Journal of Theoretical and Applied Information Technology

www.jatit.org

SOCIAL FACTOR IMPLICATIONS ON NETBILL IN E/M-COMMERCE

Yang Mu

East China University of Science and Technology, China
Email: yanmumcs@yahoo.com

ABSTRACT

Many electronic payment systems have been developed to facilitate the purchase of goods and services over the internet. However none of the system was able to be implemented on large scale [1]. This paper identifies the factors which cause failure to these payment systems. Moreover this paper presents the NetBill protocol and describes its social aspects along with its security and transactional features.

Keywords: NetBill, e-commerce, payment, social aspects, security.

I. INTRODUCTION

"History repeats itself", lets take a look at the history of commerce. Long before the written history has been started Barter system was there. In Barter system the exchange of goods was conducted face-to-face between two parties. Eventually, as trade became more complicated and inconvenient, humans invented abstract representations of value. As time passed, representations of value became more and more abstract. After Barter System, Metal Coins were used for trade where Goods are purchased and sold for metal coins.The value represented by these coins is equal to the value of the coin. After Metal coins, Paper money was issued by the governments. In beginning paper money was fully backed by gold but now it is fractionally backed. Now its time for Electronic payment systems to be implemented.

Although computers are there for more than four decades but electronic Payment systems were not able to eradicate the traditional transaction Procedures[2]. This is because there were certain factors which provides Hinderence in achieving the above goal. These factors comes from three major

aspects of electronic payment systems namely Social Aspects, Network Security Aspects, Business Aspects, which are described in Section 2, 3 and 4 of the paper. In the end an electronic payment protocol NetBill is explained.

II. SOCIAL ASPECTS

Following are the social aspects related to electronic payment systems

A. Trust

Trust is one of the mainstays of commerce. Trust is a belief or expectation that the word or promise by the seller can be relied upon and the seller will not take advantage of the buyer's vulnerability Trust and risk are closely interrelated. However, as more and more individuals and businesses participate in electronic commerce, it is becoming apparent that much of what supports trust in the traditional commerce setting is unavailable online.[3]
First party information, i.e., information that businesses provide concerning themselves is critical to developing trust online. Presenting

user feedback reduces the customer's perception of risk and enhances trust [4]. Third party ratings are important in developing trust[5]. In the online setting, seals of approval from trusted third parties such as Visa, TRUSTe, and BBBOnline, have been found to reassure consumers that they can trust a site or online business[6]. In fact, third party ratings may be even more important online than offline, due to the absence of visual and social cues traditionally found in the bricks and mortar world.[3]

B. Integrity and Reliability

A payment system with integrity allows no money to be taken from a user without explicit authorization by that user. It may also disallow the receipt of payment without explicit consent, to prevent occurrences of things like unsolicited bribery[7]. Payment transactions must be atomic: They occur entirely or not at all, but they never hang in an unknown or inconsistent state. Recovery from crash failures requires some sort of stable storage at all parties and specific resynchronization protocols[8].

C. Privacy and Confidentiality

Some parties involved may wish confidentiality of transactions . Confidentiality in this context means the restriction of the knowledge about various pieces of information related to a transaction: the identity of payer/payee, purchase content, amount, and so on. Where anonymity or untraceability are desired, the requirement may be to limit this knowledge to certain subsets of the participants only[**9**]. Untraceable payment systems can be developed by using blind signature cryptosystems which were first proposed by David Chaum for implementing DigiCash E-Cash[**10**].

D. Repudiation

Repudiation is that the originator of a message falsely deny later that they were the party that sent the message. It is much easier to repudiate an electronic business transaction then in the Cash based system[11].

Thus the protocol should prevent the denial of previous commitments or actions. This can be achieved through digital signatures. Digital Signatures are bit patterns that depend upon the message being signed and use some information unique to the sender[12].

III. NETWORK SECURITY ASPECTS

Buyers and sellers increasingly want to use the internet to conduct their businesses electronically. As a base for commerce , the internet poses special challenges due to its lack of standard security mechanisms[13]. Viruses, Trojan Horses and Dos attacks are the most prominent ones as explained in [14].

