# E-Commerce Trust Metrics and Models

**DANIEL W. MANCHALA**
*Xerox Research and Technology*

Traditional models of trust between vendors and buyers fall short of requirements for an electronic marketplace, where anonymous transactions cross territorial and legal boundaries as well as traditional value-chain structures. Alternative quantifications of trust may offer better evaluations of transaction risk in this environment.

The Internet gives vendors an easy way to set up shop for electronic commerce throughout the world. In an extension of mail- and phone-order transactions, customers usually pay for e-commerce goods and services through a credit card. The transmission of credit card numbers requires vendor sites to support encryption through the Secure Sockets Layer[1] or Transport Layer Security[2] protocols. SSL can be enhanced through special payment protocols such as Secure Electronic Transactions[3] or Electronic Commerce Modeling Language.[4] However, as millions of customers begin to participate in e-commerce, we can expect increased transactions of extremely varied quantity and value; in addition, the goods and services transacted will be subject to very different legal regulation and economic risk. This emerging marketplace will require the ability to make distinctions that the credit-card transaction model does not support.

How do we set measurement criteria to make these distinctions? One way is to quantify *trust*. This fundamental concept in managing commercial risk refers broadly to the assurance that someone or something will act in exactly the way you expect.

Research on this problem in e-commerce has focused on *authentication*—that is, associating a public key with its owner. Reiter and Stubblebine[5] derived principles for developing authentication metrics by studying systems that used trust metrics. However, all their models were based on *transitive trust* along a transaction path of entities that trust the key to different extents. E-commerce, on the other hand, requires *mutual trust* among a vendor, a customer, and all transaction intermediaries. This article introduces a notion of quantifiable trust and then develops models that can use these metrics to verify e-commerce transactions in ways that might be able to satisfy the requirements of mutual trust. The article uses two examples in illustrating these concepts: one for an e-commerce printing enterprise and the other for Internet stock trading; see the sidebar, "Trust Concept Examples," for general descriptions.

## RISK EVALUATION WITH TRUST METRICS

Although no single unit of measure is adequate to the definition of trust, several dependent variables, such as cost, can be used to describe it. These variables in turn influence action protocols that can be used to evaluate the risk involved in an e-commerce transaction.

 1089-7801/00/$10.00 ©2000 IEEE 

## Trust Variables

The overall risk of a transaction is a function of trust variables that fit into several categories.

**Transaction cost.** First, risk is a function of the cost of goods and services: a careful buyer gives more thought to expensive purchases. Similarly, a vendor might not worry about losing revenue on a single microtransaction of negligible cost value, but the risk increases with the cost of a single transaction or the number of microtransactions, and so does the vendor's attention to revenues and expenses.

Consider the Internet Commerce Printing example: a vendor may disregard the revenue loss from a microtransaction such as printing a single page. However, if a customer requests several single-page print jobs or one large job, the potential losses require attention. Such risk is not limited to the quantity of consumables, but extends to their quality and cost. For example, the use of expensive color toner and glossy paper must also enter into the equation.

**Transaction history.** Transaction history is similar to a person's credit history. Just as a bank checks a person's credit history before issuing a loan or increasing a credit limit, a customer's transaction history measures trust and can be consulted to evaluate a potential transaction. In Internet commerce, transaction history could include a customer's profile of transactions with several vendors and a vendor's profile of transactions with several customers. A vendor can disregard occasional transactions in which a customer claims to have received unacceptable goods and then returns them. However, habitual customer returns, especially those involving software, should raise a red flag for vendors.

Vendors also have transaction histories: If customers are dissatisfied with a product and/or the vendor's return policies, they can report their experience to a trusted authority that maintains vendor profiles.

In the Internet Stock Trading example, a broker might accept revenue lost in honoring a customer's one-time complaint and request for compensation over tardy stock quote information. Customers who repeat this complaint, however, might be required to show a nonrepudiated proof of verification, perhaps in the form of a time-stamped receipt of stock information.

**Indemnity.** The trust level of a transaction is increased when a trusted intermediary guarantees against loss. This is especially true for new customers or vendors without transaction histories: they cannot perform expensive transactions unless guaranteed by a trusted intermediary. In the Internet Stock Trading example, a broker can stand as a guarantee against loss. Intermediaries also provide an additional safeguard against uncontrolled, expensive, large-volume transactions.

