Chapter 1

Cryptographic Tools and Techniques

THE FOLLOWING COMPTIA CASP EXAM OBJECTIVES ARE COVERED IN THIS CHAPTER:

✓ 1.1 Distinguish which cryptographic tools and techniques are appropriate for a given situation.

- Cryptographic applications and proper implementation
- Advanced PKI concepts
  - Wildcard
  - OCSP vs. CRL
  - Issuance to entities
- Users
- Systems
- Applications
- Implications of cryptographic methods and design
  - Strength vs. performance vs. feasibility to implement vs. interoperability
- Transport encryption
- Digital signature
- Hashing
- Code signing
- Non-repudiation
- Entropy
- Pseudorandom number generation
- Perfect forward secrecy
- Confusion
- Diffusion
This chapter discusses cryptography. Cryptography can be defined as the art of protecting information by transforming it into an unreadable format. Everywhere you turn you see cryptography. It is used to protect sensitive information, prove the identity of a claimant, and verify the integrity of an application or program. As a security professional for your company, which of the following would you consider more critical if you could choose only one?

- Provide a locking cable for every laptop user in the organization.
- Enforce full disk encryption for every mobile device.

My choice would be full disk encryption. Typically the data will be worth more than the cost of a replacement laptop. If the data is lost or exposed, you'll incur additional costs such as patient notification and reputation loss.

As a security professional, you should have a good basic understanding of cryptographic functions. This chapter begins by reviewing a little of the history of cryptography. Next, I discuss basic cryptographic types, explaining symmetric, asymmetric, hashing, digital signatures, and public key infrastructure. These are important as we move on to more advanced topics and begin to look at cryptographic applications. Understanding these topics will help you prepare for the CompTIA exam and to implement cryptographic solutions to better protect your company’s assets.

The History of Cryptography

Encryption is not a new concept. The desire to keep secrets is as old as civilization. Some examples of early cryptographic systems include the following:

**Scytale**  This system functioned by wrapping a strip of papyrus or leather around a rod of fixed diameter on which a message was written. The recipient used a rod of the same diameter on which he wrapped the paper to read the message. While such systems seem basic today, it worked well in the time of the Spartans. Even if someone was to intercept the message, it appeared as a jumble of meaningless letters.

**Caesar's Cipher**  Julius Caesar is known for an early form of encryption, the Caesar cipher, used to transmit messages sent between Caesar and his generals. The cipher worked by means of a simple substitution. The plain text was rotated by three characters (ROT3) so that before a message was sent, it was moved forward by three characters. Using Caesar’s cipher to encrypt the word *cat* would result in *fdw*. Decrypting required moving back three characters.
Other Examples  Substitution ciphers substitute one character for another. The best example of a substitution cipher is the Vigenère polyalphabetic cipher. Other historical systems include a running key cipher and the Vernam cipher.

Cryptographic Services

As a security professional, you need to understand cryptographic services and how they are applied. You also need to understand the goals of cryptography and basic terms. While your job may not require you to be a cryptographic expert, you should be able to explain how specific cryptographic functions work to pass the CASP exam.

Cryptographic Goals

Cryptography includes methods such as symmetric encryption, asymmetric encryption, hashing, and digital signatures. Each provides specific attributes and solutions. These goals of these cryptographic services include the following:

Privacy  Also called confidentiality. What is private (confidential) should stay private, whether at rest or in transit.

Authentication  There should be proof that the message is from the person or entity we believe it to be from.

Integrity  Information should remain unaltered at the point it was produced, while it is in transmission, and during storage.

Non-repudiation  The sender of data is provided with proof of delivery and the recipient is assured of the sender’s identity.

An easy way to remember these items for the exam is to think of PAIN. This simple acronym (privacy, authentication, integrity, and non-repudiation) should help you remember the basic cryptographic goals.

Knowing these basic goals can go a long way in helping you to understand that cryptography can be used as a tool to achieve confidentiality, integrity, and availability. For example, consider how encryption can protect the privacy and confidentiality of information at rest or in transit. What if your CEO has been asked to travel to the Far East for trade negotiations? Think about the CEO’s laptop. If it is lost or compromised, how hard would it be for someone to remove unencrypted data? Strong encryption offers an easy way to protect that information should the equipment be lost, stolen, or accessed by unauthorized individuals. Applications such as TrueCrypt and BitLocker offer the ability to encrypt a hard drive.
During a trip to Beijing in December 2007, it was discovered that someone had accessed a laptop used by former Commerce Secretary Carlos Gutierrez and had placed monitoring programs on it designed to secretly remove information. Read more at http://techinsider.nextgov.com/2008/05/china_hacks_commerce_secretary.php.

Authentication is another key goal of cryptography. First, *authentication* is associated with digital signatures. Authentication provides a way to ensure that any message is from who we believe it's from. In its basic form, authentication is used to determine identity. It is also part of the identification and authentication process.

Integrity is another cryptographic goal. Integrity is important while data is in transmission and in storage. *Integrity* means that information remains unaltered. Imagine the situation of needing to download a patch. Although the patch is available on the developer's site, you also have a copy on DVD that was given to you by a colleague. Is the version on the DVD the same as the one on the developer's website? Integrity verification programs such as MD5 or SHA can help you determine this.

*Non-repudiation* is assurance that an entity to a communication cannot deny authenticity. It is proof of the veracity of a claim. Non-repudiation means that a sender of data receives proof of delivery and the recipient is assured of the sender’s identity. Neither party should be able to deny having sent or received the data at a later date. In the days of face-to-face transactions, non-repudiation was not as hard to prove. Today, the Internet makes many transactions faceless. We may never see the people we deal with; therefore, non-repudiation becomes all the more critical. Non-repudiation is achieved through digital signatures, digital certificates, and message authentication codes (MACs).

**Cryptographic Terms**

As a security professional, you need to understand basic cryptographic terms. You will encounter these terms when examining a vendor's security solution, discussing security controls with colleagues, and implementing a security solution. Here are some basic cryptographic terms:

- **Plain Text** Clear text that is readable.
- **Cipher Text** Encrypted text that is unreadable.
- **Encryption** Transforming data into an unreadable format. For example, using Caesar's cipher to encrypt the word *cat* would result in *fdw*. Encryption here has moved each character forward by three letters.
- **Cryptanalysis** The act of obtaining plain text from cipher text without a cryptographic key. It is used by governments, the military, enterprises, ethical hackers, and malicious hackers to find weaknesses and crack cryptographic systems.
- **Digital Signature** A hash value that has been encrypted with the private key of the sender. It is used for authentication and integrity.
Cryptographic systems can be broadly classified into symmetric, asymmetric, and hashing:

**Symmetric Cryptography**  This type uses a single private key.

**Asymmetric Cryptography**  This type uses two keys: a public key known to everyone and a private key that only the recipient of messages uses.

While both concepts are discussed in more detail later in the chapter, at this point it’s important to understand that both symmetric and asymmetric cryptography make use of a key. The key is input into the encryption algorithm as data on which to perform mathematical operations such as permutation, substitution, or binary math.

**Hash**  A hash is a defined mathematical procedure or function that converts a large amount of data into a fixed small string of data or integer. The output of a hash is known as a hash value, hash code, hash sum, checksum, fingerprint, or message digest.

For the CASP exam, more than one term may be used to describe a hash.

Here are some other terms that you will need to know:

- An *algorithm* is a set of rules or ordered steps used to encrypt and decrypt data. The algorithm is a set of instructions used with the cryptographic key to encrypt plain text data. Plain text data encrypted with different keys or dissimilar algorithms will produce different cipher text.

