Digital Signatures and Electronic Signatures

Dr. Stephen Tate, *University of North Carolina, Greensboro*
Dr. Raymond R. Panko, *University of Hawaii, Manoa*

INTRODUCTION

When we send letters, we sign them to indicate that they are from us. When we sign contracts, we are expressing our willingness to abide by the terms of the contract. We cannot later repudiate the contract because our signature binds us. Signing is also possible in the electronic world, and it generally serves the same purposes.

There are three related terms we use in this article. An *electronic signature* (e-signature) is any signing method that is used with computers and networks. It is the broadest concept. It includes such things as clicking a button to indicate that we accept the terms of a program's end-user licensing agreement.

The other two terms refer to methods for making electronic signatures—specifically, for adding signature blocks to outgoing messages. *Digital signatures* are signature blocks created with techniques from public key cryptography. *Message authentication codes* (MACs) also are per-message signature blocks, but they are created using symmetric key cryptographic techniques. MACs are also called keyed hash functions.

In our discussion, we begin with the most familiar technology, digital signatures. We then discuss MACs and, finally, electronic signatures broadly.

BACKGROUND

Applicant, Verifier, and True Party

A prime reason for electronic signing is authentication. In authentication, there are two main parties. The *verifier* wishes to determine the identity of the *applicant*—the party wishing to have his or her identity authenticated. Applicants are sometimes called supplicants.

In addition, the *true party* is the person the applicant claims to be. (The applicant may be an impostor.) The person who signs the document is the *signatory* or *signer*; this may be the true party or someone authorized by the true party to sign for the true party.

Key-Based Authentication

Authentication can be based on something the person knows (such as a reusable password), something a person has (such as a smartcard), something a person is (biometrics), or some other distinguishing characteristic.

Digital signatures and MACs are based on the applicant knowing a secret key. Digital signatures are based on public/private key pairs and require the applicant to know the true party's private key. MACs require the person wishing to be authenticated to know the symmetric key the true party shares with the verifier.

Since any entity with knowledge of the secret key can successfully authenticate in such a system, it is vital that keys are protected and proper key management techniques are used. In cases where there are significant risks related to transactions protected with digital signatures, secret keys may be protected by specialized hardware devices, such as a cryptographic accelerator, a smartcard, or a trusted platform module (TPM) embedded in a system. In less risky situations, such as routine e-mail signing, the secret key may be stored in a standard file on a computer, but it is then almost always protected by encrypting the file and requiring a passphrase be entered by a user in order to decrypt and use the secret key.

Other key management issues to consider include questions of key backup and recovery, and key escrow (having a copy of a secret key kept by a third party, such as a security officer for a company, in case something happens to the original copy of the key or to the original party that controls access to that key). Key management is an involved subject, and there are many sources for more information, such as NIST Publication 800-57.

Threat Model

In normal authentication, the biggest danger is that the *applicant* may be an impostor who tries to impersonate the true party in one or more transactions. This danger is a key element in electronic signature threat models.

In addition, there is a danger that the true party will later falsely repudiate messages and contracts that he or she signed electronically, claiming that these were signed by an impostor. Against this threat, we would like to have nonrepudiation, that is, the ability to provide proof that the true party actually did sign the messages or contracts.

In simple authentication, the *verifier* is assumed to be the "good guy." However, electronic signatures should also protect the true party from verifier malfeasance. For instance, the verifier might fabricate a message or contract, add a false signature, and then claim that the true party sent the signed message or contract. Only certain types of electronic signatures provide protection against verifier malfeasance—for example, digital signatures do provide this protection, while MACs cannot.

While the true party and verifier are communicating, an *attacker in the middle* may insert a single fabricated message into an ongoing dialog, or might delete a message or simply replay an earlier message. Initial authentication at the start of a dialog will not protect against such attacks.

DIGITAL SIGNATURES

Digital signatures are used in message-by-message authentication. A digital signature is a block of bits attached to each outgoing message to prove the sender's identity. This greatly reduces attacker-in-the-middle threats. A digital signature also provides nonrepudiation on a message-by-message basis. Figure 65.1 illustrates the process of creating and verifying digital signatures.

A basic digital signature scheme consists of three separate functions:

1. ***Generate key pair:*** Produces a random key pair, consisting of a public key and a private key. The private key will be held and kept secret by the true party, while the public key can be widely distributed.
2. ***Sign:*** Uses the private key to produce a signature for a given message. Use of the private key means that only the true party can perform this operation.
3. ***Verify:*** Uses the public key to verify that the given signature is valid for the given message. Use of the public key means that anyone can verify a signature.

