Chapter 16
DIGITAL CERTIFICATES

Digital certificates and digital signatures are signed (private key encrypted).

In Chapter 12, Alice signed (used her private RSA key to encrypt) a newsletter; only her matching RSA public key verifies (decrypts) the newsletter into meaningful text. Recall that digital signing authenticates the message author and ensures message integrity, as shown in Figure 16-1.

Digital signing is also used to make *digital certificates*. The person who creates a digital certificate proclaims something like this: "I attached H. X. Mel's public key to this digital certificate and then signed (a hash of) it with my private key."

We'll see that any user of H.X. Mel's digital certificate must completely trust the competency and honesty of the person who creates H.X. Mel's certificate. We'll also see that for anyone to confidently use H.X. Mel's digital certificate, they must also trust they have a validated copy of the certificate creator's public key. But before we examine these trust points, let's first show the two component parts of every digital certificate.

Digital certificates have two parts: plaintext and the same plaintext hashed and digitally signed. Figure 16-2 shows a digital certificate created by Alice for her daughter, Dawn. The plaintext, in the top part of the certificate, openly states that Alice created the certificate for Dawn and that Dawn's public key is attached. The plaintext, hashed and signed (in the bottom part), completes Alice's certification of the contents shown at the top.

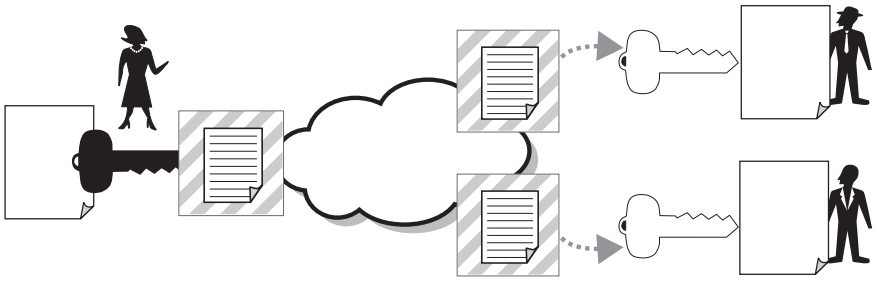


Figure 16-1 Alice signs her newsletter.

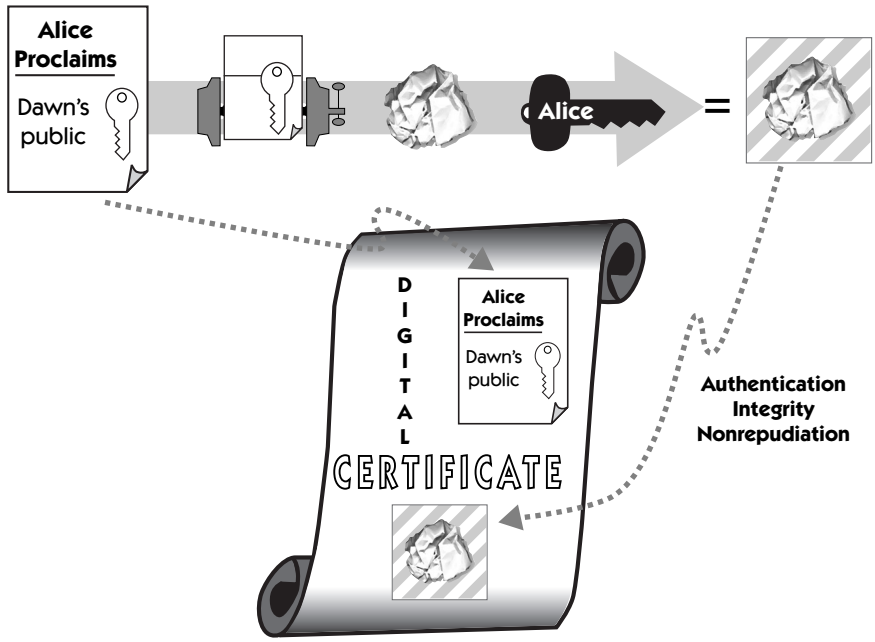


Figure 16-2 Alice makes a digital certificate to hold Dawn's public key.