**Other variables.** Two other variables to be considered in risk evaluation are

- *Spending patterns*. If a customer's host computer were compromised or the customer's smart card or currency were stolen, it might be possi-

## Trust Concept Examples

For consistency, two examples are used to illustrate the concepts of trust metrics and models in the main text.

### Internet Commerce Printing

In this example, several users send a request to print copies of user-supplied documents to a commercial print shop server that distributes both the print requests and user-supplied documents to various printers at different locations. The print request details the number of copies to be printed, whether to print as black and white or as color, the paper size and quality, and the desired print location. For instance, someone planning to give a conference presentation in Washington, D.C., might send a print request from Los Angeles to a commercial print shop, asking that the presentation copies be available at the conference location in Washington on a specified date and time. The print shop makes the arrangements and collects payment by charging the customer's credit card or account upon completing the print task.[1]

### Internet Stock Trading

In this example, several traders buy and sell stocks from a stock exchange through a stockbroker. Each stockbroker collects information on how many shares or bonds to trade and at what time and price from each of the traders. The broker collects a certain fee for the services and attempts to trade on behalf of the traders based on each trader's constraints. The broker may also help in online trade transaction services by providing timely stock price information based on the trader's profile of what shares are owned and how much profit would be made when sold at a particular time.

### Reference

1. R. deBry et al., "Internet Printing Protocol—Model and Semantics, Version 1.0," RFC 2566, Apr. 1999, available at ftp://ftp.isi.edu/in-notes/rfc2566.txt. More information on IPP can be found at http://www.ietf.org/html.charters/ipp-charter.html.

## Trust Metric Terms and Definitions

- *Transacting entity:* Any entity that engages itself in an electronic commerce transaction is a transacting entity. This entity could be a customer, a vendor, a broker, an intelligent agent, a payment server, or any intermediary.
- *Trust authority:* Trust matrices are used to evaluate the trust on a certain transaction or on the next set of transactions. Unless these trust matrices are protected against manipulation and are maintained by certain authorities, transacting entities cannot trust them. These authorities are called trust authorities (TA). Transacting entities use trust protocols to access trust matrices. A TA maintains trust matrices by updating them based on the information received from each completed transaction. TAs should be able to provide proof to trust matrix updates using nonrepudiation services and to provide each of the transacting entities the level of trust index to be placed on a certain transaction.
- *Agreement Framework*[1]*:* A relationship binding all the transacting entities involved in a single set of transactions. The relationship usually includes various policies for conducting transactions and is usually placed at a TA. Each set of transactions is interpreted based on the policy, and the results are used to update trust matrices.

### Reference

1. M. Roscheisen and T. Winograd, "A Communication Agreement Framework of Access/Action Control," *Proc. IEEE Symp. Security and Privacy*, IEEE Computer Society Press, Los Alamitos, Calif., May 1996, pp. 154-163.

ble to detect suspicious activity by observing changes in spending patterns.

- *System usage*. Increasing the number of transactions increases the tax on system resources; for example, more storage is required to check against double spending and to verify authenticity. Similarly, more transactions may mean more CPU time spent on verification. These resource requirements might be an overhead to transaction costs, since vendors and trust intermediaries should provide adequate resources and security against compromise.

**Variable parameters.** Two parameters used in measuring trust can be applied to fine-tuning all these trust variables. The first is *time:* the number of transactions conducted during a certain period of time, in which the transaction frequency could reflect a change of trust state.

The second is *location:* the transactions routed through intermediaries that have perhaps been compromised in some way would likely lower trust.

## Trust Actions

Once variables for quantifying trust are defined, a transaction can be acted upon according to the value of trust so determined. The most common actions are verification—of either the customer's or vendor's credentials—and authorization.

**Verification.** Verification includes running a complete check of a customer's authentication credentials, including ability to purchase goods, payment information, and background. Verification can sometimes expose the identity of a person involved in an anonymous transaction.

Vendors' natural tendency is to verify each customer payment. If the cost of verification is too high, however, they will usually avoid verifying small-cost transactions or microtransactions—although if the verification cost is inexpensive, the tendency to verify increases. With increasing numbers of transactions, the verification cost in terms of computation, storage, and time becomes unmanageable. One solution to this paradox might be to verify randomly selected microtransactions, especially for trustworthy customers.