- *Cipher text* is data that is scrambled and unreadable. When plain text is converted into cipher text, the transformation can be accomplished in basically two ways:

  - **Block Ciphers**  Function by dividing the message into blocks for processing.

  - **Stream Ciphers**  Function by dividing the message into bits for processing.

- **Cryptographic Key**  How strong the encryption process is relies in part on the cryptographic key. The *cryptographic key* is a piece of information that controls how the cryptographic algorithm functions. It can be used to control the transformation of plain text to cipher text or cipher text to plain text. For an attacker to brute-force the cryptographic system, he would need to guess the key. That is why the more possible keys or combinations, the longer it will take for an attacker to gain access to your encrypted data.

- **Entropy**  Although key size is important, the randomness of the key is also critical. You may have been asked to create a random key before and not have realized what you were actually doing. For example, many security products begin the process of generating a pseudorandom key by having the user tap random keys on a keyboard or randomly moving the mouse. Such activity is known as entropy. *Entropy* is a measure of the randomness of data collected by an application or an operating system and used to create a cryptography key.

While having a random key is a good start, the key must also remain secret. This is no different than thinking of your password as a key. If everyone knows the password to your
computer, anyone can access it at any time they please. High-value data requires strong protection, which typically means longer keys that are exchanged more frequently, to protect against attacks.

Not all cryptosystems are of the same strength. For example, Caesar’s cipher seemed quite strong when created, but it is insecure today. As a security professional, you should always be asking how strong an encryption process should be.

Cryptographic systems may also make use of a nonce. A nonce is a number used once—that is, as random a number as a cryptosystem can generate. The programs that create these are known as pseudorandom number generators. Such systems use algorithms to generate a sequence of numbers that approximates the properties of random numbers. Pseudorandom numbers are unique and different each time one is generated.

If you are interested in seeing programs that can be used to create pseudorandom numbers, take a moment to check out www.agner.org/random/.

An initialization vector (IV) is an example of a type of nonce. An IV is used to create a unique cipher text every time the same message is encrypted using the same key.

Table 1.1 highlights some of the strengths and weaknesses of symmetric and asymmetric encryption.

<table>
<thead>
<tr>
<th>Encryption type</th>
<th>Advantage</th>
<th>Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric</td>
<td>Faster than asymmetric</td>
<td>Key distribution is difficult and must be done out of band; symmetric encryption only provides confidentiality.</td>
</tr>
<tr>
<td>Asymmetric</td>
<td>Easy key exchange</td>
<td>Can provide confidentiality and authentication; slower than symmetric.</td>
</tr>
</tbody>
</table>

**Cipher Types and Methods**

Let’s now continue with our discussion of block and stream ciphers.
Block Ciphers

Block ciphers are widely used in software products. Most modern encryption algorithms implement some type of block cipher.

Block ciphers operate on blocks or fixed-size chunks of data; 64-bit blocks are a commonly used size. One type of block cipher is a transposition cipher. A transposition cipher shifts units of plain text in a consistent manner so that the cipher text constitutes a permutation of the plain text. An example of this can be seen in a rail fence cipher. This example of a transposition cipher encrypts the message in a downward pattern on successive rails of an imaginary fence; then it moves up toward the top when the bottom is reached. This pattern repeats itself over successive rails. The message is then encrypted by being read off in rows. Figure 1.1 shows how a rail fence cipher of the message of “WE ARE DISCOVERED. FLEE AT ONCE” would appear.

**Figure 1.1** A rail fence cipher (an example of a transposition cipher)


| WXCL THRD SEEF SNOCA IVDEN |

There are various encryption methods used in block ciphers. During the encryption and decryption process, the message is divided into blocks of bits. These blocks are then put through functions such as substitution, transposition, confusion, and diffusion.

**Substitution** Using this method means to put in the place of another, such as one letter for another or letters for numbers, etc.

**Transposition** This method scrambles a message by reordering the plaintext in some definite way.

**Confusion** This method uses a relationship between the plain text and the key that is so complicated an attacker can’t alter the plain text and determine the key.

**Diffusion** In this method a change in the plain text results in multiple changes spread out throughout the cipher text.

The substitution box (s-box) is one technique that is used to introduce confusion. When properly implemented, s-boxes are designed to defeat cryptanalysis. An s-box takes a number of input bits, \( m \), and transforms them into some number of output bits, \( n \). S-boxes can be implemented as a type of lookup table and used with symmetric encryption systems such as the Data Encryption standard (DES) and Triple (DES).
Stream Ciphers

A stream cipher inputs digits, bits, or characters and encrypts the stream of data. The one-time pad is an example of a stream cipher. The onetime pad works on each letter of the plain text message independently. A stream cipher combines the plain text bit with a pseudorandom cipher bit stream by means of an exclusive OR (XOR) operation. Stream ciphers operate at a higher speed than block ciphers and in theory are well suited for hardware implementation.

Symmetric Encryption

Symmetric encryption uses a single shared key for encryption and decryption. These are known as dual-use keys, as they can be used to lock and unlock data. Symmetric encryption is the oldest form of encryption. Historical systems such as scytale and Caesar’s cipher are types of symmetric encryption. Symmetric encryption offers users privacy by keeping individuals who do not have the key from having access to the true contents of the message. Figure 1.2 shows the symmetric encryption process.

**Figure 1.2** Symmetric encryption

Notice how the plain text is encrypted with the single shared key and is then transmitted to the recipient of the message, who goes through the same process to decrypt the message. The dual use of keys is what makes this system so simple but also introduces weakness. Symmetric encryption is fast, and with a small key it can be used to encrypt bulk data very quickly. It is also strong and hard to break if the key is of sufficient size. However, symmetric encryption does have disadvantages.

The problem is key distribution. For symmetric encryption to be effective there must be a secure method by which to transfer keys. In our modern world, there needs to be some type of out-of-band transmission. *Out of band* means using a different means to transmit the key. As an example, if Bob wants to send Alice a secret message but is afraid that Mike can monitor their communication, how can he send the message? If the key is sent in clear text, Mike can intercept it. Bob could deliver the key in person, mail it, or even send a courier. All of these out-of-band methods are highly impractical in the world of e-commerce and electronic communication because they do not scale well.
Symmetric Encryption

Even if the problems of key exchange are overcome, there are still other concerns. Another problem is key management. If, for example, ten people needed to communicate using symmetric encryption, the number of keys needed would be 45. As the number of people using symmetric encryption rises, so does the required number of keys. To determine the numbers of keys needed in symmetric encryption, the following formula is used:

\[ \frac{n!}{2!(n - 2)!} \]

which simplifies to

\[ \frac{n(n - 1)}{2} \text{ or } 10(10 - 1) + 2 = 45 \text{ keys} \]

Our third and final flaw with symmetric encryption is that it only provides confidentiality.

For the CASP exam, you should understand the three primary issues with the use of symmetric encryption. These include issues with key exchange and key management, and the fact that symmetric encryption only offers confidentiality.

While it is true that symmetric encryption is not perfect, it does offer some great features that make it an excellent choice for securing data and providing confidentiality. Symmetric encryption is fast. It can encrypt and decrypt very quickly and is considered strong. Symmetric encryption is very hard to break if a large key is used. Here are some well-known symmetric algorithms:

**DES** The Data Encryption Standard was once the most common symmetric algorithm used. It has now been officially retired by the National Institute of Standards and Technology (NIST). Its short-term replacement was 3DES. Today, all versions of DES have been replaced by the Advanced Encryption Standard (AES).