A secure digital signature scheme is one in which it is infeasible to create a valid signature (one that passes

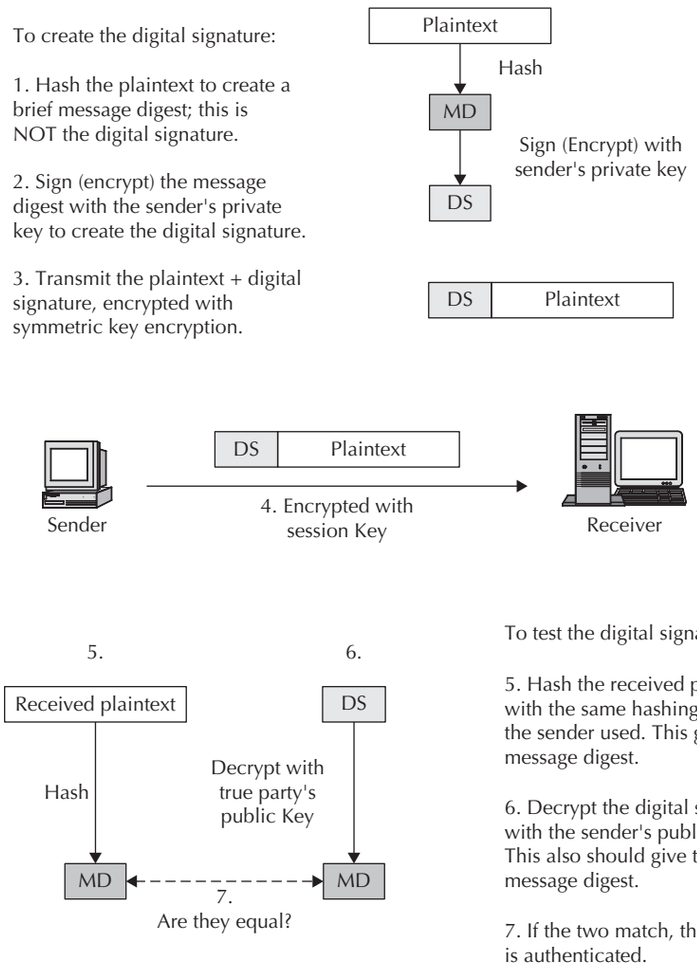

Figure 65.1: Digital Signature Creation, Transmission, and Verification.
Source: From *Corporate Computer and Network Security* (p. 258), by R. Panko, 2004, Upper Saddle River, NJ: Prentice-Hall. Reprinted with permission.

the verify test) without knowledge of the private key. So, for example, if the private key is stored within a secure hardware device that performs signatures without ever releasing the key outside the device, then a valid signature could only have been made by a party with access to that physical device.

Since keys are produced in pairs, of which one part is private and the other part public, digital signatures fall into the general category of public key cryptosystems. Commonly used digital signature algorithms include RSA, DSA, and ECDSA, all of which are believed to be secure when used with appropriate key sizes and parameters. There is also a public key encryption system known as RSA, but it's important to note that while based on the same core idea the two systems work differently and are not interchangeable—a device or piece of software that is designed to do RSA encryption cannot be used directly for creating RSA signatures. DSA and ECDSA are digital signature standards based on a design from the National Institute of Standards and Technology (NIST), and produce more compact signatures than RSA. Most systems using digital signatures (such as web browsers) will support at least RSA and DSA.

Creating the Digital Signature

The sender creates a message to be sent. The sender/applicant will have to sign something using his or her private key for authentication to be possible. However, as a public key cryptography operation, signing is very processing-intensive, so it should only be used on small blocks of bits—not on large messages.

Hashing to Produce a Message Digest

To create something small to sign, the sender's software first hashes the original message. Hashing is a mathematical process that can be applied to a string of bits of any length and that will produce a result (called a hash) that has the same length no matter how long the input string is. Hash functions are used in a variety of applications in computing, but those required for use in security-sensitive applications, such as digital signatures, must satisfy stringent requirements and are referred to as cryptographic hash functions. The most commonly-used hash functions today are SHA-1, which produces a hash of 160 bits, and SHA-256, which produces a hash of 256 bits (an earlier popular hash function, MD5, has been found to have security weaknesses, so should be avoided). So message digests will be either 160 bits or 256 bits, depending on which of these hashing algorithms are used.

The hash of the original message is called the message digest. The message digest is not the digital signature itself but rather the basis for creating the digital signature.

Signing the Message Digest to Produce the Digital Signatures

The applicant/sender wishes to authenticate himself or herself using something only the true party should know. This is the true party's private key. If someone is given a public key/private key pair, he or she should guard the private key jealously. However, their public key is not secret and can be shared freely.

Therefore, the applicant/sender signs the message digest using the *sign* function with his or her private key. The result of this operation is the digital signature.

Transmission with Confidentiality

After creating a digital signature, the applicant/sender creates a composite message by concatenating the bits of the digital signature to the bits of the original message. We will call this the composite message. (Terminology here is not standardized.)

Next, the applicant/sender normally encrypts the composite message using traditional secrecy-oriented cryptography. This provides confidentiality, meaning that no one can read the original plaintext en route. Note that this step has nothing to do with authentication, and it is possible to have authentication without confidentiality. However, confidentiality is normally desired during transmission.

While any encryption technique can be used, typically the sender will use symmetric key cryptography, with a random session key protected using public key cryptography. Symmetric key encryption is used for the message text rather than public key encryption because the composite message may be quite long. As noted earlier, public key encryption is inefficient for long messages. Symmetric key encryption, in contrast, is efficient enough for longer messages.

