Mugged on the Information Superhighway: Security Problems and Solutions

by
Robert Adams
of
Intel Corporation

Abstract

With the information superhighway speeding toward us, how will we protect our secrets? Suppose you use your credit card to order a pizza. How will we keep your neighbor from seeing the number? How will we keep the pizza house from using your credit card for a month's worth of pepperoni? Security, privacy, and encryption are key elements of the emerging cable-based broadband networks.

This paper describes the infrastructure necessary to provide security as part of the underlying infrastructure for broadband networking. It will provide a brief tutorial on the key elements of security, with a focus on the implications in a broadband, cable-based network and participation in a global information network.

Data Networking Comes to Cable

or It's 5pm and Do You Know Where Your Data Is?

Connecting a computer to the cable system is different then connecting a television. And the computers, whether embedded in settop devices or general purpose personal computers, will bring with them a multitude of new applications and new ways to steal.

Computers in the home will be talking at high speeds over the cable plant to services. The services will provide information, entertainment, and commerce. But it will be valuable information, paid for entertainment, and monetary commerce. There must be some way to protect the data.

When connecting computers to servers over the cable system, the first impulse is to use existing computer networking systems. This solves some problems (routing, interconnection, and cost) but the design of computer networks does not include private and personal data.

Existing Computer Networks to the Home

or Send Lawyers, Guns, and Money

Existing computer networks rely on physical security, trust, and passwords for its data security. These methods have their strengths and weaknesses.

Physical security is using walls and locks works for many businesses using computers. If the computer network is all in one building and the server is in a locked room, data security is easy. Outside connections (phones with modems) can be the weak link and we've all heard stories of computer breakins.

Agreements with all who have access to the data (see "physical security" above) tie legal and moral bands on a very weak link in any security system -- the people. These agerrments name all users *trusted users* of the computers. The trust creates needed accountability because only very specialized computer systems have audit trails to remember who did what, when.

The lowly *password* is the first line of defense for nearly all computer systems (and thus networks). Passwords fail to protect all too often because of the people who use them. When picking something easy to remember, most people choose something that's easy for others to guess -- first name, spouse's name, last name backwards, etc.

These have been sufficient because computer networks were developed and used in small or single business environments (where physical security was sufficient) or by the academic, research community (where trust and passwords were enough and, besides, who wants the overhead of audit trials and accountings?).

Today, the "Information Superhighway" does not exchange money, protect private conversations, or tell you if you are communicating with who you think you are. If you tap into the Internet today, you will find plain text mail messages and adhoc administration.

Existing computer networking does not solve the problems of security over the cable plant and we are challenged to address an area of legal entanglements, unknown assailants, and complex problems -- send lawyers, guns and money.

Security Problems

Let's take the data network outside the building. Assume you have a server that people need to communicate with and anyone else can snoop on the bits that are sent between the server and a client. This is how networking to your server over the cable plant would be -- the RF goes into everyone's house and business.

Sending data between the client and the server as "plain text" is certainly out of the question. Anyone listening to the coax cable could receive and look at what I sent -- whether it be my VISA card number or a letter to Aunt Minny.

I could modify the data I'm sending to the server by some algorithm. The server would have to know to use the inverse of the algorithm to create the plain message. Anyone listening to the conversation could not know what was being sent. Well, not until they found out or figured out the algorithm.

A better way would be to use a data modifying (or data "encryption") algorithm that has a little part that is easy to change and is not generally known. This is called a "keyed encryption algorithm" -- the encryption algorithm includes a small part (called a *key*) that I can choose. So, even if the listener knows the encryption algorithm, unless he has the key, the plain message can not be extracted.

I can now send data to the server by encrypting it by the algorithm that uses my key. The server has the decrypting algorithm and the proper decrypting key so it can understand what I sent. The client and the server have a *shared secret* that they use to send data between them. Anyone listening could not receive the data even if they knew the encryption algorithm because they wouldn't have the keys.