Digital certificates are openly distributed and include plaintext.

Digital certificates can include plaintext because there's usually nothing secret about their contents—the digital certificate creator (e.g., Alice), someone's public key (e.g., Dawn), and so on. Digital certificates must include plaintext because, as mentioned in Chapter 12, many digital signature methods, such as DSA, don't recover plaintext as a byproduct of the verification process. Plaintext is included so that the recipient of the digital certificate knows who issued the certificate and whose public key is enclosed. Also, as you'll soon see in Chapter 17, because Bob (and other Internet users) collects many digital certificates, he can look through the plaintext part of a digital certificate before he decides to spend computer time verifying the authenticity of the certificate and public key enclosed.

Include signed hash for integrity.

A digital certificate must include a signed hash because, as explained in the introduction to Part III, e-mailing a plaintext public key is susceptible to a man-in-the-middle attack. Including a signed message digest over the plaintext in the top part allows the recipient to detect tampering.

Verifying a Digital Certificate

Definitions: issuer, subject

Figure 16-3 shows Alice issuing a digital certificate for Dawn. Alice is the *issuer* (the one who signs with her private key); Dawn is the *subject* (the certificate contains her public key). Then Alice sends the digital certificate to Bob, who wants a trusted copy of Dawn's public key.

Bob verifies a digital certificate in the same way that he verified the newsletter in Chapter 12.

Bob verifies that Alice signed the digital certificate just as he verified Alice's newsletter in Chapter 12. The difference is that now, instead of a newsletter, he gets Dawn's public key.

Bob hashes plaintext in the top part and uses his copy of Alice's public key to verify (decrypt) the bottom part. Because, as shown in Figure 16-3, both parts are equal, Bob has verified that Alice created the digital certificate and that it has not been altered since Alice made it. Just as Alice's signed newsletter assured her customers, Alice's signed digital certificate assures Bob that he has Dawn's public key—assuming that Bob trusts Alice and trusts that he has Alice's public key.

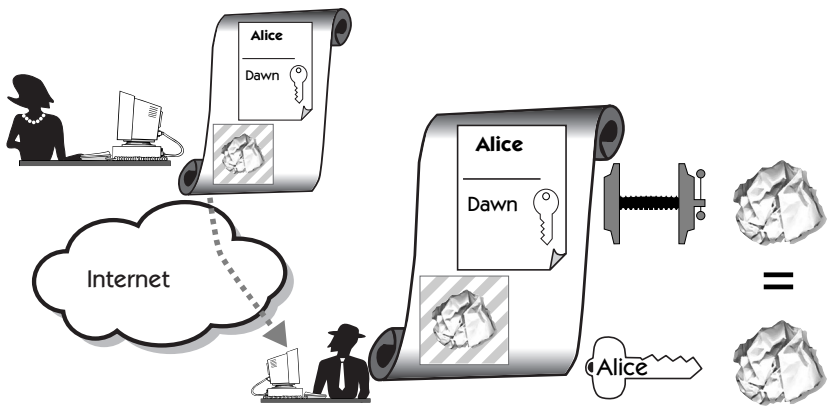


Figure 16-3 Bob verifies a digital certificate. Bob uses his trusted copy of Alice's public key to verify that Alice made the digital certificate. (Shown is RSA verification; DSA is slightly different.)

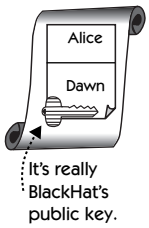
Attacking Digital Certificates

Even though digital certificates and their attached public keys are openly distributed, they can be compromised if handled without proper safeguards. Let's see four ways that BlackHat can attack digital certificates.

Attacking competency

Attacking the Creator of the Digital Certificate

If BlackHat can successfully masquerade as Dawn, he can convince Alice that his public key is Dawn's public key. Then Alice will mistakenly put BlackHat's public key on the digital certificate she creates for Dawn. BlackHat can then masquerade as Dawn to Bob or anyone who receives Alice's mistakenly created digital certificate. Bob must completely trust Alice's competency.



