Chapter 7
Information Security

Not a day goes by without news of a computer security failure: data leaked or breached, files encrypted and held for ransom, financial or other credentials cracked. These failures are, or should be, preventable.

Figure 7-1
Blaise de Vigenère, French diplomat, cryptographer, and alchemist
© Trustees of the British Museum

Everyone who works in the information technology field or works with information technology to accomplish their jobs has some responsibility for security. In the United States and elsewhere, Federal and state laws may impose obligations with regard to information security. This chapter provides an introduction to the concepts, terminology, principles, and practices of information security. It will help prepare you for one or more courses in information security, or to understand the concepts as you use information technology.

The foundation of an information security program is an information security policy that describes which information assets are to be protected and the constraints on the use of those assets.

7.1 What Does "Secure" Mean?

A system is **secure** if it "does what it is intended to do *and nothing else*." (Pfleeger & Pfleeger, Security in Computing Fourth Edition, 2006) [51]

51 Simson Garfinkel and Eugene Spafford wrote much the same thing in *Practical UNIX Security* in 1991: "A computer is secure if you can depend on it and its software to behave as you expect."

That captures a big idea in a very few words. Every security failure is the result of a computer system doing something other than what was intended.

Through unpatched software and configuration errors, Kennesaw State University's Center for Election Systems allowed some 6.5 million voter records to be available on the Internet although the records were intended to be confidential. [52] (Torres, 2017)

Sometimes, failure of intentionality is more subtle. Microsoft designed the Windows Metafile graphics format to include executable code that was run when an image was viewed. That was for Windows 3.0, released in 1990, when there was little thought of connecting personal computers to the Internet. In late December, 2004, malicious actors discovered that they could execute arbitrary code on victims' computers if the victim simply viewed a specially-crafted image with a Web browser. Microsoft absolutely intended that displaying WMF images would cause code to be executed. What they did not intend was for malicious actors to exploit this capability to do damage. [53]

The Windows Metafile example shows that the intended use of computer systems must be specified carefully and in depth. That is done through the establishment of security policies, as discussed below.

Often when people think of information security, they think of malicious actors, or "hackers." As the examples above show, security problems are at least as likely to occur from error. A well-designed information security program protects against error, environmental failure such as a power outage, and other inadvertent causes as well as malicious attack.

7.2 Properties of a Secure System

There are three principal properties of a secure information system: confidentiality, integrity, and availability. The essence of security is protecting these

52 The Center for Election Systems was formerly operated by KSU under a contract with the Georgia state government.

53 Microsoft released a patch for the vulnerability on January 6, 2005.

three properties. Of course, there's a lot more to security than three things, but the tools focus on protecting one or more of those three properties.

7.2.1 Confidentiality

Confidentiality is the condition that information not be revealed to unauthorized parties. An organization's information security policy must identify what information is confidential and who is authorized to have access to that information.

Confidentiality and privacy

Privacy can mean freedom from surveillance or the right to control information about oneself. The first definition addresses things like license plate readers and facial recognition systems. The second definition addresses things like financial, student, or patient databases. It can also address the data collected by surveillance systems.

Organizations may have a legal, contractual, or moral duty to defend the privacy of those whose information they collect and store. This duty is fulfilled by maintaining the confidentiality of such information, but privacy itself is not addressed by information security. Privacy is addressed by law, contract, and moral obligation.

7.2.2 Integrity

Integrity is a measure of how much we can trust data, software, or systems. Since integrity is a measure of "how much," it is much harder to quantify than confidentiality.

Data integrity is the state that information agrees with the source from which it was derived and has not been altered or destroyed in an unauthorized matter. As with confidentiality, the information security policy describes how and by whom data may be changed or destroyed.

Origin integrity tells how much we trust that information came from the source it is supposed to have come from.

Program integrity tells how much we trust a computer program to do what is intended and how certain we are that it hasn't been modified in an unauthorized manner.

7.2.3 Availability

Availability is the property that information systems are available to authorized users when and where needed. Availability is usually defined in terms of "quality of service," in which authorized users are expected to receive a specific level of service stated in terms of metrics like uptime and response time.

7.2.4 Properties, States, and Controls: the McCumber Model

John McCumber (1991) proposed a model for information security that relates the three properties – confidentiality, integrity, and availability – to the states of information and the controls that can be applied to protect the three properties. [54] The premise of the McCumber Model is that to develop an effective information security program, one must consider the properties of information security, the states of information, and the controls available, and not focus on one or a few items.

