Notes from the Second USENIX Workshop on Electronic Commerce

Michael Harkavy, Andrew Myers, J. D. Tygar, Alma Whitten and H. Chi Wong

Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

These are notes taken from the Second USENIX Workshop on Electronic Commerce from November 1996. They record presentations and questions from this workshop.

Introduction

The Second USENIX Workshop on Electronic Commerce was held in Oakland, California at the Claremont Hotel, with a day of tutorials on November 18, 1996, and three days of technical program beginning on November 19.

The best paper award was won by Ross Anderson and Markus Kuhn for "Tamper Resistance — A Cautionary Note." The best student paper award was won by David Wagner for his submission with Bruce Schneier, "Analysis of the SSL 3.0 Paper." (Please check with the authors for final version of this paper.)

The proceedings of the First USENIX Workshop on Electronic Commerce (which appeared several months after the workshop was completed) included a set of notes prepared by Peter Honeyman and his students. This guide was useful to researchers, so J. D. Tygar and his students organized a similar note-taking exercise for the Second Workshop. Since the proceedings for the Second (and Third) Workshops were distributed during the workshop, the program chairs for the Second and Third Workshops agreed to have these notes appear in the proceedings of the Third Workshop.

Electronic commerce has changed a lot since November 1996. We hope that the attendees to the Third USENIX Workshop will enjoy seeing this snapshot of the research state of our field from two years ago.

Session I: Hardware Tokens

Tamper Resistance – A Cautionary Note

Ross Anderson, Cambridge University; Markus Kuhn, Purdue University

Markus Kuhn began by pointing out that, while cryptographic security usually assumes that attackers can't get at the secret keys or observe the computations, current distributed and mobile applications such as pay TV access control give attackers plenty of access to the hardware. He stated that he would discuss hardware security and tamper resistance in terms of three classes of potential attackers: clever outsiders, knowledgeable insiders, and funded organizations.

Markus then described a host of simple attacks on the physical security of smart cards. The tamper resistant coating on the Motorola smart card chip can be dissolved with $30 worth of fuming nitric acid and acetone. Access to software stored in standard microcontrollers is generally prevented by setting an irreversible security fuse bit, but a UV EPROM eraser can be used to reset the security fuse and the software can then be read. Special smart card security processors usually have a melt fuse as the security bit, but a well-equipped lab can often repair the fuse. For many microcontrollers voltage attacks can successfully reset the security bit. Other techniques for accessing the software include timing analysis, applying heat gradually to toggle EEPROM bits, and recording current leakage. It is also possible to change single instructions by signal glitches such as increasing the clock frequency; for example, a loop control variable can be changed causing additional memory content to be output.

Markus stated that all these attacks are feasible even for clever outsiders. Knowledgeable insiders and funded organizations who have resources up to $50,000 may have access to tools such as microprob-

ing workstations and laser cutters for breaking connections and removing the passivation layer. If they have up to $1,000,000 available, they may use electron beam testing for reading bus signals, focused ion beam workstations for making new connections on the chip, and selective dry etching. If they have even more resources, they may use tools like automatic layout reconstructions to recreate circuit diagrams, electro-optic sampling and IR rear access.

Markus concluded by stating that the moral is not to blindly trust manufacturer claims about tamper resistance, avoid global secrets, reduce the importance of tamper resistance whenever possible, use fault-tolerant machine code in smart cards, implement fallback modes and insist on in-depth hostile review of designs.

Token-Mediated Certification and Electronic Commerce

Daniel E. Geer and Donald T. Davis, Open Market, Inc.

Dan began by noting that public key cryptography usually expects the use of certification authorities (CAs) to remove the need for real-time authority participation; he then suggested that we consider what could be done if certificates were made really cheap and disposable. For example, a smart card in the wallet could be trusted to act as a CA and generate certificates on a per-transaction basis, not to supplant existing CAs and certificates, but as a supplement. This could allow us to get highly scalable access control and simpler payment protocols.

As one possibility, we could have a delayed purchase scenario, in which the goods are not currently available at the time of purchase. We generate a certificate for the merchant which is a public key, an authorization to deliver (download) goods, and an expiration. The raw material required for this would be a smart card with reader, a cryptographic coprocessor, a secure key store, browser support for smart cards such as a Netscape plug-in or Microsoft cryptographic API, and certificates such as X.509v3. No certificate directory or revocation lists would be required. The card owner would begin by generating two key pairs, one for the owner and one for the card, and would certify the card's key with the owner's key. The private keys could then be deposited with a key recovery center, cross-encrypted with the public keys.

Dan argued that there would be many advantages to such a system. For authorization, he stated that such certificates correspond naturally to roles, and that roles scale better than access control lists and are easier to think out properly than capabilities. He pointed out that revocation would be unnecessary because these certificates are both short-lived and not generally published. He listed advantages of this scheme for electronic commerce including: ease of set up, design and management; the ability to do delayed fulfillment such as prime-time purchases with off-time deliveries; fewer on-line parties needed for each transaction; no access control list management; new services that rely on asynchronous delivery, such as magazine subscriptions; and consumer ease and safety because delivery can take place securely without the participation of either the customer or the smart card.

Marvin Sirbu pointed out that a Kerberos ticket and session key could be used instead of a public key; Dan agreed that public key cryptography was unnecessary here and that Kerberos would work fine. Greg Rose asked why it's necessary to create a new key pair; Dan's answer was that it's a containment issue. Terry Ingoldsby asked why the bank can trust the merchant to take payment from the customer's account; Dan explained that the certificate given to the merchant is signed with the customer's private key.

Smart Cards in Hostile Environments

Howard Gobioff, Carnegie Mellon University; Sean Smith, IBM Research; J. D. Tygar, Carnegie Mellon University; Bennet Yee, University of California, San Diego

Howard Gobioff described a security problem with smart cards due to the lack of direct I/O to the customer; that is, in an untrusted environment all communication between the customer and the smart card must go through an untrusted card reader. As an example of the problems this can create, Howard described a point-of-sale scenario in which the merchant's terminal reports the transaction to the smart card as $100 while displaying it to the customer as $10.

We would like to have communication between the customer and the smart card be secure (private and trusted) in both directions. Howard outlined possible additional capabilities that we could assume for the smart card, and the benefits that would result from each.

If we assume that we have a one-bit private input channel from the customer to the smart card,

then we can also have private output from the smart card to the customer, by having the customer input a key through the private channel which the smart card can then use to encrypt its output. This might be used to allow the customer to check the balance in the smart card without revealing it to the merchant. Similarly, if we assume we have a one-bit private output channel from the smart card to the customer, then the card can provide the customer with a key and the customer can input encrypted values for privacy. This is similar to work by Abadi, Burrows, Kaufman and Lampson in which the card presents a random value which the customer then adjusts using arrow keys.

If we assume we have trusted input plus one bit of trusted output, then we can have general trusted output: the customer feeds the displayed value back to the card via trusted input and the card uses the one bit of trusted output to signal any discrepancy. Likewise, if we assume we have trusted output plus one bit of trusted input, the card can display and the customer can signal discrepancies, so again we have general trusted output.

Bob Gezelter pointed out the importance of having a timeout equal not-okay, since otherwise the merchant can attack by distracting the user at the right time. Simon Kenyon argued that in a closed-loop system fraud will be caught when the books are balanced; Howard responded that this is true in present systems but fraud is still a problem.

Session II: Protocol Analysis

Analysis of the SSL 3.0 Protocol

David Wagner, University of California, Berkeley; Bruce Schneier, Counterpane Systems

SSL is a protocol for practical application–layer security, mostly for web traffic. The most recent version, 3.0, is an attempt to fix problems with version 2.0 and to add support for more cryptographic algorithms.