On the other hand, customers may doubt the credibility of items billed by the vendor as well as the quality of the goods. Customers might find it difficult to test goods, such as software, online before completing the transaction, but they cannot trust the vendor to such an extent that they simply assume that the downloaded software will install successfully and include all the advertised features.

Referring to the Internet Commerce Printing example, customers might want a third party to verify certain facts about an order such as timely delivery and print quality before posting payment.

**Authorization.** When a customer has been authenticated and authorized to buy services or goods, the question remains whether they can be trusted not to misuse them. For example, can they be trusted not to resell information, software components, copyrighted material, and so forth?

One way to curtail misuse is to restrict rights while delegating them.[6] Greater restrictions can be placed on less-trusted consumers. For example, a customer's ability to allow other customers access to an information database, or a vendor's ability to allow access to clients, could be restricted by delegating a subset of rights. Similarly, restricted access rights to a printer could be delegated to prevent such misuse such as exceeding a print quota, printing copyrighted or objectionable material, or

overusing expensive resources.

## RISK ANALYSIS USING TRUST MODELS

Trust variables and actions are the basis for the four different types of trust models presented here. The key underlying terms are explained in the sidebar, "Trust Metric Terms and Definitions."

### Models Based on Boolean Relationships

Two or more trust variables and parameters can be used to describe the level of trust on a particular transaction. These variables should be meaningfully related to each other to provide a semantic definition of the model. The relationship can be captured by a trust matrix, where matrix actions—entities—relate to the row and column labels.

Figure 1 captures the trust relationship between a customer and a vendor. The figure describes a trust matrix with a single matrix action, V, which signifies that a particular transaction should be verified. Actions that need not be verified are grouped into a *trust zone*, the boundary of which zone is a *trust contour*. As Figure 1 shows, for customers with the worst transaction history, every transaction is verified (for authenticity, proof of origin, and so on) regardless of transaction cost. Customers with excellent transaction histories need only have more-expensive transactions verified.

### Models Based on Fuzzy Logic

Linguistic terms such as "microcost transaction" or "excellent transaction history" let transacting entities such as vendors easily describe their measurement units. The actions may also have to be weighted to distinguish the various degrees of measurement. For example, it makes little sense to verify a transaction for someone with a good history to the same extent as for someone who has a poor history, even when making the same high-cost transaction.

Figure 2 shows how weighting affects trust levels for customers with varying transaction histories. For example, a customer with an excellent transaction history has one in 50 transactions verified (V/50), whereas the customer with the worst transaction history has every transaction verified through a variety of methods, including thorough consultations with other vendors, trusted intermediaries, and reviews of previous transactions. This might be represented in the matrix by something like 20V.

Such weighting forms a *weighted trust surface*, with peaks at 20V and valleys at V/70. A weighted trust surface is a three-dimensional surface generated by smoothly interpolating the values in the matrix. The numbers in this trust matrix are difficult to determine, however, and it is unclear how 20V compares to, say, 10V. As Figure 3 shows, we must develop a fuzzy logic trust matrix to deal with such issues.[7] One of fuzzy logic's benefits is that using linguistic terms (such as "normal," "excessive," and "worst") allows for easier interpretation of the matrix entities by trust intermediaries and authorities. These linguistic terms cover a range of values

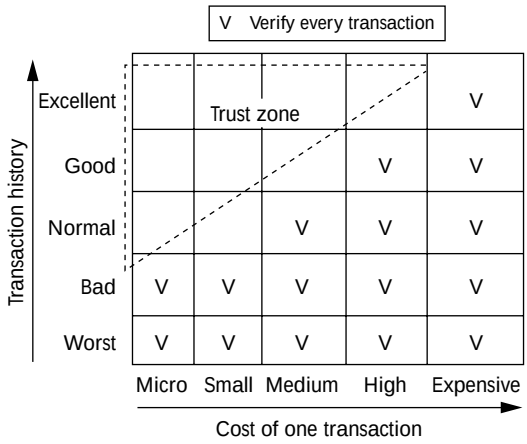


**Figure 1. Trust matrix showing a trust zone. Vendors are highly likely to verify transactions for customers with bad transaction (credit) histories. For customers with good credit histories, the likelihood increases only as transaction costs increase.**