A.. Viruses

Viruses are the most publicized threat to client systems. They are effective because of the built-in insecurity of client systems . Subverting a PC system requires access to the system and no special privilege is needed to write code or data into sensitive system areas. The more publicized viruses such as Melissa, ILOVEYOU, Resume, KAK and IROK have no effect on Unix systems. Viruses need "system privilege" in order to be effective. In general, the multiple privilege access schemes present in Unix, VMS and other multi-user operating systems prevents a "virus" from damaging the entire system. It will only damage a specific user's files.[15]

B. Trojan Horses

Trojan horses the most popular and traditional way

Trojan horse programs launched against client systems pose the greatest threat to e-commerce because they can bypass or subvert most of the authentication and authorization mechanisms used in an e-commerce transaction. These programs can be installed on a remote computer by the simplest of means: email attachments.[15]

The **BackOrifice**, **Netbus**, **BO2K** hacker tools allow a remote user to control,

examine, monitor any information on the target PC. What makes them especially beguiling is that they are also capable of using the target PC to send information to the net *as if the legitimate user had done so.* There are commercial tools like CUCme, VNCviewer that perform the same function.[15]

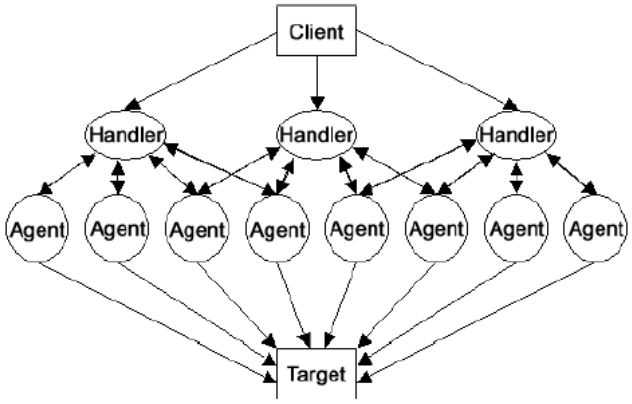
C. DOS Attacks

While DoS attack technology continues to evolve, the circumstances enabling attacks have not significantly changed in recent years. DoS attacks remain a serious threat to the users, organizations, and infrastructures. [16]

Businesses that rely on web-based transactions are and will continue to be vulnerable to Denial of Service (DoS) attacks. DoS attack scripts are the most common, effective and easiest to implement attacks available on the WEB. No actual damage is done to the victim site. The access paths to it are simply overwhelmed with incoming packets.[15]

The Distributed Denial of Service (DDOS) attacks are the latest evolution of DoS attacks and their success depends on the inability of intermediate sites to detect, contain and eradicate the penetration of their network. [16]

These DDoS attacks are a two-phase assault. The attacker will spend a large amount of time preparing for the first phase of an attack. This phase of the assault involves compromising as many systems as possible. The second phase is the actual Denial of Service attack. The compromised systems will generate the network traffic to bring down a targeted site. These compromised systems are considered secondary victims of the Denial of Service attack.[17]



With increase in the user demand of more flexible, robust and facilitating software has greatly enhanced the risk of intrusion and infections and making the software more facilitative has made it more prone to attack. The issue is further enhanced by the technical standards increasing at a drastic rate and the Syatem and Network Administrators knowledge being outdated in a span of a few months with emerging new thechnology[18], while the attack technology and tool deployment is international in scope. Furthermore the difficulty of cybercrime investigation, apprehension and prosecution means that it will remain a challenge for conducting commerce in the network environment.

This forces the ecommerce service providers to check the client's credibility thoroughly for preventing any fraud etc. and the clients using the facility of ecommerce sites have to compromise on the right of the customer privacy to some extent and to let the ecommerce service providers to examine their credit history and they should be provided with information about who gets their historical data and information[19].