SSL version 3.0 is, overall, an improvement. The MAC keys have been expanded to 128 bits, even in the exportable version. Separate keys are now used to perform encryption and authentication. Finally, SSL now uses HMAC as its message authentication algorithm. These improvements help to stop replay and connection truncation attacks. Some standard, simple attacks were tried, but none compromised 3.0's security.

One attack that is successful against 3.0 is a traffic analysis attack, based on the observation that the ciphertext length usually reveals the plaintext length. An eavesdropper on web traffic could record the encrypted request for a URL and a server's encrypted reply. If the attacker can get an index of all the documents on a web server, he can compare the encrypted lengths of the documents and their request strings to the lengths of the observed request and document. A match in lengths between the lengths of an encrypted document and the observed document as well as the encrypted document request and the observed request indicates a very likely match between the unencrypted forms of the documents. This attack reveals which documents a client received from a web server.

In addition, there is an attack on the handshake layer of the protocol. A field used to select between the RSA and Diffie–Hellman algorithms is not part of the signed section of a message. If an adversary were to change the value of that field, the client could become confused about the meaning of signed data, eventually compromising the session's security. If the client performs a sanity check, it could detect this attack, but the sanity check is not mandated by the protocol. The suggested fix is to expand the signature to also cover the field.

There are still many questions for future study. David is not sure whether or not all nonces are properly signed, and the protocol should be checked to make sure that it is not vulnerable to session resumption and version rollback attacks. In general, this analysis was informal, not formal, meaning that it can only illustrate flaws in the protocol, not prove that it's correct.

Martin Abadi asked what the maximum amount of an SSL transaction in which David would be willing to participate was. David answered that SSL is probably acceptable for encrypted credit card transactions. Dan Geer was curious about where the most likely points for an implementation to fail would be. David responded that sanity checks and key management were likely places. Another question was whether cataloging all the documents on a web site is really possible. David did not know how much of a typical web site is made up of dynamically generated documents, but he suspected that the amount is non–trivial.

Fast, Automatic Checking of Security Protocols

Darrell Kindred and Jeanette Wing, Carnegie Mellon University

An increasing number of security protocols are becoming more important. These protocols are being developed extremely rapidly as well, and it's hard to

get these protocols right. We would like to automate the protocol–checking process to save time and gain more confidence in the results. There are several approaches to verifying that a protocol is correct. Handwritten proofs are easy to get wrong. Another approach which is effective, though tedious, is to reason about a protocol in high–order logic and use theorem proving software to verify correctness. Finally, small logics have been developed for reasoning about parts of protocols, e.g. BAN. The usual answer from such logics is limited to yes, no, or cannot decide.

The focus of this work is on improving automated support for using small logics. In such logics, a protocol message is represented as a logic formula reflecting the result of the message being received, a set of assumptions, and a set of inference rules to express familiar concepts. Usually, the premises of the rules are larger than the conclusions. Thus, the process of generating conclusions will terminate eventually.

We can exhaustively produce all truths in the protocol. With these truths, we can check that properties hold and explore what effect changes in the protocol will have on the truths generated.

A logic checker was implemented which can verify that a logic satisfies certain restrictions. In the actual checker, there are three types of rules: shrinking, growing, and rewrites. The checker uses shrinking rules to generate truths, only applying other rules when no more shrinking rules can be applied. This method does not generate all truths, but typically, interesting truths are generated. We can also check whether a specific formula is derivable.

Currently, the logic checker should be useful for protocol developers; it is automatic and fast, with most runs being on the order of one to two minutes. In the future, more logics will be developed to reason about other properties such as anonymity or safety from man–in–the–middle attacks, and support for temporal logics will be added.

Nevin Heintze was curious as to whether, given a property, a system could work backwards, giving changes to a protocol which would be required for the property to hold. Darrell pointed to model checking, which gives a counter–example when a property does not hold, and said that getting the smallest set of changes to a protocol that would make a property hold would be possible.

Verifying Cryptographic Protocols for Electronic Commerce

Randall W. Lichota, Hughes; Grace L. Hammonds, AGCS, Inc.; Stephen H. Brakcin, Arca Systems, Inc. Presented by Jack Wool.

The aim of this work is to develop methods for verifying protocol correctness that will aid protocol designers at the beginning of the design process, rather than later, after the protocol is already in use. The focus is on determining what each party in a transaction can be proved to believe. The tool should be usable by protocol designers, not mathematicians. The tool is meant to be used to iteratively refine a protocol design.

At the same time, human factors were very important to the tool's developers. The tool provides a common front end to a variety of back end formal methods engines. The front end consists of a "software through pictures" user interface, allowing the user to view the software as a set of interacting objects. The back end uses a version of belief logic whose libraries have been inspected on the source code level by members of the theorem–proving community. The back end receives the protocol specification through an intermediate language produced by the front end. Feedback is returned to the front end and displayed to the user.

To model a protocol, the user inputs a description consisting of beliefs, assumptions, and initial conditions from which the tool produces high level, graphical diagrams. The user can then see what goals are reachable. As a demonstration of the system, a somewhat simplified version of public key Kerberos was modeled. The usual problems in verification, insufficient initial conditions and wrong associations of messages and assumptions were experienced. The whole analysis took three days of tool use.

This tool provides a way to clearly specify the assumptions in a cryptographic protocol. It decreases the analysis time through automation, and it places formal methods in the designer's hands.

Invited Talk

Legal Signatures and Proof in Electronic Commerce

Benjamin Wright, Attorney and Author – The Law of Electronic Commerce

Benjamin Wright is a lawyer and the author of "The Law of Electronic Commerce." He started his talk by saying that his background is in law and society, and not in any scientific area. His talk would address, from a lawyer's point of view, an area that lies in the intersection of law and digital signatures.

Ben stressed that, when regulating uses of a technology, one needs to be humble and recognize that

people may use the technology differently from how one has assumed. Different uses may require different legal interpretations and therefore require different legislations. So, he wanted to rename this talk "The Confessions of an Electronic Commerce Lawyer: My Fear of Curves on the Information Super Highway."

The law of signatures in the US has been around for a long time, and the legal community has been dealing with the question "what is a signature," for legal purposes, for centuries. Ben's own personal interpretation of "what is a signature" is liberal compared to that of some of his colleagues. His interpretation is: a signature, for legal purposes, is simply a symbol that someone adopts for the purposes of taking responsibility for a transaction. Generally speaking, the law of signatures in the United States has never required that signatures be secure in any way. This implies that signature, security, proof, and evidence are different things.

He then gave two examples of disputes that have happened with traditional signatures. The first example involved an autograph received through a fax machine. Its supposed signer argued that fax transmissions offered a low level of security, and the autograph was not his or her signature. The court ruled, however, that the lack of security did not interfere with the signatureness of the autograph. Authenticity, the question of whether or not the supposed signer signed it, was a separate issue and would be the one to look at.

The second example involved a real estate sale contract. The buyer backed out after sending a document that said: "I, X, agree to ..." At court, the buyer claimed that the document was not signed (no autograph was written at the bottom of the document). The judge, however, ruled that there was a signature in the document. The signature in this case appeared in the content of the message, where the buyer identified himself or herself as X. So, in fact, the document was signed, and the contract was legally effective.

Ben then went on to discuss what is currently happening with electronic signatures. Several states have been working on legislation regarding electronic signatures, and they don't agree with one another. Florida, Texas, Kansas and Iowa have adopted the liberal interpretation that Ben uses and separated the issue of signature from the issues of security, proof and evidence. He stressed that he personally likes this approach, but that there are disagreements in the legal community.