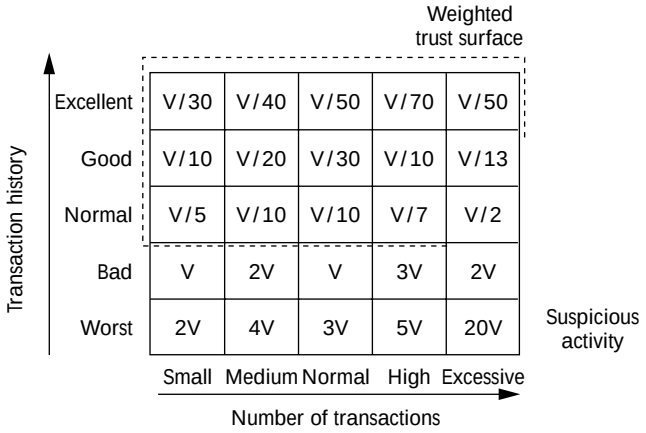


**Figure 2. Weighted transaction verifications. The dotted line encloses the weighted trust surface. Clearly, the higher the number of transactions (microtransactions, in this case) and the worse the customer's transaction history, the more intensive the verification effort will be.**

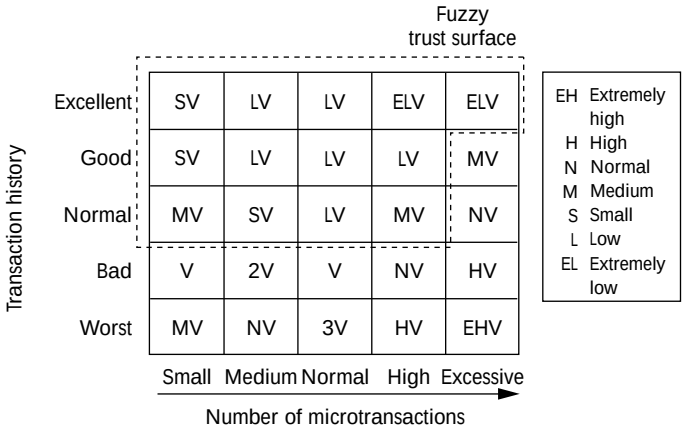


**Figure 3. A fuzzy trust matrix. The trust matrix is "fuzzy" because discrete numbers are replaced by linguistic values that could be either standardized or defined by each intermediary. In this example, a person with an excellent transaction history has low verification (LV) for a "normal" number of transactions.**

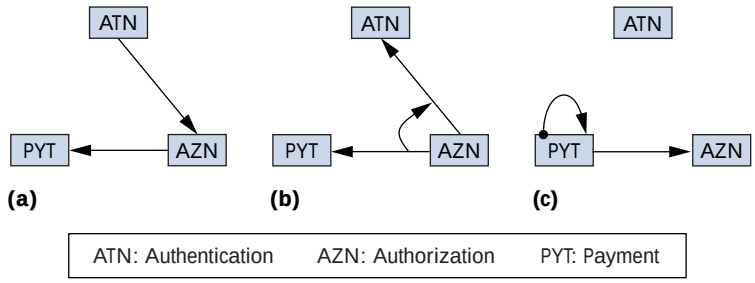


**Figure 4. Three trust models based on transaction processes: (a) the authenticate-first; then authorize, pay, and deliver (AAP) model; (b) the authenticate-if-trust-violated (ATV) model; and (c) the pay-first (PF) model.**

rather than a single "discrete" value, which enables their use in a knowledge processing system, such as a fuzzy logic expert system. A fuzzy logic-based expert system allows these linguistic values to be represented by mathematical functions called membership functions. Thus, the fuzzy trust matrix could be replaced by a set of fuzzy membership functions that could be useful in reasoning.

## Models Based on Transaction Processes

E-commerce transaction protocols in general—and security protocols in particular—follow a handshake procedure before delivering goods of any kind to the customer. This handshake procedure usually follows the authenticate-first, then authorize, pay, and deliver trust model (called the AAP model). The customer initiates this process after determining the trustworthiness of the vendor by examining a trust-authority-signed advertisement from the vendor. The signed trust advertisement would contain a customer satisfaction index, what the product promises to have, and so on. In this model, the customer and vendor mutually authenticate each other; the vendor verifies if the customer is authorized to purchase goods, checking against age restrictions, trust restrictions, and so on. The vendor ultimately accepts the payment and delivers the goods.