D. Encryption

It is the process of translation of data into a secret code. **Encryption** is the most effective way to achieve data security. To read an encrypted file, you must have access to a secret key or password that enables you

to *decrypt* it. Unencrypted data is called *plain text*; encrypted data is referred to as *cipher text*. It is of two main types of **encryption**: asymmetric **encryption** (also called public-key **encryption**) and symmetric **encryption**. [20]

The security and integrity of clients confidential data provided by him on his own will is the responsibility of the ecommerce service provider, and in any network environment a good Encryption strategy is the best mechanism available for this purpose

E. Identity and message authentication

It refers to encryption as "the process of changing a digital message (from plain text to cipher text) so that it can be read only by the intended parties (also called enciphering), or to verify the identity of the sender (authentication), or to be assured that the sender really did send that message (non-repudiation)." The dictionary also clearly distinguishes private from public keys.[21]

F. Secure capability semantics

Electric Communities has designed and implemented a capability-based security infrastructure for certificates that goes far beyond simple digital signature. Capability semantics. E offers a sophisticated security model that allows convenient but extremely detailed control over sensitive functions within a single machine or across a network[22]

IV. BUSINESS ASPECTS

Commerce always involves a payer (buyer)and a payee(seller)— who exchange money for goods or services—and at least one financial institution—which links "bits" to "money."[9].

Traditionally the broker is the middle man or party responsible for the issuing and validation of some sort of payment mechanism implementation in all ecommerce systems regardless of the system being crypto less or implementing cryptography.

The following stakeholders are usually present in almost all of the ecommerce systems. Lets take an critical look at how their interests are protected in the implementation and how this protocol effects them if anything goes out of order.

A. Buyers

Chargebacks occur when a card/certificate holder refutes a transaction. Typical chargeback types include situations where the cardholder claims he or she did not participate in the transaction, did not receive the goods, or believed the goods were not as represented by the merchant. Consumers typically are not responsible for the actual chargeback amounts in these situations, but they fear that it could lead to impact their credit history. Victims of this type of fraud lose confidence in the credit card system when they notice fraudulent charges on their credit card or receive calls from their card issuer indicating that their account has exceeded the typical velocity of charges or that suspect authorizations had occurred.[18]

Another type of fraud, "Friendly fraud", occurs when a cardholder did make a transaction, but wants to deny that he made a potentially embarrassing type of adult purchase[19]

B. Sellers

Merchants are protected from certain chargeback reasons during card-present transactions. But merchants with excessive numbers or percentages of chargebacks face steep fines, despite the fact that internet merchants have no means to authenticate a cardholder online. Moreover Merchants pay a one to five percent fee per credit card transaction. Merchants are forced to facilitate mechanisms of electronic transactions which are lesser profitable to them because if they don't then they will have to loose the buyers. Merchant fraud

occurs when merchants authorize and capture fraudulent charges against credit card numbers without cardholder authorization. Merchants are often victims of blackmail when hackers steal credit card number databases. CD Universe was one of the first publicized cases of such attempted fraud.[19]

C. Financial Institutions

Financial institutions intend to minimize the risks and maximize the profits Since the protocols for electronic transactions are actually implemented and facilitated by financial institutions therefore protocols developed are biased towards the financial institution. However, financial institutions take care of buyers because they intend to sell the credit cards to them.

D. Electronic credentials

Modern approach is the use of electronic credentials, to prove their trustworthiness. Electronic credential is the counterpart of paper credential of the real world. For example, X.509 is one of the most popular formats. E-strangers are those who have no previous knowledge of each other but prove their authenticity and trustworthiness by disclosing electronic credentials to establish trust.[23]

V. NETBILL

The NetBill transaction model involves three parties: the customer, the merchant and the NetBill transaction server. A transaction involves three phases: price negotiation, goods delivery, and payment. For information goods which can be delivered over the network, the NetBill protocol links goods delivery and payment into a single atomic transaction. In a NetBill transaction, the customer and merchant interact with each other in the first two phases; the NetBill server is not involved until the payment phase.