Attacking reliability

Malicious Certificate Creator

Bob completely trusts that Alice will never issue a fraudulent digital certificate. This means that Bob trusts that Alice would never issue a digital certificate in Dawn's name that contains BlackHat's public key instead of Dawn's public key.

If Alice did perpetrate this kind of fraud, BlackHat could easily masquerade as Dawn to anyone who trusts Alice.

Attacking the Digital Certificate User

Weak link: Bob must get and protect Alice's public key.

Bob must protect his copy of Alice's public key. If BlackHat substitutes his public key for Alice's public key on Bob's computer, BlackHat can masquerade as Alice and send Bob digital certificates supposedly from Alice. A weak link of public key cryptography is that Bob must have and protect his copy of Alice's public key.

The Most Devastating Attack

Unauthorized use of private key

If BlackHat somehow gets Alice's private key, he can forge Alice's signature on digital certificates. Anyone retrieving BlackHat's forged certificates has no way of knowing that BlackHat perpetrated the fraud. For example, with a copy of Alice's private key BlackHat can create and sign a digital certificate, which then appears to have been made by Alice and appears to contain Dawn's public key. In reality, it has been made by BlackHat and contains any public key BlackHat chooses. Anyone using BlackHat's forged digital certificate (e.g., Bob) has no way of knowing otherwise—until Bob is notified that Alice's private key has been compromised.

Nevertheless, all the major e-commerce communication programs use digital certificates as the de facto technology for distributing public keys. Secure Socket Layer, secure e-mail, virtual private networks, and IPsec all use digital certificates.

Understanding Digital Certificates: A Familiar Comparison

Trying to keep track of all these people and their keys can be a little confusing. Here's an analogy that may help. A digital certificate is not very different from another common means of identification: a driver's license. Refer to Figure 16-4 as we compare a digital certificate to a state-issued driver's license.

Issuer and Subject

After H. X. Mel satisfies the requirements, Minnesota makes and issues a driver's license that contains Mel's name and picture. Minnesota is the authority that issues a license for the subject, in this case Mel.

Similarly, with digital certificates, Alice issues (certifies and makes) a digital certificate that contains Dawn's name and Dawn's public key. Alice is the authority who issues a certificate for the subject, Dawn.

Issuer Authentication

Cryptography can prove ID better than pictures can.

The Minnesota Department of Motor Vehicles (DMV) laminates and binds the DMV seal and Mel's photograph to the license. The DMV seal asserts that the DMV believes Mel's photograph is accurate; making alterations to Mel's picture is difficult.

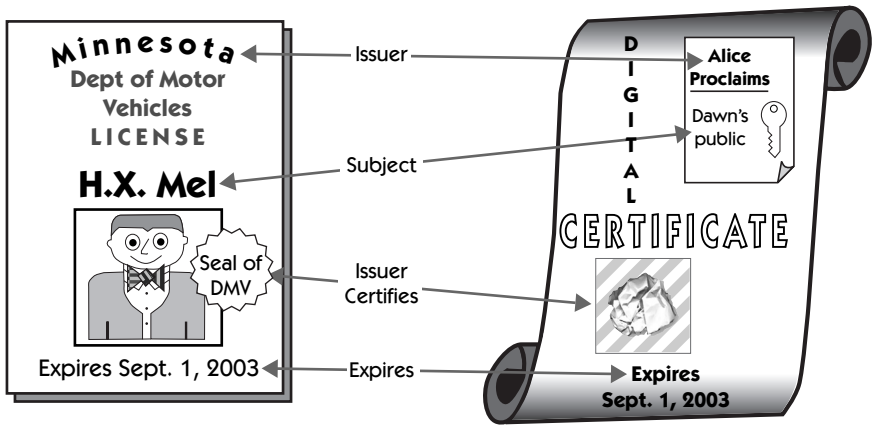


Figure 16-4 A digital certificate is like a driver's license.

Similarly, Alice signs (private key encrypts) Dawn's public key in the certificate. Alice's digital signature authenticates her assertion that Alice believes Dawn's public key to be accurate. Cryptography makes altering the digital certificate much more difficult than altering Mel's photograph on his driver's license.