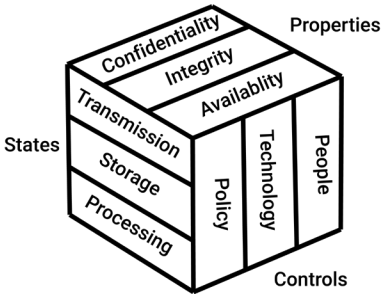

Figure 7-2
The McCumber Model

The states of information are transmission, storage, and processing. The McCumber Model shows that the properties of confidentiality, integrity, and availability must be protected for all three states of information.

On the last face of the cube are the types of controls available: policy, technology, and people. The McCumber Model shows that all three controls types

54 The original illustration had a 3 × 3 grid on each visible face of the cube. "People" was labeled "Education, training, awareness," and the three sets of labels were on different faces.

should be applied to all three properties. The McCumber Model helps us think about information security in three dimensions.

The "people" part to the controls dimension refers to those who use or are responsible for the organization's information assets. Assuring that everyone understands their responsibilities with respect to information security and also has the knowledge to fulfill them is an important part of an information security program.

7.2.5 Other "Properties"

Some authors discuss "properties" in addition to confidentiality, integrity, and availability. Among them are non-repudiation, authentication, and auditability.

Non-repudiation means that someone cannot effectively deny supplying data, sending a document, or taking an action. Non-repudiation is a consequence of high origin integrity. If one has high confidence in the origin of a document or other information, it is difficult or impossible to deny having supplied the document or information.

Authentication and auditability are controls rather than properties. An authentication mechanism requires a subject to prove identity before being granted privileges on a system. Auditability is a control on integrity. If a system maintains a secure record of all changes, it becomes possible to answer the question, "Was this changed, and if so, when and by whom?"

7.3 Goals of an Information Security Program

An information security program must have three **goals**: **prevention**, **detection**, and **response**.

Clearly, the first goal of an information security program is to prevent disclosure, alteration, or denial of the information resources being secured.

News reports of information security failures often include information that an intrusion or other attack started months before it was discovered. We must accept that not all security incidents will be preventable. An effective program will have mechanisms in place to detect security failures as they occur or soon after.

Once a security incident has been detected, it is too late to plan a response. *Ad hoc* responses lead to mistakes. Instead, an effective information security program plans response and recovery in advance of incidents occurring, and tests the plans.

The three goals are addressed through policy, technical controls, and procedures, as shown by the McCumber model. An effective program provides for **defense in depth**. That means multiple layers of defense, the failure of any one of which will not compromise the asset being protected. For example, firewalls and anti-malware software can be deployed to prevent files from being encrypted by ransom-ware. An effective system of backups, stored where they are safe from infection, provides a recovery mechanism if prevention fails.

7.4 Policy, Controls, and Procedures

Every information security program must be grounded in an information security policy that describes the intended use of the system, including what information assets are to be protected and the constraints on the use of those assets. The *definition* of a secure system is one that does not violate the organization's security policy. That means the policy must fully and correctly capture the security requirements of the organization. The policy describes the "what" of information security and should be technology-neutral.

For example, a policy might state, "Student grades are to be viewable by the student, by the student's parents or guardians if the student is under 18, and by university employees or agents with a legitimate educational purpose to have access to grades." [55]

55 This isn't enough for a comprehensive policy. The policy must also describe who can record grades and who can change grades after they have been recorded.

Procedures describe how people should implement policies and might be connected with how technical controls are implemented and used. Given the example above, those employees or agents who should have access to grades might have a value set in their university login records. There would be specific procedures for setting and removing that value. Procedures, along with technical controls, describe the "how" of information security.

Technical controls include things like access control mechanisms for logging in, firewalls, anti-malware software, and access control lists that provide fine-grained control of access to information technology assets.

Information security policy should be developed based on a risk assessment, as described in Section 7.5. Policies and technical controls can be developed from industry best practices and adjusted for the organization's own assessment of assets and risks. In evaluating best practices, it is important to be sure those adopted meet the requirements of the organization.