**Advanced Encryption Standard** The symmetric algorithm chosen as a replacement for DES. It was adopted from the Rijndael algorithm and is used for sensitive and secret data. It’s key sizes are 128, 192, and 256 bit.

**Blowfish** A general-purpose symmetric algorithm intended as a replacement for DES, Blowfish has a variable block size and up to a 448-bit key.

**CAST** Carlisle Adams/Stafford Tavares (CAST) is a 128- or 256-bit block cipher that was a candidate for AES.

**IDEA** The International Data Encryption Algorithm (IDEA) is a block cipher that uses a 128-bit key to encrypt 64-bit blocks of plain text. It is used by Pretty Good Privacy (PGP).

**Rijndael** This is a block cipher adopted as the AES by NIST to replace DES.

**RC4** Rivest Cipher 4 is a stream-based cipher. Stream ciphers treat the data as a stream of bits.

**RC5** Rivest Cipher 5 is a fast block cipher. It is different from other symmetric algorithms in that it supports a variable block size, a variable key size, and a variable number of rounds.
A *round* is a sequential repetition of a series of math functions. Allowable choices for the block size are 32, 64, and 128 bits. The key can range up to 2040 bits.

SAFER Secure and Fast Encryption Routine (SAFER) is a block-based cipher that processes data in blocks of 64 and 128 bits.

Skipjack Promoted by the U.S. National Security Agency (NSA), Skipjack uses an 80-bit key and operates on 64-bit blocks of text. Skipjack faced opposition because the government would maintain a portion of the information required to reconstruct a Skipjack key so that legal authorities could decrypt communications between the affected parties when approved by a court.

Twofish Twofish is a block cipher that operates on 128-bit blocks of data and is capable of using cryptographic keys up to 256 bits in length.

Now let’s look at some of the popular symmetric encryption standards in more depth.

**Data Encryption Standard**

DES was originally developed by IBM and then modified by NIST. The NSA endorsed the revised standard. It was published in 1977, and was released by the American National Standards Institute (ANSI) in 1981.

DES is a symmetric encryption standard that is based on a 64-bit block that processes 64 bits of plain text at a time. DES outputs 64-bit blocks of cipher text. The DES key size is 56 bits, and DES has four primary modes of operation:

- Electronic codebook (ECB) mode
- Cipher block chaining (CBC) mode
- Output feedback (OFB) mode
- Cipher feedback (CFB) mode

All four modes use the 56-bit key, and while the standard lists the key as 64 bits, 8 bits are used for parity checking so the true key size is actually 56 bits. Parity checking is a simple form of error detection. Each 64-bit, plain-text block is separated into two 32-bit blocks and then processed by the 56-bit key. The plain text is processed by the key through 16 rounds of transposition and substitution.

Any CASP exam questions that mention DES should be examined closely. Remember that while DES operates on 64 bit blocks, the effective key length is only 56 bits long.

**Electronic Codebook Mode**

Electronic codebook (ECB) mode is the default mode of encryption used by DES. If the last block is not a full 64 bits, padding is added. While ECB produces the greatest throughput, it is also the easiest implementation of DES encryption to crack. If used with
large amounts of data, it is easily broken because the same plain text encrypted with the same key always produces the same cipher text. This is why if ECB is used, it should only be on small amounts of data.

When you’re using ECB, keep in mind that a fixed key and a known repeating plain text message will always produce the same cipher text.

**Cipher Block Chaining Mode**

When DES is operating in cipher block chaining (CBC) mode, it is somewhat similar to ECB except that CBC inserts some of the cipher text created from the previous block into the next one. This process is called XORing. It makes the cipher text more secure and less susceptible to cracking. CBC is aptly named because data from one block is used in the next, and the blocks are chained together. This chaining produces dependency but also results in more random cipher text.

**Output Feedback Mode**

Output feedback (OFB) mode is implemented as a stream cipher and uses plain text to feed back into the stream of cipher text. Transmission errors do not propagate throughout the encryption process. An initialization vector is used to create the seed value for the first encrypted block. DES XORs the plain text with a seed value to be applied with subsequent data.

**Cipher Feedback Mode**

Cipher feedback (CFB) mode can be implemented as a stream cipher and used to encrypt individual characters. CFB is similar to OFB in that a previously generated cipher text is added to subsequent streams. Because the cipher text is streamed together, errors and corruption can propagate through the encryption process.

How secure is DES? Not as secure as it once was. Computing power has increased over the years, and that has decreased the time required to brute-force DES. In 1998, the Electronic Frontier Foundation was able to crack DES in about 23 hours.

**Triple-DES**

Triple DES (3DES) was designed to be a stopgap solution. DES was initially certified on a five-year basis and was required to be recertified every five years. While easily passing these recertifications in the early years, DES began to encounter problems around the 1987 recertification. By 1993, NIST stated that DES was beginning to outlive its usefulness. They began
looking for candidates to replace it. This new standard was to be referred to as the Advanced Encryption Standard (AES).

While AES was to be the long-term replacement, something else was needed to fill the gap before AES was ready to be deployed. Therefore, to extend the usefulness of the DES encryption standard, 3DES was adopted. It can use two or three keys to encrypt data, depending on how it is implemented. It has an effective key length of 112 or 168 bits and performs 48 rounds of transpositions and substitutions. Although it is much more secure, it is as slow as a third the speed of 56-bit DES.

**Advanced Encryption Standard**

In 2002, NIST chose Rijndael to replace DES. Its name is derived from its two developers, Vincent Rijmen and Joan Daemen. It is a fast, simple, robust encryption mechanism. Rijndael is also known to resist various types of attacks. The Rijndael algorithm uses three layers of transformations to encrypt and decrypt blocks of message text, including the following:

- Linear mix transform
- Nonlinear transform
- Key addition transform

Rijndael uses a four-step, parallel series of rounds. Rijndael is an iterated block cipher that supports variable key and block lengths of 128, 192, or 256 bits:

- If both key and block size are 128 bit, there are 10 rounds.
- If both key and block size are 192 bit, there are 12 rounds.
- If both key and block size are 256 bit, there are 14 rounds.

Each of these steps is performed during each round. They include:

**Byte Substitution** Each byte is replaced by an s-box substitution.

**Shift Row** Bytes are arranged in a rectangle and shifted.

**Mix Column** Matrix multiplication is performed based on the arranged rectangle.

**Add Round Key** Each byte of the state is combined with the round key.

On the last round, the fourth step is bypassed and the first is repeated.

**International Data Encryption Algorithm**

The International Data Encryption Algorithm (IDEA) is a 64-bit block cipher that uses a 128-bit key. It is different from others, as it avoids the use of s-boxes or lookup tables. Although IDEA is patented by a Swiss company, it is freely available for noncommercial use. It is considered a secure encryption standard and there have been no known attacks against it. It operates in four distinct modes, like DES. At one time, it was thought that IDEA might replace DES, but patent royalties made that impractical.
Asymmetric Encryption

Rivest Cipher Algorithms

The RC series ciphers are part of a family of ciphers designed by Ron Rivest. Rivest ciphers include RC2, RC3, RC4, RC5, and RC6. RC2 is an older algorithm. RC2 maintains a variable key size, 64-bit block cipher that can be used as a substitute for DES. RC4 was implemented as a stream cipher. The 40-bit version is what was originally available in Wired Equivalent Privacy (WEP). It is most commonly found as the 128-bit key version. RC5 is a block cipher in which the number of rounds can range from 0 to 255 and the key can range from 0 bits to 2,048 bits in size. Finally, there is RC6. It features variable key size and rounds and added two features not found in RC5: integer multiplication and 4-bit working registers.

