V-ONE™

Security for a Connected World™

Electronic Commerce and Security

Marcus J. Ranum
Chief Scientist
V-ONE Corporation

Introduction

The promise of Electronic Commerce is one of the major factors that is contributing to the rapid growth of the Internet as a communications medium. As with any commercial activity, it is important to consider its security implications.

In the last two years we have seen a great deal of press coverage devoted to Internet security, and many of the businesses adopting electronic commerce are confused and worried by the threat of losing money through electronic crime or credit card fraud. In this paper, we'll examine some of the security issues in Internet-based electronic commerce.

What is Electronic Commerce?

The term "electronic commerce" has been used to describe all steps of the commercial process that are managed via computer. We prefer to use the term in a more limited scope, specifically referring to computerization of the selling process. In other words, advertising is not electronic commerce per se, though clearly it is important to commercial success. When doing business over the Internet first became possible, it was mainly restricted to electronic advertising and marketing, with users browsing on-line catalogs and purchasing goods and services via credit cards over the phone. Transfer of physically purchased goods is still handled via the U.S. mail or express mail services, since electronic transfer of matter is still exclusively the domain of science fiction writers.

For the purposes of this paper, we will consider electronic commerce as *the process of arranging transfer of goods or services, including arranging or performing payment and exchanging customer information*. If you imagine this in terms of telephone-based mail order, the Internet electronic commerce role replaces the transactions that occur between the point at which the phone service agent answers the phone, and the phone service agent schedules the customer's product shipment and hangs up. During that time, the customer places an order including the desired items, their quantity, and a credit card or account number and shipping address. Internet electronic commerce attempts to automate this process wherever possible.

From a security perspective, there are several important things to take into account during the customer transaction process, which apply to "real life" or telephone commerce as well as to electronic commerce:

How do customers know they are dealing with a legitimate business? In "real life" a major store is difficult and expensive to fake. On the Internet it is not. Long-established businesses and their name recognition factors have a powerful market clout that newcomers do not. Does the electronic commerce revolution threaten this? In a sense it may no longer be a question of "how big you are" it may now be a matter of "how big you look".

How does the business know it is dealing with a legitimate customer? In some transactions the merchant does not need or wish to know the identity of the customer. In the current market, customers who wish to remain anonymous can use cash or money orders instead of credit cards, and the merchant is protected by the relative difficulty of forging cash. When a customer wishes to pay by credit card, the approval process tries to verify the customer's identity by checking that the card is active, not overdrawn, that the holder knows the expiration date, and often that the shipping address matches the billing address.

How does the customer arrange payment? In "real life" commerce there are a number of options for payment. Electronic commerce almost always assumes some kind of electronic identity (usually a credit card) that is exchanged as a promise to pay. Electronic cash technologies exist, but are less popular than credit card based systems, and are a concern to governments that fear anonymous transactions may make money laundering easy.

How does the customer specify or change the address where goods are to be shipped? The shipping address for goods is often used to reduce credit card fraud by cross-referencing it with the credit card billing address. Electronic commerce systems that make it easy to change billing or shipping addresses may be vulnerable to attack by redirecting goods or invoices.

What aspects of the transaction does the customer expect or the law require to be private? Many aspects of a transaction a customer may not wish disclosed. Home addresses and telephone numbers, for example, may be protected for the customer. Law may protect other transactions such as medical record lookups or bank balances, and an electronic merchant may liable for damages in the event of disclosure.

What are the indemnifying factors that protect the merchant and the customer? Many Internet-based electronic commerce applications rely on credit cards for payment. As a result, the regulations limiting damages from credit card abuse may apply. It is unlikely that electronic commerce will enjoy wide market acceptance unless the extent of end-user and vendor liability is well understood by both parties.

