Chapter 15

Representing Identity

AEMELIA: Most mighty duke, behold a man much wrong'd.
All gather to see them.
ADRIANA: I see two husbands, or mine eyes deceive me!
DUKE SOLINUS: One of these men is Genius to the other;
And so of these, which is the natural man,
And which the spirit? Who deciphers them?
DROMIO OF SYRACUSE: I, sir, am Dromio: command him away.
DROMIO OF EPHESUS: I, sir, am Dromio: pray, let me stay.
— *The Comedy of Errors*, V, i, 332–338.

The theme of identity runs throughout humanity's experience, and computers are no exception. In computer science, an identity is the basis for assignment of privileges and is integral in the designation of a protection domain. This chapter discusses the many different types of identity and the contexts in which they arise. It begins with the identity of a principal on a system, first singly and then as defined by function. Designation of identity for certificates follows, as does identity on a network with respect to both individual processes and individual hosts. The chapter concludes with the notion of an anonymous user.

15.1 What Is Identity?

Identity is simply a computer's representation of an entity.

> **Definition 15–1.** A *principal* is a unique entity. An *identity* specifies a principal.

Authentication binds a principal to a representation of identity internal to the computer. Each system has its own way of expressing this representation, but all decisions of access and resource allocation assume that the binding is correct.

Identities are used for several purposes. The two main ones are for accountability and for access control. Accountability requires an identity that tracks principals across actions and changes of other identities, so that the principal taking any action can be unambiguously identified. Access control requires an identity that the access control mechanisms can use to determine if a specific access (or type of access) should be allowed.

Accountability is tied to logging and auditing. It requires an unambiguous identification of the principal involved. On many systems, this is not possible. Instead, the logged identity maps to a user account, to a group, or to a role.

Most systems base access rights on the identity of the principal executing the process. That is, all processes executed by user *bishop* have some set of rights. All processes executed by user *holly* have a set of rights that may differ from those that *bishop*'s processes have. However, a process may have fewer rights than the principal executing it, and in fact there are substantial reasons to reduce privileges. Chapter 16, "Access Control Mechanisms," discusses this topic in more depth.

15.2 Files and Objects

The identity of a file or other entity (here called an "object") depends on the system that contains the object.

Local systems identify objects by assigning names. The name may be intended for human use (such as a file name), for process use (such as a file descriptor or handle), or for kernel use (such as a file allocation table entry). Each name may have different semantics.

EXAMPLE: The UNIX operating system offers four different types of file names. The *device number* and *inode* uniquely identify a file. The inode contains file attribute information such as access control permissions and ownership information, and identifies the specific disk blocks that contain the file's data. Processes read files using a *file descriptor* that abstracts the inode into a representation that the process can read from, write to, and so forth. Once created, the file descriptor cannot be rebound to a different file. Processes (and users) can also use *file names* that identify files by describing their positions in the file hierarchy. UNIX file names may be *absolute path names* that describe the locations of files with respect to the root of the UNIX file hierarchy, or *relative path names* that describe the locations of files with respect to the directory in which the current process is executing.

The semantics of the names differ in important ways. Most critically, when a process or user operates on a file, the kernel maps the file name to an inode using

EXAMPLE: On the DG/UX system, a multilevel secure system, system administration privileges belong to the *sysadmin* role, not the *root* user [2186]. That user's rights are restricted. The *sysuser* user can assume the *sysadmin* role to administer the host, or the *netadmin* role to administer the network. Several such roles are defined.

15.5 Naming and Certificates

Chapter 11 described certificates as a mechanism for binding cryptographic keys to identifiers. The identifier corresponds to a principal; it must uniquely identify the principal to avoid confusion.

Suppose the principals are people. The identifiers cannot be names, because many different people may have the same name. (How many people named "John Smith" or "Pierre LeBlanc" are there?) The identifiers must include ancillary information to distinguish the "Matt Bishop" who teaches at UC Davis from a different person named "Matt Bishop" who works at Microsoft Corporation.

