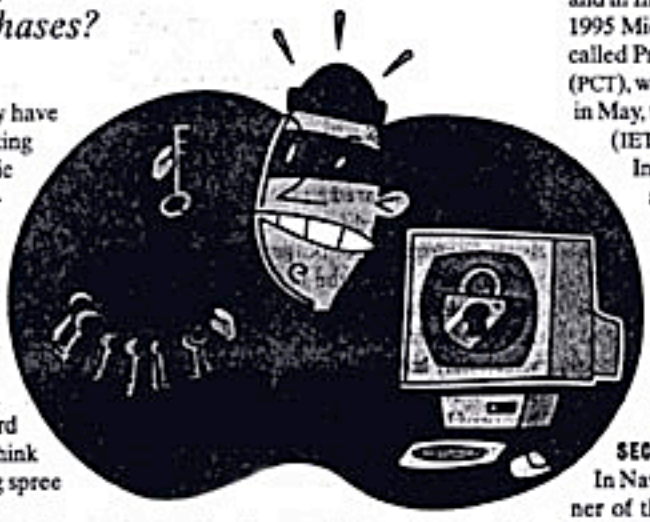
Internet Security Standards

How safe is it to send your credit card number over the Internet to make purchases?

By Michele Rosen



The Internet phenomenon may have happened quickly, but it's taking quite a bit longer for electronic commerce to take hold. Although surveys indicate that almost a quarter of Internet users have made purchases by sending their credit card numbers over the Internet, the same surveys show that many more people still believe that doing such a thing is just plain foolish. Although federal law limits consumer credit card liability to $50, it's disconcerting to think of some hacker going on a shopping spree with your credit card.

Whether or not the fear is irrational, it is certainly understandable. Many also accuse the Internet of being difficult to navigate, but the concepts behind hypertext are child's play compared with the technology needed to protect data from prying eyes. So what really happens when you type in that 16-digit number and press *Submit*? How do Microsoft's and Netscape's browsers stand between unscrupulous hackers and your credit card? The short answer is *encryption*. But of course, as always with technology, it's much more complicated than that.

ENCRYPTION BASICS

You may remember sending notes to your grade-school friends in which you replaced each letter of the alphabet by the number representing its position (A was 1, B was 2, and so on). You didn't know it then, but you were encrypting your message. The key to interpreting the message was to know what the numbers represented. In cryptography jargon, the *key* is the number that, when plugged into an equation or algorithm, allows you to encrypt and decrypt data.

Unless you're a mathematician, however, that's where the simplicity ends. Scrambling data to make it incomprehensible is difficult enough. Doing this in a way that permits returning the data to its original form requires complicated algorithms that employ esoteric mathematical equations.

There are two broad categories of encryption algorithms—*private-key* and *public-key*. In private-key encryption, users accomplish both encryption and decryption with the same key. For data that remains in one location, this is not a problem. A difficulty arises, however, when data needs to be transmitted. If I want to send you a secure message, I use my private key to encrypt it. Then I send it to you. Now you have the encrypted message. But I need a secure way to send you the key.

Cryptographers Whitfield Diffie and Martin Hellman proposed a solution to this problem in 1976 when they invented public-key cryptography. Their solution is deceptively simple: Use two different but related keys to encrypt and decrypt the data. The encryption key is called the *public key*. Since the key used to encrypt the data cannot be used for decryption, there is no risk of its being discovered by unscrupulous individuals. Decrypting the data requires the *private key*—which, thanks to the existence of the public key, can be maintained in a secure location. If I want you to send me an encrypted message, I send you my public key, which you use to encrypt the message. When I receive the encrypted message, I use my private key—which has never left the (relative) security of my computer—to decrypt it. Problem solved!

As always, there is a catch. Public-key cryptography algorithms can take 1,000 times as long as private-key algorithms to encrypt or decrypt data. Public-key cryptography also requires keys up to 10 times as long as those for private-key cryptography to provide an equal level of security.

For these reasons, the security protocols used by both Microsoft's and Netscape's browsers take advantage of the benefits of both public- and private-key cryptographic methods, while avoiding the disadvantages. The most widely used security protocol is called Secure Sockets Layer (SSL). It has been included in Navigator since the first version and in Internet Explorer since Version 3.0. In 1995 Microsoft proposed another protocol, called Private Communications Technology (PCT), which was also included in IE 3.0. And in May, the Internet Engineering Task Force (IETF)—the organization that codifies Internet standards—began considering a new protocol based on SSL, called Transport Layer Security (TLS). Since all three are similar, we'll illustrate how SSL works step by step and then explain how PCT and TLS differ from it, so that you can make a more informed judgment about the security your browser provides.

SECURE SOCKETS LAYER

In Navigator 3.0 and 4.0, the lower-left corner of the screen is reserved for a security icon—a chain in the earlier version and a padlock in the current one. When the broken chain becomes whole or the padlock closes, you know you have entered a secure session with the server you are contacting.

When the browser first connects to a secure Web page, the server sends a "hello request" message. To initiate the secure session, the browser must respond with a message called a "client hello," and the server must answer that with a "server hello." During this initial phase, the browser and server are communicating security information using the handshake protocol, the first part of SSL. The client "hello" message contains a number, called a *session ID*, that uniquely identifies this session between the browser and the server. The message also tells the server which cryptographic algorithms, SSL version, and compression methods the browser supports. Finally, it includes a random number generated by the browser. The server "hello" message responds with the compression method and encryption algorithm it has selected from the choices provided by the browser, the appropriate SSL version, a different random number, and an acceptable session ID number.