The State of Internet Electronic Commerce Today

When the Web exploded into acceptance as a new medium for marketing and commerce, it was initially used primarily as a forum for electronic marketing, with early adopters beginning to accept credit cards as payment. Many of the early adopters ignored security concerns about credit card information being intercepted, and gained useful experience and market leadership over their competitors. A number of highly publicized security incidents raised public awareness of the potential for credit card fraud, and many vendors began examining options for providing security.

Netscape Communications, Inc.'s SSL (Secure Socket Layer) is the most widely deployed Internet transaction protection technology. SSL is supported by Netscape's browser, which has hastened its market penetration. A competing protocol is Enterprise Integration Technologies, Inc.'s SHTTP (Secure HTTP) protocol. The SSL protocol recently received bad press relative to a security flaws that have been widely published. One flaw was a basic mistake in how public keys were generated for session encryption. This flaw has been fixed in subsequent releases, but many security experts have been concerned that such an elementary error was made in such an important part of the system. Another flaw in SSL that received a great deal of attention was the cracking of an encrypted transaction by a French student who used a number of high performance workstations to crack a key for the encryption algorithm used to protect transactions.
SSL and SHTTP both provide basic encryption of session contents, to prevent an eavesdropper from being able to intercept credit card or other personal information. SSL and SHTTP use a digital signature scheme to authenticate both parties.

Internet Electronic Commerce and Prior Art

To a great extent, Internet-based electronic commerce solutions have largely ignored prior art embodied in other electronic payment systems. This is because many EDI (electronic document interchange) applications assume that the parties transmitting data are already well known to each other, or that the data is being transmitted over a secure channel or network. The assumption that both parties in a transaction already know each other does not work on the Internet, where there are millions of users, with identities being added, changed, and deleted all the time. It's also quite obviously unwise to assume that the Internet is a secure channel that is tamper-proof. Any prior art in existing EDI systems is also likely to be ignored in the rush to get new software out of the starting gate, and to stake out new technological territory.

It is unfortunate, since we are doubtless going to see many past mistakes repeated because few software developers are taking the time to adequately research problems before bringing products to market. The Netscape public key generation bug is a good example: an elementary mistake was made that anyone with expertise in cryptography would have avoided. Staffing and time-to-market considerations prevented Netscape from researching the problem adequately, or having their approach reviewed by experts.

Security Relationships and Communications Channels

When considering electronic commerce applications, it's important to identify what parts of the transaction need to be protected, and how they are protected. Undue attention paid to one part of the complete system will not result in improved overall security, but rather a false sense of security. Consider as an example the manner in which large amounts of cash are transferred: every step of the process of moving money between vault, street-curb, armored car, street-curb at destination, and vault at destination has appropriate security. The security practices employed in transferring large amounts of cash are a direct result of years of refinement spurred by innovative criminal action. When we look at electronic commerce systems we must assume that a similar process of refinement will take place. During the early 18th century, explosives lacked the shattering power and compact size required permitting small safes to be blown open without destroying the contents. As a result, great emphasis was placed on multiple complex locks with carefully guarded keys. Not surprisingly, criminal activity centered on theft of keys. *Criminal activity will usually be directed against the most effort/cost effective target.* One implication is that "insider jobs" will continue to be a significant threat to electronic commerce. Eventually, the most cost-effective form of attack will be getting a job with the victim. We need to maintain our perspective and build computer security into electronic commerce processes *systematically*, not simply into the obvious and attractive points of attack.

Data Pipes

For Internet-based electronic commerce, the data pipe—the TCP/IP connection between the client's computer and the commercial server—is currently the main focus of our attention. A variety of encryption and authentication schemes are being used to protect the data in transit from being altered or monitored. This is important, yet it is only a small part of the complete security spectrum we need to address in order to build robust electronic commerce systems. For all intents and purposes, a data stream that is protected with reasonably high-quality encryption (56 bits of key size or better) is safe against attack. There is a potential for cryptographic attacks on the data stream but an attacker who is not a cryptographer will be effectively barred from playing. If the financial benefit to attacking the cryptography used is sufficient, it may justify mounting an attack. European pay-per-view satellite systems have had their cryptography compromised several times, at a cost of millions of dollars in lost

revenues. The attackers who compromised the cryptography have more than recouped their investment in effort by selling falsified access cards.