The applicant/sender now transmits the composite message encrypted with symmetric key encryption to the verifier/receiver. This message cannot be read en route by an attacker in the middle.

The verifier/receiver decrypts the transmitted message in order to restore the composite message.

Verifying the Digital Signature

Now it is time for the verifier/receiver to verify the authenticity of message. The verifier first recomputes the message digest by hashing the received message.

The applicant/sender hashed the original message to create the digital signature. The verifier/receiver rehashes the original message using the same algorithm the applicant/sender used. Hashing is a repeatable process, meaning that the verifier/receiver will get the same resulting hash the applicant/sender obtained. This, of course, is the message digest.

The verifier then uses the *verify* function of the digital signature scheme, using the true party's public key, to test whether the received signature is valid for the computed message digest. As described earlier, a valid signature can only be made using the true party's private key, so passing this test means that the signature must have been made by a party with knowledge of the true party's private key. Assuming the private key was appropriately protected and managed, only the true party (or a duly authorized signatory) would know the true party's private key, so the message is authenticated.

Benefits and Issues

Benefits

Digital signatures provide three important benefits. One is message-by-message authentication. This guards against

the insertion of a fabricated message in a dialog's message stream by an attacker in the middle.

A second benefit is message integrity, that is, proof that a message has not been tampered with en route. If an attacker has deliberately modified a message or if there has been a technical transmission error, the two message digests will not match. The verifier/receiver will discard the message.

The third benefit is nonrepudiation. If the message was signed by the true party, then the true party cannot disclaim responsibility for the message without arguing that he or she lost control of the private key, which itself may be considered negligence. If a private key is stolen, of course, it can be used as a rubber stamp to sign documents. However, for nonrepudiation, the verifier/receiver must keep the original composite message so that authentication can be verified in court or by an expert.

Issues

Digital signature verification requires the verifier/receiver to know the true party's public key. This seems simple, but it is fraught with danger. For instance, suppose the applicant/sender sends the verifier/receiver a public key claiming that it belongs to the true party. If the applicant/sender is an impostor, of course, he or she will send his or her own public key rather than the true party's public key. If the verifier/receiver accepts this impostor's public key as the true party's public key, the impostor will be "verified" as the sender of all messages. This is public key deception.

To guard against such deception, the verifier/receiver must get or verify the true party's public key using a trusted third party. Organizations called certificate authorities (CAs) provide such information in the form of digital certificates. Digital certificates are in turned signed by the CA, so a verifier can verify certificates using the CA's public key, regardless of where the certificate was obtained. Since there are only a handful of trusted CAs, software that uses digital certificates typically comes with public keys of CAs built-in so that this verification can be performed.

The idea of what a CA does is simple, but the practical matters are often extremely complex. A CA often uses a Registration Authority to verify the identity of a party requesting a digital certificate, and while this might be simple for a small organization, large-scale authorities must utilize reliable methods for verifying that a certificate applicant is authorized to obtain the certificate they are requesting. For example, when receiving a request for an organizational certificate, a Registration Authority might check third-party business directories and use a validated phone number to call the organization in question. The policies for verifying the identity of a certificate applicant have a strong impact on how much trust to place in a digital certificate. CAs generally publish their policies for public scrutiny in a document called a "Certification Practice Statement" (CPS). Due to the complex nature of implementing a reliable CA service, CPSs can be very involved—for example, version 3.3 of the CPS for Verisign (a widely used CA) is 109 pages long.

There is a great deal of confusion about digital certificates. The main thing to keep in mind is that the essential information that digital certificates provide is the true party's name and the true party's public key. Anyone claiming to be the named true party should be able to create digital signatures that can be tested with the public key enclosed in the digital certificate.

A digital certificate generally does not vouch for the trustworthiness of the party named in the digital certificates. Although some CAs provide compensation to victims if a named party behaves badly, few do. Vouching for trustworthiness is not what digital certificates are designed to do. Digital certificates are designed to tell you the public key of the named party.

At the same time, companies and individuals who do behave badly may have their certificates revoked before the expiration date on the digital certificate. CAs maintain certificate revocation lists (CRLs) of such digital certificates. CRLs are also used at the request of the true party to revoke certificates when a private key has been compromised or exposed, meaning that signatures created using this private key can no longer be trusted.

It is crucial for the verifier/receiver who gets the digital certificate from another party, for instance the applicant/sender, to check the certificate authority's CRL to be sure that the certificate has not been canceled.

The certificate authority also has a private key and a public key. The CA adds a digital signature to every certificate it creates, signed with the CA's private key. Popular CAs have public keys that are well known, so it is easy for verifier/receivers to check any digital certificate sent to them. As noted earlier, digital signatures provide message integrity, ensuring that the digital certificate has not been modified, say, by entering an impostor's public key in place of the true party's public key.