Other states, like CA, have adopted a more constrained approach. According to CA's Digital Signature Law, a digital signature is effective as a traditional signature if the signer has some kind of unique, verifiable code, if the code is under the signer's sole control, and if after the signer uses the code, one can determine if what was signed has been altered. (In addition, digital signatures need to comply with some state regulations that are still being elaborated.) So far, this legislation only regulates digital signatures involving the state of CA.

According to Ben, this view has advantages: it does not specify the technology, only the criteria. On the other hand, traditionally nothing has required that a signature has to achieve any degree of reliability or verifiability, so CA's legislation is less flexible in this sense.

Ben continued by talking about two completely different paradigms of electronic signatures, and the implications that their uses would have.

The first is the paradigm adopted by Utah's Digital Signature Act. Utah is a pioneer in regulating the use of public key cryptosystems in digital signatures. According to Utah's law, before I can use a private key as my signature, I need to go to a licensed certification authority (CA) to have my public key certified. After the certification, the private key has universal powers and I am responsible for keeping it secure. There is a presumption that any document that is signed with my private key is presumed to be my responsibility. It is not impossible to repudiate it, but it is difficult. This means that: 1) recipients of a signature have high evidence that the signer is going to be responsible for it; and 2) the owner of a key needs to be very careful with it, because the key can be used to sign any legal transaction, and the burden of proving the non-authenticity of the signature is on the person that owns the key. Ben's warning: "Don't let your spouse get it!"

Utah's digital signature law is an example of a concentrated transaction, where all the evidence is in one place: the signature. Dispersed transactions are the opposite of concentrated transactions. In dispersed transactions, the evidence that is needed for the receiver of a document to feel confident about it is dispersed across a number of factors and circumstances that are external to the signature. Previous relationships and the amount of money involved in the transaction are examples of such factors. When asked about Utah's failure to recognize the importance of other elements of a transaction, Ben said that he is concerned that the Utah law is too narrow and too detailed, and that there is no flexibility. However, he sees some areas in which signatures can be regulated this way. Electronic cash is one such example.

A second paradigm was then presented to contrast with the Utah approach. (Ben emphasized that he is not defending either approach, but is only giving us elements by which to judge for ourselves.) It is called PenOp, and uses the notion of dispersed transactions. It has been adopted by the IRS for electronic tax returns.

In PenOp, the taxpayer is shown the document he or she is about to sign. For the signature itself, he or she uses a digital pen to write on a digital tablet. The image of the signature, the speed with which it was written, and other biometric, "act-of-signing" data are then stored and constitute the signature. This biometric signature is then combined with the cryptographic hash value of the document, and the result is the signed document. For the IRS, having the document signed does not imply the document's non-repudiability. Instead, it is an evidence that you looked through your tax return and knew that you were legally responsible for it. In the case of a dispute, the IRS can not claim based only on the signed form that you were the person who signed it. Lots of other evidence is needed before they can prove that you were the signer.

Ben mentioned that the IRS+PenOp approach is probably not the Cypherpunks' favorite. Cypherpunks favor strong cryptography and oppose governments' intrusions into their citizens' private affairs. "What the IRS wants," Ben clarified, "is to get a little information about you, but not a whole lot. The security is mild, but things work in the context of other factors and circumstances." Ben said that for the IRS, security is a different issue and is ensured in different ways. For instance, the IRS has a private network to transmit its information.

An advantage of PenOp is that there are no keys, no passwords, and no training needed. One only writes one's signature.

Robert Gezelter asked about the shift in the burden of proof when adopting Utah's approach. Ben answered that there is in fact a shift. Traditionally, the receiver of a signature was the one to prove that the supposed signer actually signed it. Under Utah's approach, the signer is the one to repudiate it.

When asked about the method that UPS and FEDEX use, Ben said that their approach is less sophisticated, because only the bitmap of the signature is captured. No biometrics are used.

Someone asked what would happen if a PenOp signature is stolen and appended to another document. Ben said that the signature does not carry too much weight and its owner can easily repudiate it.

Ben Fried commented about the huge power that a signature has under Utah's approach and the ease with which this power can be transfered. Ben Wright agreed and said it is unprecedented in modern Western culture. He added that a private key is not a credit card. One can use private keys not only for financial transactions, but also for all other legal affairs like divorce, child custody, etc. One way to limit this power is to append disclaimers to the keys, restricting their use to specific purposes.

Session III: Policy and Economics

Digital Currency and Public Networks: So What If It Is Secure, Is It Money?

John du Pre Gauntt, London School of Economics

John posed the question "what is money?" He feels that electronic commerce is ready, or nearly so, from a technical standpoint, but that there has been insufficient consideration of what will "back" digital currency. He listed five properties of electronic markets (attributed to J. Yannis Bakos). One: the cost of communicating products and prices is lower. Two: the benefits of participation increase with the number of participants. Three: there are substantial initial switching costs. Four: they require large investments and benefit greatly from economies of scale. Five: potential participants are faced with uncertainties regarding the benefits.

He argues that given the importance of telecommunication for electronic markets, the market financial service providers will demand significant control over the telecommunications infrastructure and capacity. There are three types of money: the inherently valuable, that which has been directly backed by something inherently valuable, and money which has no direct value. Digital money is clearly not the first, and it is unlikely that states will initially give it the authority needed for the third, so what will back digital money? John suggested telecommunications bandwidth as a possibility. This concept raises many unaddressed issues. He detailed FLAG (Fibre-optic Link Around the Globe) as an example of the broadening connection between commerce and telecommunications. FLAG is a telecommunications cable which is principally funded not by telecommunications interests, but by investment houses.

While digital currency is being used on a small scale today, John argues that significant policy issues have been obscured by concern over the technical challenges.

Modeling the Risks and Costs of Digitally Signed Certificates in Electronic Commerce

Ian Simpson, Carnegie Mellon University

The risks and costs of electronic commerce have not been sufficiently analyzed. Ian devised a quantitative model for the risks and costs of electronic commerce with certain parameters and assumptions. The certifying authority and the merchant are trusted. Cheaters are modeled by an initial compromise of the system, followed by chances for detection both over time and due to spending rate. This model was used to explore the state space of cheating, and enables a quantified assessment of risks. The model is still in a primitive form, and further analysis and refinement is expected.

Eric Bach asked if the participants in the model as described were behaving rationally. Ian responded that since the model is still in development, the participants don't always behave rationally.

Session IV: Standard Payment Interfaces

Generic Electronic Payment Services: Framework and Functional Specifications

Alireza Bahreman, EIT

Electronic commerce is becoming more popular. Increased demand is beginning to strain the infrastructure. The current solutions are not complete. The question is how we should build a framework for these applications to allow interoperability. One good approach is to identify the set of services necessary for electronic commerce and build these services and supporting infrastructure simultaneously. This work will focus on payment services, specifically transactions that involve peers who exchange values. This is only a subset of electronic commerce.

Alireza does not believe that SET should be the only solution. Payment models other than credit cards should be supported. Even if we only look at SET, it is likely that multiple vendors will add enhancements to it, so multiple solutions are almost unavoidable. The goal is to create a framework that will allow these solutions to interoperate. This will allow innovation, making it possible for new systems to be introduced easily. A framework would present an organized view of the confusion of payment systems to the user. And application developers would not have to rewrite their applications every time a new payment protocol was introduced. Instead, there would be a generic API for payment systems.

In GEPS, there are three layers: applications, services, and resources. Applications use services and services use resources. Any layer can change without affecting the other two layers. Applications come in two flavors: normal, which use services, and specialized, which are part of the payment infrastructure. Some examples of specialized applications are brokers for digital cash, banks, traders, and the government, which might back a payment scheme.