Internet-based electronic commerce systems presently are placing a great deal of emphasis on cryptographic protections that are being applied to the data pipes. This emphasis is a direct result of the attack cost-effectiveness of TCP/IP sniffing. TCP/IP sniffing, in which an attacker passively monitors traffic on a network, is a known problem, and a relatively easy attack to defeat. We need to be concerned that in our rush to protect against sniffing that we don't ignore the other points of attack and potentially more serious threats.

End Points

When a commerce application uses an encrypted data pipe to communicate with the server on the other end, what do we know about the security of the end point systems? In this case, there are two end point systems: one is the server at the merchant's electronic place of business, the other is on the user's desktop. Both end points are vulnerable to attack if not secured adequately. In some cases the merchant's server is behind a firewall or some form of network protection. In others it is sitting directly on the Internet on a (hopefully) "hardened" UNIX system. What happens to the data once it gets across the secure pipe to the server? Often it is simply stored in a file or database, which may contain customer credit card information, telephone numbers, home addresses, etc. Usually, the data gathered is stored unencrypted, "in the clear" and is a very attractive target for an attacker. We have already seen many cases where Web servers have been attacked and broken into. Usually, this has been a harmless nuisance, but eventually a Web server with customer credit card information will be compromised. Most likely, this has already happened several times but nobody has noticed or notified their customers.

The second end point in electronic commerce is the user's desktop system. Users on multi-user systems have a particularly serious problem since it is not difficult to exploit security holes in most commercial operating systems, to access other users' data. This may include information such as users' keystrokes as they type them, passwords, credit card information, etc. If an attacker can read the data out of a user's running program before it is placed into an encrypted data pipe, there is no need to attack the encryption at all. Which is easier: breaking DES and RSA encryption, or breaking into UNIX? PCs or other computers are no more secure than UNIX systems; Trojan horses or attack programs are potentially capable of accessing users' information from their hard disks without anyone being the wiser. As Web browser meta-language technologies such as Java, and Visual Basic grow more powerful and prevalent, the likelihood of someone developing a credit-card number stealing browser applet increases.

To protect the end points of electronic commerce we will need tools that protect not only the data pipes but the data at the ends of the pipes. This means that security will have to be taken into account at every step of the design process, rather than added as an afterthought once the application is in beta-test. Applications will have to be "smarter" about their default operations, and will need to be designed to prompt users when dangerous operations are attempted. Smart card technologies, which are capable of protecting data even from untrustworthy applications, will become more important, as they allow users to "unplug" private information from the system and carry it away with them.

Applications

Electronic commerce promises to herald the dawn of desktop banking. What will happen when everyone's PC is a personal automatic teller machine? Clearly, some kind of authorization needs to be applied so that the PC software "knows" that the correct person is using it. The danger is that early versions of desktop banking might rely only on a password (or worse: no password at all) to authorize access to an individual's account. Presumably, the situation will improve once a few highly publicized

incidents serve to illustrate the risks. Imagine what would happen if an annoyed spouse or child decided to sell someone's stock portfolio short, or to stop payment on a mortgage check. To build electronic commerce systems, we will need to walk a fine line between making the system *easy to use* and making the system *easy to fool*. Accomplishing this goal will require careful and consistent thinking about security and privacy issues.

Cryptosystems

One of the principal protections available for electronic commerce is encryption. Presently, a great deal of attention is being paid to various algorithms, key exchange mechanisms, and certification technologies. Cryptography is an important tool but is not, by itself, a solution. *Simply adding encryption is not sufficient to make an application "secure".* The use of encryption in commercial systems has to be correct and relevant to what needs protection.