One of the best lists of requirements for procedures and controls was published nearly 50 years ago. (Saltzer & Schroeder, 1973) Here is their list, with brief comments by this author:

- *Economy of mechanism:* Simple controls and simple procedures work best.
- *Fail-safe defaults:* Policies, procedures and controls list those things that *should* be allowed, and controls must deny everything else. A mistake will cause access to fail, and the subject whose access fails will complain. Trying to list those things that should not be allowed is a recipe for disaster. Marcus Ranum called that "enumerating badness." (Ranum, 2005) [56]
- *Complete mediation:* Control mechanisms must check every access to an information asset.
- *Open design:* A protection mechanism must work even when an attacker knows the design of the mechanism. This was first stated as Kerckhoffs' Principle by Dutch cryptographer Auguste Kerckhoffs in the 19th century. [57] Depending on keeping the design of a security mechanism secret is called

56 Dr. Ferrol Sams described enumerating badness this way: "He's not a bad boy, he minds well; I just can't think of enough things to tell him not to do." (Sams, 1982)

57 Claude Shannon restated this in 1949 as "assume the enemy knows the system."

"security by obscurity" and it fails as soon as a malicious actor learns the mechanism.

- *Separation of privilege:* A mechanism that requires two actions is safer than a mechanism requiring only one. This is a kind of defense in depth. An example is two-factor authentication, a login that requires both a password and a security key. Compromise of either one alone will not allow access.
- *Least privilege:* Users of information assets should have access to those resources required to do their jobs and nothing more.
- *Least common mechanism:* Controls protecting multiple assets must meet the requirements of the asset needing the most protection. The implication is that assets needing different levels of protection should have separate, independent controls.
- *Psychological acceptability:* Protection controls should be so easy and so natural that they are routinely used correctly.

7.5 Assets, Risks, and Risk Management

The existence and use of information technology assets creates risks to those assets. Consider an automobile as an example. A car parked in a secure garage can be crushed and destroyed by a falling tree. It's rare, but it can and does happen. Driving the car increases the risk by exposing it to threats like collision. The same thing is true of information technology assets. An information security program involves identifying the organization's information technology assets and the threats to those assets, then managing the risks present.

7.5.1 Identifying Assets and Threats

The threats to information assets are failures of the three properties of a secure system. **Disclosure** is the threat that compromises confidentiality, **alteration** compromises integrity, and **denial** compromises availability.

An organization's information technology assets can be broadly classified as hardware, software, data, documentation, people, and infrastructure.

People from all parts of the organization should be involved in building the list of assets because there are likely to be information assets unknown to the information technology department. The inventory of assets should be reviewed each time an information system is put into production or decommissioned and also at regular intervals. [58] Asset identification should include the asset name, use within the organization, and asset owner. The asset owner is the person or position in the organization authorized to set policy for the asset. Users of the asset must also be identified. If there is an asset custodian, often the information technology department, identify the custodian. Depending on the type of asset, one may need to collect things like manufacturer name and serial number.

When the asset list has been developed, and each time it is reviewed, a list of threats to the asset should be produced or updated. As with asset identification, this works best if people from throughout the organization participate.

7.5.2 Risk Management

A **risk** is the probability that a threat is realized and damages the asset. Risk management is the process of identifying and controlling the risks facing an organization. In the insurance industry, risks are characterized by annualized loss expectation, or ALE. The ALE is the cost if a threat is realized multiplied by the number of times the threat is expected to be realized each year. The multiplier will be a fraction for events expected to happen less frequently than once a year. To continue with the automobile example, your insurer does not know whether you will be in an accident next year, but they know to several decimal places how many people of similar age, sex, driving record, and type of car have been in accidents in the past. They know how much it cost to pay the resulting claims. They use that information to compute an ALE for you and use that to price your insurance.

58 In an earlier career, the author conducted an annual asset review as part of the preparation for each year's financial audit.

It is important to remember that the ALE is applicable only over numbers of similar assets. For any given asset, the annual cost is zero if a threat is not realized and is the full economic cost if a threat is realized.

It is seldom possible to be this precise in an information security program because there's not sufficient data. It may be necessary to characterize the probability of a risk being realized as low, medium or high. It is worthwhile to try to put a dollar range on the cost of a risk being realized.

Risks are classified along several dimensions:

- Direct or indirect risks [59]
- Risks to information
- Environmental risks
- Physical risks.

With risks classified, it is possible to design policies, technical controls, and people controls like procedures to reduce the risks.