At this stage the client and server can exchange digital certificates, which verify that the two parties are who they say they are. The server's certificate can also include a public key appropriate to the public-key encryption algorithm selected during the handshake protocol. This key will be used only for a short time, however; the actual transaction (read: credit card information) will be encrypted

using a private-key encryption algorithm.

To implement this kind of algorithm, both sides must have a single private key, which is generated by the browser. Rather than simply using the public key to encrypt this master key for transmission to the server, however, the browser sends a premaster secret key instead. Based on a predetermined protocol and using the random numbers exchanged during the handshake protocol, the server can use the premaster secret key to determine the true master key. This avoids the necessity of transmitting the actual master key. Once this process is complete, both browser and server have copies of the master key and can communicate securely.

INTERNET EXPLORER SECURITY

As mentioned earlier, Internet Explorer 3.0 supports both SSL and PCT. Like SSL, PCT uses public-key cryptography to encrypt a private key, which is used for the rest of the session between the browser and server. The major difference between SSL and PCT is in the handshake protocol phase. According to the Internet draft proposal written by Microsoft and presented to the IETF, PCT requires fewer messages to negotiate a compatible set of protocols, supports more encryption algorithms, and provides additional security by using different keys for authentication and encryption. Microsoft evidently plans to continue to support both SSL and PCT in future versions of Internet Explorer.

In addition to supporting these security protocols, Internet Explorer 4.0 uses its security zones to let users configure their browsers' security levels at different sites. Each zone is assigned a security level that allows only certain activities to take place. For example, you could assign your company's intranet site to the trusted zone, in which case you could surf the site without encrypting transmissions. On the other hand, you could assign Internet sites you are visiting for the first time to the untrusted zone, which would require the server to provide SSL authentication before the browser uploaded any information.

Internet Explorer 4.0 ships with four defined zones: local intranet, trusted sites, Internet, and restricted sites. Using the Options dialog box, users can alter a zone's security level or create new zones (see Figure 1). A fourth option allows the user to configure a custom security zone.

THE FUTURE OF INTERNET SECURITY

The Transport Layer Security protocol derives its name from the IETF working group charged with developing an Internet standard for a secure, authenticated channel between hosts. Version 1.0 of the TLS protocol was presented to the IETF in May. The protocol is currently based on SSL, but the differences that have been introduced make it incompatible with SSL 3.0.

According to Netscape, the IETF is close to according TLS the status of an Internet standard. This doesn't mean that vendors will be obliged to implement it, of course. But at least there will be a standard for secure transactions against which other protocols can be compared.

The major credit card companies have been developing another standard, called the Secure Electronic Transaction standard (SET), which may have an important effect on the security of Internet transactions. SET wouldn't eliminate the need for protocols such as TLS; rather, it focuses on confidentiality and authentication. SET-compliant software not only will make sure that thieves cannot steal a credit card number; it will also keep a merchant from seeing the number while still providing assurances that the card is valid. The transmission will pass through the merchant's hands directly to the credit card issuer, which will then decrypt it and credit the merchant's bank account.

FIGURE 1: *The security dialog in Microsoft Internet Explorer, Version 4.0, lets you set security levels for four types of Internet and intranet sites.*

But SET's significance goes beyond its ability to protect credit card transactions from prying eyes. That known and trusted companies like MasterCard and Visa created it may instill more confidence among consumers than any strong encryption.

SO IS IT SAFE?

When creating a new algorithm, a cryptographer has no way of knowing for sure that it is airtight against thieves. The only way to increase confidence in any encryption algorithm is through trial and error: Confidence improves as the number of people who try and fail to break it increases. This is why only a few algorithms, such as RSA and DES, are used in most business and government applications; they have stood the longest test of time. But even these algorithms may have weaknesses that cunning hackers can exploit.

Apart from using any weakness in the algorithm, the only way to decrypt encrypted data without the key is a brute-force attack. This method is similar to trying to open someone else's padlock by trying 0-0-0, 0-0-1, 0-0-2, and so on, until the correct combination is found. The longer the combination, or key, the harder it is to find the right number.

This is why so much debate centers around the issue of key length. The large majority of keys range from 40 bits to 1,024 bits; obviously, it's a lot easier (though still not easy) to find the right combination of 40 ones and zeros than to find the correct string of 1,024 ones and zeros. There have been several cases in which people have successfully identified 40- and 48-bit keys. DES 56-bit has also been cracked, but only with an immense brute-force effort by tens of thousands of people. This kind of resource is *not* going to be available to Joe Hacker. Your swimwear purchase at Land's End is almost certainly still safe.

Although encryption techniques continue to improve, cryptographers emphasize that strong encryption isn't the answer to every security issue. Buggy software, human error and greed, and poor server administration leave the door open wide for unscrupulous hackers. On the other hand, a recent review of Internet security breaches indicates that most systems will never experience a break-in and those that do will not be severely damaged. Rather than taking heart in these results, though, one leading cryptographer points out that as long as little valuable data is on the Internet, thieves will stay away; when electronic commerce picks up steam, it's likely that more people will be tempted to try their hands at cracking.

So if the Internet is relatively safe now, what's stopping consumers from buying? Fear of the unknown and the tenacity of old habits are two factors. But like the tortoise, electronic commerce will slowly but surely accelerate. Whether it wins the race depends in part on how well software developers and system administrators protect the process.

Michele Rosen is a freelance technical writer.