EXAMPLE: The X.509v4 public-key certificates use identifiers called *Distinguished Names* [2032, 2176]. A Distinguished Name identifies a principal. It consists of a series of fields, each with a key and a value. When written as strings, the fields are separated by "/" and the key and value by "=".[3] To use our earlier example, the "Matt Bishop" who teaches at the University of California might have the Distinguished Name

```
/O=University of California/OU=Davis campus/
OU=Department of Computer Science/CN=Matt Bishop/
```

(where the key "O" means organization, "OU" means organizational unit, and "CN" means common name) and the "Matt Bishop" who works at Microsoft might have the Distinguished Name

```
/O=Microsoft Corporation/OU=Quality Assurance/
CN=Matt Bishop/
```

Although the names are the same, the individuals, and hence the Distinguished Names, are different.

Certification authorities (CAs) vouch, at some level, for the identity of the principal to which the certificate is issued. Every CA has two policies controlling how it issues certificates.

[3] When compiled into a binary format, in many cases the key is implied by the data structure.

Definition 15–2. A *CA authentication policy* describes the level of authentication required to identify the principal to whom the certificate is to be issued.

Definition 15–3. A *CA issuance policy* describes the principals to whom the CA will issue certificates.

The difference between these two policies is that the first simply establishes the level of proof of identity needed for the CA to accept the principal's claim of identity whereas the second answers the question, "Given the identity of the principal, will the CA issue a certificate?"

EXAMPLE: In 1996, Verisign Corporation ran several CAs. Each had its own policies of issuance and authentication for certificates [746].

Individuals obtained certificates (called "Digital IDs") from one of three CAs.[4] The class 1 CA authenticated the individual's electronic mail address. This CA provided a certificate for sending and receiving electronic mail securely. The class 2 CA required that the individual supply his real name and address, which was verified through an online database. This CA provided a certificate suitable for online purchasing and was (roughly) equivalent to the level of authentication for a credit card. The class 3 CA required a background check from an investigative service. The certificate from this CA provided a higher level of assurance of identity than the other two certificates. All three CAs had the same issuance policy: that certificates were issued to individuals. A fourth CA provided certificates to web servers. This CA had the same issuance policy as the class 3 CA. Consumers who did business with the website had a high degree of assurance that the website was whom it claimed to be.

In many cases, a CA delegates to a third party, the *registration authority* (RA), the checking of data to be put into the certificate, such as identity. When the RA determines that the CA's requirements for issuing a certificate are met, the RA instructs the CA to issue the certificate. CAs can issue certificates to other organizations. The hierarchical certificate-based key management architecture demonstrates how such an organization can lead to a simple hierarchical structure of policies [1033].

EXAMPLE: The infrastructure organizes CAs into a hierarchical, tree-based structure. Each node in the tree corresponds to a CA. Consider a node that is the root of a subtree. The CAs under that root are constrained by the policies of that root; the subordinate nodes may issue certificates with more restrictive policies, but not with more liberal policies.

The root of the tree is the *Internet Policy Registration Authority* (IPRA). It sets policies that all subordinate CAs must follow, and it certifies other CAs

[4] Actually, a single CA issued multiple types of certificates. Conceptually, the single organization is acting as though it were multiple CAs.

called *policy certification authorities* (PCAs). Each PCA has its own issuance and authentication policies, but those policies must not conflict with the policies set by the IPRA. The PCAs issue certificates to ordinary CAs, which can then issue certificates to organizations or individuals. The IPRA and PCAs do not issue certificates to individuals or organizations. All CAs, PCAs, and the IPRA have unique Distinguished Names.

The elegance of this approach is twofold. Because all PCA policies are public, on receiving a certificate one can determine how much trust to place in the identity in the certificate (authentication policy) as well as the requirements that the holder had to meet to have the certificate issued (issuance policy).

To understand how this works, suppose the University of Valmont wishes to establish a CA for both students and staff. The requirements for certification for these groups are different. Students must present valid registration cards to obtain certificates. These certificates would be considered low-assurance certificates (because of the nature of the registration process) and so would be signed using the university's low-assurance certificate. This certificate, in turn, is signed by a PCA that requires its subordinate CAs to make a good-faith effort to verify the identities of those to whom it issues certificates. But the university requires staff members to present proof of employment and fingerprints, which are compared with the fingerprints obtained when each employee was hired. This provides a high level of assurance of identity, and so the University of Valmont signs these certificates with its high-assurance certificate, obtained from a different PCA that requires the use of biometrics for verification of identity.