How did the client and server institute this shared secret (the keys)? They couldn't send the information over the network un-encrypted because everyone listening would know it also. They could send it through the mail or by messenger. Should there be one shared secret or many? Should I use the same key whenever I talk to this server or would I need a new key every time we exchanged data? Would there be different keys for all the servers I communicate with or would they all use the same? The creation and juggling of all these keys requires some *key management.*

In addition to encrypting the data between my client and the server, there is the problem if the server knowing if it is really me talking to it. For instance, if I send a VISA number, how does the server know it's my VISA number and not a stolen number? Existing manual systems require a PIN or a signature to verify identity. So, we need some sort of *authentication* -- the ability to authenticate the identity of a data sender or receiver.

Authentication identifies a person before something is done, but what about leaving proof that something was done. How do you "sign" an electronic document so that it could be proven in court that it was indeed you that

signed it? Thus there is the need for *certificates* that can be used to identify a person or an association (this person signing this document).

Bulk Encryption

or How To Keep People Out of Your Diary

Bulk encryption refers to algorithms that encrypt blocks of data. They are used to encrypt a message or a stream of data. The encryption algorithm usually includes a key -- the sender knows the encryption key and the receiver has the decryption key. So, for Mary to send a message to Bob, Mary and Bob somehow set up their keys. Mary then encrypts the message with the encryption key and sends it to Bob. Bob decrypts the message with the decryption key. Anybody with the decryption key can decrypt the message so the secrecy of the key is very important.

The most popular encryption algorithm is DES ("Data Encryption Standard") that was defined and endorsed by the US Government in 1977. DES is a "secret key, symmetric" cryptosystem meaning the sender and receiver share a secret key and that the key is the same for encryption and decryption. Researchers have tried to break DES for more then a decade and no one has been successful, so it is felt that DES is reasonably secure.

Skipjack is a new algorithm that is the bulk encryption algorithm chosen for the U.S. government's Capstone project[1]. Skipjack is new and unknown (the algorithm is classified) but it could be the next de facto standard the same way that DES became one -- the U.S. government mandates it as their standard.

Beyond these two algorithms, there are thousands of algorithms that can be and are used for bulk encryption that range from fancy, classified algorithms to Captain Midnight decoder rings.

Public Key Encryption

Public-key cryptographic systems are a relativily new invention (1976) that adds new capabilities to encryption. In this system, each person gets a pair of keys -- one the public key and the other a private key. The public key is published for anyone to see and the private key is kept secret. The encryption algorithm is asymmetric: data encrypted with the public key can be decrypted with the private key.

Say Mary wants to send a message to Bob using public-key encryption. Mary would encrypt the message with Bob's public key (kept in her address book or looked up in some directory). The message would be sent to Bob who could decrypt it with his private key. Anyone could send encrypted messages to Bob and the messages could only be read by him.

The advantage over shared secret cryptosystems is that Mary and Bob didn't have to arrange the shared, secret key in advance. In this public-key example, there was no key exchanged. This has definate advantages for uses like email where the relationship with the receiver might be transitory.

Public-key systems can be used for authentication -- Mary would "sign" the message with her *digital signature*. The digital signature includes the message encrypted with Mary's private key plus other identifying information[2]. Mary would append the digital signature to the message to Bob and then encrypt the message with Bob's public key. Bob, after he decrypted the message, can use Mary's public key to verify that only Mary could have created the signature.

---

[1]Capstone is the is the U.S. government's long term project to develop a set of standards for publicly-available cryptography. There are four major components: a bulk data encryption algorithm, a key exchange protocol, a digital signature algorithm, and a hash function.

[2]To make the size of digital signatures managable, the whole message is not encrypted but a hash function is used to generate a few hundred bit code that represents the document. Several hash codes exist that create unique codes for nearly any document.