Figure 7-3 provides guidance on which controls to design and implement first. Clearly, one should address those risks with both a high probability of occurrence and a high cost if the threat is realized. However, Figure 7-3 does not account for the cost of controls. It may be advantageous to implement some less important controls early in the process if they are easy and cheap. It may also be necessary to forego some controls entirely if their cost or technical complexity exceeds what the organization can support.

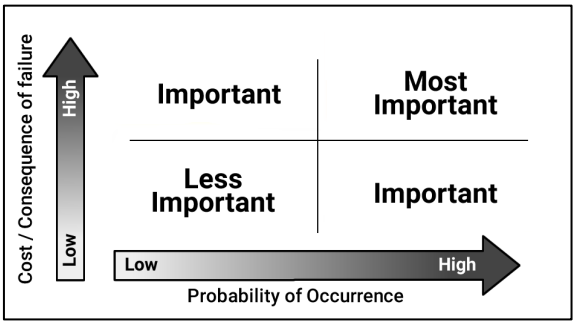

Figure 7-3
Priority of designing and implementing controls.

59 Direct risks compromise the asset itself if realized; indirect risks are things like business interruption, damage to reputation, and legal liability.

As controls are implemented, the level of risk will decrease, but it will never reach zero. The remaining risk is called **residual risk**. The purpose of an information security program is to reduce residual risk to a level compatible with the organization's resources and tolerance for risk.

7.6 Identification, Authentication, and Access Control

In this discussion, we use **principal** to refer to a unique entity, perhaps a person or an information system. An **identifier** specifies a principal with respect to an information system. Each of you probably has several identifiers, things like logins to various systems. Your legal name is an identifier.

Depending on the security requirements of a system as expressed in policy, a process of **identity proofing** may be necessary. Identity proofing is assuring that the claimed identity of a principal is correct. The documents necessary to get a driver's license are a form of identity proofing.

Authentication is the binding of a principal to a specific identifier internal to the system. Authentication is crucial to security because all decisions on access to a system's resources must assume that the binding is correct.

A simple authentication example is a system with login based on user ID and password. The login prompt asks "Who are you?" and the correct reply is an identifier, the user ID. Often the identifier is widely-known or easy to guess. [60] Then the system asks, "Prove it!" and the correct response is a password matching the identifier. In this scenario, the password is assumed to be known only to the principal and not able to be obtained or guessed by others. If the password is correct, the assumption for security decisions is that the principal associated with the identifier is the one using the system, and access decisions are based on that assumption.

A strong system of authentication and access control protects confidentiality by assuring that only those authorized by policy have access to data. It protects

60 For many years my user ID at my university was "bbrown." For several other systems, it is my email address.

integrity by assuring that only those authorized by policy to change data are able to do so.

Authenticating factors

There are three [61] factors that can be used for authentication: something one knows, something one has, or something one is.

Something one knows: typically, this is a password, but might be a pattern traced on a touch screen or another challenge, like an algorithm. A simple example of a challenge might be that the system presents a random integer and the principal must add 27 to it.

Something one has: In the 21st century, this is often a security key like the YubiKey. It could be as simple as a metal key used to unlock doors or start automobiles. It could also be a mobile phone to which the system sends a text message that must be entered to complete the login. [62]

Something one is: This is properly called biometric authentication. [63] Typical uses are recognition of faces or fingerprints, but there are systems based on other characteristics, including hand geometry and scans of iris or retina. A handwritten signature is a form of behavioral biometric authentication that can be validated automatically. (Rosso, Ospina, & Frery, 2016)

Each of these factors has strengths and weaknesses, a discussion of which is beyond the scope of this chapter.

Multi-factor authentication

For much of the history of remote access to information systems, the standard for identification and authentication was user ID and password. The problem is that people can't remember multiple strong passwords, so they choose weak

61 Some authors place things like signatures and unlock patterns in their own category, something one does, and add somewhere one is for authentication by IP address, geolocation, etc.

62 If you are using a mobile phone in this way, a Web search for "SIM hijacking" will be enlightening.

63 Not to be confused with biometric identification, a different and much harder problem. Biometric authentication asks "Does this face or fingerprint belong to Alice?" Biometric identification asks, "To whom does this face or fingerprint belong?"

passwords or reuse passwords across multiple systems, or both. Weak passwords are easily compromised. If passwords are reused, a compromised password on one system compromises all of them.