Algorithms

Encryption algorithms are hard to design; the history of cryptography is filled with "unbreakable" systems that turned out to have fatal flaws. For electronic commerce, most systems rely on the U.S. Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA), or RSA Data Security Inc.'s RC4 algorithm.

DES typically uses 56 bits of key space, IDEA 128 bits, and RC4 40 bits. The number of bits used for keys is important since it indicates the level of effort required performing a brute-force search for the correct key. Recently, a college student with a large number of workstations cracked a 40-bit key used by Netscape to encrypt data with RC4. Within two years, it is safe to assume that about $10,000 worth of hardware will provide adequate processing power to crack 40 bits worth of keys. Keys with a length of 56 bits will, for some time to come, be outside of the reach of all but national government agencies and is, for practical purposes, immune to brute-force search. The time required to brute-force search a cryptosystem does not necessarily mean that the cryptosystem is unbreakable: security flaws in cryptosystems might permit an attacker to reverse-engineer a key in significantly less time than a brute-force search. Most of the cryptanalysts skilled enough to crack codes are currently employed by national government agencies and probably do not represent a threat to electronic commerce applications. Because of the risk of flaws in cryptography, most commercial applications use one of the well-known cryptosystems, rather than "home brew" systems.

Key Management: Public and Secret Key

The actual encryption algorithm used is only part of the picture when using encryption systems. Key exchange, the process of safely getting a shared encryption key to both parties, is one of the weaknesses all encryption systems share. Secret key exchange relies on some secure means of pre-exchanging a key for future use. The pre-exchange of the key has to be done securely, and is usually done out-of-band or face-to-face. A good example of a simple secret key exchange is the way in which a bank issue personal identification numbers (PINs) for automatic teller cards. The customer appears in person at the bank and is issued the card and PIN, or the card and PIN is sent via the (*presumably* secure) U.S. mail.

Another way in which secret key can be used effectively is by encapsulating the secret key in a smart card or similar device, where the issuing authority never gives the customer access to the key information at all. The main disadvantage of secret key technology is that each key must be managed securely, and each user has a unique key-pair that is used to authenticate and encrypt traffic to the central authority. As a result of the unique key-pair requirement, it is impossible to engage in commerce

with someone to whom you have not already been “introduced” to. Neither of you has a shared secret key and there’s no secure channel over which to exchange one. Secret key works best when a single issuing authority is maintaining a service for a customer base where there is some kind of registration process that takes place prior to becoming a customer. Possibly the biggest advantage of secret key approaches is that the registration and unique per-user key means a merchant can be fairly certain as to the identity of the customer.

The primary tool of public key encryption is the RSA encryption algorithm. RSA takes advantage of some clever mathematics to allow users to split a key into two parts, one of which is kept secret, the other of which is safely published. Anyone, using the published part of a user’s key, may send encrypted messages to them without having to have access to a pre-arranged shared secret. This is important since it means that parties that have not been “introduced” or pre-registered can carry on a transaction without a third party being able to eavesdrop.

For performance reasons, RSA is usually used only to exchange keys, and conventional (DES or whatever) encryption is used for the bulk of the message. There is still an open issue of reliably knowing who is on the other end of the transaction. For most of the electronic commerce systems in operation today, the user’s credit card and the credit card verification process is used to authenticate them as an authorized consumer. In a sense, with current electronic commerce systems, the basis of trust is still in the validity of the credit card. The main disadvantages of public key technology are the lack of standards for exchanging public keys, the lack of means for verifying the identities of key holders, and the performance of the algorithms: public key encryption is many times more compute intensive than traditional encryption. At present, the main role that public key encryption plays in electronic commerce is as a technique for setting up encrypted data pipes between parties which have never been introduced. In an environment like the Internet, where the population is growing constantly, this is an extremely important capability.