One disadvantage of public-key systems is the speed of the algorithms. Because it must be difficult to compute the private key from the public key, the keys are very long (500, 1000, or more bits). This makes encryption and decryption slow. Thus, public-key cryptosystems are usually used for authentication and for key exchange and a bulk encryption algorithm is used for the majority of the data.

The predominate public-key system in use today is RSA, invented in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman (the last name initials gives the "RSA" name). This system is based on large primes and the difficulty in factoring same. Several companies and agencies exist to create and certify public and private keys for individuals.

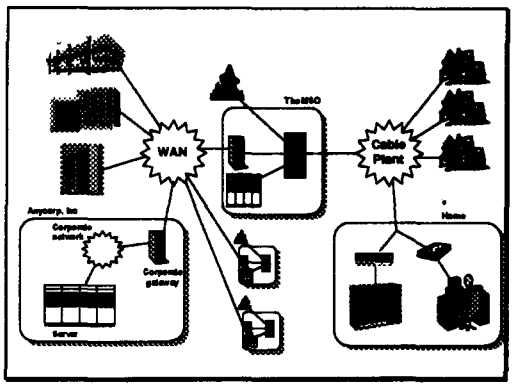
The U.S. government's Capstone Project has chosen an algorithm named DSS ("Digital Signature Standard") for creating digital signatures. It is not as general as RSA and cannot be used for key exchange, for example.

Secure Telecommuting

or "Humm, I wonder what my competition is up to."

Let's look at an application: telecommuting.

In the picture below, Anycorp has employees scattered over the metropolitan area that can telecommute to the companies' offices. Like most metropolitan areas, there are several MSOs serving the region so the company is connected through some wide area data network ("WAN") to the different MSOs. The MSOs provide the data connection from the WAN to the computers in the home.



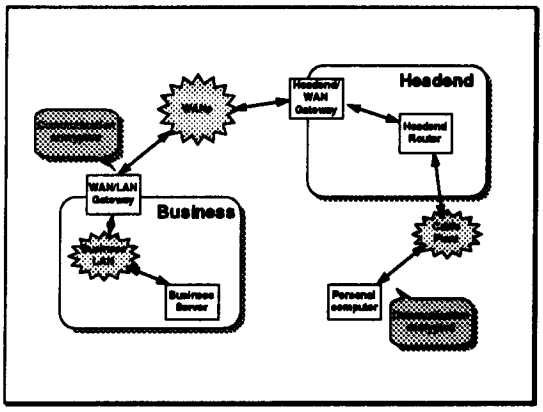
There are many companies in the area that have telecommuting employees so there are multiple companies connected to the WAN and employees from competing companies connected to one MSO. It's obvious that data from one telecommuter will get multiplexed with data from another telecommuter as the data moves between the companies and the employees at home.Additionally, inside each home, competitors' data will be on the cable.

The multiplicity of companies and employees and MSOs adds an interoperability requirement. For instance, if the person in the home is a consultant, she might need to telecommute to two different companies. This means that any security system must be compatible and, if there is specialized hardware to do security, that hardware must be either compatible or everywhere or the person in the home will need different hardware for everyone they're communicating with.

With corporate data moving through various networks and even into competitor's homes, physical security, trust, and passwords won't protect the data.

The way an employee is connected to the company can vary: phone lines (for calling in while on the road), wireless (when using their PIM), and cable (when connected at home). Each of these has its advantages and disadvantages but, to provide useful connectivity, all must be supported and, ideally, the security solution for one should work for all. The security solution cannot just work for connecting a computer to a cable modem.

Any company would make sure that any data leaving the company is encrypted until it gets to the employee as shown in the following simplified drawing.



In a simple system, the employee trys to connect to the company and is authenticated (this could be as simple as a password). All communication thereafter would be encrypted by a shared secret key. A twist on this would be to make the shared secret key the password. In this case, the employee would encrypt all communication to the company with the password and the company could decrypt it because it had the password in its user databases.