The certificates for student John and professor Marsha are both signed by the same organization, but they are signed using different cryptographic keys. John's certificate is signed by the key corresponding to a low-assurance certificate (because the first PCA signed it), and Marsha's certificate is signed by the key corresponding to a high-assurance certificate (because the second PCA signed it). By checking the policies of each of the PCAs, and (possibly) the CA, the recipient of one of these certificates can tell what the policies of issuance and assurance are. (A potential conflict arises because the CA has the same Distinguished Name for two different types of policies. Section 15.5.1 discusses this topic further.)

As another example of how the certificates encode policy, note that Marsha's certificate implicitly identifies her as being affiliated with the University of Valmont. This type of certificate is called an *organizational certificate*. The Internet infrastructure defines a second type of certificate, a *residential certificate*, that identifies the principal's residential address. Marsha has one of these, issued by the post office, and identifying her as a citizen residing in the city of Valmont:

```
/C=US/SP=Louisiana/L=Valmont/PA=27 Russell Blvd./
  CN=Marsha/
```

Here, "C" is the country code, "SP" is the province or state name, "L" is the locality (city, town, or village), and "PA" is the street address.

The principals need not be people or organizations; they can be roles.

EXAMPLE: A company wishes to have its comptroller authorized to digitally sign documents. To this end, it issues a certificate to the role:

```
/O=Hodgepodge Corporation/
OU=Office of Big Money/RN=Comptroller/
```

Even if the current comptroller leaves and a new one is hired, the same certificate can be used. Here, "Comptroller" is a role (and the use of the "RN" key, for "Role Name," reflects this).

The identifiers in a certificate need not be formal Distinguished Names. The certificates used with PGP, for example, allow the subject to provide any identifier he or she wishes. The convention is to use a name and an electronic mail address [340], but this permits a high level of ambiguity, especially when mail addresses change frequently. This leads directly to conflicts; how can a CA ensure that the certificate it issues does not conflict with another?

15.5.1 Conflicts

Both X.509 and PGP are silent about certificate conflicts. They assume that the CAs will prevent conflicts. The CA's Distinguished Name is in the certificate, so if no two CAs have the same Distinguished Name and each CA requires that principals be identified uniquely among the set of principals certified by that CA, no conflicts will arise.

The PEM certification hierarchy uses the same approach: the IPRA requires that each PCA have a unique Distinguished Name, and no PCA may certify two CAs with the same Distinguished Name. But in practice, there may be conflicts. For example, suppose John A. Smith and John B. Smith, Jr. both live at the same address. John B. Smith, Jr. applies for a certificate, based on his residence, from the post office, which issues one:

```
/C=US/SP=Maine/L=Portland/PA=1 First Ave./
CN=John Smith/
```

His father, John A. Smith, applies to the Quick Certificate Company for a residential certificate. His Distinguished Name would be identical to his son's, but the Quick Certificate Company would have no way to know this because there is no central repository of certificates. The PEM infrastructure deals with this problem in two ways. First, it requires that all CA Distinguished Names be "superior" to the Distinguished Name of the principal.

EXAMPLE: In the University of Valmont case, if Marsha's certificate were

```
/C=US/O=University of Valmont/
OU=Computer Science Department/CN=Marsha/
```

then the University of Valmont's CA would be either

```
/C=US/O=University of Valmont/
OU=Computer Science Department/
```

if the issuer were the Computer Science Department, or

```
/C=US/O=University of Valmont/
```

if the issuer were the university itself. The University of New York, with a Distinguished Name of

```
/C=US/O=University of New York/
```

could not issue a certificate to Marsha as an employee of the University of Valmont, because its Distinguished Name is not superior to that of Marsha.

This works for organizational certificates, since each organization can be its own CA, or can empower subordinate units to be their own CAs. However, it is unrealistic to expect that only one entity will issue residential certificates. This immediately leads to a conflict.

EXAMPLE: Suppose Heidi Smith's daughter is named Heidi O. Smith (the mother has no middle name). Heidi O. Smith needs a residential certificate to apply for college. She goes to the post office and obtains one with the following Distinguished Name:

```
/C=US/SP=California/L=San Rafael/
PA=1 Forbes Ave./CN=Heidi Smith/
```

Because CA Distinguished Names are superior to those of the principals, the post office must have a Distinguished Name that is one of the following:

```
/C=US/
/C=US/SP=California/
/C=US/SP=California/L=San Rafael/
/C=US/SP=California/L=San Rafael/PA=1 Forbes Ave./
```

Heidi's mother must fill out a financial aid package and needs a certificate to sign it. Because the line at the post office is too long, she goes to Quick and Cheap Certs, Inc. and obtains a residential certificate from them:

```
/C=US/SP=California/L=San Rafael/PA=1 Forbes Ave.
  /CN=Heidi Smith/
```

But by the same rule, the Distinguished Name that Quick and Cheap Certs, Inc. uses in the certificate could be the same name as that of the post office.

