & Whinston solution

With similar idea as the one proposed by Steel, Kalakota's book describes a new EDI that uses the document type definition (DTD) used in SGML and HTML as an exchange with tagged or marked-up documents so that browsers can understand the document structure easily. This will require several steps, defined as follows [12]:

1. Shift the focus of the EDI standardization process away from the low-level interchange structure and onto more high-level business work flows involving many low-level interchange activities.
2. Allow customization (or massaging) of information by enabling application programs to use the interchange structures that best suit their local environment.

To make new EDI, more work is need to address a standard bridge between the language of business and the programming languages used in expressing the interchange standards. It standardizes the way programmers handle the business specifications in the programs they write to make the computer do what the business managers want them to do.

Another aspect of new EDI is the interactive query response (also called interactive EDI), which is a form of EDI used by travel agents to book airline flights. Interactive EDI is aimed at starting and completing the business process using an open channel of communication (point-to-point) between the customer and the supplier for the period of the business transaction(s). This type of interaction eliminates the current intermediary, namely, the value-added network.

6.1.2 New solution

TPX

TPX (Transaction Protocol using XML) is a new e-commerce transaction protocol similar to EDI. However, this protocol provides flexible message format which can easily be changed to fit the needs of the application. Because of this flexibility, TPX is capable of allowing one-to-many and many-to-many communications among all parties involved which is not feasible with EDI. TPX is built on open standards, provides for self-describing transactions (XML), allows for object-based documents (data and rules resided together), and provides access to a greater number of trading partners.

TPX Concept

The generated purchase order from the buyer is first fed into the TPX system. This system takes this electronic order form and converts it into an XML (Extensible Markup Language) format with either an internal or external DTD (Document Type Definition).

A DTD is a set of rules that the XML document, which contains the information about the purchase order, must obey. An internal DTD is embedded inside an XML document or file. An external DTD is stored in a separated file and referenced by the XML document by its file name.

The generated XML document and companion DTD are both encrypted and sent to the seller by some communication mechanism. After receiving the encrypted XML files, the seller decrypts them and feeds them into the TPX system. Using an XML parser, the system will first parses the XML file using the DTD received. It then uses the DOM (Document Object Model) APIs to access the data resulted from the parse and either generate the same purchase order as the one created by the buyer or directly process the parsed data.

This protocol supports one-to-many and many-to-many communications since there is no data-exchange standard needed among parties involved. Each party only needs to be able to read the DTD that is sent with the XML document in order to translate the data contained in the XML document. And the DTD can contain any set rules desired by the sending party. XML is a standard from W3C. However, using XML for TPX protocol makes it more extensible because of the flexibility that the DTD provides.

6. CONCLUSION

In e-commerce, data communication plays the key role that drives businesses over the Internet. Being able to accurately and securely transport data (ordering or payment information) between two parties is very important. A lot of work is needed in the design of an efficient transaction protocol to ensure proper message formatting that can be understood by all parties involved and data integrity and authentication which protect the data and verify that the receiving party is indeed the party that was meant to receive the data. Because of the growing popularity of e-commerce, many organizations and standards committees have proposed and implemented many transaction protocols to help make doing business over the Internet possible with efficiency and speed.

The four most talked-about transaction protocols existing today were discussed in this paper: EDI, SET, OBI, and OTP. Although the designers and architects have put a great deal of effort into designing and implementing these protocols, there is no avoiding of making loopholes which set limitations on the protocol. EDI, for example, is a very simple protocol to use with straightforward design and implementation. However, because it uses fixed standard for formatting the message, it limits itself from being able to support one-to-many or many-to-many communication among trading partners. Also, because it is so simple, it does not provide support for security of the data and authentication of the trading partners. Nor does it provide any support for additional purchasing features such as customer profile or receipt acknowledgement and delivery.

SET is a little more efficient than EDI in that it does provide for strict security and authentication of trading partners. It also provides error handling capability. However, it is still missing other e-commerce features such as customer profile. Both OBI and OTP which are the latest protocols and seem to be the most efficient so far are still lacking some features. OBI which follows the fixed standard of ANSI X12 for message data formatting is having the same limitation as EDI of not being able to support many-to-many communication. The benefit with OBI is that it supports customer profile information which helps to uniquely identify the buyer in order to construct a specialized catalog view from which the buyer can select items.

OTP, with its usage of XML for its message format, is the most flexible protocol among the four discussed. However, the problem of OTP is that it is complicated to use and understand since every message contains many blocks and components. The processing of OTP messages can also be very difficult if security is also included as one of the message components.

For every protocol designed, there are always some benefits and limitations. It is very difficult to come up with a transaction protocol that can satisfy all kinds of e-commerce users. The best way to make the best use of these existing protocols is do pick the one that will work in a particular e-commerce environment most efficiently. Work is ongoing for a yet better transaction protocol that will fit well and work most efficiently in the growing and most competitive businesses on the Internet.