Certification and Registration

In public key encryption, one party can “introduce” another by using their public key to “sign” the new public key. A key, with other public key signatures attached, is often referred to as a *public key certificate*. Usually a certification authority, such as a merchant, ISP, bank, or other trusted source countersigns a certificate. The role of a public key certificate is not unlike a credit card: it is a portable object with some minimal information about the user, which is issued by some organization that can be called upon to verify its authenticity. Credit card authorization systems are a means of verifying the validity of a given card when a merchant wants to perform a purchase. Just as credit cards can be stolen, public key certificates need to be protected, since someone who gains illicit access to the secret part of the key can subsequently impersonate the user. Public key certification systems share many common antecedents with credit card systems. Just as credit card authorization databases may be able to mark a stolen card as “bad: seize on sight” a public key certificate system must address certificate revocation and must permit the issuer to assign a limited lifetime to a certificate, if desired.

Microsoft, Inc.’s recent announcement of an electronic commerce initiative with Visa includes a public key certification hierarchy as part of its architecture. The details have not yet been sorted out, but presumably issuing banks will create what amount to electronic credit cards. When this happens, we need to remember that just as with real credit cards, there will be a danger of having them stolen. The secret part of an RSA public key is usually 1024 bits or larger: 128 characters—more than most people can remember or care to type. Therefore, it must be stored someplace safe where nobody can access it. Current systems such as Pretty Good Privacy (PGP), an Email encryption program, and store the secret data encrypted on a user’s hard disk. If it is not carefully guarded, it makes a good target for attack.

Smart card technologies, some of which incorporate the ability to do RSA signatures and storage for keys, are an attractive technology for protecting this important information in a portable manner.

Relevant Technologies

Integrating secure electronic commerce solutions requires a consistent and unified view of the problem domain. Web servers, clients, firewalls, and encryption/authentication systems must all work together to ensure that there are no weak links. Security must address the complete path between the user and the end point processing system, protecting private data as it is transmitted and stored.

Web Servers

Many sites have difficulty deciding where to put their Web server. For organizations with a security perimeter protected by a firewall, there's usually a strong desire to put it behind the firewall to shield it from attack. Many firewalls, however, do not support "outside" access to the Web server behind the firewall, without significantly weakening the firewall. There is also often a concern that attackers, which would effectively put the attackers behind the firewall, where they could easily attack the rest of the network, may compromise the Web server. Putting the Web server in front of the firewall leaves the server on its own to defend against attack, and increases the difficulty of managing the pages on it by isolating it from the protected network. For commerce activities, the position of the Web server vis-a-vis the firewall is especially crucial if the Web server stores customer information such as credit card numbers. If at all possible, sensitive information should be quickly handed off the Web server and through the firewall to a safe place. If a Web server cannot resist attack, can the contents of the transactions it passes through the firewall be trusted?

Many commercial sites are running their Web servers on UNIX systems, using public domain or off-the-shelf HTTP servers. There are several common points of attack against Web servers: HTTP servers, CGI scripts, and host platforms. Several security flaws have been identified in various versions of HTTP servers. Some of these flaws would permit an attacker to execute arbitrary commands on the Web server itself, in some cases permitting the attacker to log in directly with system administrator privileges. As CGI scripts become more powerful, they also represent a potential avenue for attack. In one well-known incident, a flaw in a perl-based CGI script permitted an attacker to execute commands on the Web server from across the Internet. The entire Web site was quickly compromised through a single weakness.

Firewalls

One technique for protecting Web servers and customers accessing the Web is to place them behind firewalls. For the sake of this discussion, we will not worry about the details of a particular firewall technology: *a firewall is a generalized access control system between two networks*. With a firewall in place, the Web server may be protected, but often at a cost in performance or complexity. Access from the user's perspective is often more complicated as well. New services are always appearing on the Internet and making them work through a firewall are a time-consuming headache. It is an unfortunate fact that most electronic commerce protocols are designed with the implicit assumption of direct point-to-point connectivity between end points. As more sites are connecting to the Internet behind firewalls, electronic commerce applications will have to become firewall-aware, and firewalls, in turn, will have to support newer and different services.