By requiring two or more of the factors – something you know, something you have, something you are – a compromise of only one of them will not be sufficient to gain access. This is another example of defense in depth.

Access control and auditability

Once authentication is complete, there exists a binding of the principal permitted to use the system and the identifier, or user ID. Except for the very simplest systems, where authority on the system is all or nothing, further decisions are necessary to determine what the principal may do. This is called **authorization** or **access management.**

Many modern authorization systems use **access control lists** (ACLs). Each object, such as a file, has a list of those identifiers (principals) allowed to access the object, and what access is allowed. For example, the Windows operating system includes access permissions for reading, writing, modifying, and executing files. Other types of objects, like disks or printers, have different sets of permissions suitable to the kind of object being protected.

The UNIX operating system and its look-alike, Linux, use a different model in which there are three permissions: read, write, and execute, and three levels of access: owner, group, and everyone. It is possible to exercise very fine-grained control using the UNIX model. Extensions to UNIX and Linux provide for access control lists.

Designers of access control systems must remember Saltzer and Schroeder's principle of fail-safe defaults. Access should be denied unless explicitly granted. Those who administer access control systems must remember the principle of least privilege. Give each principal the rights necessary to perform that principal's permitted functions and no more.

Access control systems in both UNIX / Linux and Windows provide for logging of changes in access permissions and of access to objects. Enabling logging provides auditability; it is possible to know what happened, when it happened, and who did it. Auditability, in turn provides accountability for those who use the information system.

7.7 Cryptography

Cryptography [64] refers to the use of **ciphers** [65] to obscure the meaning of text or other data by scrambling the bits of a document, message, or other data so that the content is impossible to read without reversing the encryption process. A cryptographic system consists of a long and random collection of bits called the key and algorithms for encrypting and decrypting. The algorithms are assumed to be known by an adversary. The key must be kept secret. The encrypting process combines the key with the data, called plain text, to produce encrypted data, called the cipher text. Encrypting data protects confidentiality.

Cryptography has been used to protect secrets since at least 1500 BCE. Around 1580, French cryptographer Blaise de Vigenère (Figure 7-1) improved an earlier cipher, now called the Vigenère cipher, and invented another. At the time, the Vigenère cipher was called *le chiffre indéchiffrable*. [66] The Vigenère cipher withstood attack for nearly 300 years. In 1863, Austrian military officer Friedrich Kasiski published a method of attacking the Vigenère cipher by deducing the length of the key.

Cryptographic techniques can be used to protect data integrity by revealing tampering and to protect origin integrity through digital signatures and cryptographic authentication. [67]

---

64 From the Greek words for "hidden writing."

65 Codes operate at the level of meaning, *e.g.* "One if by land and two if by sea." Ciphers operate at the level of symbols or small groups of symbols, transform data using transposition, substitution, or both, and are reversible.

66 The undecipherable cipher.

67 The Certified Information Systems Security Professional (CISSP) Common Body of Knowledge says cryptography also protects availability. That is something of a stretch to say the least.

Effectively unbreakable cryptosystems exist, are available, and are well-documented. Attempting to use home-grown cryptosystems is very dangerous and should never be considered as part of an information security program. While it might seem very cool to design and implement your own cryptosystem, it turns out to be surprisingly hard to do right. Unless you have the equivalent of a Ph.D. in mathematics with an emphasis on cryptology and a dozen years' experience, it's an extremely bad idea to trust important information to home brewed crypto. Experiment all you want, but when you are serious about protecting information, use cryptosystems developed by experts, examined by other experts, and that have withstood the test of time.

7.7.1 Symmetric Key Cryptography

Symmetric key cryptography uses the same key for both encrypting and decrypting. It is an appropriate tool for protecting the confidentiality of data in storage. Even if encrypted data are compromised, they are of no use to an attacker. The encryption key must be protected, but this may be easier than protecting a large store of data. A suitably strong key will be 128 or 256 bits, 16 or 32 bytes, and can be stored offline until data in storage must be decrypted. Care must be taken in the handling of data that has been decrypted.

Symmetric key cryptography is less useful for data being processed because the key must be available. That might mean that a compromise of the data also results in a compromise of the key.

Symmetric key cryptography is not useful by itself for data in transmission because both sender and receiver must have a copy of the key. The problem of how to transmit a key securely over an unsecure channel is called the **key exchange problem**. Key exchange is addressed by asymmetric key and hybrid cryptography, both of which are discussed below.