The PEM infrastructure contains an explicit exception that allows multiple residential CAs to have the same Distinguished Name. But this issue also arises when the same CA wishes to issue certificates under two different policies, and hence under two different PCAs. Because the CA uses the same Distinguished Name for all its certificates, how does one determine under which policy a certificate was issued?

EXAMPLE: John's certificate was issued under a low-assurance policy. He uses it to sign a letter to Eve. When Eve gets John's certificate, she validates it. She cannot determine whether the high-assurance authentication policy or the low-assurance authentication policy was used.

The PEM infrastructure handles these conflicts with a Distinguished Name conflict detection database. Before a PCA may issue a certificate to a CA, it must determine if a conflict exists. It sends a query to the database containing the following information:

- A hash value computed on a canonical representation of the CA's Distinguished Name
- The CA's public key in the certificate
- The Distinguished Name of the PCA

If the first two fields conflict with any other entry in the database, the IPRA returns the conflicting entry. (The two PCAs must then resolve the conflict.) Otherwise, the information is entered into a new record and a timestamp is added.

This mechanism does not ensure uniqueness of Distinguished Names. It *does* ensure uniqueness of the pair (Distinguished Name, public key), and therein lies the answer to the above-mentioned conflicts. In the residential certificate example, the post office and Quick and Cheap Certs, Inc. have different public keys, so the CA for the certificates could be determined at validation time. In the University of Valmont example, the different public keys used to sign the certificate would indicate under which policy the university issued the certificate.

15.5.2 The Meaning of the Identity

The authentication policy defines the way in which principals prove their identities. Each CA has its own requirements (although they may be constrained by contractual requirements, such as with PCAs). All rely on nonelectronic proofs of identity, such as biometrics (fingerprints), documents (driver's license, passports), or personal knowledge. If any of these means can be compromised, the CA may issue the certificate in good faith to the wrong person.

This hearkens back to the issue of trust. Ignoring the trust required for cryptography to work, the certificate is the binding of an *external* identity to a cryptographic key and a Distinguished Name. If the issuer can be fooled, all who rely on that certificate may also be fooled.

With the erosion of privacy in many societies comes the need for anonymity. This conflicts with the notion of a certificate binding an identity to a Distinguished Name and a public key. The conflict arises when the anonymous principal needs to send a set of integrity-checked, confidential electronic messages to a recipient and to ensure that the recipient realizes that all of the messages have come from the same source (but the recipient cannot know what the source is).

EXAMPLE: A government plans to require all citizens with a specific gene to register, because anecdotal evidence suggests that people with that gene commit crimes slightly more often than other people. The government plans to make the law without publicity, because aside from the civil liberties issues, there is no reputable scientific evidence to back up the belief. A government employee decides to alert the media. She realizes that the government will promptly deny the plan and change its approach to getting the law passed. She feels that she will be fired (or charged with a crime) if the government determines who she is, and would therefore be unable to reveal any changes in the plan. So she decides to publicize the plans anonymously.

Anonymous, or *persona*, certificates supply the requisite anonymity. A CA issues a persona certificate under a policy that makes the Distinguished Name of the principal meaningless. For example, a persona certificate with a principal Distinguished Name of

```
/C=US/O=House of Representatives/CN=Jessica Rabbit/
```

does not imply that the certificate was issued to someone named Jessica Rabbit. PGP certificates can have any name to identify the principal, and can innately provide anonymity in this sense.

EXAMPLE: Continuing, our heroine obtains a persona certificate and sends a copy of the government's plan to the media, using electronic mail, as described above. The government denies the plan and secretly changes its strategy. It has some employees leak verifiably false information so that if the original whistleblower sends another message, it is less likely to be believed. But she does, and she uses the same certificate to authenticate the message. Now the media can check that the two messages came from the same source (or at least were signed with the same certificate), whereas the false messages were signed by different certificates.