In practice, every browser today comes with the ability to read digital signatures automatically, without the user's intervention. Browsers also come with the public keys of several root certificate authorities, so they can also handle most digital certificates. What the user sees is merely a notification that a particular document came from a particular, named entity. When this notice appears, the user can feel confident that the message really did come from that person or organization.

MESSAGE AUTHENTICATION CODES (MACS)

Message authentication codes (MACs) are similar to digital signatures. Both are blocks of bits appended to original messages. However, while digital signatures are created using public key cryptography, MACs are created using symmetric key cryptography. Figure 65.2 illustrates the creation, transmission, and use of message authentication codes.

Unlike digital signatures, MACs typically consist of only a single function, which computes the MAC using a symmetric key for a given message. Unlike public key cryptography, which requires complex techniques to create a linked public/private key pair, symmetric key techniques such as MACs typically just use random strings of bits for keys, so no special key generation function is required.

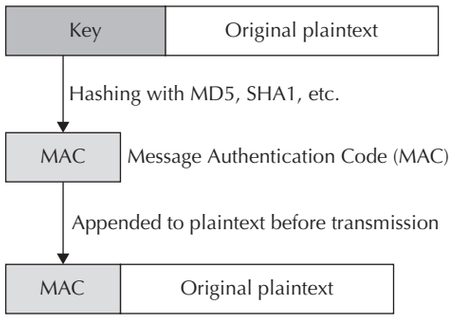

Figure 65.2: The Creation, Transmission, and Verification of Message Authentication Codes (MACs).
Source: From *Corporate Computer and Network Security* (p. 255), by R. Panko, 2004, Upper Saddle River, NJ: Prentice-Hall. Reprinted with permission.

MAC Properties and Advantages

Speed

The main advantage of MACs over digital signatures is processing speed. The public key cryptographic techniques used in digital signatures are typically several orders of magnitude slower than the techniques used for MACs, and so MACs are preferred when the application requires particularly fast responses, has a high volume of requests, or is performed by a device with limited computational capability. For example, if each small packet in a communication must be authenticated, as in IPsec, then MACs are the only feasible solution.

Symmetric Key

The applicant/sender and the verifier/receiver share a single key in symmetric key cryptography. Each of them uses the same key to create MACs and to verify MACs.

When both authentication and confidentiality are required, good cryptographic practice requires that the symmetric key used for MAC authentication is different from the symmetric key used for confidential transmission. In such cases, the two parties have at least two symmetric keys that they share.

Creating a Message Authentication Code

To create a MAC, the applicant/sender again begins with the message to be sent. Next, the applicant/sender calls the MAC function using the pre-arranged symmetric key to compute the authentication code.

One common technique used for computing MACs is known as HMAC, which is an algorithm based on cryptographic hash functions that has a strong theoretical basis. For example, HMAC-SHA1 uses the SHA-1 hash algorithm, and produces MACs that are 160 bits long. Similarly, HMAC-SHA256 produces 256 bit long MACs. Also possible is the use of CBC mode with a symmetric cipher, such as AES, to compute a MAC (sometimes called CBC-MAC or CMAC). Since AES is a 128-bit block cipher, a MAC computed in this manner will be 128 bits long.

To transmit the message, the applicant/sender appends the MAC to the original message. This is the composite message that actually will be transmitted.

Next, for confidentiality, the applicant/sender normally encrypts the composite message. The applicant/sender then transmits the resultant ciphertext. Interceptors will not be able to decrypt the ciphertext back to plaintext because they will not have the decryption key.

Verifying the MAC

The verifier/receiver first undoes the encryption for confidentiality by decrypting the cipher. This gives the verifier/receiver the composite message consisting of the original message plus the MAC.

Verifying the MAC is simple. The verifier/receiver simply repeats the MAC operation on the message and compares the result with the attached MAC. If the composite message has been properly constructed and has not been tampered with, the newly computed MAC should match the one appended by the sender.

If this process successfully reproduces the MAC, then the sender must know the symmetric key used for authentication. Only the true party and the verifier/receiver should know this key. The MAC must have been created by the true party.

Benefits and Issues

Benefits

Like digital signatures, MACs provide authentication. Also like digital signatures, MACs provide message integrity. If the message is altered en route either deliberately or by transmission errors, the verification process will not reproduce the MAC, and the message will be discarded.

Issues

MACs really prove that someone who knows the symmetric authentication key created the MAC. Obviously, this could be the true party acting as the applicant/sender.

However, although it is easy to overlook the fact, the verifier/receiver could also have created the MAC because the verifier/receiver also knows the symmetric authentication key. Why would a verifier/receiver fabricate a MAC? The answer is that the verifier/receiver may be dishonest and wish to claim that the true party sent a message that the true party never sent, for instance a message agreeing to a dubious contract.

Due to the possibility of verifier/receiver misbehavior, MACs cannot provide nonrepudiation. A dishonest true party can repudiate a legitimate message claiming that the verifier/receiver really created it. In court cases, jurors would have to decide who to believe—hardly an easy undertaking.

Where nonrepudiation is an issue, MACs are dangerous.