There are several symmetric key algorithms that are effectively unbreakable. One of them is AES, the Advanced Encryption Standard, established by the National Institute of Standards and Technology in the United States in 2001.

7.7.2 Asymmetric Key Cryptography and Key Exchange

In the mid-1970s, four [68] groups of researchers independently invented a mechanism that uses two different keys with the same data, one to encrypt and one to decrypt. The key usually used for encrypting is called the **public key**, and the key usually used for decrypting is called the **private key**. Data encrypted with a particular public key can only be decrypted using the corresponding private key. You can give the public key to anyone, and they will not be able to decrypt messages that others may have encrypted with the same public key. Such a system is called **asymmetric key cryptography**, or public key cryptography.

This eliminates the key exchange problem because it is no longer necessary to exchange keys. One looks up the public key of a recipient – it's available publicly – and encrypts. The recipient decrypts using the corresponding private key.

The problem with asymmetric key cryptography is that it is computationally intensive. Keys are thousands of bits long and encryption requires exponentiation using those long keys. Encrypting or decrypting a message with the **RSA** [69] **algorithm** can take 10,000 times as long as encryption with a modern and secure symmetric key algorithm.

Key exchange algorithms and hybrid cryptography

It is possible to obtain the benefits of public key cryptography while paying only a part of the computational penalty. Instead of using asymmetric key cryptography to encrypt an entire message for transmission, it is used to solve the key exchange problem. The sender generates a random number 128 or 256 bits long and encrypts the random number using asymmetric key cryptography. Since the random number is perhaps 16 or 32 bytes, encryption is fast even with a computationally intensive algorithm. Using the random number as the key, the sender uses a symmetric key algorithm like AES to encrypt the message. The key, encrypted with an asymmetric key algorithm, and the message, encrypted with a symmetric key algorithm, are sent to the recipient together. The recipient

68 Three of these groups published their work at the time. The work of the fourth was classified by the military of the United Kingdom and not declassified until 1997.

69 Named for its inventors, L.M. Adleman, R.L. Rivest and A. Shamir.

first uses the private key of the asymmetric key pair to decrypt the generated key, then uses that key to decrypt the message. Such a scheme is called hybrid cryptography. This mechanism is effective only is the private key used to encrypt the session key remains secure.

Forward Secrecy

If a public key is used to encrypt a session key and the corresponding private key is compromised,[70] the adversary can decrypt the session key and with that, decrypt the communications. All future communications using the same public / private key pair are also compromised. If past communications have been captured and stored, they, too can be decrypted.[71]

There are key exchange algorithms that cannot be used to encrypt messages but that can enable two parties to negotiate a shared secret key over an unsecure channel. One such algorithm is the elliptic curve Diffie-Hellman key exchange algorithm. If asymmetric key cryptography is used to validate the identities of parties as described below, but a key exchange algorithm that doesn't depend on the security of the channel is used to generate a shared symmetric key, compromise of the private key no longer compromises the communication itself. Such a mechanism is called **forward secrecy** or sometimes, optimistically, perfect forward secrecy.

Quantum-resistant asymmetric key cryptography

Quantum computers apply fundamentally different approaches to computation. It would take decades for the fastest von Neumann architecture computers to find the prime factors of a 2,048 bit number if it were possible at all. A quantum computer and Shor's Algorithm could find the prime factors in a few seconds or less. That isn't an immediate concern because no quantum computer capable of running Shor's Algorithm on such large numbers currently exists.

---

70 A private key can be compromised by carelessness, but in the United States and elsewhere, it may also be possible to compel production of a private key with a court order or other legal process.

71 The U.S. National Security Agency has built a data center near Bluffton, Utah to store encrypted communications for possible future decryption. The data center was completed in May, 2019.

Experts estimate that it will be 2030 or later before current public key cryptography algorithms can be successfully attacked with quantum computers.

However, the year 2030 is not that far in the future. In 2016, the U.S. National Institute of Standards and Technology announced a contest that would select quantum-resistant cryptography algorithms for standardization. In 2019, NIST narrowed the field of entries to 26 possible candidates. NIST plans to publish a final draft by 2024.

7.7.3 Cryptographic Hashes and Digital Signatures

Part of defense in depth of the integrity of information is to be able to detect tampering. This is accomplished using a **cryptographic hash** [72] algorithm. A cryptographic hash algorithm takes variable-length data as its input and produces a small, fixed-size output. For SHA-256, the output is 256 bits, or 32 bytes. The output of such a hash algorithm is called a **digest**.