There are five different services: Transaction Management, Capability Management, Preference Management, Payment Method Negotiation, and Payment Interface Management. Transaction Management, used by the other services, is the most basic service. Its responsibilities include logging, maintaining transaction status, failure recovery, etc. Capability Management provides an interface between a user's services and various payment providers. It acts as a layer of abstraction between services and payment systems. Preference Management handles a user's configuration details, e.g. whether to use cash or credit, the maximum amount to spend per day, etc. It can be used to rank payment systems according to user–defined criteria. The purpose of Payment Method Negotiation is to negotiate which payment system will be used between peers. Finally, Payment Interface Management is an abstraction of GEPS for applications which do not wish to become embroiled in all GEPS' details. It presents a uniform interface consisting of configuration, a user's wallet, value transfer, and error handling.

Related work includes UPAI, IBM Zurich's Secure Electronic Market Place for Europe (SEMPER), and Java Electronic Commerce Framework (JECF).

U-PAI: A Universal Payment Application Interface

Steven P. Ketchpel, Hector Garcia-Molina, Andreas Paepcke, Scott Hassan, and Steve Cousins, Stanford University

There is a diversity of payment mechanisms, all with different properties and different protocols. This diversity is a large problem for the application developer since applications will need to support all these protocols. All protocols have a general framework in common: the customer pays, some mechanism is invoked which handles the transaction, and the merchant receives the payment.

It would be useful to have an entity, called an Account Handle, that negotiates between the customer and merchant, handling details such as which pay-

ment system to use. The Account Handle would act as a proxy between a party and its actual commerce system–specific accounts. An entity known as the Payment Control Record would be used to control the transaction. It consists of a series of records of payment from one party to another. Each record would have transaction and control information. The customer and merchant would talk to the Record and the Record would talk to their Account Handles. Finally, entities known as Monitors will oversee the transaction and notify their owner, either the customer or merchant, through a callback mechanism if the status of a transaction changes. This removes the need for polling and gives the customer and merchant an entity to query about a transaction's status.

The system, which is object/method based, uses CORBA. Each entity has a set of methods which it can perform. Account Handles can create, open, close, or delete accounts as well as performing other maintenance tasks. The Payment Control Record can set the amount, source, destination, authorizations, etc. The Monitor, the implementation of which is left to the application programmer, has several types of status messages which it sends to the application: transaction completion, failure, or still pending. Additional programmer–definable subtypes of these messages can also be used.

This system can be adapted to support the delivery of electronic goods. The methods are the same, but the role of the customer and merchant are reversed. Instead of using an electronic commerce protocol, we would use an electronic delivery protocol.

Work in progress consists of adding more functionality such as support for pay per view, subscriptions, and shareware. Also, existing commerce protocols should be integrated into the system. Finally, security, which is currently absent, will be taken care of in future versions of CORBA.

In response to a question about what UPAI has to do with a digital library, Steve stated that the library needed a method of billing its customers for electronic payment services. Doug Tygar wished to know how this system related to a credit card payment via SSL. Steve responded that this would be just another payment protocol. Another person pointed out that UPAI does not present the user with all the options available in the First Virtual protocol. Agreeing, Steve said that while support for these options could be added, it would require application support as well. In general, the goal is not to support every foible of every protocol. Another question was whether only a few protocols would be in use eventually. Steve believes that there would be more than one protocol in use, so there is value in a common interface. Also, a common interface would make it easy to try new protocols.

Payment Method Negotiation Service: Framework and Programming Interface

Alireza Bahreman and Rajkuman Narayanaswamy, EIT

Payment Method Negotiation is one of the five services of GEPS. The purpose of Payment Method Negotiation is for a merchant and a customer to come to an agreement on all details of the method of payment when they wish to transfer funds. The service deals with a variety of details about a payment, not just the choice of cash or credit. Some of the motivations for using negotiation are to make good payment method choices in a situation with too many options and to speed up the payment process through automatic negotiation.

Negotiation can take several rounds. In the prepayment stage, while a customer is shopping but before he wants to make a purchase, he could negotiate with the merchant just to make sure that there is a common payment method between them. This could be done in the background as the customer shops. When it comes time to pay, negotiation should be accomplished quickly. Negotiation could be symmetric, or one party could simply name a method. Then comes a finalization stage, in which the method is picked and the transfer of funds begins.

Related work includes the Joint Electronic Payment Initiative (JEPI), which addresses the protocol used to exchange messages. It includes the Universal Payment Protocol from Cybercash and the Protocol Extension Protocol over HTTP. JEPI is concerned with syntax and message flow, but it is not tied to a particular payment system. It is hoped that the payment method negotiation work will use JEPI for message syntax.

Negotiation is closely linked to the payment preferences and payment capability services in GEPS. It is assumed that both parties in a transaction have GEPS. There are three components in the implementation: input and output, payment negotiation, and a user interface. The user interface is specified, but left to the application to implement.

There are several policy issues. It is not clear who should or would want to reveal their capabilities. Also unknown is how long a negotiation will take. Finally, a method of picking the party who is to choose which payment system will be used has

not been defined. Some policy decisions can be controlled by the way the application interacts with GEPS. Specifically, the application can control the order of negotiation, how many or few capabilities to reveal, how long negotiation will take, and in which phase the negotiation is. For further policy control, the GEPS objects can be subclassed.

Implementation of this system is in progress, though currently, the emphasis is on promoting the system. Alireza hopes that a very large and influential company will adopt GEPS as a standard, urging other companies to adopt it as well.

Session V: Atomic Transactions

Anonymous Atomic Transactions

Jean Camp, Sandia National Laboratory; Michael Harkavy and J. D. Tygar, Carnegie Mellon University; Bennet Yee, University of California, San Diego

Jean Camp began by pointing out that money sent over open networks is subject both to attacks and cheating by untrusted parties, and to network failures. She then defined several terms: money atomicity, which means that money is conserved in the system; goods atomicity, which means money atomicity plus guaranteed delivery of goods; and certified delivery, which means goods atomicity plus proof of precisely what goods were delivered. This discussion refers strictly to information goods, not to physical goods.

Jean went on to define anonymity as meaning that the identity of the consumer is not revealed, and to note that previous anonymous transaction protocols were not atomic: after the purchase both the customer and the merchant have the token and may be racing to cash it.

Jean stated that she would offer a protocol which was both anonymous and atomic, and give a proof by example. Her assumptions were: secure communication channels that don't reveal the consumer's identity; blinded signatures (Chaum) that enable signing of unseen data so that signature verification can't be linked to the initial signature; and a transaction log of messages from the customer, the bank, the merchant, and the log itself, which is an agent recording to publicly readable, reliable storage.

This is a two-part protocol. First there is a withdrawal or exchange of the token via a blinded request to the bank, which the bank signs; then the customer unblinds the request to get the token. The token is itself a public key, which the customer can prove has value, like a single-use certificate. Second, there is a purchase. At the customer's request, the merchant sends to the customer the goods in encrypted form along with a contract with the price and description of the goods. The customer sends an approval to the bank, the bank sends an approval to the merchant, and the merchant sends the encryption key to the log, which makes it publicly available.

The log is the transaction coordinator, and issues either a global commitment of the transaction or a global abort. We assume that the merchant trusts the transaction log not to release the key without issuing a commitment, and that the consumer trusts the transaction log to publish the key in a timely manner. The bank and the transaction log could potentially be combined – separation helps to minimize how much each party needs to be trusted. The protocol is anonymous because each token is a public key that is not linked to the customer's identity.

Jean then discussed some possible variations. Reusable consumer keys would be more efficient, but less anonymous as they would allow the bank to link a series of transactions. Cryptographic timestamps in the log would help to reduce the risk of delay and minimize the damage if the log were compromised. Encrypting the log would get rid of observers. Two-sided certified delivery would allow the merchant to take the burden of proof.

Questions for future development include how to make the protocol more efficient and more scalable, and how to reason formally about anonymity. Policy issues relevant to this work include legal requirements on transaction sizes and aggregate data collection, and key escrow.