IPsec

One reason to consider MACs is that they are the default message-by-message authentication mechanism in IPsec (IP security) standards. These are increasingly being used

in virtual private networks (VPNs) to ensure confidentiality, authentication, message integrity, and other benefits in dialogs between partners. If IPsec does become very widely used and if MACs continue to be the default authentication/integrity method, it must be remembered that the packet-by-packet authentication provided by MACs are not sufficient to provide nonrepudiation, and a different authentication technique (such as a digital signature) must be layered on top to provide this capability.

OTHER ELECTRONIC SIGNATURE TECHNOLOGIES

Digital signatures accompanied by digital certificates are the gold standard in electronic signature technologies, and MACs are good despite their lack of nonrepudiation. However, quite a few other types of e-signature technologies are possible and may make sense in particular situations.

Typed Signatures and Scanned Physical Signatures

In the simplest e-signature methods, the sender merely types his or her name at the end of a message or includes a scanned copy of his or her written signature. Although typed signatures and scanned physical signatures are allowed under most e-signature regulations, they are not likely to stand up in court because they are so easily forged.

Click Agreements

When you purchase software, you typically are required to click on a dialog box button to show that you accept the user licensing agreement. Click agreements generally are difficult to enforce in court because of the difficulty proving who actually clicked on the button.

However, enforceability often is not the goal. Rather, click agreements often serve primarily to create a ceremony in which the person formally makes a commitment. This brings the seriousness of the situation to the person's attention. In addition, when people make explicit commitments, they may be more likely to keep them.

Authenticated Sessions

Many transactions are sessions in which the two parties exchange a long series of messages. With various mechanisms, it is possible to authenticate the user at the beginning of the session and perhaps occasionally during the session to ensure that the person is still there. Although less secure than digital signatures and MACs, which authenticate every message, authentication at the beginning of a session (initial authentication) provides some assurance that a certain party is sending the messages.

Most initial authentication systems rely on reusable passwords, which people (or software processes) use each time they log in for a certain period of time. The problems with reusable passwords are well known. Unless the system enforces strong passwords, people tend to use easily guessed passwords that can be cracked in a few seconds by a password-cracking program. People also tend to write their passwords somewhere, often on their computer monitors.

A more subtle problem is lost passwords. About a quarter of all calls to help desks are for lost passwords. The help desk operator can perform a "password reset," giving the account a new password. However, there is danger in giving out new passwords over the telephone. The caller may be an imposter, not the real account holder. Although some password reset systems require the caller to answer questions that only the true account holder should know, most of these questions are easily guessed if the impostor has done his or her research.

Another means of authentication is access cards and token. If you have stayed at a hotel room recently, you probably were given an access card that allowed you into your room. Most such access cards have magnetic stripes containing information that allows access to the room; smart card versions have microprocessors and memory for more sophisticated identity checking.

Even if you get a key, the key is likely to be a physical token that contains access information. Plugging it into the door allows the door reader to query a central system for access permission. There also are tokens that plug into the USB ports of personal computers for access to those machines.

Another type of token requires the user to enter a PIN number on a small (generally) numerical keypad. The token then shows a temporary password on its display. The user must use this temporary password to log into a computer system.

Access cards and tokens provide good security, but they are easily lost or stolen. There must be a quick way to disable lost access devices as well as a way to reissue them in ways that impostors cannot exploit.

Another approach to handling authenticated access sessions at Web sites is visit traces, recording the paths users take through them, including click agreements they have made. Documentation that a person saw certain information can be convincing evidence in court and may prompt a person to drop repudiation claims. However, visit trace logs must be secured against tampering by the logkeeper if they are to be useful in court.

Biometrics

One form of authenticated session technology is biometrics. It is new and complex, so we will give it its own section.

"Biometrics" comes from the words "bio," meaning biological life, and "metrics," meaning measurement. Some types of biometrics measure bodily dimensions, such as fingerprints, iris patterns in the eye, and facial features. Other types measure activities, such as motions and pressures involved when signing a name or the temporal patterns of password typing.

The advantage of biometric authentication is that it does not require the applicant to carry something that can be lost and (usually) does not require the applicant to remember anything. Despite jokes to the contrary, we never actually forget our heads at home. Many people hope that biometrics will replace reusable passwords as the dominant form of authentication.

There are several problems related to biometrics that have stood in the way of widespread adoption. First, the

immutability of a biometric, meaning that it is tied to the individual being authenticated, is both its strength and a weakness. The problem with this is that if a measured biometric can be captured and replayed, then the system can be spoofed. Such a capture might take place by monitoring communication in a poorly-designed protocol, or by physical capture of biometrics—one amusing research project showed how fingerprints could be captured and replicated using gummy bear candy so that early fingerprint scanners were fooled. Making this situation much worse is the inability to revoke a biometric—a person can't simply revoke their fingerprints and get a new set.

Another major problem with biometrics is that there are serious disagreements over the error rates involved in biometric measurements. Many vendors make impressive claims about accuracy, but these often are based on tests conducted under ideal conditions and may not be representative of accuracy in the real world.