A cryptographic hash algorithm has two important properties. First, even a single-bit difference in inputs will produce very different hash values. Second, it should be impossible to construct two inputs so that both produce the same hash value. That's called a **collision**, and a hash algorithm that allows collisions can't produce a unique "fingerprint" of its input.

It is not possible to determine the original input given only the hash value. However, for small inputs like passwords, it may be possible to mount a **brute force attack** in which many or all possible values are used as inputs to the hash algorithm to see whether any results in the same hash value as the one under attack.

Cryptographic hashes can be used to detect changes in data. To do so, create a hash digest of the data when it is a known-good state. The digest must be stored separately and securely. The integrity of the data can then be tested by re-computing the digest and comparing it with the original. If they're equal, the assumption is that no tampering has occurred. The strength of that assumption depends on the resistance to collisions of the chosen hash algorithm.

72 It's called a "hash" because the output has no intrinsic meaning.

It is important to know that there are different kinds of hashing algorithms for different purposes. Hash algorithms for integrity checking or digital signatures, like SHA-256, are designed to be fast. Hash algorithms for storing passwords, like BCRYPT or SCRYPT, are specifically designed to be slow to make brute force attacks time-consuming. There are also cryptographic hash functions, like MD5, that were once thought to be secure but have since been found to have flaws.

Digital signatures

Some asymmetric key cryptography algorithms have the property that the public and private keys are cryptographic inverses of one another. Data encrypted with a given public key can only be decrypted using the corresponding private key. However, data or a message encrypted using the private key can be decrypted using the public key. Such a message is not confidential because anyone can decrypt it using the public key. However, there is a strong assumption that it could only have been encrypted by the owner of the corresponding private key. It has been digitally signed. How strong is the assumption? As strong as our knowledge or belief that the holder of the private key has kept it private. If the private key has been compromised, the digital signature is no longer reliable.

Rather than encrypting an entire message using the sender's private key, which would be very slow, digital signatures are implemented by first computing a hash digest of the data to be signed, then encrypting the digest with the signer's private key.

To validate a digital signature, compute a digest of the signed data using the same algorithm used to sign the data. Decrypt the signature using the signer's public key, then compare the decrypted signature, really a message digest, to the digest just computed. If they're equal, one can conclude that the data were signed using the signer's private key, *and also* that the data haven't been tampered with or corrupted. If the data had been changed, even though the digital signature was correct, the computed digest would be different.

Digital certificates

Suppose Bill wants to send a confidential message to Alice, with whom his only method of contact is email. Bill looks up Alice's public key, perhaps on the OpenPGP key server. But Bill has a problem, namely how to be certain he really has Alice's public key. If evil Eve the eavesdropper had somehow compromised the key server and substituted her own public key for Alice's, Eve would be able to decrypt and read any messages she could intercept.

The solution to Bill's problem is a digital certificate. The purpose of a digital certificate is to bind an identity, in this case Alice's, to a public key. Figure 7-4 is a simplified diagram of a digital certificate. The identifier (Alice's email address), the public key, and the other information, including a certificate version number and expiration date, are the body of the certificate. They're all in plain text; anyone can read them with no deciphering necessary.

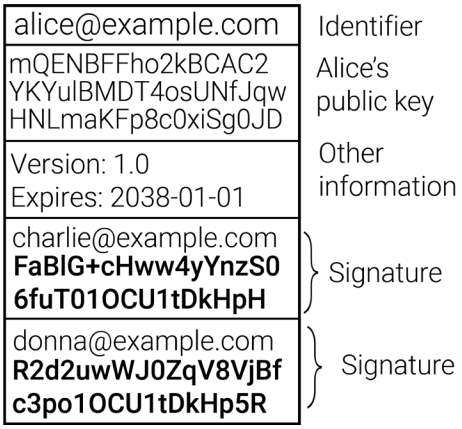

Figure 7-4
Simplified diagram of a digital certificate.