Dan Geer asked Jean to elaborate on how she sees the role of anonymity, and whether it is possible to achieve privacy through other means. Jean answered that it is possible to get privacy through policy mechanisms to some extent, but that this is subject to violation since you have to trust others to maintain your privacy. Jean was asked if any prototype of this system has been built, and replied that none had, nor had any analysis been done, and that public key systems are generally expensive. Doug Tygar then added that storage requirements would be quite large as they are for all digital cash systems which must keep log and bank records.

Strongboxes for Electronic Commerce

Thomas Hardjono and Jennifer Seberry, University of Wollongong

Thomas Hardjono suggested that an electronic counterpart to physical strongboxes could be a useful mechanism for electronic commerce. These elec-

tronic strongboxes would be used for secure storage of anything digital, including certificates, contracts, cash, coins and checks. Access to an electronic strongbox service would ameliorate the need for users to have large storage spaces personally for storing all their electronic cash, and would facilitate trade and exchange.

Thomas further suggested that physical valuer services might generate unforgeable digital representations of the value of existing physical goods. With the real goods in secure physical storage, on-line valuers could be used to verify the authentic ownership of items in the system. to split items into sub-items, and to help the exchange facilitator, who would mediate exchanges between customers to ensure that they were honest and irrefutable. A formal association would oversee the entire system and its participants, and a notary would handle disputes in cooperation with the association.

Such a system would require proof of the retrieval or storage of an item in a strongbox, proof of the ownership of an item with anonymity, proof of the submission of an item for valuation, proof of an exchange transaction, and detection of illegal and duplicate items.

Bob Gezelter asked if this implied that a user would in effect create a shadow currency by depositing something of value; Thomas answered that yes, it did. It was then pointed out that anonymously tradeable certificates equal cash.

Model Checking Electronic Commerce Protocols

Nevin Heintze, Bell Labs; J. D. Tygar, Jeanette Wing and H. Chi Wong, Carnegie Mellon University

Chi Wong outlined an electronic commerce system consisting of a customer, a merchant, a bank, and goods that can be sent over a network. She then defined two main properties of the system: money atomicity, which requires that the total money in the system remains constant; and goods atomicity, which requires that the customer receives the goods if and only if the merchant receives the money. She described the possible failure modes of the system as either network or processor failure, and cheating, which would be an attempt by the customer to double-spend, or an attempt by the merchant to double-deposit.

Chi then explained that model checking, an approach based on exhaustive search of finite state spaces, could be applied to this system to verify its properties. A model of this system and a property specification could be given as input to a model checker, which would return a yes, meaning that the properties were verified, or provide a counterexample.

In FDR model checking, which stands for "Failures Divergence Refinement," the system model and the property specification are both state machines represented in the same language. The model checker then implements a refinement relation to see if the state space given by the model is a subset of the state space given by the property specification. Chi described building FDR models of simplified versions of the NetBill and Digicash systems, which were then run through a model checker; she referred the audience to the paper for the results, noting that while model checking has been useful for hardware verification, and recently also for software verification, this is the first time it has been applied to electronic commerce protocols. As future work, Chi hopes to create a more complete failure model, do more complex runs, add more properties, and explicitly represent the role of cryptography, which is currently abstracted away.

Dan Geer asked whether it was possible to use these techniques during design rather than just analytically; Chi thought probably not with FDR model checking, but that automatic program generation refinement tools might be useful. Eric Hughes asked how the size of the state space affected the process; Chi referred him to the appendix, noting that the example given had about 3000 states. Eric then asked about asymptotic properties, and Chi answered that the model checker looks at all possibilities and thus is exponential; Eric asked if symbolic modeling had been considered as a technique to cut it down, and Chi replied that it had, and that a tech report would follow.

Jeanette Wing then followed up on Dan Geer's question by stating that yes, you can use these model checkers for iterative checking and design of protocols, and that it would be pushbutton technology so the iteration would be fast. She noted that SMV is a richer language for expressing properties, and that FDR is tuned to check deadline detection and is more limited.

Session VI: Experience

BigDog: Hierarchical Authentication, Session Control, and Authorization for the Web

Benjamin Fried, Andrew Lowry, and Morgan Stan-

ley

The goal of BigDog is to use the WWW to interact (e.g. deploy applications, exchange data) with existing clients, not to recruit new clients or establish new relationships. BigDog incorporates different levels of security to accommodate different levels of data sensitivity. The SSL is used to encrypt all data flow. "Home site" (i.e. IP address) information is also used. An access control list is maintained on a per use, per resource basis. The work evolved, and Ben stated that their experience is that plug-ins were troublesome. The model of separate communicating protocols provided more freedom and worked better. He mentioned that this work is related to OM-Access.

Ben was asked if user input was used to design BigDog. He answered that some input was used, but commented that users are not necessarily educated about security issues. Eric Hughes asked what plans there were for risk analysis. Ben indicated that it would be nice to be able to indemnify to auditors. Bob Gezelter asked why the IP address information was used, since it can be spoofed, to which Andrew responded that it was used as only a minor security component – a "half step" in security. Steve Jones asked how users or resources were grouped. Andrew said that that such grouping took place in the administration, not in BigDog itself. When asked, Ben said that it was hard to estimate exact costs, but that around three person months went into the project. Ed Uielmetti asked if there was support for out of band authentication. Andrew replied that one application does use it. In response to a question from Andy Rabagliati, Andrew stated that the system operated on SunOS/Solaris.

Financial EDI Over the Internet: Case Study II

Arie Segev, Jaana Porra, and Malu Roldan, University of California, Berkeley

Bank of America, for which this work was done, is a multi-billion dollar corporation and operates in 36 countries, so size and scale presented extra challenges to the project. The planning involved meetings of many people from many different fields. The project was approached as a learning experience. Privacy Enhanced Mail (PEM) was used to provide security. During phase 1 of the project, the scale was kept small. No messages were lost and no tampering was detected, but very few transactions were processed. After this early success, the limits and volumes were increased, and a more substantial number of transactions were processed without detected tampering. Testing showed that the FEDI system (rather than the network) caused most of the delays in messages. Public perceptions of security are an obstacle to this sort of project.

Dan Geer asked if a large part of the problem was from the lack of value added networks; Jaana replied "yes". Dan followed by noting that VANs are a small part of the total cost of existing financial transactions. Jaana responded that this is true, and that the cost benefit analysis for this approach is not entirely clear. Bob Gezelter commented that over-batching of messages could be responsible for much of the latency, and Jaana agreed that this was possible.

Scalable Document Fingerprinting

Nevin Heintze, Bell Labs

In part as a response to being plagiarized, Nevin became interested in finding a scalable means to catch copied (or partially copied) papers. The fingerprinting should be sufficiently robust to catch moderate variations of the same paper. A sliding window of chunks of a certain length (e.g. 30 characters) is used to go through the document, creating a fingerprint against which other documents' fingerprints can be matched. Each document's full fingerprint can't be stored if there are many documents, so the method used was to keep a fixed number (say 100) of chunks from each document in the database. In order to reduce false positives infrequently occurring chunks were used in the fingerprints. This was combined with hashing to select certain chunks and increase matches.

The method worked well when tested on a corpus of CMU technical reports and on a larger corpus of reports available electronically. Even a one percent match indicated a likely similarity. In practice, techniques such as ignoring headers, introductions, references, and other highly repetitious information helped to reduce false positives and provide more precise detection. In order to defeat an iterative attack, in which a plagiarized document is repeatedly modified until the matches are eliminated, occasional random resamplings should be performed, updating the chunks associated with each document in the database.