There are two basic types of errors in biometrics, indeed in all access control methods. False acceptance rates (FARs) measure how often a system verifies or identifies someone who should be rejected. High FARs means impostors are getting in and forming a strong basis for repudiation. A failure to test FARs may make access data difficult to defend in court.

At the opposite end of the spectrum is false rejection rates (FRRs). FRRs tell you what percentage of legitimate applications is rejected. Although FRRs do not harm security, significant FRRs may make a system unacceptable to users. Many systems allow applicants to attempt to authenticate themselves several times to reduce FRRs.

Another issue in biometrics is user acceptability. For instance, some users may refuse to use a fingerprint system because of its criminal connotations. Others will reject systems they fear may harm them; for instance, some people believe that eye identification systems shoot laser beams into their eyes. Still others reject systems that are difficult to use, such as iris scanning systems, which require proper eye placement. In general, if a significant number of users refuse to use the system, the loss in revenues and other values may make the system completely cost ineffective.

Another problem with biometrics is that the different technologies vary widely in cost and accuracy. Selecting a biometric method is an important task. Not surprisingly, the most expensive methods tend to be the most accurate. In addition, the cost of biometric readers and other system components may be prohibitive.

A final problem is that we do not yet have comprehensive standards for biometrics. As a consequence, choosing biometric authentication today generally means getting locked into a single vendor.

SELECTING AN ELECTRONIC SIGNATURE METHOD

Selecting an electronic signature method is not a technical decision. It is a business decision. Like any business decision, it requires the selector to understand the business situation before considering anything else.

Some e-signature systems serve closed communities, such as individual corporations or consortia of firms. Others serve open communities, such as a vendor and its customers. In open communities, it is difficult to impose stringent e-signature requirements. For instance, in consumer e-commerce, it is traditional in the SSL/TLS methodology that is used in almost all transactions to require the merchant but not the consumer to have a digital certificate and produce a digital signature. Note that e-signature implementations can be asymmetrical, with different requirements imposed on the two sides. In addition, in an open community, the general lack of e-signature and PKI standards and the need to coordinate rollouts in many firms tends to require long lead times.

There are two forms of authentication: verification and identification. In verification, the person claims to be a particular person, for instance by typing in an account name. The authentication system then only has to see whether the password typed or the other authentication data given is correct for the account. If so, the applicant is verified as the true account holder. In identification, the applicant does not claim a particular identity. The applicant provides authentication data, and the identification system matches that data against that of *all* the accounts in the identification database. If the best match that is selected meets closeness-of-fit criteria, the applicant is identified as that person (or software process) in the database. Identification is more difficult than verification and thus has higher error rates. It also may lack the intentionality that is normally present in signing activities and so may not be enforceable in court.

Security Requirements

Signatures of any kind exist to validate agreements. To be effective, they must be safe from use by impostors and from other security threats.

In security, one must always consider threat severity, which is the likely cost of a security incursion times the probability an incursion will take place. It is, in other words, the expected value of loss from a threat. If the threat severity is low and is likely to remain smaller than the cost of implementing an e-signature system, then implementing the system will not make economical sense.

In general, e-signatures should be viewed as security techniques, and security always must consider risk management—the balancing of risks and countermeasure costs. Generally speaking, however, the more expensive the transaction, the more expensive the e-signature.

Key length is an important security concern when private keys are used to sign documents or message digests. Documents that must be kept secure for many years must be signed with longer keys than documents whose signatures only have to be verifiable for a few years. However, longer keys mean longer processing times and therefore higher costs. The period of sensitivity is a crucial determinant of key length.

Legal Goals

Another consideration in selecting an electronic signature technology is a firm's legal goals. Many countries require that contracts worth over a certain amount of money be signed to be valid. In U.S. commercial law, for instance,

contracts over $500 or lasting a year or more must be signed to be valid. If the goal is to meet this requirement, even the least-secure e-signing methods may be acceptable.

As noted earlier in this article, one consideration is whether or not to create a ceremony of commitment, in which a person must explicitly, through an action, acknowledge ownership of a document. Again, if this is the main goal, even nonsecure e-signing methods may be acceptable.

If the goal is nonrepudiation, then a stronger e-signature method is needed. The only method available today that provides strong technical nonrepudiability is use of digital signatures, which require a PKI for digital certificates. The signature and digital certificate should be retained so that validity can be tested in court if necessary.

Although technical nonrepudiability is good, juries are likely to decide contested cases. If the complainant is more believable than the defendant, the jury may disregard the defense's argument that the e-signature method used does not provide technical nonrepudiation. If the defendant is more believable, the jury may side with the defense even if a digital signature and digital certificate are used.

Suitability

Another consideration in selecting an e-signature methodology is whether the system can be implemented. One question to ask when considering the various methods is what, in any given firm, is technically feasible. Some choices may not be feasible, and some that are may be outside the firm's resources to implement. PKIs are especially problematic.

Of course, the firm must consider the cost of implementing a system, including the cost of the technology itself and the cost of installing it. In addition, electronic signature processing may slow computer processing enough to require upgrading to faster hardware.