The certificate in Figure 7-4 has two signatures, by Charlie and Donna. The important parts are the identifiers, in this case email addresses, and a cryptographic hash digest of the body of the certificate, encrypted with the signer's private key. By signing, Charlie and Donna attest that the public key in the body of the certificate is really Alice's. [73]

To verify a signature, Bill computes his own digest over the body of the certificate. He then uses Charlie's or Donna's public key, obtained separately, to decrypt the signature. If the decrypted digest matches the digest Bill computed, Bill can trust that he has Alice's true public key because Alice has

73 Keys are often exchanged in person at "key signing parties." Perhaps you and friends or classmates want to organize a key signing party. You can find detailed instructions on the Web.

been "introduced" by Charlie or Donna. The amount of trust depends upon how much Bill trusts Charlie or David.

If Bill doesn't know, or doesn't trust, Charlie or Donna, he can look at who has signed *their* public keys. Bill's asymmetric key software will expand the search automatically, looking for someone whom Bill has marked as a trusted introducer. This concept is called the **web of trust**, in which parties vouch for one another.

Certificate authorities and the public key infrastructure

Another kind of digital certificate, an X.509 certificate, usually has only one signature, that of a certificate authority. A certificate authority is a company, a non-profit, or in some countries, a government agency that takes on the task of verifying that public keys belong to the named principal. The certificate authority acts as a *trusted third party*. Certificates issued in this way are used by TLS [74] to encrypt traffic between browsers and servers and to provide a measure of assurance that a person who wants to connect to, *e.g.*, apple.com has actually reached Apple's servers.

Such an arrangement is not without problems of its own. Not all certificate authorities are both equally careful and equally trustworthy. There have been a few instances of certificate authorities issuing fraudulent certificates. A big threat is that a certificate authority's "signing key" – the private key used to sign digital certificates – becomes compromised. There are ways to mitigate these problems, but they are beyond the scope of this chapter.

7.8 Summary

Security problems can be caused by malicious actors, but they can also arise from error or environmental failure.

The three properties of a secure information system are confidentiality, integrity, and availability. The threats to information assets are disclosure, alteration, and denial.

74 Transport Layer Security, the facility used to encrypt traffic on the World Wide Web.

Every information security program must be grounded in an information security policy that describes the intended use of the system, including what information assets are to be protected and the constraints on the use of those assets. An information security program is a risk management program that manages risks to the organization's information assets.

A principal is a unique entity that may attempt access to an information system. The authentication process binds a principal to the internal representation of that principal. Authentication is crucial to security because all decisions on access to a system's resources must assume that the binding is correct. Once a principal has been authenticated, it is possible to make authorization decisions about what the principal is allowed to do.

Cryptography is the use of ciphers to obscure information until the encryption is reversed by decryption. Cryptography can protect the confidentiality and integrity of information. Classes of cryptographic systems are symmetric key cryptography, asymmetric (public) key cryptography, and cryptographic hash functions. Digital signatures and digital certificates protect the integrity of information, including cryptographic keys.

7.9 References

McCumber, J. (1991). Information Systems Security: A Comprehensive Model. *Proceedings 14th National Computer Security Conference*. Baltimore, MD: National Institute of Standards and Technology.

Pfleeger, C. P., & Pfleeger, S. L. (2003). *Security in Computing*. Hoboken, NJ: Prentice Hall Professional.

Pfleeger, C. P., & Pfleeger, S. L. (2006). *Security in Computing Fourth Edition*. Hoboken, NJ: Prentice Hall Professional.

Ranum, M. (2005, September 1). The Six Dumbest Ideas in Computer Security. Retrieved June 12, 2020, from https://www.ranum.com/security/computer_security/editorials/dumb

Rosso, O. A., Ospina, R., & Frery, A. C. (2016). Classification and Verification of Handwritten Signatures with Time Causal Information Theory Quantifiers. PLoS One. doi:10.1371/journal.pone.0166868

Saltzer, J. H., & Schroeder, M. D. (1973). The Protection of Information in Computer Systems. *Fourth ACM Symposium on Operating System Principles* (p. 119). New York, NY: Association for Computing Machinery. doi:10.1145/800009.808059

Sams, F. (1982). *Run with the Horsemen.* Atlanta GA: Peachtree Press.

Torres, K. (2017, July 14). Georgia to shift elections work in-house, away from Kennesaw State. *Atlanta Journal-Constitution*. Retrieved June 12, 2020, from https://www.ajc.com/news/state--regional-govt--politics/georgia-shift-elections-work-house-away-from-kennesaw-state/JAazLxUB0SODnPMqEGNWdJ/