The AAP trust model is shown in Figure 4a. In the Internet Commerce Printing example, the customer mutually authenticates to a commercial print server using authentication credentials, which lets the print shop vendor know the customer's identity. The customer then sends authorization credentials along with a print request. If the customer has been preregistered, the print server needs to check its access control list instead of verifying the authorization credentials. Once the customer has been authorized and cleared against quota limits, the vendor accepts payment and processes the print order.

The AAP model does not suit all e-commerce transactions for reasons of efficiency and redundancy. For example, transaction entities within a physical trusted zone such as within a firewall (say, a virtual mall) need not be strongly authenticated with each other. A typical scenario might find a customer entering a virtual bookstore at a virtual mall and printing off the desired books on a printer also located at that virtual mall. Privacy or confidentiality obtained through encryption is not necessary since the actual content is not sent across the Internet. This information could, of course, be encrypted to protect the customer's privacy in buying habits. Otherwise, entities need only be authorized, checked for privileges, billed (paid), and the goods delivered.

When suspicious activity is noticed, the server or vendor should insist on proper authentication. Suspicious activities would include a spurt in transaction activity or an unanticipated request for access to confidential information. This is called the authorize-first, then pay and deliver, or authenticate-if-trust-violated (ATV) model. This model also applies to noncommerce-related network communication in which authorization but not pay-

ment is required. Figure 4b shows this model.

A third trust model is the pay-first (PF) model, shown in Figure 4c, which is useful for customers interested in anonymity or new customers who have no trust relationship. Anonymous customers who want to remain that way prefer to pay using electronic currency (for example, Digicash) to pay before receiving goods. Similarly, customers without a trust relationship would fit the PF model by making payments to the vendors before entering into a trust relationship.

The PF model becomes the ATV or AAP model when trust is established for nonanonymous customers. Payment is authorized, and a customer is registered into the trust relationship with the identity they provide.

### Model Based on a Transaction Automaton

A transaction automaton models transaction behavior in the form of state transitions. Thus, this trust model describes e-commerce trust on the basis of the transaction's state: fail, success, in-progress, attack. A transaction is successfully completed when the trust authority (TA) receives a complete acknowledgment from all entities involved in the transaction. Acknowledgment does not, however, guarantee that the current transaction set will not change to, say, an attack state. The trusted intermediary (TI) or the TA can later determine that the current transaction is under attack and can change the state of all related transactions from success to attack state.

A transaction is in-progress if the TA has not received a complete acknowledgment from any of the transacting entities. The in-progress state can also occur if the customer-vendor transaction completed, but the TA hasn't yet received a complete message from at least one of the transacting entities.

A transaction fails if the TA has not received a complete message during the time allocated for the transaction, as defined in the agreement framework or contract to complete. The transaction also fails when the TA receives a complaint or suspicion acknowledgment from a transacting entity. If the failure results from a time-out, the TA or TI sends out probe signals to each transacting entity to try to reinitiate the transaction. If the failure results from a complaint or suspicion, however, the transaction state changes from in-progress (or failure) to attack.

When the transaction state is attack, the transaction entities switch from the current mode (buy, sell, and so on) to preventive mode, which blocks the attacker from causing further damage. Preventive mode suspends the transacting entities so that they cannot process other transactions in the transaction set. The transaction can simultaneously spawn other modes to recover from transaction losses if possible and to perform corrections that would enable the victims to resume from the suspended state.

## COMMERCE-RELATED ATTACKS

By applying trust models to examine e-commerce-related attacks, we can better understand how to detect, prevent, correct—and recover from—them.

**The pay-first model is useful for customers interested in anonymity or who have no trust relationship.**

### Stolen Token

Known cryptographic methods can prevent password sniffing over a network, thus preventing the theft of passwords or identity as they travel across the wire. But such security provides little benefit in the case of a stolen password or token such as a smart card. The thief can use the stolen password or token to impersonate the genuine legitimate customer and make transactions over the Internet. The impersonating customer can pay for goods, but these payments are not genuine since they were stolen. Hence, in-time payments may not be proof enough to authenticate the customer for this commerce-related attack; the best that could be done is to limit losses from such transactions.