The last major consideration in determining the suitability of a method is whether users will accept it. As noted earlier, fingerprints, iris scanning, and other techniques may offend users or make them uncomfortable. The company must also consider the cost of lost business if some users refuse to use e-signatures and therefore stop doing business with the firm. On the other hand, a firm may gain revenues if it develops a reputation for having strong security in general.

Obviously, individual home consumers are unlikely to be willing to get digital certificates unless they see strong benefits from doing so. As a result, most firms are unlikely to require consumers to use digital certificates for fear of losing business.

LEGAL AND REGULATORY ENVIRONMENT

One consideration that is especially difficult to discuss clearly is the legal and regulatory environment of electronic signatures. Laws vary around the world and even within countries. Few of these laws, furthermore, have been tested in court. Regulation of certificate authorities and other aspects of electronic signatures barely exists, even in countries that have begun to back legislation with regulation.

In the United States, a number of states created electronic signature laws before the federal government created the Electronic Signatures in Global and National Commerce Act, better known as E-SIGN, in 2000. As the name states, E-SIGN governs only national (interstate) commerce. No court cases have yet appeared to test this area.

Quite a few states have created their own electronic signatures laws. In 1999, before Congress acted, the National Conference of Commissioners on Uniform State Laws adopted the Uniform Electronic Transactions Act (UETA), and a number of states based their laws on this act. However, past court rulings have found that many intrastate activities affect interstate commerce and so are governed by federal laws.

In 1999, the European Parliament and the Council of the European Union created Directive 1999/93/EC, On a Community Framework for Electronic Signatures. This directive did not create laws but rather directed the member countries to create e-signature regulation. The first country to do so was Ireland, in July 2000. Germany followed in May 2001.

The simplest element in electronic signature laws is whether legal validity has been established. As noted earlier, most jurisdictions require that contracts worth more than a certain amount of money be signed to be valid. Almost all e-signature laws provide the weak protection of saying that contracts cannot be invalid simply because they are signed electronically. This does not ensure that a particular e-signature methodology will stand up in court if one side repudiates the document.

One consideration in selecting an electronic signature technology is whether a country's e-signature law allows all types or only some types of electronic signatures. The U.S. E-SIGN Act is intentionally vague, not mentioning specific e-signature technologies. The EC Directive mentions several specific technologies but does not limit itself to them.

The EC Directive defines an electronic signature in general as "data in electronic form which are attached to or logically associated with other electronic data and which serve as a method of authentication." The EC Directive also specifies *advanced e-signatures,* which basically are digital signatures based on qualified certificates—the strongest type of electronic signature. These advanced signatures are given a certain degree of privileged status, which is reasonable because of their strength. Most important, advanced electronic signatures are viewed as equivalent to hand signatures in legal proceedings.

Another consideration is whether e-signatures are permitted for all types of documents or only some. For instance, in the United States, the federal E-SIGN law forbids certain documents to be signed electronically because of potential harm to consumers. These exclusions were not present in early versions of the laws, which failed to pass because of insufficient consumer protection. Among the documents excluded are wills, trusts, adoptions, divorces, other family court matters, court documents, utilities cancellation notices, notices of foreclosures, eviction notices, insurance cancellations, warnings about the transportation of hazardous materials, and the repossession of primary residences.

Another key consideration in selecting an electronic signature technology is whether one side can force the other side to accept electronic signatures or whether using e-signatures is voluntary. Early versions of the U.S. E-SIGN law were rejected by Congress because they failed to say explicitly that people cannot be forced to accept electronically signed documents. The final E-SIGN Act says that consumers must be notified of options, including the mandatory option of paper-only transactions, must give consent, and must demonstrate the ability to store and access digital documents.

Digital signatures require digital certificates from certificate authorities. A major issue is whether a country will regulate certificate authorities. The U.S. E-SIGN Act leaves everything to industry. The EC Directive, in contrast, specifies that each country should establish a regulatory framework within the country for certificate authorities and related services, such as registration services, time stamping services, and directory services (Article 1a). In addition to all these, cyber crime has no physical boundaries, and questions of jurisdiction can be significant. A person from any country can actually commit a crime to a party of another country of which there is no legal framework in that country to punish them.

CONCLUSION

Electronic signatures exist for legal reasons. We have long been able to send documents electronically. However, many documents must be signed to be legal in court proceedings. Until recently, the legality of electronic signatures sent with documents was uncertain.

Now, however, most countries have adopted electronic signature legislation, which permits certain types of electric signing. The U.S. E-SIGN legislation is especially broad in terms of what forms of electronic signatures it will recognize.

One reason for signing documents is to provide nonrepudiation, which means that the signer cannot claim not to have signed the document if they really had done so. Only one form of electronic signature provides nonrepudiation. This is the digital signature, in which the sender signs a message digest with his or her private key. If this digital signature can be verified with the true party's public key provided by a digital certificate from a reliable certificate authority, the only rational basis for repudiation is that the true party's private key was stolen.