A. Neill Architechture

NetBill uses a single protocol that supports charging in a wide range of service interactions. NetBill provides transaction support through libraries integrated with different client-server pairs. These libraries use a single transaction-oriented protocol for communication between client and server and NetBill; the normal communications model between client and server is unchanged. Clients and servers can continue to communicate using protocols optimized for the application for example, video delivery or database queries while the financial-related information is transmitted over protocols optimized for that purpose. The client library is called the *checkbook* and the server library is called the *till.These libraries are* well-defined API allowing easy integration with a range of applications. The libraries incorporate all security and payment protocols, relieving the client/server application developer from having to worry about these issues. All network communications between the checkbook and till are encrypted to protect against adversaries who eavesdrop or inject messages.

B. The NetBill Transaction Protocol

Before a customer begins a typical NetBill transaction, she will usually contact a server to locate information or a service of interest. For example, the customer may request a Table of Contents of a journal showing available articles available, and a list price associated with each article. The transaction begins when the customer requests a formal price quote for a product. This price may be different than the standard list price because, for example, the customer may be part of a site license group, and thus be entitled to a marginal price of zero. Alternatively, the customer may be entitled to some form of volume discount, or perhaps there is a surcharge during the peak hour.

The customer's client application then indicates to the checkbook library that it would like a price quote from a particular

merchant for a specified product. The checkbook library sends an authenticated request for a quote to the till library which forwards it to the merchant's application. The merchant then must invoke an algorithm to determine a price for the authenticated user . He returns the digitally signed price quote through the till, to the checkbook, and on to the customer's application. The customer's application then must make a purchase decision. The application can present the price quote to the customer or it can approve the purchase without prompting the customer. For example, the customer may specify that her client software accept any price quote below some threshold amount; this relieves her of the burden of assenting to every low-value price quotes via a dialog box.

Assume the customer's application accepts the price quote. The checkbook then sends a digitally signed purchase request to the merchant's till. The till then requests the information goods from the merchant's application and sends them to the customer's checkbook encrypted in a one-time key, and computes a cryptographic checksum on the encrypted message. As the checkbook receives the bits, it writes them to stable storage. When the transfer is complete, the checkbook computes its own cryptographic checksum on the encrypted goods and returns to the till a digitally signed message specifying the product identifier, the accepted price, the cryptographic checksum, and a timeout stamp, this information is known as the *electronic payment order (EPO)*. Note that, at this point, the customer can not decrypt the goods; neither has the customer been charged.Upon receipt of the EPO, the till checks its checksum against the one computed by the checkbook. If they do not match, then the goods can either be retransmitted, or the transaction aborted at this point.

This step provides very high assurance that the encrypted goods were received without error. If checksums match, the merchant's application creates a digitally signed invoice consisting of price quote, checksum, and the decryption key for the goods. The application sends both the EPO and the invoice to the NetBill server.

The NetBill server verifies that the product identifiers, prices and checksums are all in agreement. If the customer has the necessary funds or credit in her account, the NetBill server debits the customer's account and credits the merchant's account, logs the transaction, and saves a copy of the decryption key. The NetBill server then returns to the merchant a digitally signed message containing an approval, or an error code indicating why the transaction failed . The merchant's application forwards the NetBill server's reply and the decryption key to the checkbook .

C. Protocol Failure Analysis

The protocol must gracefully cope with network and host failures. One of the goals is to tightly link two events: charging the customer and delivering the goods. The customer should pay exactly when she receives the information goods. The NetBill server is highly reliable and highly available. All transactions at the NetBill server are atomic: they either finish completely or not at all. NetBill is never in doubt about the status of a purchase. We cannot make similar assumptions about the reliability of the merchant's and customer's software; they must maintain a state consistent with the NetBill Server.