Trust models could be used to understand such attacks and reduce the associated transaction loss or risk, although none of the trust models described earlier can prevent or detect this attack. Instead, the following methods could be employed:

- *Detection by analyzing spending pattern*. Impersonators typically make as many purchases as possible before the theft is detected and reported. For a not-so-frequent user, detecting a stolen or lost token is not fast enough to prevent the impersonator from making many transactions. By observing these purchase patterns, the trusted intermediaries or authorities

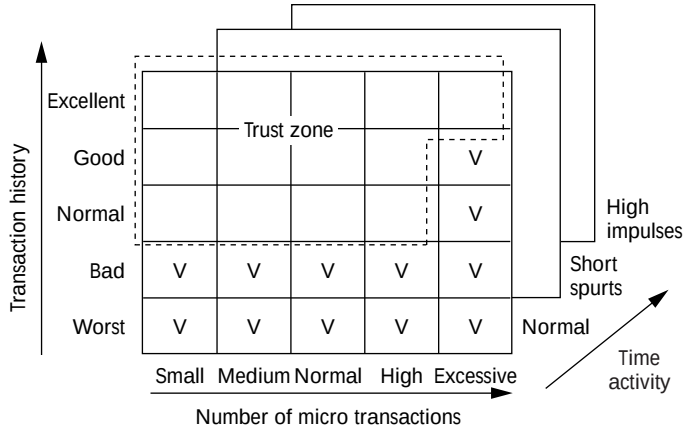


**Figure 5. A Boolean trust model enhanced to include time activity as a trust parameter.**

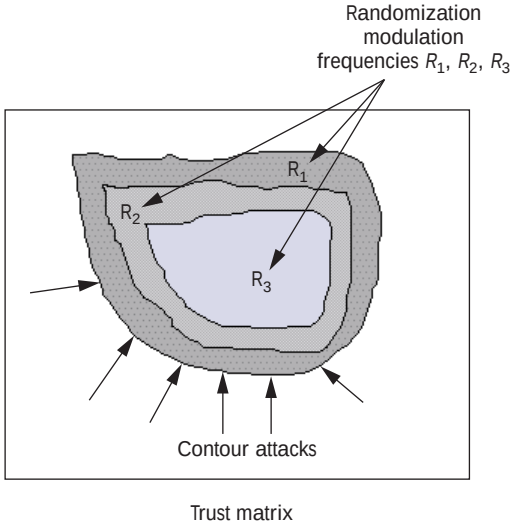


**Figure 6. Contour attacks and random modulation on a trust matrix. It is hard for an attacker to determine the contour, since the boundaries of trust regions keep changing randomly at different frequencies (R1, R2, R3 in this case).**

might detect such fraud before the genuine customer realizes the theft and reports it. For the trust authorities to detect such activities, the basic Boolean Trust Model described earlier can be enhanced by adding a trust parameter: time.

- *Prevention by timer-delay key recovery*. After detection of a suspicious spending pattern, delaying the delivery of a decrypting key to the impersonating customer can prevent further loss. This key is usually delivered after the customer pays the vendor,[8] but this step can be delayed until the customer (here, the impersonator) provides more secret or sensitive information—such as biometrics—not contained in the stolen smart card or computer. This step is called reauthentication under suspicious circumstances. If the customer cannot provide proper authentication credentials, the previous transaction set is treated as having been hijacked. The trust process model switches from AAP to the ATV model, and the transaction switches from attack mode to preventive.
- *Correction and transaction recovery*. Once identity is reestablished, the trust authority issues the customer a new token. Losses to the customer and vendor are covered under an agreement framework that is similar to an insurance policy, though information or services confiscated by the impersonator cannot be easily recovered.

Figure 5 shows that mutual trust results when a large number of transactions are conducted over time, with a corresponding increase in customer loyalty to a vendor. However, an excessive number of transactions occurring in a short time should trigger suspicion, with every ensuing transaction verified regardless of the customer's transaction history.

## Contour Discovery

A key weakness of the fuzzy trust models is that they are prone to contour discovery attacks. Contour discovery means that the boundary of a trust zone has been discovered by an impersonator, for example. An impersonating customer might penetrate the trust zone by watching several transactions between a trusted customer and the transacting vendor (thus exposing the person's privacy). The impersonator might use network monitoring tools and hacker programs to watch these transactions as they pass across the network. The impersonator later uses this information so that none of the transactions come under strict verification, thus cheating the vendor by making several transactions within the discovered trust boundary.