Similarly, if someone falsely claims that a person signed a contract, a digital signature will prove that assertion false because verification with that person's public key will fail.

Consequently, although many forms of electronic signature are permitted by law, only digital signatures provide strong legal protections.

GLOSSARY

Access cards: Have magnetic stripes containing information that allows you into, for instance, a hotel room. Smart card versions have microprocessors and memory for more sophisticated identity checking.

Advanced e-signatures: In the European Union electronic signature directive, digital signatures based on qualified certificates.

Applicant: The party wishing to have his or her identity authenticated.

Attacker in the middle: Party who may insert a single fabricated message into an ongoing dialog, delete a message, or replay an earlier message.

Authentication data: Where an identification system matches user-supplied data against all accounts in the identification database to determine who the user is.

Biometric authentication: The authentication of a person based on body measurements.

Ceremony of commitment: Act of signing a document that results in heightened awareness of the gravity of the situation on the part of the signer.

Certificate Authorities (CAs): Organizations that create and distribute digital certificates.

Certificate Revocation List (CRL): List of a certificate authority's certificates that have been revoked before the termination date listed on the certificate.

Click agreements: Electronic signatures created by the user clicking on a button that says, for example, "agree."

Digital certificates: Documents that give a named party's public key and other information.

Digital signatures: Signature blocks created with public key cryptographic techniques. They are created by signing message digests with the applicant's private key.

Directive 1999/93/EC: European Union electronic signature directive to the EU member nations.

Electronic signature (e-signature): Any signing method that is used with computers and networks.

E-SIGN: U.S. electronic signature law.

False Acceptance Rates (FARs): Percentage of times a person is authenticated when he or she should not be.

False Rejection Rates (FRRs): Percentage of times a person is not authenticated when he or she should be.

Hashing: A mathematical process that can be applied to a string of bits of any length and that will produce a result (called a hash) that has the same length no matter how long the input string is. Security-sensitive applications typically require a special type of hash function known as a cryptographic hash function.

HMAC: A popular technique for calculating a message authentication code (MAC).

Identification: Process in which the applicant does not claim a particular identity. When the applicant provides authentication data, the identification system matches the data against all the accounts in the identification database to determine who the applicant is.

Message authentication codes (MACs): Per-message signature blocks, created using symmetric key cryptographic techniques. MACs are also called keyed hash functions.

Message digest: The result of hashing a message. This is the first step in creating a digital signature.

Message integrity: Proof that a message has not been tampered with en route to its destination.

Nonrepudiation: When the sender cannot plausibly claim that a message did not really come from him or her.

Password reset: Gives an account a new password when a reusable password is forgotten.

Period of sensitivity: Period of time during which a file must be kept confidential.

Public key deception: Process in which an impostor sends his own public key, claiming that it is the true party's public key.

Repudiate: When a party claims that he or she did not send a message that was apparently sent by him or her.

Reusable passwords: Passwords used repeatedly. Most passwords are reusable passwords.

Scanned physical signatures: Electronic signatures formed by scanning a written signature and inserting the image in the document as a signature block.

Signatory: Party who actually signs the document. May be the true party or someone or something to which the true party delegates signing authority.

Supplicant: Another name for an applicant.

Tokens: Physical devices used to authenticate a person.

True party: The person or object the applicant claims to be.

Typed signatures: Electronic signatures in which the sender merely types his or her name.

Validity: Characteristic of a document that allows it to be presented as evidence in a court.

Verification: When the person claims to be a particular person, for instance when typing in an account name.

Verifier: The party wishing to determine the identity of the applicant.

Visit trace: A list of the locations a user has visited at a Web site.

CROSS REFERENCES

Digital Identity; E-Commerce Vulnerabilities; Electronic Payment Systems;
PKI (Public Key Infrastructure); Internet Security Standards; Public Key Standards: PKCS (Public-Key Cryptography Standards); Encryption Basics; Hashes and Message Digests; Public Key Algorithms

REFERENCES AND SUGGESTED READINGS

American Bar Association. 1996. *Digital signature guidelines: Legal infrastructure for certification authorities and electronic commerce.*

Ford, W, and M. S. Baum. 2000. *Secure electronic commerce: Building the infrastructure for digital signatures and encryption.* 2nd ed. Englewood Cliffs, New Jersey: Prentice-Hall PTR.

Grant, G. L. 1997. *Understanding digital signatures: Establishing trust over the Internet and other networks (CommerceNet)*. Columbus, Ohio: McGraw-Hill Professional.

Hammond, B. 2002. *Digital signatures*. Emeryville, CA: McGraw-Hill Osborne Media.

Panko, R. 2004. *Business computer and network security.* Englewood Cliffs, NJ: Prentice-Hall.

Piper, F., S. Blake-Wilson, and J. Mitchell. 2000. *Digital signatures: security and control*. New York: Information Systems Audit and Control Foundation, 2000.

Pfitzmann, B. 1996. *Digital signature schemes: General framework and fall-stop signatures*. New York: Springer-Verlag.