Transfer of Trust from the Issuer to the Subject

Mel uses his license photograph, created by the DMV, to prove his identity to a stranger, the clerk at a 24-hour convenience store. The clerk trusts that she knows what a real license looks like and trusts that the DMV put Mel's correct picture on the license. The clerk must also trust that her eyes can correctly match Mel's photograph to Mel's physical appearance. If the clerk trusts the DMV and her eyes, she takes Mel's personal check.

Bob trusts that Alice signed Dawn's public key *and* that he has Alice's correct public key.

Similarly, Dawn uses her digital certificate, issued by Alice, to prove her public key to Bob. Bob trusts that Alice put Dawn's correct public key on the digital certificate. Just as the clerk trusts her eyes to verify a driver's license, Bob must also trust that he has Alice's correct public key. Alice's public key is used to accept (or reject) the integrity of a digital certificate (and Dawn's attached public key).

Bob trusts Alice, so he trusts that he has Dawn's public key. For example, suppose that Dawn e-mails a signed (private key encrypted) report to Bob.[1] Bob trusts that the report originated with Dawn after he verifies (decrypts) it with Dawn's public key. How can Bob get Dawn's public key? Alice e-mails Bob Dawn's public key on a digital certificate issued by Alice. How does Bob know that BlackHat didn't intercept the e-mail and substitute BlackHat's public key?

Definition: chaining

Bob verifies that the digital certificate and Dawn's attached public key were not altered in transit because the digital certificate verifies with his copy of Alice's public key. Bob has *chained* from his trusted copy of Alice's public key to a trusted copy of Dawn's public key. You'll learn more about chaining in Chapters 17 and 18.

Table 16-1 summarizes these analogies. In Chapters 20 and 21 you'll see how digital certificates are used to prove identity to e-commerce vendors and Internet correspondents.

Note that digital certificates are verified by software that's inherently much more accurate than the matching skills of a human eye.

1. Recall that private key encryption ensures that she authored the report. Private key encryption does not ensure confidentiality.

Table 16-1 Trust: Comparing a driver's license picture to a digital certificate.

License: What the Clerk Trusts	Digital Certificate: What Bob Trusts
DMV put Mel's accurate picture on license.	Alice put Dawn's key on the digital certificate.
Clerk knows what a license looks like.	He has Alice's public key.
Her eyes can match Mel's DMV photo to his physical appearance.	His copy of Alice's public key will correctly verify (or reject) the integrity of the digital certificate.

Issuer's Limited Liability

Bob can't (easily) sue Alice for an incorrectly issued digital certificate.

The DMV certifies only that Mel has successfully completed a driving test and that Mel's picture is on the front of the license. A victim of Mel's bounced checks or reckless driving can't sue the DMV.[2]

Similarly, Alice certifies only that Dawn identified herself and that Alice signed Dawn's public key on Dawn's certificate.[3] A victim of Dawn's improper and/or unethical use of her public/private key pair can't sue Alice. For example, if Dawn signs a message and denies authorship by falsely claiming that her private key was stolen, Alice is not legally responsible for Dawn's deception.

Time Limits

The DMV requires that Mel renew his license every few years. Although Mel is a law-abiding citizen, car rental firms won't rent to him if his license has expired.

Digital certificates also need to be renewed. Bob should never trust the public key on Dawn's digital certificate after the expiration date (see the front of the digital certificate in Figure 16-4).

Revoking Trust

If Mel violates driving laws, the DMV can revoke his license before the expiration date. Car rental companies likely find a DMV revocation list very useful.

2. Of course, anyone can sue anyone for anything. Winning is another story.
3. In Chapter 17 you'll see that Alice and Dawn can choose different ways for Dawn to prove her identity to Alice. For example, presenting yourself in person is stronger than presenting yourself via e-mail.

Revoking a digital certificate

If Dawn violates Alice's guidelines, Alice can revoke the digital certificate she made for Dawn before the expiration date. But how will Bob know that Alice has revoked Dawn's certificate? How to handle distribution of certificate revocation lists is the subject of on-going discussions and is covered in the next two chapters.