While symmetric encryption does offer speed, if you’re looking for a cryptographic system that provides easy key exchange, you will have to consider asymmetric encryption.

Asymmetric Encryption

Asymmetric encryption, or public key cryptography, is different from symmetric encryption. It overcomes one of the big barriers of symmetric encryption: key distribution. Asymmetric encryption uses two unique keys, as shown in Figure 1.3. What one key does the other key undoes.

**FIGURE 1.3  Asymmetric encryption**

Here’s how asymmetric encryption works: Imagine that you want to send a coworker a message. You use your coworker’s public key to encrypt the message. Your coworker receives the message and uses a private key to decrypt it.

Public key cryptography is made possible by the use of one-way functions. A one-way function, or trap door, is a math operation that is easy to compute in one direction, yet
almost impossible to compute in the other. Depending on the type of asymmetric encryption used, this difficulty is based on either the discrete logarithm problem or the factoring of a large number into its prime factors. While the math behind the encryption process is not needed to pass the CASP exam, in algebra, discrete logarithms are group-theoretic analogues of ordinary logarithms. For example, if you are given two large prime numbers, it is easy to multiply them. However, if you are given only their product, it difficult or impossible to find the factors with today’s processing power.

If the message is encrypted with the public key, only the matching private key will decrypt it. The private key is kept secret, whereas the public key can be given to anyone. If the algorithm is properly designed, it should not be possible for someone to easily deduce the private key of a pair if that person has only the public key. Cryptographic systems may also make use of zero knowledge proof. This concept allows you to prove your knowledge without revealing the fact to a third party.

Consider the following: Given the prime numbers of 397 and 823, it is easy to multiply them together and get 326,731. However, if you are given the number 326,731, it’s quite difficult to extract the two prime numbers of 397 and 823. Anyone who knows the trapdoor can perform the function in both directions, but if you are lacking the trapdoor, you can only perform the function in one direction. Trapdoor functions can be used in the forward direction for encryption and signature verification, whereas the inverse direction is used for decryption and signature generation. To help ensure your success on the CASP exam, symmetric and asymmetric cryptographic systems are compared in Table 1.2.

<table>
<thead>
<tr>
<th>Table 1.2</th>
<th>Attributes of symmetric and asymmetric encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Symmetric</strong></td>
<td><strong>Asymmetric</strong></td>
</tr>
<tr>
<td>Confidentiality</td>
<td>Confidentiality, integrity, authentication, and non-repudiation</td>
</tr>
<tr>
<td>One single shared key</td>
<td>Two keys: public and private</td>
</tr>
<tr>
<td>Require an out-of-band exchange</td>
<td>Useful for in-band exchange</td>
</tr>
<tr>
<td>Not scalable, too many keys needed</td>
<td>Scalable, works for e-commerce</td>
</tr>
<tr>
<td>Small key size and fast</td>
<td>Larger key size required and slower to process</td>
</tr>
<tr>
<td>Useful for bulk encryption</td>
<td>Best for small amounts of data, digital signatures, digital envelopes, digital certificates</td>
</tr>
</tbody>
</table>

**Diffie–Hellman**

Dr. W. Diffie and Dr. M.E. Hellman released the first public key-exchange protocol in 1976. They developed it specifically for key exchange and not for data encryption or digital
signatures. The Diffie–Hellman protocol was designed to allow two users to exchange a secret key over an insecure channel without any prior communication. The protocol functions with two system parameters: \( p \) and \( g \). Both parameters are public and can be used by all the system’s users. Parameter \( p \) is a prime number, and parameter \( g \), which is usually called a generator, is an integer less than \( p \) that has the following property: For every number \( n \) between 1 and \( p – 1 \) inclusive, there is a power \( k \) of \( g \) such that \( g^k = n \mod p \). Diffie–Hellman is used in conjunction with several authentication methods, including the Internet Key Exchange (IKE) component of IPSec.

While Diffie–Hellman was groundbreaking in its ability to allow two parties to securely exchange encryption keys, it is not without its problems. It is vulnerable to man-in-the-middle attacks because the key exchange process does not authenticate the participants. Digital signatures should be used to alleviate this vulnerability.

**RSA**

The RSA algorithm is named after its inventors. Ron Rivest, Adi Shamir, and Len Adleman developed RSA in 1977. Although RSA, like other asymmetric algorithms, is slower than symmetric encryption systems, it offers secure key exchange and is considered very secure. RSA supports a key size up to 3,072 bits. The design of RSA is such that it has to use prime numbers whose product is much larger than 129 digits for security; 129-digit decimal numbers are factored using a number field sieve algorithm. RSA public and private keys are generated as follows:

1. Choose two large prime numbers, \( p \) and \( q \), of equal length and compute \( p \times q = n \), which is the public modulus.
2. Choose a random public key, \( e \), so that \( e \) and \( (p – 1)(q – 1) \) are relatively prime.
3. Compute \( e \times d = 1 \mod [(p – 1)(q – 1)] \), where \( d \) is the private key.
4. Thus, \( d = e – 1 \mod [(p – 1)(q – 1)] \).

From these calculations, \( (d, n) \) is the private key and \( (e, n) \) is the public key. The plain text, \( P \), is encrypted to generate cipher text, \( C \), as follows:

\[
C = P^e \mod n
\]

and is decrypted to recover the plain text, \( P \), as follows:

\[
P = C^d \mod n
\]

RSA functions by breaking the plain text into equal-length blocks, with each block having fewer digits than \( n \). Each block is encrypted and decrypted separately. Anyone attempting to crack RSA would be left with a difficult challenge because of the difficulty of factoring a large integer into its two factors. Cracking an RSA key would require an extraordinary amount of computer processing power and time. The RSA algorithm has become the de facto standard for industrial-strength encryption, especially since the patent expired in 2000. It is built into many protocols, such as PGP; software products; and systems such as Mozilla Firefox, Google Chrome, and Microsoft Internet Explorer.
Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) can be found in smaller, less-powerful devices such as smartphones and handheld devices. ECC is considered more secure than some of the other asymmetric algorithms because elliptic curve systems are harder to crack than those based on discrete log problems. Elliptic curves are usually defined over finite fields such as real and rational numbers, and they implement an analog to the discrete logarithm problem.

El Gamal

El Gamal was released in 1985, and its security rests in part on the difficulty of solving discrete logarithm problems. It is an extension of the Diffie–Hellman key exchange. El Gamal consists of three discrete components: a key generator, an encryption algorithm, and a decryption algorithm. It can be used for digital signatures, key exchange, and encryption.

Merkle–Hellman Knapsack

Merkle–Hellman Knapsack is another example of an asymmetric algorithm. It is different in that it is based on fixed weights. While it was once popular, it is not easily found in current products as it was broken in 1982.