Lunch with Invited Speaker

Designing New Rules of the Road for Electronic Commerce in Digital Information

Pamela Samuelson, University of California, Berkeley

Currently, there are three new sets of rules being proposed for the information superhighway. One set is about the validity and meaning of contracts, another is on revisions to copyright law, and a third concerns new intellectual property law for database contents. All three areas will change the law (both national and international), strengthening the rights of information vendors. These changes are being made to encourage the electronic commerce market and to help United States information providers continue to dominate the market. These rules are close to being implemented.

One of the proposed laws would validate the terms of "shrink wrap" licenses. These are traditionally debatable since a licensee does not see the license until after the shrink wrap has been broken. A federal court ruled in favor of this view, finding that a shrink wrap license was indeed invalid. However, an appellate court overturned this ruling, stating that the license was valid and could be enforced if a user continued to use the product after seeing the license. This decision implies that a license to use software, rather than ownership of a copy of the software, can be sold. If a person violates the license, then that person loses the license to use the software.

Another change is coming in the area of implied warranties. Currently, an implied warranty basically says that a product should work. But under the proposed changes, unless there is an explicit statement of quality, the implied warranty would be that the manufacturer did its best to make the software correct.

In the area of copyright law, a large expansion over the control of reproduction has been proposed. Today, a copy has to be "tangibly fixed" to be considered an infringement. In the future, temporary copies, even caching an image in RAM, could be considered an infringement and thus can be controlled by the owner of the original. Any digital transmission of an object would be considered a communication of the work to the public and could be controlled by the object's owner. This would mean that, contrary to traditional copyright law, a person can be restricted from giving a document to a friend when he no longer wants it. Also, web crawlers, since they keep temporary copies of documents, would become illegal.

All of these proposals are part of a Clinton administration white paper. The strategy behind these proposals is to garner international support, which will force these changes to be adopted in the United States. Congress would not need to adopt any international treaties that are proposed, but could simply implement equivalent legislation.

In the area of database content, there is a proposal to grant the producer of a database, if a significant amount of effort was invested in creating it, 15 or 25 years of exclusive control over the extraction of information from the database without exceptions for fair use or research. The goal of this proposal is to protect the United States database industry against people who pirate information.

Bob Gezelter predicted that not allowing caching would drive the Internet into the ground with high load. He wondered whether there would be an exception for delivery mechanisms. Pam answered that if the Clinton administration had thought about delivery mechanisms, they would probably view them as infringing. Another opinion is that caching is an aid to people who deliver information, so under fair use, it should be acceptable. But in the Clinton administration white paper, caching in RAM was specifically mentioned as a cause for infringement. Another person asked whether these proposals were sparked by "banality or stupidity." Pam replied that the motivation was to protect the entertainment industry, but that it's time we moved away from the attitude that something that's good for a specific business is good for the whole country.

One question was whether everything on the World Wide Web would need to explicitly state what rights were granted to users. Pam replied that the proposals would change the ground rules and we would no longer be able to assume rights such as keeping temporary copies. One person wondered whether people would own their personal information, such as data traditionally used for marketing. Twenty years ago, Pam replied, the answer would have been a flat "No." Today and in the future, information is becoming more like property. In fact, the proposed database treaty would make information into property.

Another question was whether a web site with many links to other sites could be considered a database. Pam said that this issue has not been addressed yet, but the database bill in the House of Representatives defines a database as a collection of information materials arranged in a systematic way. In addition, copyright law would still apply to the content of the web site.

One person was curious about whether the Europeans would have fair use exceptions to copyright laws. Pam responded that there are several rules about this. In one approach, users would have the right to take insubstantial parts. Another approach would be to disallow taking any part, no matter how small. The last approach is that a piece, even a substantial one, could be extracted for illustrative purposes such as education, but not for analysis. The drive for new database legislation was unknown at first in the scientific and education communities. Recently, these communities are coming out against the new legislation.

One person wondered whether we could use cryptography and fair competition laws to take care of concerns about people copying CD-ROMs. Pam replied that most violations of the proposed laws would already be violations of current laws. The goal is to stop a slight leakage from becoming a hemorrhage.

Session VII: Protocols

A Protocol for Secure Transactions

Douglas H. Steves, Chris Edmondson-Yurkanan, and Mohamed Gouda, University of Texas, Austin

Douglas Steves opened the session on protocols. He and his co-authors are interested in secure transaction protocols as a means of achieving secure electronic commerce. They proposed a protocol with strong relational properties.

Doug started by contrasting secure communication protocols with secure transaction protocols. In secure communication protocols, the main concerns are privacy, authentication, integrity and nonrepudiation. PGP and PEM are the best known examples of this class of protocols. SSL, SHTTP and SET, although they have the notion of sessions, do not establish relationships between the messages in a session, and are therefore considered examples of secure communication protocols.

According to Doug, message security properties are not enough for secure transactions. Relational properties that link the multiple actions in a transaction are crucial. He then identified three relational properties: atomicity, isolation, and causality. They are all present in the standard database (DB) theory and legal contracts. Atomicity and isolation have been discussed by Tygar and his colleagues, but causality is new here. Basically it says that it is not enough for two (or more) messages to be part of the same transaction; the ordering of these messages is important.

At this point, he opened a parenthesis and said that secure transaction protocols should lie underneath electronic commerce protocols. The question of role playing (who is the customer and who is the merchant), as well as forms of exchange media (credit cards or electronic cash) should all be part of this higher level. Close parenthesis.

Returning to the main focus of the talk, Doug said that the way that he looked at atomicity was different from the way that Tygar looked at it. Atomicity, for Tygar, appears in a concentrated form: both the commit and exchange of goods and money takes place in one point in time and space. Their view of atomicity is dispersed: commit and exchange are physically and logically separated, the exchange being dependent on the commit.

With respect to isolation, Doug Steves and his colleagues' definition is that all or none of the transaction messages are valid. In DB operations, isolation is guaranteed by the DB manager, which only allows the result of a transaction to be seen by the outside world when the transaction is complete. In a message exchange system, isolation is hard to guarantee, since messages sent over the network can be caught, copied and stored at will.

Causality was first discussed by Lamport, who introduced the notion of vector stamps to indicate the ordering of messages in distributed systems. Under this approach, the receiver of a message only knows the number of messages that have been sent and received previously by the sender, but does not know the contents of the messages. Authentication was added to vector stamps by Tygar and Smith. In 1993, Ken Birman proposed piggybacking messages on top of other messages, thus introducing a new form of causality where one can talk about the contents of previous messages. In 1996, Gong and Reiter combined Birman and Tygar and Smith's proposals and obtained secure causality. It is this form of causality that Doug Steves uses in his transaction protocol. Thus, when committing to a transaction, the protocol commits to the messages of the transaction and to the order of the messages in the transaction.

The protocol that Doug Steves and his colleagues have implemented proceeds in three phases (initiation, exchange and termination) and uses half-duplex communication. Atomicity and isolation are achieved via the two-phase commit mechanism, and causality is achieved via Gong and Reiter's mechanism.

During the question period, someone stated that SET also satisfies isolation, atomicity and causality. When asked how his protocol differs from Gong and

Reiter's, Doug said that Gong and Reiter did not address atomicity and isolation.

PayTree: "Amortized-Signature" for Flexible MicroPayments

Charanjit Jutla, IBM; Moti Yung, Banker's Trust

Charanjit Jutla presented the PayTree payment mechanism. He started with a summary of the major steps in micro-payments systems and pointed out that it is crucial to make payments from customers to merchants computationally efficient.

Public key signatures are the most reliable way of authenticating or verifying payments, but they are computationally expensive. The idea is therefore to minimize the number of public key signatures that are required in issuing or authenticating (a sequence of) certificates for payments.