More than One Certificate

It's often useful to have more than one digital certificate.

If Mel lives in The Netherlands for part of each year, he might want to get a Dutch driver's license. The two licenses might come in handy if, for example, Mel has to prove his identity to a British bank.

Similarly, Dawn might want more than one person (say, Trusty Tom) to issue her a digital certificate. For example, she might want to prove her public key to someone who doesn't completely trust Alice but does trust Trusty Tom.

Fees for Use

Issuing digital certificates for profit

By issuing a license to (young man) Mel, the DMV enables him to drive legally. The DMV gets a few dollars (and a smile) from Mel. A few dollars from every driver can amount to a substantial sum.

By signing Dawn's digital certificate, Alice enables Dawn to send it to anyone who has and trusts Alice's public key. If Alice's public key has been widely distributed, she can issue digital certificates to help *other* people deliver their public keys. Alice may collect a few dollars from Dawn. A few dollars from many Internet users can mount into a substantial sum. In Chapter 17 you'll see an example of businesses built on issuing digital certificates.

Note that Alice's issuance of Dawn's certificate does not give Alice any new way to deliver her own public key because anyone using Alice-issued digital certificates must already have Alice's public key.

The Needs of Digital Certificate Users

Retrieve, certify, notify

Digital certificate users have three basic needs.

1. They want to be able to retrieve digital certificates from online merchants and others with whom they want encrypted communications. In Figure 16-5(a), Bob retrieves some certificates.
2. They want to get their own public key certified and posted to Internet repositories, where others can retrieve it. In Figure 16-5(b), Bob gets his digital certificate verified and posts it to an Internet repository.

3. They want to know whether a digital certificate is still trustworthy—that is, whether the public key attached to a digital certificate is no longer considered secure. In Figure 16-5(c), Bob receives notification that he can no longer trust the public key attached to a digital certificate.

In Chapters 17 and 18 you'll see how two digital certificate systems (X.509 and PGP) satisfy these needs of certificate users.

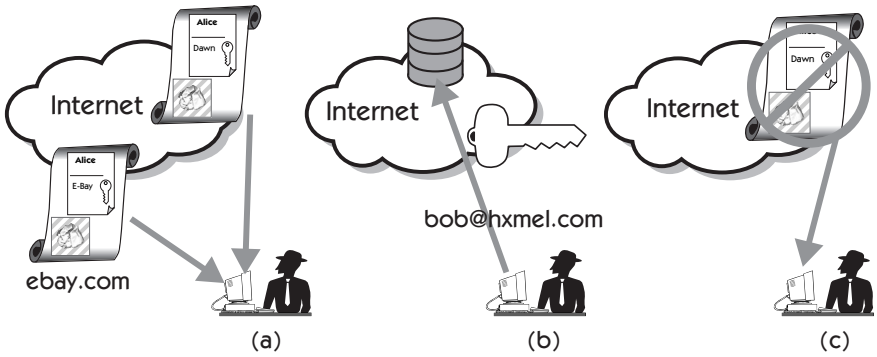


Figure 16-5 Bob needs (a) certificates from online merchants (such as e-Bay.com) and others (such as Alice); (b) someone who will certify Bob's public key and a way to post his digital certificate; and (c) notification if someone's digital certificate is no longer secure.

Getting Your First Public Key

How might Bob get Alice's public key? Professional digital certificate companies (VeriSign, ATT, and the like) negotiate with software sellers to have their public keys put in special digital certificates included in shrink-wrap software. For example, Internet browsers such as Internet Explorer and Netscape include dozens of so-called root digital certificates.[4]

4. See Chapter 17 for a discussion of root digital certificates and how to use them to get more digital certificates.

Certificates Included in Your Browser

Figure 16-6 shows how you can see the digital certificates put on H. X. Mel's disk by Microsoft's IE 5.0 when the browser was installed. In Chapter 17 and Part IV, you'll see how these digital certificates (and their public keys) are used.