Hybrid Encryption

Sometimes mixing two things together makes good sense. Do you remember the commercial, “You got your chocolate in my peanut butter”? While you may not consider cryptography as tasty as chocolate, there is a real benefit to combining both symmetric and asymmetric encryption. Symmetric encryption is fast, but key distribution is a problem. Asymmetric encryption offers easy key distribution, but it’s not suited for large amounts of data. Combining the two into hybrid encryption uses the advantages of each and results in a truly powerful system. Public key cryptography is used as a key encapsulation scheme, and the private key cryptography is used as a data encapsulation scheme. Here is how the system works. If Bob wants to send a message to Alice, the following occurs:

1. Bob generates a random private key for a data encapsulation scheme. This session key is a symmetric key.
2. Bob encrypts the message with the data encapsulation scheme using the symmetric key that was generated in step 1.
3. Bob encrypts the symmetric key using Alice’s public key.
4. Bob sends both of these items, the encrypted message and the encrypted key, to Alice.
5. Alice uses her private key to decrypt the symmetric key and then uses the symmetric key to decrypt the message. This process is shown in Figure 1.4.
Almost all modern cryptographic systems make use of hybrid encryption. This method works well because it utilizes the strength of symmetric encryption and the key exchange capabilities of asymmetric encryption. Some good examples of hybrid cryptographic systems include IPSec Secure Shell, Secure Electronic Transaction, Secure Sockets Layer, Pretty Good Privacy, and Transport Layer Security. With hybrid systems, can we achieve perfect secrecy? This depends on items such as the algorithm, how the key is used, and how well keys are protected. The concept of perfect forward secrecy (PFS) is based on the concept that the exposure of a single key will permit an attacker access to only data protected by a single key.

Hashing

Hashing refers to a broad category of algorithms that are useful for their ability to provide integrity and authentication. Integrity ensures that the information remains unchanged and is in its true original form. Authentication provides the capability to ensure that messages were sent from those you believed sent them and that the message is sent to its intended recipient.

Hashing and Message Digests

Hashing algorithms operate by taking a variable amount of data and compressing it into a fixed length value referred to as a hash value. Hashing provides a fingerprint or message digest of the data. A well-designed hashing algorithm will not typically produce the same hash value or output for two different inputs. When this does occur, it is referred to as a collision.

Collisions can be a problem in the world of hashing. A collision occurs when two different files create the same hashed output. One way to deal with collisions is to increase the size of the hashing algorithm output—for example, moving from SHA 160 to SHA 256 so that a larger hash is created.
Hashing can be used to meet the goals of integrity and non-repudiation depending on how the algorithm is implemented. While one of the advantages of hashing is its ability to verify that information has remained unchanged, it is also used in authentication systems and digital signatures. Figure 1.5 gives an overview of the hashing process.

**FIGURE 1.5** Hashing process

A hash is a one-way process and is not intended to be used to reproduce data. When a message or data file is hashed, the hashing algorithm examines every bit of the data while it is being processed. This means that if two files are close yet not exactly the same, their hashes will be different. For example, if I gave you a copy of a software program that had CASP study questions and you went to the Sybex website and downloaded the same software, hashing both files should result in the same value. An example of a cryptographic hash is shown in Figure 1.6. It can be seen after the text “MD5.”

**FIGURE 1.6** An example of a cryptographic hash on a software product

If there were even a slight change between the two files, the hashed values would be different. Comparing the hashes for the two files would indicate that the software I gave you had been altered. This same process is how programs such as Tripwire, MD5sum, and Windows System File Checker (sfc.exe) work. These kinds of programs can be used to monitor a file, folder, or an entire hard drive for unauthorized changes. You also see this process used for functions such as code signing. Code signing is the process of digitally signing executables and scripts to confirm the software author. Code signing also guarantees
that the code has not been altered or corrupted since it was signed by use of a hash. Listed here are some examples of hashing algorithms:

- Message Digest Algorithm (MD5) series
- Secure Hash Algorithm (SHA) series
- HAVAL
- RIPEMD
- Tiger
- MAC
- HMAC

**MD Series**

The MD algorithms are a series of cryptographic algorithms that were developed by Ron Rivest. These have progressed through the years as technology has advanced. The first was MD2, which is considered outdated. Another reason for its demise is that it was prone to collisions. MD4 was the next in the series. MD4 processes data in 512-bit blocks. As with MD2, MD4 was found to be subject to collisions and could potentially be vulnerable to forced collisions. These issues helped lead to the development of MD5. MD5 processes a variable-size input and produces a fixed 128-bit output. As with MD4, it processes the data in blocks of 512 bits. However, MD5 has also somewhat fallen from favor as it too has been shown to be vulnerable to collisions.

**SHA**

A Secure Hash Algorithm (SHA) is similar to MD5. Some consider it a successor to MD5 because it produces a larger cryptographic hash. SHA outputs a 160-bit message digest. SHA-1 processes messages in 512-bit blocks and adds padding, if needed, to get the data to add up to the right number of bits. SHA-1 has only 111-bit effectiveness. SHA-1 is part of a family of SHA algorithms, including SHA-0, SHA-1, and SHA-2. SHA-0 is no longer considered secure and SHA-1 is also now considered vulnerable to attacks. Some of the strongest versions currently available include SHA-256 and SHA-512.

**HAVAL**

HAVAL is another example of a one-way hashing algorithm that is similar to MD5. Unlike MD5, HAVAL is not tied to a fixed message-digest value. HAVAL-3-128 makes three passes and outputs a 128-bit fingerprint, and HAVAL-4-256 makes four passes and produces a fingerprint that is 256 bits in length.
Message Authentication Code

A message authentication code (MAC) is similar to a digital signature except that it uses symmetric encryption. MACs are created and verified with the same secret (symmetric) key. There are four types of MACs that you may come across in your career as a security professional: unconditionally secure, hash function–based, stream cipher–based, and block cipher–based.

HMAC

Sometimes hashing by itself is not enough, and in such situations a hashed message authentication code (HMAC) may be needed. HMAC was designed to be immune to the multi-collision attack. This functionality was added by including a shared secret key. Basically, HMAC functions by using a hashing algorithm such as MD5 or SHA-1 and then alters the initial state by adding a password. Even if someone can intercept and modify the data, it’s of little use if that person does not possess the secret key. There is no easy way for the person to re-create the hashed value without it.

Digital Signatures

*Digital signatures* are a category of algorithms based on public key cryptography. They are used for verifying the authenticity and integrity of a message. To create a digital signature, the message is passed through a hashing algorithm. The resulting hashed value is then encrypted with the sender’s private key. Upon receiving the message, the recipient decrypts the encrypted sum and then recalculates the expected message hash using the sender’s public key. The values must match to prove the validity of the message and verify that it was sent by the party believed to have sent it. Digital signatures work because only that party has access to the private key. Let’s break this process out step by step to help detail the operation:

1. Bob produces a message digest by passing a message through a hashing algorithm.
2. The message digest is then encrypted using Bob’s private key.
3. The message is forwarded to the recipient, Alice.
4. Alice creates a message digest from the message with the same hashing algorithm that Bob used. Alice then decrypts Bob’s signature digest by using Bob’s public key.
5. Finally, Alice compares the two message digests, the one originally created by Bob and the other that she created. If the two values match, Alice can rest assured that the message is unaltered.

Figure 1.7 illustrates the creation process. It shows how the hashing function ensures integrity and how the signing of the hash value provides authentication and non-repudiation.
The digital signature is hashed with the sender's private key. This helps prove that only the sender could have completed the signing process.

To help ensure your success on the CASP exam, integrity verification methods are reviewed in Table 1.3.