First, consider the protocol from the perspective of the customer's application. When the customer application acknowledges receipt of the information goods, the customer application knows that no transaction has occurred. That is, the customer does not have access to the product and the merchant does not have the customer's money. Once the application sends the EPO, the customer is *committed* to the transaction and must be prepared to accept the purchase. If the customer's application

does not receive a response from the merchant's application, then it is the responsibility of the customer's application to determine what happened: the customer's application can poll either the merchant application or the NetBill server to determine the status of the purchase request. If the merchant's application did not successfully forward the EPO to the NetBill server, then the EPO will have expired and the NetBill server will respond to the customer's application that the purchase has failed. Of course, the customer still does not have the one time key, so while the customer still has her money, she also does not have the goods. If, on the other hand, the transaction succeeded before communication failed, then the customer's application can find the status of the purchase and, if appropriate, the decryption key from either the merchant's application or the NetBill server (which has registered the key). If both are unreachable, the customer's application must continue to poll.

Now consider the protocol from the perspective of the merchant's application. Before it forwards the EPO and invoice to the NetBill server, the merchant's application knows that the transaction has not occurred. After it forwards the EPO and invoice, however, the merchant's application is *committed* to the transaction and must obtain the result from the NetBill. If the merchant's application does not receive a response from the NetBill server, the merchant's application must poll the NetBill server.

The protocol is much simpler for the NetBill server than for the other parties. The NetBill server is never in a state in which it depends on a response from another entity to determine the status of a transaction. Until the NetBill server receives the EPO and invoice from the merchant's application, it knows nothing about the purchase. Once it receives the EPO and invoice it has all the information necessary to approve or reject the purchase.

The NetBill transaction protocol also exhibits a number of other desirable features:

1) *Support for flexible pricing.*

By including the steps of offer and acceptance, we provide an opportunity for the merchant to calculate a customized quote for an individual customer. In the process we also generate signed messages that can later prove that there was a contract at the quoted price.

2) *Scalabilit*y.

The bottleneck in the NetBill model is the NetBill server which supports many different merchants. Our transaction protocol minimizes the load on the NetBill server and distributes the burden over the many customer and merchant machines.

3) *Protection of user accounts* against unscrupulous merchants.

In a conventional credit card transaction, the merchant learns the customer's credit card number and can submit fraudulent invoices in the customer's name. In a NetBill transaction, the customer digitally signs the EPO using a key that is never revealed to the merchant,thus eliminating this threat. Moreover, the customer has proof of the exact nature of the information goods received, providing evidence in case a dishonest merchant attempts to deliver faulty information goods.

D. NetBill Account Management

NetBill supports a many-to-many relationship between *customers* and *account*s. A project account at a corporation can have many users authorized to charge against it. Conversely, an individual customer can
maintain multiple personal accounts. Every account has a single user who is the account *owne*r; and the account owner can grant various forms of access rights on the account to other users.

An authorized user can view and change a NetBill account profile, authorize funds transfer into that account, or view a current statement of transactions on that account, using a standard WWW browser.

Authentication and security are provided by treating account information as "billable" items. NetBill provides account information to users using the NetBill protocol. NetBill can be configured to provide this information for free or for a service charge, as desired.Automating account establishment for both customers and merchants is important for limiting costs. To begin the process, a customer retrieves, perhaps by anonymous FTP, a digitally signed NetBill security module that will work with the user's WWW browser. Once the customer checks the validity of the security module, she puts the module in place. She then fills out a WWW form, including appropriate credit card or bank account information to fund the account, and submits it for processing. The security module encrypts this information to protect it from being observed in transit. The NetBill server must verify that this credit card or banking account number is valid and that the user has the right to access it.

VI. CONCLUSION

This paper has presented the basic principles for the protocols for secure electronic commerce, and a NetBill protocol which is used for micro payments. In principle, the technology exists to secure electronic payments over the Internet. It is now possible to achieve security for all parties, including the perfect intractability of the payer. However, no electronic payment system is currently deployed on large scale. There is a little chance that the world will agree on a single scheme for electronic payments in near future. However, the world needs one card holder scheme, not one per brand or one per country.

VII. REFERENCES

[1]. J. D. Tygar. “Atomicity in Electronic Commerce”. Carnegie Mellon University Pittsburgh, January 1996.