The impersonator need know only the price, the customer's name, and the transaction's success or failure. This information is available outside the secured (confidential) portion of the transaction. Combined with a stolen token, the impersonator could pretend to be honest by applying techniques such as indemnity, prepayment, or overpayment.

Once a malicious transaction was discovered, the vendor would have no choice but to verify every transaction from the impersonated customer for a certain predetermined trust-regain time, thus

increasing the processing time for every transaction during the period.

Other security measures like creating new encryption keys for the trust base (set of trust matrices) prevent further attacks by the same impersonator.

One technique for preventing contour discovery attacks is *random perturbation*, using an audited trust zone; that is, we can randomly verify transactions within the trust zone instead of committing the transaction without performing adequate security checks for authentication, authorization, or trust. A random modulated fuzzy set (shown in Figure 6) models such a preventive measure.

## PROPAGATION OF TRUST

Electronic commerce generally requires a customer to interact with several trusted intermediaries before actually contacting the vendor. Some of these intermediaries may not have had a trusted relationship among themselves, in which case a default relation is set between them. Otherwise, the existing relations are invoked to participate in the commerce exchange.

Figure 7 is an interaction diagram (as derived from Ketchpel[9]). It establishes trust relationships via matrices between entities. Thus, there is a trust relationship between the customer and the intermediary, and a trust relationship between the trusted intermediary (TI) and the broker. The number of these trust relationships increases with the number of trusted intermediaries between customer and vendor.

Differing trust relationships exist among the customers, intermediaries, and vendors. To calculate a single trust value between the customer and vendor requires forming an overall trust relationship that governs the transactions between them. The customer-vendor trust relationships (called the chain of trust relationships) must be reduced to a single overall trust relationship that accounts for the trust values in each and every relationship. This process of reducing the chain of trust relationships into a single value using merge operators is shown in Figure 8.

Figure 9a shows a trust matrix that exists between the customer and the trusted intermedi-

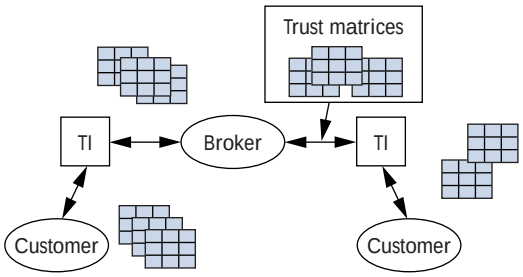


**Figure 7. An overall trust relationship. The relationship is shown here as a set of trust matrices between two entities: a customer and a trusted intermediary (TI), and a TI and a broker.**

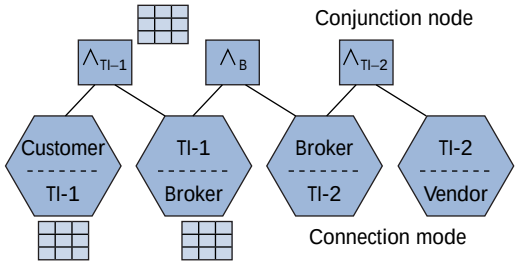


**Figure 8. Sequence diagram created from interaction diagram. The trust relationships are reduced at the site of the merge operators (^).**

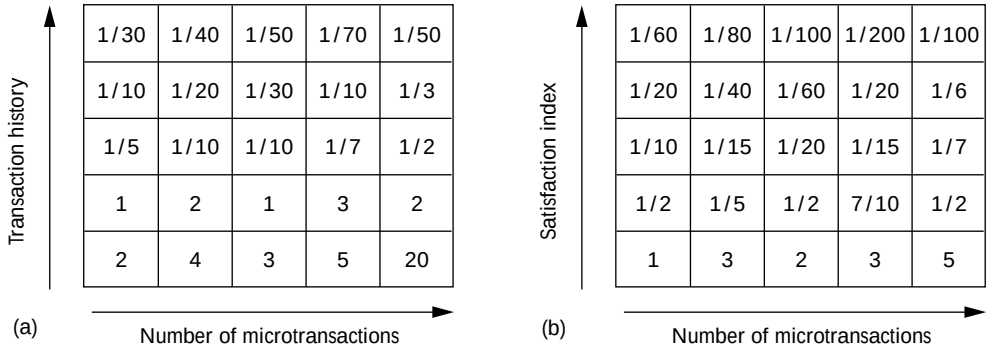


**Figure 9. Related trust matrices in two relationships: (a) trust relationship between customer and trusted intermediary; (b) trust relationship between trusted intermediary and broker. Each of these values represents a level of verification.**


**Figure 10. Trust matrix formed by merging the two trust matrices in Figure 9.**

ary. The variables used in the matrix are transaction history and number of microtransactions. Figure 9b shows a similar trust matrix between the TI and the broker.