**TABLE 1.3** Attributes of symmetric and asymmetric encryption

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parity</td>
<td>Simple error detection code</td>
</tr>
<tr>
<td>Hashing</td>
<td>Integrity</td>
</tr>
<tr>
<td>Digital signature</td>
<td>Integrity, authentication, and non-repudiation</td>
</tr>
<tr>
<td>Hashed MAC</td>
<td>Integrity and data origin authentication</td>
</tr>
<tr>
<td>CBC MAC</td>
<td>Integrity and data origin authentication</td>
</tr>
<tr>
<td>Checksum</td>
<td>Redundancy check, weak integrity</td>
</tr>
</tbody>
</table>
Digital signatures are typically used within the Digital Signature Standard. The Digital Signature Standard makes use of the Digital Signature Algorithm. It makes use of SHA-1 and public key encryption.

Public Key Infrastructure

*Public key infrastructure* (PKI) allows two parties to communicate that were previously unknown to each other. It facilitates e-commerce. Consider how different dealing with brick-and-mortar businesses is from transactions over the Internet. Dealing with brick-and-mortar businesses gives you plenty of opportunity to develop trust. After all, you can see who you are dealing with, talk to the employees, and get a good look at how they do business.

In the modern world of e-commerce, transactions are much less transparent. You may not see whom you are dealing with yet might have full trust in them. PKI addresses these concerns and brings trust, integrity, and security to electronic transactions. One issue with key distribution is the nontechnical issue of controlling access to keys. Any PKI system has to be carefully controlled to ensure the wrong individuals don’t get access to secret keys.

From a user’s perspective, PKI may look seamless—yet in reality, it is made up of many components. PKI consists of hardware, software, and policies that manage, create, store, and distribute keys and digital certificates. The basic components of PKI are the following:

- The certificate authority (CA)
- The registration authority (RA)
- The certificate revocation list (CRL)
- Digital certificates
- A certificate distribution system

**Certificate Authority**

The CA is like a passport office. The passport office is responsible for issuing passports and is a standard for identification for anyone wanting to leave the country. CAs are like passport offices in that they vouch for your identity in a digital world. VeriSign, Thawte, and Entrust are some of the companies that perform CA services. The most commonly used model is the hierarchical trust model. An example is shown in Figure 1.8. In small organizations, a single trust model may be used. Its advantage is that it’s not as complex and has less overhead.
While those companies are external CAs, companies may also decide to tackle these responsibilities by themselves. Regardless of who performs the services, the following steps must be performed:

1. The CA verifies the request for certificate with the help of the RA.
2. The individual’s identification is validated.
3. A certificate is created by the CA, which verifies that the person matches the public key that is being offered.

**Registration Authority**

If the CA is like a passport authority, the RA is like a middleman. Think of it as one of the rush services you can use when you need to get your passport right away. The RA is positioned between the client and the CA. Although the RA cannot generate a certificate, it can accept requests, verify a person’s identity, and pass along the information to the CA for certificate generation.

RAs play a key role when certificate services are expanded to cover large geographic areas. One central CA can delegate its responsibilities to regional RAs such as having one RA in the United States, another in Canada, another in Europe, and another in India.

**Certificate Revocation List**

Just as with passports, digital certificates do not stay valid for a lifetime. Certificates become invalid for many reasons, such as someone leaving the company, information
changing, or a private key being compromised. For these reasons, the *certificate revocation list* (CRL) must be maintained.

The CRL is maintained by the CA, which signs the list to maintain its accuracy. Whenever problems are reported with digital certificates, they are considered invalid and the CA has the serial number added to the CRL. Anyone requesting a digital certificate can check the CRL to verify the certificate’s integrity.

**Digital Certificates**

Digital certificates are key to the PKI process. The digital certificate serves two roles. First, it ensures the integrity of the public key and makes sure that the key remains unchanged and in a valid state. Second, it validates that the public key is tied to the stated owner and that all associated information is true and correct. The information needed to accomplish these goals is added into the digital certificate.

Digital signatures play a key role in proving your identity when performing electronic transactions.

Digital certificates are formatted to the X.509 standard. The most current version of X.509 is version 3. One of the key developments in version 3 was the addition of extensions. Version 3 includes the flexibility to support other topologies such as *bridges* and *meshes*. It can operate as a web of trust, much like PGP. An X.509 certificate includes the following elements:

- Version
- Serial number
- Algorithm ID
- Issuer
- Validity
- Not before (a specified date)
- Not after (a specified date)
- Subject
- Subject public key information
- Public key algorithm
- Subject public key
- Issuer-unique identifier (optional)
- Subject-unique identifier (optional)
- Extensions (optional)

Figure 1.9 is an example showing some of these elements.
Different entities can use a certificate. The act of issuing certificates to end users and services is known as issuance to entities. *Issuance to entities* identifies who the CA issues certificates to. The certificate might be issued to a user, system, or an application. The CA not only issues the certificate but also vouches for the authenticity of entities. While it is not mandatory that you use an external CA to issue certificates, they are widely used. An organization may decide to have themselves act as a CA. Regardless of whether a third party handles the duties or your company performs them, digital certificates will typically contain the following critical pieces of information:

- Identification information that includes username, serial number, and validity dates of the certificates.
- The public key of the certificate holder.
- The digital signature of the signature authority. This piece is critical as it validates the entire package.

If you decide to use a third party to issue a certificate, there is the issue of cost. These organizations are generally for profit and will charge fees for you to maintain your certificate in good standing. Some organizations may choose to use wildcard certificates to cut costs. A *wildcard certificate* allows the purchaser to secure an unlimited number of subdomain certificates on a domain name. The real advantage is that you must buy and maintain one certificate. However, the drawback is that you are using just one certificate and private key on multiple websites and private servers. If just one of these servers or websites is compromised, all of the others under the wildcard certificate will be exposed.

If a private key is exposed or another situation arises where a certificate must be revoked, the PKI has a way to deal with such situations. That is when CRL is used. These lists can be checked via the Online Certificate Status Protocol. The *Online Certificate Status Protocol* (OCSP) is an Internet protocol used for obtaining the revocation status of an X.509 digital certificate. This process is much the same as maintaining a driver's license. Mike may have a driver's license, yet if he gets stopped by a police officer, the officer may still decide to run a check on Mike’s license; he’s checking on the status of Mike’s license in the same way that the OCSP is used to check on the status of X.509 certificate.
Chapter 1 • Cryptographic Tools and Techniques

If the topic of OCSP and certificates interest you, be sure to check out RFC 2560. This RFC details CRL and OCSP.

Certificate Distribution

Certificates can be distributed by a centralized service or by means of a public authority. The use of a CA is an example of centralized distribution: A trusted CA distributes a public key to another party. The certificate is signed by means of a digital signature of the CA to prove it is valid.

A second way to distribute keys is directly to a third party. This is called a web of trust. For example, if you email me with a question about the book, my return email will include my public key. It’s an easy way to distribute keys but does not offer the level of trust that would be obtained from a third-party CA such as VeriSign or Thwart. PGP and GnuPGP are examples of web-of-trust certificate distribution.

The Client’s Role in PKI

While the CA is responsible for a large portion of the work, in the world of PKI, the client also has some duties. Clients are responsible for requesting digital certificates and for maintaining the security of their private key. Loss, compromise, or exposure of the private key would mean that communications are no longer secure. Protecting the private key is an important issue because for the attacker it may be easier to target the key than to try to brute-force or crack the certificate service. Organizations should concern themselves with seven key management issues:

- Generation
- Distribution
- Installation
- Storage
- Recovery
- Change
- Control
- Disposal

Key recovery and control is an important issue that must be addressed. One basic recovery and control method is the \( m \) of \( n \) control method of access. The \( m \) of \( n \) control method is designed to ensure that no one person can have total control; it is closely related to dual control. Therefore, if \( n \) administrators have the ability to perform a process, \( m \) of those administrators must authenticate for access to occur. \( m \) of \( n \) control should require physical presence for access. Here is an example: Let’s say that a typical \( m \) of \( n \) control method requires that four people have access to the archive server and at least two of them must be
present to accomplish access. In this situation, $m = 2$ and $n = 4$. This would ensure that no one person could compromise the security system or gain access.