Charanjit briefly explained the workings of PayWord, a scheme that explores this idea. Based on Lamport's one-time password scheme, PayWord only requires one public key signature to issue a number of payment certificates. By linking the validity of future payments to the validity of a previous payment through cheap hash functions, the cost of the signature operation is amortized.

PayWord, however, is not able to determine who the cheater is when fraud occurs. For instance, if a certificate is presented more than once for redemption, the bank does not know if the customer double spent it, or if the merchant colluded with another merchant. Also, it is not well suited for web surfing, because payment certificates generated by a signature can not be spent with different merchants.

The idea of PayTree was then presented. It is based on Merckle's authentication tree scheme, and uses the following data structure: a tree whose leaf nodes are labeled by secret random values, whose internal nodes are labeled by the hash value of the nodes' successors, and whose root is signed. Because of the tree structure, PayTree is more flexible and allows payments to different merchants to be made using different parts of the tree. This means that multiple merchants can now share the cost of a public key signature. Charanjit proceeded to describe ways of issuing and verifying payments in three different scenarios. In the first scenario, a tree is used to pay only one merchant; in the second scenario, a tree is used to pay multiple honest merchants; in the last scenario, multiple merchants with arbitrary behaviors are considered. The computational complexity of each of the cases is presented.

In the basic PayTree mechanism, the individual payments all share the same value and each tree has a pre-defined total value associated with it. But PayTree can be slightly modified to implement trees with multiple denominations, unlimited payment potential or divisible coins. It can also be used as a module of other payment systems.

Someone in the audience commented that the flexibility of PayTree increases bandwidth and storage consumption. Charanjit agreed.

Agora: A Minimal Distributed Protocol for Electronic Commerce

Eran Gabber and Abraham Silberschatz, Bell Labs

Eran Gabber presented Agora, a micro-payment protocol that relies on the assumption that public key signatures are not that expensive. The goal was to design a micro-payment protocol that is compatible with existing tools and protocols, scalable and distributed, and whose overhead per transaction is minimal. The overhead that matters is given by the number of messages and signatures required by the protocol for a typical transaction. Micro-payments that have been proposed so far are not designed with the number of messages in mind. They are not concerned with compatibility with existing web protocols either.

Agora fulfills the requirements mentioned above. It is minimal: a typical transaction does not generate more messages than what is required for web browsing. It is distributed: a typical transaction can be verified by the merchant without contacting any online authority. It enables online purchases: one does not need to go to a broker first to get scrips.

Eran went on to describe the protocol. There are five kinds of entities: a central authority, who certifies the banks' public keys; banks, who manage accounts; customers; merchants; and arbiters, who also need to be registered with the central authority. Both customers and merchants have accounts (with expiration dates) at the bank. The expiration date helps in housekeeping billings and payments, and lessens the effect of brute force attacks. A billing period is associated with the lifetime of each account. Everybody has public keys and private keys. New sets of keys are generated for each billing period. Whenever the merchant starts with a new pair of keys, he or she takes them to the central authority for certification. When customers change their keys, they go to their banks to get new certificates for the next billing period. A certificate is a customer ID, signed by the customer's bank, and includes the expiration date, an account number and the customer's public key. The certificate is used as a promise of payment. Banks would only issue new certificates

to customers that paid their bills for the previous period.

For a typical transaction, the protocol uses four messages. The first message has the customer asking for price quotations; the second message has the merchant reply with the quotations; the third message contains the purchase order from the customer; and the last message, from the merchant, contains the goods (or error messages). This is the same number of messages that free web browsing takes. The first message is generated when the customer clicks on a link containing a page of price quotations; the page is displayed to the customer when the second message is received; the third message is generated when the customer clicks on an specific item; the item is then returned in the last message and displayed to the customer.

Note that the merchant sends more quotations than have been asked for. This is an advantage, because if the customer decides to purchase a second item on the same page of quotations, only two message are needed in the transaction. Also, only one signature is used for each page of quotations.

Because it is assumed that merchants distrust banks with whom they have not had relationships before, the first time the merchant receives a certificate issued by a bank that he or she does not know, the merchant may go to his or her own bank and verify the unknown bank's public key. (Every bank's key certification is broadcast to all the other banks by the central authority.)

Merchants can therefore accept certificates in a distributed fashion, and only submit them at the end of each billing period. This offline mechanism does allow merchants to be defrauded in situations where they accept certificates from accounts that have been revoked or expired.

Eran continued with a security analysis. Because all messages are signed, authenticity, tamper-proofness, and non-repudiability are guaranteed. Because all messages come with sequence numbers, replay attacks and double charging by merchants can be prevented. But, as mentioned before, full distribution makes fraud possible in this protocol. To lessen the possibility of fraud, it is possible to enhance the protocol and have merchants talk to customers' banks now and then. (Banks need to keep track of all revoked certificates.)

If the customer sends a purchase order, which constitutes a promise to pay, and never gets anything back or gets something else, then he or she can go to the arbiter and present the quotation (which includes the description of the goods) and the purchase order. The arbiter can then demand the goods. If the merchant does not comply with the demand, the arbiter has the power to revoke the transaction.

During the question period, Mark Manasse asked about the impact of doing digital signatures on the latency of transactions. Eran replied that the customers' machines are assumed to be idle, so there is no problem there. Merchants, however, should have a farm of PCs dedicated to signature generation and verification. Also, one can always use optimized signatures to make things more efficient. Marvin Sirbu commented that Agora reduces the number of messages by transferring liability to the merchant. Eran agreed.

Panel Discussion

Electronic Commerce in Practice – What Have We Learned?

Clifford Neuman, University of Southern California (moderating); Ed Vielmetti, First Virtual Holdings, Inc.; Marc Briceno, DigiCash; Steve Crocker, Cybercash; Daniel Geer, Open Market, Inc.; Malu Roldan, University of California, Berkeley; David Van Wie, InterTrust

The members of the panel each spoke briefly about their experience with building electronic commerce systems in the real world. Steve Crocker described Cybercash's automation of the customer/merchant payment authorization process, and said that they have about 150 merchants using their main system, and about 20 merchants using their newer small payment system. Dan Geer stated that OpenMarket has moved its focus to automating web commerce between businesses, rather than between merchants and customers. Malu Roldan described how existing companies move toward using the web, saying that there's a lot of interest in cheap initial experimentation. David Van Wie said that InterTrust is focusing primarily on information commerce, distributing movies, newspapers and software electronically, and that InterTrust also believes that business to business is the place to start. Ed Vielmetti described First Virtual as a global, low cost payment system in which anyone can be the buyer and anyone can be the seller, with about 200 merchants and 180,000 customers. Finally, Marc Briceno described Digicash's e-cash system for providing the buyer with complete anonymity.

General discussion and questions from the audience followed. Some points of agreement seemed to be: that customer service actually tends to be a bigger cost than fraud, at least for retail systems; that

our understanding of what we want in an electronic store is still evolving; that there is plenty to be done in automating existing real world systems; and that merchants are looking to reduce costs more than to increase business. Opinions were divided on the usefulness of risk modeling. The difficulty of building global electronic commerce systems while different countries have different laws and expectations was acknowledged.

Session VIII: Security

Organizing Electronic Services into Security Taxonomies

Sean Smith, IBM Research; Paul Pedersen, Los Alamos National Laboratory

As the world moves to depend more on electronic services, it is desirable to have a method to analyze the tradeoffs being made. We wish to know the vulnerabilities and points of attack of any given system. Sean suggests a structured approach, building a taxonomy of the vulnerabilities from the inherent structure of the provided services. They placed a partial order on the various services which can be provided, and looked at the differences between the two steps. Services inherit vulnerabilities from below (i.e. weaker services), and stronger services can introduce new vulnerabilities as well. This taxonomic structure also works to model points of attack, which can be thought of as "inadvertent services." An example case is kiosks. There were difficulties resolving the levels of services into quantum steps. A variety of properties (e.g. spatial extension, input privacy) were used to describe the provided services and build the structure of vulnerabilities. Sean emphasized that this is a prototype, and that it is being refined and extended.