For this simple system the MSO is not involved in the security aspects of the communication and thus does not incur extra hardware, administration, or liability. Even if the MSO is selling the "service" of telecommuting, the implementation is mere connectivity between the employee and the company.

Secure Correspondence

or electronic envelopes

The shared password worked in the above telecommuting example because it is reasonable that the employee has to be "registered" with the company before he can connect. This wouldn't work, though, if you wanted to send secure mail to a friend or a business associate. It would be unreasonable to have to arrange a password with everyone you wanted to send mail to.

A public-key system works well for electronic mail. The sender does not have to pre-arrange a shared secret (e.g., a password) with the recipient. The message is encrypted with the recipients public key and sent.

So, while passwords (or fixed, shared secrets) can be used for pre-arranged connections, they are not suitable for the multitude of connections that can be made over a large community or for relationships that are transitory.

Money and Contracts

or digital money can move faster than paper money

If you always went into the same stores, they'd know you and know to accept your checks. If there's only one store available on the cable network, you can have a password to that store. If there are many stores, you need something better then passwords.

When you do buy something and exchange money, merchandise and warrenties, both side of the transaction will want more then just a forgable note saying it was done (digital bits are easily changed). It's not that important when buying a toaster but someday it will be possible to buy houses. Transactions need more then just un-snoopability.

There are also two types of monetary transactions: transactions with a third party (e.g., existing plastic money where someone authroizes and records the transaction) and between only two parties (e.g., paper money). Some people like the latter because of the anonymity. It's easier to implement the former, though.

Digital signatures allow someone to send an unforgable message to a merchant to purchase and item and to exchange money. These can be considered mini-contracts. Also, communication with a credit vendor (a plastic money provider) can be authenticated and transmitted without fear of being overheard.

Digital signatures and the related public-key cryptosystems enable much of the commerce the new interconnected world is supposed to make happen.

Conclusions

Existing computer network systems solve most connectivity problems but they don't address security, privacy or accountability. When the computer network is "up on the poles", the security systems that have worked in the past don't work any more.

Privacy, security, and accountability are required for communication over the cable system. Even if it only supports video-centric services (e.g., video on demand), the requirements of commerce can't allow breakends and forgeries.

With greater interconnectivity, where the real power of having computers at both ends of the connection creates compelling environments, the security needs also include inter-operability and flexibility. A fixed, position dependent, subscriber dependent solution does not create the base for the growth we all envision.

Glossary

authentication - is the process of verifying that the person communicating with is the person expected. In the physical world, an example is showing your driver's license to use a check.

bulk encryption - encrypting large amounts of data.

Capstone - the name of a U.S. government project to define and standardize a bulk encryption scheme, a digital signature standard, and a key exchange protocol for government agencies.

certificate - in public-key systems, they are digital documents that attest to the binding of a public key to an individual. Certificates rely on (contain) the certificates of the attesting agency. This creates a certification chain up to some well trusted authority.

Clipper - the silicon chip that implements the Skipjack bulk encryption algorithm. Part of the Capstone project.

digital signature - data that is unique to the sender and un-forgable. This is usually implemented with a public-key cryptosystem.

key - a "seed" for an encryption or decryption algorithm. The algorithm can be widely known but, without the keys, encrypted data cannot be decrypted.

key management - the processes and protocols used to distribute and store encryption and decryption keys.

password - usually a work or phrase that is the shared secret between a client and a server.

physical security - using physical barriers and monitoring to create

shared secret - some information that the sender and receiver of data knows that no one else knows. This is usually the keys that go with some encryption algorithm.

Skipjack - the bulk encryption algorithm chosen for the U.S. government's Capstone Project.

trusted third party - two people can rely on another agency to distribute and keep communication keys secret. This third party generates keys that are used to communicate. For instance, the Kerberos system has a server on a computer network that will authenticate users on the network and hand out session encryption keys.