The security of public keys distributed this way is based on the following assumption. If you got your browser from a CD, your certificates should be genuine because BlackHat would have to forge a CD with bogus copies of digital certificates, forge a copy of Internet Explorer with Microsoft's logo, stock the shelves of your local retailer, and so on. Although BlackHat's work is a little easier if you download your browser from an Internet site, it's still difficult. The list in Figure 16-7 shows some of the certificates you're probably already trusting.

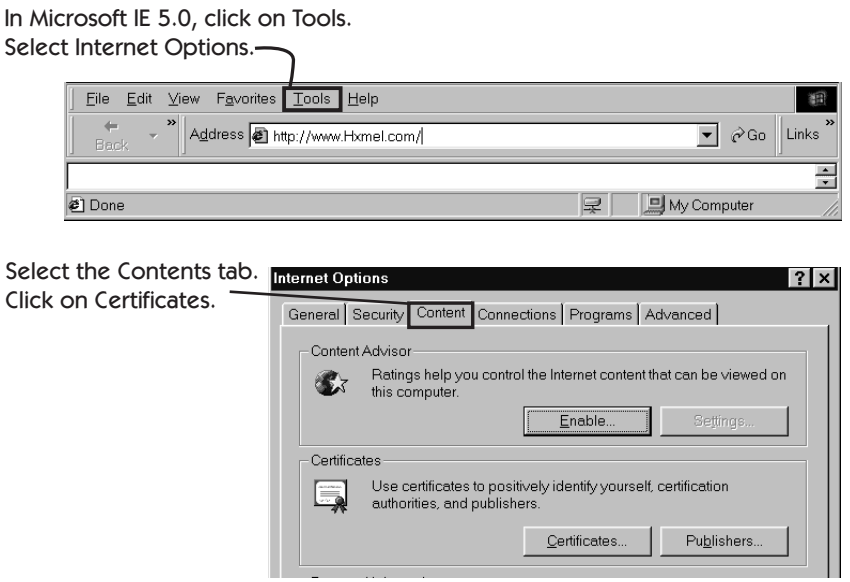


Figure 16-6 To see digital certificates packaged with your Microsoft IE 5.0, click on Tools, Internet Options, Content.

Review

Digital certificates are the preferred way to securely deliver public keys. A digital certificate is a specialized document signed by a trusted third party. The top part of a digital certificate contains plaintext identifying the issuer (signer), the subject (the entity whose public key is attached), the subject's public key, and the

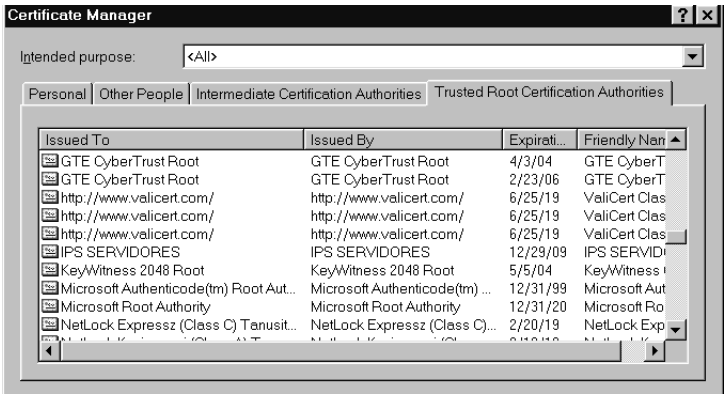


Figure 16-7 Some of the digital certificates Microsoft's IE 5.0 put on H. X. Mel's computer. (Netscape installs digital certificates, too.)

expiration date of the certificate. The bottom part of a digital certificate contains the issuer's signed hash of the top part.

The consumer of a digital certificate must have a trusted copy of the issuer's public key to correctly verify the certificate. The issuer, subject, and consumer are reviewed in Table 16-2.

Table 16-2 Digital certificate players.

Player	Action	Analogy
Issuer	Creates and signs certificate	DMV
Subject	Sends issuer his or her public key	Licensee
Consumer	Uses subject's public key (on certificate)	Car rental firm that confirms licensee ID