Doug Tygar asked if there were hopes for making the process more general. Sean said that the system has some generality, more than shown in the example. Ed Uielmetti asked about weaknesses that are the result of combined services, and are not weaknesses in the components. Sean responded that this is not covered by the method, but that work is in progress.

WWW Electronic Commerce and Java Trojan Horses

J. D. Tygar and Alma Whitten, Carnegie Mellon University

Alma brought up ways in which WWW commerce can be attacked that are based on the way people browse the web and weaknesses in the security model. The attacks presented do not rely on implementation faults, but rather are weaknesses in the way the system is designed. The two attacks presented are bogus remote pages and local Trojan horses.

The bogus remote page attack relies on the lack of verification of who operates a particular page or electronic storefront. Users do not usually check address names, and domain names are available that could be used to plausibly impersonate a real site. Given the ease of copying electronic information, an attacker simply creates a site which looks like a trusted site. When the user's browser is pointed to the bogus page address, an applet which spoofs the trusted page takes control, and thus the bogus remote page enables the local Trojan horse attack.

Once the attacker applet has control, it can spoof secure dialog boxes and act like the spoofed site while obtaining potentially sensitive information (such as a password or credit card number) through the user's entries. This information can be sent back to the attacker's site by hiding it in the page access requests. After this is done the applet passes control to the real site, and the attack goes unnoticed.

Code signing is not a sufficient fix for this problem, since it requires a trust basis and it will still be desirable to run unsigned applications, since code verification is expensive. A solution is window personalization. Make the trusted aspects (such as the background of the dialog boxes) 1: distinctive and easy to recognize for the user and 2: difficult for a prospective attacker to predict. To make this method work, Alma suggests the following: require selection at installation, educate users, offer many choices with randomized defaults, and avoid company logos or other predictable designs. There are extensions of this approach to ATM and POS applications.

Alma was asked why location couldn't be used to indicate genuine dialog boxes, and responded that pages may occupy the entire display. Bob Gezelter mentioned the importance of good randomization, and Alma agreed. Alma was asked why the local Trojan horse was necessary. She replied that the local Trojan horse attack is more general than just the bogus remote page. Ben Fried suggested that code signing with trust determined by the vendors would alleviate the problem. Alma said that even if trust assumptions were given, code signing is inherently subject to potential flaws.

On Shopping Incognito

Ralf Hauser, McKinsey Consulting; Gene Tsudik, University of Southern California

Gene presents a system for anonymously performing electronic commerce. The system involves four phases, browsing, obtaining offers, payment, and delivery. Privacy is important in all phases. Identity information can be used, for example, in junk mailing lists or unfair pricing. It is important that transactions be unlinkable. In the first phase, prepurchase browsing, the consumer collects signed offers of price and description, which may or may not be transferable. Gene presented two protocols, prepurchase browsing (PPB) and electronic merchandise delivery (EMD). The PPB protocol supports anonymous browsing, but identity information is revealed during delivery. EMD supports anonymous merchandise delivery. They can be combined to form a completely anonymous system. The protocols provide signatures to the participants which enable them to prove in court what transpired. Gene referred to this as a cop out.

Eric Hughes asked why Gene considered the court system a cop out. Gene clarified that he simply means that actual court involvement would be very costly, but that, as in paper transactions, the backing of the court system is necessary. Andy Rabagliati mentioned that the wide prevalence of transatlantic caching would help support privacy, and Gene agreed that it would help support browsing. In response to a question, Gene indicated that merchants would want to support this to provide for their customers, and that it might have applications in situations of political oppression as well as the obvious application in pornography. Bob Gezelter mentioned that upcoming copyright legislation might interfere with some of this protocol.

Session IX: Software Agents

Market-Based Negotiation for Digital Library Services

Tracy Mullen and Michael P. Wellman, University of Michigan

Tracy Mullen presented work on market-based negotiations for digital library services. This work is based on two assumptions: 1) available resources are limited, and 2) what people may want to do in a digital library is unpredictable. Given these assumptions, digital libraries should provide a flexible, open and extensible infrastructure that supports different market practices.

The University of Michigan Digital Library (UMDL) project is designing and implementing a digital library based on a system of software agents that interact with each other and with end users. Software agents are used to perform various activities needed to deliver goods and services. Because there are multiple competing agents trying to accomplish multiple tasks as efficiently and as cheaply as possible at a given time, resource competition needs to be resolved. UMDL sets the agents to negotiate with each other, so that globally optimal agreements can be reached. Instead of hardwiring pre-defined negotiations, UMDL uses user-definable auctions that can be dynamically established when goods or services are to be sold or bought. Auction mechanisms need a number of parameters to be fully specified. These parameters include the type of goods, price quote policy, price quote interval, clearing policy, and tie breaking, among other things.

UMDL's digital library is a market place with dynamically evolving configurations that include goods that are being sold and the mechanisms by which they are negotiated and exchanged. When given buyers' demand profiles and the current resource congestion profile, the system will decide on a level of service for each buyer. Currently, there is an auction server working.

During the question period, someone asked about the danger of shill-bots. Tracy answered that it is possible to use certificates of "honesty" to decide who is allowed to transact in the system. Christian Frank wondered about the scarcity of information goods. Tracy pointed out that one should look at other resources as well. Users' time and the system's computational resources can both be scarce. Someone asked about protections that the system offers against unexpectedly high demands. Does UMDL prevent crashes from happening in such scenarios? Tracy answered that the marketplace is self-regulating. As the demand increases, the price of the goods or service also increases, which can drive the demand down. But UMDL can also resort to distributed auctions. The last comment came from Doug Tygar: "The notion of having auction markets for information goods is terrific, because that means that we may have a futures market and I can short my colleagues' papers!"

Information and Interaction in MarketSpace – Towards an OpenAgent-based Market Infrastructure

Joakim Eriksson, Niclas Finne, and Sverker Janson, Swedish Institute of Computer Science

Joakim Eriksson's talk described an agent-based infrastructure that he and his colleagues are building to automate market interactions, such as searching for business partners, negotiating, and settling a deal. Although the Web can be used for such purposes, it is not quite adequate because: 1) the data on the Web is unstructured (there is text, graphics, video, etc.), and 2) the type of interaction offered by web browsers is not tailored to business transactions.

The main focus of the work is to develop information and interaction models. The information model should satisfy the following requirements: 1) the data should be presented as structured knowledge; 2) the participants' interests and their potential business deals should be adequately represented by well-defined description languages; and 3) the approach should be object-oriented. With respect to the interaction model, it must be simple, but able to model a wide range of types of market interactions. Examples of primitives one can use to carry out an interaction in their model are: ASK, TELL, NEGOTIATE, OFFER, ACCEPT, and REFUSE. Joakim then showed how information is represented and how interactions are modeled under their approach through a number of examples. The examples included direct transactions between two individuals, someone seeking the help of a broker, and an auction.

Joakim then briefly talked about other components of Market Place, the project that the work in this paper is part of. One of the components connects Market Place with the Web, allowing users of one system to use the other. The Market Place group is also developing agents that have roles (brokers, auctioneers, buyers, etc.). The system will be tested in Spring 1997.

During the question period, Eran Gabber asked about the use of English as a universal language in their system. Joakim said that the Web also has this problem, and that the Market Place is not trying to solve it. Instead, they plan to just plug in solutions, once they are available.

A Peer-to-Peer Software Metering System

Bruce Schneier and John Kelsey, Counterpane Systems

Bruce Schneier was delayed by bad weather and the presentation of this paper was canceled.