Real World Scenario

Trust in the World of PKI

Trust isn’t a problem in small organizations, but the need to communicate within large organizations or with external clients and third parties requires developing a working trust model. Organizations typically follow one of several well-known trust models, such as single-authority trust, hierarchical trust, or web of trust.

Each model has its advantages and disadvantages, and as a CASP, you may be asked to recommend a method to your organization. You should keep in mind that while a single authority model is simple, it’s not well suited for large organizations; if it is managed by the company, cross-certification to other entities can be an issue. A hierarchical model is typically provided by a commercial entity. While much more robust, there are associated fees that are ongoing.

Finally, there is the web of trust. This is the least complex of all models. While it may work well for an individual or small groups, it has a low level of trust. Which model will you choose for your company?

Cryptographic Solutions

Has this chapter got you thinking about all the ways cryptography can be used and how valuable it is to a security professional? I hope that it has. The real question is now that you’re armed with some very specific cryptographic solutions, where might you apply them? Cryptography can be used in many different situations to help build a true defense in depth. If we think of cryptography in reference to the TCP/IP model, we can see where cryptographic solutions can be applied, from the application layer all the way down to the physical frame. Let’s start at the top of the TCP/IP stack and work down through the layers, highlighting a few cryptographic solutions.

Application Layer Encryption

The following application layer protocols are just a few examples that can be used to add confidentiality, integrity, or non-repudiation:

Secure Shell (SSH)  SSH is an Internet application that provides secure remote access. It serves as a replacement for FTP, Telnet, and the Berkeley “r” utilities. SSH defaults to TCP port 22.
Secure Hypertext Transfer Protocol (S-HTTP)  S-HTTP is a superset of HTTP that was developed to provide secure communication with a web server. S-HTTP is a connectionless protocol that is designed to send individual messages securely.

Pretty Good Privacy (PGP)  PGP was developed in 1991 by Phil Zimmermann to provide privacy and authentication. Over time, it evolved into open standards such as OpenPGP and GnuPGP. PGP builds a web of trust that is developed as users sign and issue their own keys. The goal of PGP was for it to become the "everyman's encryption." Popular programs such as HushMail and Veridis are based on PGP.

Transport Layer Encryption

The transport layer of the TCP/IP stack can also be used to add cryptographic solutions to data communications. Some common examples follow:

Secure Sockets Layer (SSL)  Netscape developed SSL for transmitting private documents over the Internet. SSL is application independent and cryptographically independent since the protocol itself is merely a framework for communicating certificates, encrypted keys, and data.

Transport Layer Security (TLS)  TLS encrypts the communication between a host and a client. TLS consists of two layers—the Record Protocol and the TLS Handshake Protocol.

Wireless Transport Layer Security (WTLS)  WTLS encrypts the communication between a wireless host and a client. WTLS is a security protocol and is part of the Wireless Application Protocol (WAP) stack. WTLS was developed to address the problems surrounding mobile network devices. These issues will become increasingly important in the next few years as more and more people move to smartphones and use them for activities such as online banking.

Internet Layer Controls

The Internet layer is home to IPSec, a well-known cryptographic solution. IPSec was developed to address the shortcomings of IPv4. IPSec is an add-on for IPv4. IPSec can be used to encrypt just the data or the data and the header. With the depletion of IPv4 addresses, look for more attention to be paid to IPSec as it is built into IPv6. The components of IPSec include the following:

Encapsulated Secure Payload (ESP)  ESP provides confidentiality by encrypting the data packet. The encrypted data is hidden, so its confidentiality is ensured.
Authentication Header (AH)  The AH provides integrity and authentication. The AH uses a hashing algorithm and symmetric key to calculate a message authentication code. This message authentication code is known as the integrity check value (ICV). When the AH is received, an ICV value is calculated and checked against the received value to verify integrity.

Security Association (SA)  For AH and ESP to work, there must be some information exchanged to set up the secure session. This job is the responsibility of the SA. The SA is a one-way connection between the two parties. If both AH and ESP are used, a total of four connections are required. SAs use a symmetric key to encrypt communication. The Diffie–Hellman algorithm is used to generate this shared key.

Transport and Tunnel Mode  AH and ESP can work in one of two modes: transport mode or tunnel mode. Transport mode encrypts the data that is sent between peers. Tunnel mode encapsulates the entire packet and adds a new IP header. Tunnel mode is widely used with VPNs.

Physical Layer Controls

Now we have worked our way down to the bottom of the TCP/IP stack. As you’ve learned, there are many places to encrypt data. Encryption can happen at any one of many different layers. The question the CASP must ask is what is actually getting encrypted. Is the data itself secured or the data and all headers? Some physical layer security solutions include the following:

Password Authentication Protocol (PAP)  I have included PAP here but is should not be used. It is weak at best. PAP is not secure because the username and password are transmitted in clear text.

Challenge Handshake Authentication Protocol (CHAP)  CHAP is a more suitable option, as it sends the client a random value that is used only once. Both the client and the server know the predefined secret password. The client uses the random value, nonce, and the secret password and calculates a one-way hash.

Point-to-Point Tunneling Protocol (PPTP)  PPTP consists of two components: the transport that maintains the virtual connection and the encryption that ensures confidentiality. It can operate at a 40-bit or 128-bit length.

Layer 2 Tunneling Protocol (L2TP)  L2TP was created by Cisco and Microsoft to replace L2F and PPTP. L2TP merged the capabilities of both L2F and PPTP into one tunneling protocol.

Cryptographic Attacks

As long as there have been secrets there have been people trying to find out what these secrets are. Attacks on cryptographic systems are nothing new. The formal name for this activity is known as cryptanalysis. Cryptanalysis is really just the study of analyzing cryptography and attempting to determine the key value of a cryptographic system. Depending on which key is targeted by the attacker, it's possible that success may mean that someone could gain access to confidential information or pretend to be an authenticated party to a communication.

Many countries seek to control cryptographic algorithms and place controls on their use. These controls fall under the Wassenaar Arrangement on Export Controls for Conventional Arms and Dual-Use Goods and Technologies. You can read more about it at www.bis.doc.gov/wassenaar/default.htm. The goal of the agreement is to promote transparency in transfers of conventional arms and dual-use goods and technologies while also promoting greater responsibility in transfers of conventional arms and dual-use goods and technologies. The idea is to keep strong cryptography out of the hands of criminals and terrorists.

Real World Scenario

How Strong Is Your Password?

As a security administrator, you’ve no doubt heard many stories about how some people do very little to protect their passwords. Sometimes people write their passwords down on sticky notes, place them under their keyboards, or even leave them on a scrap of paper taped to the monitor. As a security professional you should not only help formulate good password policy but also help users understand why and how to protect passwords. One solution might be to offer password manager programs that can be used to secure passwords. Another approach is migration to biometric or token-based authentication systems.

For this scenario, you’ll need to put yourself in the position of an attacker wanting to see how strong your password is. From this perspective, you will test passwords with the following attributes:

- Create a password that is seven lowercase characters.
- Create a password that is seven upper- and lowercase characters.
- Create a password that is 14 upper- and lowercase characters and includes at least one special character.