[2]. M. Sirbu and J. D. Tygar. “ NetBill: An Internet Commerce System Optimized for Network Delivered Services “. In IEEE Personal Communications, 2(4) pages 34Ð39, August 1995.

[3]. M. Daignault, M. Shepherd, S. Marche, C. Watters.” Enabling trust online”. In Proceedings of the International Symposium on Ecommerce, October 2002.

[4]. M. Sirbu and J. D. Tygar. “ NetBill Security and Transaction Protocol “. Carnegie Mellon University Pittsburgh, January 1996.

[5]. e-Commerce and Trustmarks: Results from the ALPINE Working Group computing.breinestorm.net/trust+www+trustmark+commerce+information/

[6]. eCommerce Trust Study.. Cheskin Research & Studio Archtype/Sapient. 1999. [online]. Available:http://www.cheskin.com/think/studies/ecomtrust.html [viewed July 30, 2001].

[7]. Security in the Internet Payments, By N.V.SanjeevSenior ConsultantCambridge Technology PartnersNew York (USA). Available at : http://www.indiainfoline.com/bisc/sepy.html

[8]. Electronic Payment over Open Networks -- A Technology Overview -- 1 P. Janson, M. WaidnerIBM Zurich Research Laboratory CH 8803 Ru"schlikon, Switzerland{pj,wmi}@zurich.ibm.com Available At :

www.zurich.ibm.com/security/publications/1995/JaWa95f.ps.gz

[9]. Electronic Payment Systems* N. Asokan, Phil Janson, Michael Steiner, Michael Waidner IBM Research Division, Zurich Research LaboratoryCH-8803 Ru"schlikon, Switzerland {aso,pj,sti,wmi} @zurich.ibm.com

[10]. D. Chaum: Privacy Protected Payments -- Unconditional Payer and/or Payee Un-traceability; SMART CARD 2000, North-Holland, Amsterdam 1989, 69-93

[11]. Some Guidelines for Non-repudiation Protocols Available at : portal.acm.org/ft_gateway.cfm?id=505676&type=pdf

[12]. Network Security and Encryption Available At : www.electronics.dit.ie/staff/mdavis/Section9_DigitalSignatures.pdf

[13]. NetBill Security and Transaction Protocol, Carnegie Mellon University Pittsburgh, PA 15213-3890 Benjamin Cox J. D. Tygar Marvin Sirbu thoth+@cmu.edu tygar@cmu.edu sirbu+@cmu.edu

[14]. Viruses, Worms, Trojan Horses Prof. Dr. Christoph Meinel Available At: http://www.tele-task.de/player/embedded.php?series=13&lecture=93&language=en

[15]. E-commerce Security Issues R.Marchany ,J. Tront Available At: http://doi.ieeecomputersociety.org/10.1109/HICSS.2002.994190

[16]. Trends in Denial of Service Attack Technology CERT® Coordination Center Kevin J. Houle, CERT/CC George M. Weaver, CERT/CC In collaboration with: Neil Long Rob Thomas

[17]. Distributed Denial of Service Trin00, Tribe Flood Network, Tribe Flood Network 2000, And Stacheldraht CIAC-2319 Paul J. Criscuolo

[18]. Emerging eCommerce Credit and Debit Card Protocols Mark E. Peters IBM Corporation mepeters@us.ibm.com

[19]. Bennett, Robert A., "I didn't do it", USBanker 12 Dec 2001, P48-52

[20]. http://www.webopedia.com

[21]. Systems Administration by S. Lee Henry SunExpert Magazine n July 1997

[22]. USING THE EC TRUST MANAGER TO SECURE JAVA AN ELECTRONIC COMMERCEWHITE PAPER AVAILABLE AT: WWW. HTTP://WWW.CROCKFORD.COM/EC/ETM.HTML

[23]. Electronic Credential based Security Management in Decentralized Computing Environment Chenxi Huang*, Jie Xu**, Keith Bennett* *Department of Computer Science, University of Durham Durham DH1 3LE, England chenxi.huang@durham.ac.uk **School of Computing, University of Leeds Leeds LS2 9JT, England