The variables in the left-hand matrix are transaction history and number of microtransactions. Variables in the right-hand matrix are the number of microtransactions and a satisfaction index. Satisfaction index could indicate the average transaction history of the customers in a category, or a quantity that the trusted intermediary determines based on the satisfaction report it has from its customers. A merge operator in this case could be one that uses the matrix on the right-hand side to lessen the verification risk used by the matrix on the left-hand side. It could be a subtraction, a min operator on the two quantities, or an averaging operator, or some combination of operators. Figure 10 shows the resulting matrix with some of the entities filled. A variety of operations can be performed on different rows of the trust matrix shown in Figure 10 using several complex operators.

## CONCLUSION

Trust models based on the credit-card system measure trust as a function of credit history. These models do not support a theoretic approach to trust, nor do they address issues such as online payment verification. In general, they are not suitable for electronic commerce. The alternative approach presented here offers a way to verify transactions, while avoiding the unnecessary computation costs of verifying every transaction.

Possibilities for future work along these lines include the study of attacks made via anonymous operation and the study of corrective and preventive methods for recovery and survival. ■


## ACKNOWLEDGMENTS

The author thanks Daniel Greene and Mark Stefik of Xerox Palo Alto Research Center for comments and suggestions that helped improve this article.


### REFERENCES


1. A.O. Freier, P. Karlton, and P. Kocher, *The SSL Protocol, Version 3.0*, Netscape specification, Nov. 1996; available at http://www.netscape.com/eng/ssl3/ssl-toc.html.
2. T. Dierks and C. Allen, "The TLS Protocol Version 1.0," Internet Engineering Task Force RFC 2246 (standards track), Jan. 1999; available at ftp://ftp.isi.edu/in-notes/rfc2246.txt.
3. Secure Electronic Transaction Specification, *Book 3: Formal Protocol Definition*, Version 1.0, May 1997; available at http://www.setco.org/download/set_bk3.pdf; general information on SET is available at http://www.mastercard.com/set.
4. D. Eastlake and T. Goldstein, "ECML, v1: Field Names for E-Commerce," IETF RFC 2706 (informational), Oct. 1999; available at http://www.ietf.org/rfc/rfc2706.txt; general information on ECML specifications is available at http://www.ecml.org/spec.html.
5. M.K. Reiter and S.G. Stubblebine, "Toward Acceptable Metrics of Authentication," *Proc. IEEE Symp. Security and Privacy*, IEEE Computer Soc. Press, Los Alamitos, Calif., May 1997, pp. 10-20.
6. B.C. Neumann, "Proxy-Based Authorization and Accounting for Distributed Systems," *Proc. 13th Int'l Conf. Distributed Computing Systems (ICDCS 93)*, IEEE Computer Soc. Press, Los Alamitos, Calif., May 1993, pp. 283-291.
7. L. Zadeh, "Fuzzy Sets as a Basis for a Theory of Possibility," *Fuzzy Sets and Systems*, Vol. 1, 1978, pp. 3-28.
8. J. Su and D. Manchala, "Building Trust for Distributed Commerce Transactions," *Proc. 17th Int'l Conf. Distributed Computing Systems (ICDCS 97)*, IEEE Computer Soc. Press, Los Alamitos, Calif, May 1997, pp. 322-329.
9. S. Ketchpel and H.M. Garcia, "Making Trust Explicit in Distributed Commerce Transactions," *Proc. 16th Int'l Conf. Distributed Computing Systems (ICDCS 96)*, IEEE Computer Soc. Press, Los Alamitos, Calif., May 1996, pp. 270-281.



Daniel Manchala is a senior member of the research and technology staff at Xerox Research and Technology, Xerox Corp. His research interests include electronic commerce systems, security, and software architectures. He received a PhD in computer science from Texas A&M University. He is a member of the IEEE and the ACM.

Readers may contact Manchala at Xerox Corporate Research and Technology, 701 S. Aviation Blvd., ESAE-231, El Segundo, CA 90245; manchala@cp10.es.xerox.com.