Web Clients

Web clients are becoming increasingly powerful and are beginning to incorporate powerful multimedia display systems and programming languages. The danger of someone developing attack code hidden in HTML documents increases as HTML extension languages give the author more control over the client system. The Microsoft Word “concept” virus is an example of the ease with which macro languages can be used to launch powerful and invisible attacks. Many security experts are justifiably concerned that unknown persons or agents might be able to execute electronic commerce transactions on behalf of the user, by reprogramming their Web clients. It makes sense to eventually separate the user-interface and display aspects of the browser from the payment system. We are facing a period of market and customer confusion as multiple competing electronic payment systems are introduced. Compatibility and interoperation will be a problem if users have to swap browsers in order to make purchases.

Current browsers have been fairly free of security flaws. An early version of one UNIX-based Web client permitted commands to be executed on behalf of the user when certain URLs were selected. Later versions of the Netscape browser had a memory overrun that potentially could have been used to create a Web page that would be able to run commands on the user’s system, if they were using the right (or in this case, wrong) version of the browser. S oftware flaws are to be expected, as browsers continue to become more complex and powerful.

The Future

It is inevitable that electronic commerce will play a role in our futures. Pay-per-view television, Internet commerce, and digital cash/credit card convergence are already being developed. Will there be security problems? Inevitably. Applications must incorporate security as a basic part of their design, rather than an afterthought.

Future Trends

The current trend seems to be toward massive integration of functionality into monolithic programs that do everything imaginable. If the trend in browsers continues, in a few years users will expect to be able to perform bank transactions, edit files, send Email, and collaborate on writing using a common interface. The competition to provide those interfaces will be intense, and with such a large market it is likely that vendors will attempt to fragment the market along functionality lines, in order to dominate specific segments. Strategic alliances and partnerships will continue to be extremely important factors in determining what is available to the customer. For a long time to come, we will be faced with a lot of systems that provide electronic commerce capabilities, but which do not overlap; much as today’s automatic teller machines and credit card systems took years to converge. Key management and certification/registration will become increasingly important and may push the market towards wider-scale deployment of smartcards, as it already has in Europe.

Future Problems

The security problems of the future will be the same as those of the present: management of complexity. Software systems suffer security problems because they are complex, large, and difficult to program; a single flaw can give an attacker a foothold into an otherwise very strong system. Good design and a solid security foundation can provide multiple levels of protection, and reduce the risk of a system being completely compromised. Future electronic commerce systems will require correct interoperation among end points, browsers, servers, firewalls, and other network devices that haven't been invented yet. As

the number of cooperating agents increases, the chances that everything will perform correctly decreases.

Today, many of the security holes in software result from the race to bring software to market with new, more desirable features. There is no reason to expect or hope that the feature-race will ever end, and every reason to expect new features will bring new security holes. Probably the most fruitful area for attackers will be in meta-languages and embedded interpreters. It is scary to ponder that we have not even come close to solving the problem of computer viruses, yet we are rapidly deploying increasing numbers of meta-programmed applications, each with their own interpreters and protection models.

Future Technologies

What security technologies will we be able to deploy to secure the electronic commerce applications of the future? The most important problem that will be solved will be key management and registration. Smart cards, which can contain multiple public keys safely and portably will provide a tamper-proof platform for authenticating purchases. Many visionaries believe that eventually everyone will carry personal electronic agents or digital assistants. These agents will become the “portable automatic teller machines” of the future, and, eventually, we may see some form of “convergence” in which credit cards, smart cards, cellular telephones, and Internet terminals are rolled into a single portable, tamper-proof unit. Until such a time as the market unifies, competitive pressures will effectively act to prevent convergence. The ancient Chinese curse, “may you live in interesting times”, will always be true for the cutting edge of electronic commerce.