RELATED WORK

Stefek Zaba. *Tools and Protocols for E-Commerce.* Information Security Technical Report, Vol. 3, No. 2 (1998) 34-40.

Nabil R. Adam, Igg Adiwijaya, Vijay Atluri (from Rutgers University) and Yelena Yesha (from University of Maryland Baltimore County). *EDI Through A Distributed Information Systems Approach.* Proceedings of the 31st Hawaii International Conference on System Sciences (HICSS'98). Published by IEEE Computer Society. 1998.

Roger Tilson, Jianming Dong, Shirley Martin, Eric Kieke. *A Comparison of Two Current E-commerce Sites.* ACM. 1998.

Taimur Aslam. *Protocols for E-Commerce.* Dr. Dobb's Journal. December 1998.

Daniel W. Manchala. *Trust Metrics, Models and Protocols for Electronic Commerce Transactions.* Xerox – Corporate Research and Technology.

Glossary

Idempotency The ability to be able to perform an operation any number of times without causing harm.

REFERENCES

[1] Nabil R. Adam and Yelena Yesha. *Electronic Commerce.* Springer-Verlag, Berlin, Heidelberg, New York, 1996.

[2] BankAmerica Corporation. *What is SET[tm] and How Does it Work?* http://www.bofa.com/spare_change/set_what_and_how.html, 1998.

[3] Holly Blumenthal, editor-in-chief of Netscape Enterprise Developer. *The OBI standard gains momentum.* http://www.netscapeworld.com/ned-12-1997/ned-12-extranet.html. December 14, 1998.

[4] Kenneth W. Copeland, C. Jinshong Hwang. *Electronic Data Interchange: Concepts and Effects.* http://www.isoc.org/isoc/whatis/conferences/inet/97/proceedings/C5/C5_1.HTM.

[5] Rik Drummond, Mats Janson, Chuck Shih. *Requirements for Inter-operable Internet EDI.* EDIINT Working Group. Internet Draft. December 1998.

[6] EDICT Systems. *EDI Benefits.* http://www.edictsystems.com, 1999.

[7] ETHOS team. *Open Buying on the Internet (OBI) standard.* http://www.tagish.co.uk/ethosub/lit5/c1aa.htm. September 10, 1997.

[8] Extensible Markup Language (XML). *W3C Recommendation.* http://www.w3.org/TR/1998/REC-xml-19980210, 1998.

[9] Federal Supply Service (FSS). *More About Electronic Data Interchange (EDI).* http://www.fss.gsa.gov/edi_mae.html.

[10] Eric S. Freibrun. *Electronic Data Interchange (EDI) and the Law,* 1993.

[11] John Evan Frook, *EDI No Longer On The Ropes,* http://www.interactiveage.com/news/news0330-2.htm, March 30, 1998.

[12] Ravi Kalakota & Andrew B. Whinston. *Frontiers of Electronic Commerce.* Addison-Wesley Publishing Company, Inc., Menlo Park, California. June 1996.

[13] Larry Loeb. *Secure Electronics Transactions: Introduction and Technical Reference.* Artech House Publishers, Boston, London, 1998.

[14] Mark S. Merkow, Jim Breithaupt, Ken L. Wheeler. *Building SET Applications for Secure Transactions.* John Wiley & Sons, Inc., New York, 1998.

[15] Netsavvy Communications. *Enabling Technologies: Secure Electronic*

Transactions (SET). http://www.sellitontheweb.com/ezine/techfaq.shtml#a6, 1999.

[16] The OBI Consortium. *Open Buying on the Internet (OBI): Technical Specifications.* Release V2.0, 1999.

[17] The OBI Consortium. *OBI Library: Open Buying on the Internet.* http://www.openbuy.org/obi/library/white-paper.html, 1999.

[18] The OBI Consortium. *About OBI.* http://www.openbuy.org/obi/about/OBIbackgrounder.htm, 1999.

[19] Ken Steel. *Matching Functionality of Interoperating Applications: Another Approach to EDI Standardization.* Department of Computer Science, University of Melbourne, and as ISO/IEC JTC1/WG3 IT117/94, Committee Draft of MAY94 (email reprint), 1994.

[20] The Open Trading Protocol Consortium. *Internet Open Trading Protocol. Part 1 Business Description.* January 12, 1998.

[21] The Open Trading Protocol Consortium. *Internet Open Trading Protocol. Part 2 Specification.* January 12, 1998.

[22] G. Winfield Treese and Lawrence C. Stewart. *Designing Systems for Internet Commerce.* Addison Wesley Longman, Inc., Reading, Massachusetts, 1998.

[23] Warwick Ford and Michael S. Baum. *Secure Electronic Commerce.* Prentice Hall, Upper Saddle River, New Jersey, 1997.

[24] David Kosiur. *Understanding Electronic Commerce.* Microsoft Press, Redmond, Washington, 1997.

[25] VISA/MasterCard. *SET Secure Electronic Transaction Specification; Book3: Formal Protocol Definition.* Version 1.0. May 31, 1997.