15.5.3 Trust

The goal of certificates is to bind the correct identity to the public key. When a user obtains a certificate, the issuer of that certificate is vouching, to some degree of certainty, that the identity corresponds to the principal owning the public key. The critical question is the degree of that assurance.

X.509v4, and the PEM certification hierarchy, define the degree of certainty in the policy of the CA that issues the certificate. If a CA requires a passport as identification, then the degree of certainty is high; if it requires an unsworn statement of identity, the degree of certainty is low. But even high-assurance CAs can be fooled. In the case of the passport, passports can be stolen or forged. So the level of trust in an identity is not quantifiable. Rather, it is an estimate based on the policy of the CA, the rigor with which that policy is followed, and the assumptions that the policy makes.

EXAMPLE: Consider the CA that requires a passport to issue a certificate. The certificate will have as its DN the name in the passport, the name of the country issuing the passport, and the passport number. There are several points of trust in this policy. First, the CA assumes that the passport is not forged and that the name has not been altered. Second, the CA assumes that the country issuing the passport issued it to the person named in the passport. Third, the CA assumes that the individual presenting the passport is the individual to whom the passport was issued.[5] Fourth, the users of the certificate assume that the CA has actually checked the passport and the individual using the passport to obtain a certificate.

PGP certificates include a series of signature fields (see Section 11.4.2.2), each of which contains a level of trust of the identity in the certificate.[6] The OpenPGP specification defines four levels [340]:

- Generic certification of a user name and a public key packet; this makes no assertions about the correctness of the name.
- Persona certification of a user name and a public key; the signer has done no verification that the user name correctly identifies the principal.
- Casual certification of a user name and a public key; the signer has done some verification that the user name correctly identifies the principal.
- Positive certification of a user name and a public key; the signer has done substantial verification that the user name correctly identifies the principal.

Even here, though, the trust is not quantifiable. What exactly do "some verification" and "substantial verification" mean? The OpenPGP specification does not define them, preferring to leave their definitions to the signer, so the same terms can imply different levels of assurance to different signers.

EXAMPLE: At a university, "substantial verification" may mean having a student identification card and a matching driver's license. The university's CA would sign the student's PGP certificate with level 4 trust. But at a high-security government installation that requires background checks before certificates are signed, the university's "substantial verification" would most likely be considered level 2 trust, "no verification."

[5] Passport photographs are notoriously poor, making visual identification questionable unless conditions are optimal.

[6] This is encoded in the signature type field of the signature.

The point is that knowing the policy, or the trust level with which the certificate is signed, is not enough to evaluate how likely it is that the identity identifies the correct principal. Knowing how the CA or signer interprets the policy and enforces its requirements is also required.

EXAMPLE: On March 22, 2001, Verisign, Inc. and Microsoft Corporation [2247] reported that Verisign had issued two certificates to someone claiming to be a representative of Microsoft Corporation. The individual was not. Both companies took steps to cancel the certificates and prevent them from being used.

If the CA delegates the validation of identity (and other information) to a registration authority (RA), then the CA trusts that the RA abides by the policy of the CA, and is not otherwise compromised. The delegation means that the CA is still ultimately responsible for the certificates it issues.

EXAMPLE: A user account on a registration authority for the certificate authority Comodo was compromised. The attacker used the RA to generate requests for certificates for Google, Yahoo, Skype, and other major Internet sites. The compromise was detected within hours, and the fraudulently issued certificates were immediately revoked [446, 851]. The compromised account was deactivated immediately.

15.6 Identity on the Web

Certificates are not ubiquitous on the Internet. Several other means attach identity to information, even though the binding may be very transient.

The Internet requires every host to have an address. The address may be fixed or may change, and without cryptography the binding is weak. Many servers send information about the state of the client's interaction, so that when the client reconnects, the server can resume the transaction or glean information about previous transactions.

15.6.1 Host Identity

Host identity is intimately bound to networking. A host not connected to *any* network can have any name, because the name is used only locally. A host connected to a network can have many names or one name, depending on how the interface to the network is structured and the context in which the name is used.

The ISO/OSI model [1859] provides a context for the issue of naming. Figure 12–6 shows the layers of the ISO/OSI model. Each host, conceptually, has a principal at each layer that communicates with a peer on other hosts. These principals communicate with principals at the same layer on other hosts. Each