Submit each of the examples to


and test the strength. What are your conclusions?
Summary

This chapter focused on cryptography. Cryptography is one of the most powerful tools that a security professional has. It offers you the ability to protect sensitive information through the use of encryption. It can also offer the ability to verify the integrity of patches, files, and important data. Cryptography also makes e-commerce possible. With cryptographic solutions such as PKI, you can have trust that a third party is who they claim to be. These are but a few of the solutions cryptography offers.

As a security professional, you need to able to communicate with others about cryptographic solutions and services. You don't have to be able to write your own cryptographic algorithm. You do need to be able to offer solutions to real problems. There is not a week that goes by without a news report that lists stolen or lost media that contained personal information. As a security professional, you may be in a position to suggest that your company use full disk encryption for all laptops. You may also have the opportunity to promote PGP as a standard to encrypt all email being used to discuss sensitive business dealings. You may even be on a team preparing to roll out a new e-commerce site and be asked to offer your opinion on PKI. These are the types of solutions that security professionals offer every day.

Exam Essentials

Be able to describe which cryptographic solution is appropriate for a given solution. Cryptographic solutions can be broadly divided into symmetric encryption, asymmetric encryption, hybrid encryption, and hashing. Each offers specific solutions such as privacy, authentication, integrity, and non-repudiation.

Be able to describe the basic operation of PKI and understand advanced PKI concepts. PKI allows two parties that are previously unknown to each other to communicate over an insecure public network. Such communications can then be used to securely and privately exchange data or for e-commerce. PKI systems make use of public and private keys. Keys are shared through a trusted certificate authority.

Know what terms such as wildcard mean when applied to PKI. A wildcard certificate allows the purchaser to secure an unlimited number of subdomain certificates on a domain name. The real advantage is that you must buy and maintain only one certificate. However, the drawback is that you are using just one certificate and private key on multiple websites and private servers.

Be able to describe transport encryption. Transport encryption is one of the two modes that IPSec can operate in. When using IPSec transport encryption, only the data portion or payload of each IP packet is encrypted. This leaves the IP header untouched and sent in the clear.
Be able to describe a digital signature. A digital signature is a hash value that has been encrypted with the private key of the sender. It is used for authentication and integrity.

Be able to describe hashing. Hashing refers to a broad category of algorithms that are useful for their ability to provide integrity and authentication. Hashing algorithms operate by taking a variable amount of data and compressing it into a fixed-length value referred to as a hash value.

Be able to describe code signing. Code signing is the process of digitally signing executables and scripts to confirm the software author. Code signing also guarantees that the code has not been altered or corrupted since it was signed by use of a hash.

Know how non-repudiation works. Non-repudiation is the ability to verify proof of identity. It is used to ensure that a sender of data is provided with proof of delivery and the recipient is assured of the sender’s identity.

Be able to define the concept of pseudorandom number generation. Pseudorandom number generators are algorithms that generate a sequence of numbers that approximates the properties of random numbers.

Be able to explain perfect forward secrecy. Perfect forward secrecy is based on the concept that the exposure of a single key will permit an attacker access to only data protected by a single key.

Define the terms confusion and diffusion. Confusion is the process that occurs when the relationship between the plain text and the key is so complicated that an attacker can’t alter the plain text and determine the key. Diffusion is the process that occurs when a change in the plain text results in multiple changes spread throughout the cipher text.
Review Questions

1. You have been asked by a member of senior management to explain the importance of encryption and define what symmetric encryption offers. Which of the following offers the best explanation?
   A. Non-repudiation
   B. Confidentiality
   C. Hashing
   D. Privacy and authentication

2. As the security administrator for your organization, you must be aware of all types of hashing algorithms. Which algorithm was developed by Ron Rivest and offers a 128-bit output?
   A. AES
   B. DES
   C. MD5
   D. RC4

3. A coworker is concerned about the veracity of a claim as the sender of an email denies sending it. The coworker wants a way to prove the authenticity of an email. Which would you recommend?
   A. Hashing
   B. Digital signature
   C. Symmetric encryption
   D. Asymmetric encryption

4. A junior administrator at a sister company called to report a possible exposed private key that is used for PKI transactions. The administrator would like to know the easiest way to check to see if the lost key has been flagged by the system. What are you going to tell the administrator?
   A. Hashing
   B. Issuance to entities
   C. Online Certificate Status Protocol
   D. Wildcard verification

5. You’ve discovered that an expired certificate is being used repeatedly to gain logon privileges. To what list should the certificate have been added?
   A. Wildcard verification
   B. Expired key revocation list
   C. Online Certificate Status Protocol
   D. Certificate revocation list
6. A junior administrator comes to you in a panic after seeing the cost for certificates. She would like to know if there is a way to get one certificate to cover all domains and subdomains for the organization. What solution can you offer?
   A. Wildcards
   B. Blanket certificates
   C. Distributed certificates
   D. No solution exists

7. Which of the following is not an advantage to symmetric encryption?
   A. It’s powerful.
   B. A small key works well for bulk encryption.
   C. It offers confidentiality.
   D. Key exchange is easy.

8. Most authentication systems make use of a one-way encryption process. Which of the following best offers an example of one-way encryption?
   A. Asymmetric encryption
   B. Symmetric encryption
   C. Hashing
   D. PKI

9. Which of the following is an early form of encryption also known as ROT3?
   A. Transposition cipher
   B. Substitution cipher
   C. Scytale
   D. Caesar’s cipher

10. Which type of encryption best offers easy key exchange and key management?
    A. Symmetric
    B. Asymmetric
    C. Hashing
    D. Digital signatures

11. SSL and TLS can best be categorized as which of the following?
    A. A symmetric encryption system
    B. An asymmetric encryption system
    C. A hashing system
    D. A hybrid encryption system
12. You're explaining the basics of cryptography to management in an attempt to obtain an increase in the budget. Which of the following is not symmetric encryption?
   A. DES
   B. RSA
   C. Blowfish
   D. Twofish

13. Which of the following is not a hashing algorithm?
   A. SHA
   B. HAVAL
   C. MD5
   D. IDEA

14. A mobile user calls you from the road and informs you that he has been asked to travel to China on business. He wants suggestions for securing his hard drive. What do you recommend he use?
   A. S/MIME
   B. BitLocker
   C. Secure SMTP
   D. PKI

15. You were given a disk full of applications by a friend but are unsure about installing a couple on your company laptop. Is there an easy way to verify if the programs are original or if they have been tampered with?
   A. Verify with a hashing algorithm
   B. Submit to certificate authority
   C. Scan with symmetric encryption
   D. Check the programs against the CRL

16. What is the correct term for when two different files are hashed and produce the same hashed output?
   A. Session key
   B. Digital signature
   C. Message digest
   D. Collision
17. You have been asked to suggest a simple trust system for distribution of encryption keys. You client is a three-person company and wants a low-cost or free solution. Which of the following would you suggest?
   A. Single authority trust
   B. Hierarchical trust
   C. Spoke/hub trust
   D. Web of trust

18. Which of the following would properly describe a system that uses a symmetric key distributed by an asymmetric process?
   A. Digital signature
   B. Hybrid encryption
   C. HMAC
   D. Message digest

19. What key is used by the sender to sign a digital signature?
   A. Symmetric key
   B. Public key
   C. Private key
   D. HMAC

20. While symmetric encryption suffers from problems such as key exchange, it offers which of the following as an attribute?
   A. Speed
   B. Scalability
   C. Integrity
   D. Availability