CISSP Video Mentor

Shon Harris

Pearson Education, Inc.

800 East 96th Street • Indianapolis, IN 46240

CISSP Video Mentor

ISBN-13: 978-0-7897-4030-4

ISBN-10: 0-7897-4030-3

Printed in the United States of America

First printing December 2009

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Pearson cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the CD or programs accompanying it.

Bulk Sales

Pearson offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact

U.S. Corporate and Government Sales

1-800-382-3419

corpsales@pearsontechgroup.com

For sales outside the U.S., please contact

International Sales

international@pearsoned.com

Publisher
Paul Boger

Associate Publisher
David Dusthimer

Acquisitions Editor
Jessica Goldstein

Development Editor
Andrew Cupp

Managing Editor
Patrick Kanouse

Technical Editor
Shawn Merdinger

Copy Editor
Water Crest Publishing

Publishing Coordinator
Vanessa Evans

Multimedia Developer
Dan Scherf

Designer
Gary Adair

Page Layout
Bronkella Publishing LLC

Contents at a Glance

Contents

About the Author

Shon Harris, CISSP, is the president of Logical Security, a security consultant, a former engineer in the Air Force's Information Warfare unit, an instructor, and an author. She has authored two best-selling CISSP books and was a contributing author to the books *Hacker's Challenge* and *Gray Hat Hacking*. Shon was recognized as one of the top 25 women in the Information Security field by *Information Security Magazine*.

About the Contributors

Crystal Bedell is a copywriter specializing in technology. She has 10 years of experience writing and editing study guides and articles for information technology professionals in security and networking. In addition, she writes B2B technology marketing materials such as white papers, case studies, and e-books. She offers marketing communications support and advice through her website, www.bedellcommunications.com.

Don Keeber is a solutions-oriented networking and security engineer with notable success delivering a broad range of corporate IT initiatives, while participating in planning and implementation of information-security solutions in direct support of business objectives.

Although most of Don's career has been on the delivery side, a significant segment was in a technical training role, during which he conducted classes for scores of IT professionals from several Fortune-500 organizations. The training programs were corporate-based certification tracks for Check Point Firewalls, RSA Security, and Nokia Security Specializations.

Don's current technical certifications include: $(ISC)^2$ CISSP #318146, Cisco CCSP, CCNP, CCDP, and CCIE-security (written only), Microsoft MCSE-2003+ Security, Checkpoint CCSE/CCSI - NG/NGX, RSA-SE/Instructor, INFOSec (NSA) Security Specialist, Novell CNE, Bluecoat-BlueNet, and CompTIA CTT (written part).

Michael J. Lester holds a master's degree in information systems security from Boston University (an NSA Center of Excellence), as well as around 20 industry certifications including CISSP, CISA, CCE #876, Security+, MCSE:Security, CCSE+, and ITIL. He is an author of *Gray Hat Hacking: The Ethical Hacker's Handbook* (McGraw Hill Publishing), a book on advanced penetration-testing techniques, and has written many other articles and presentations. He is the chief instructor for Shon Harris's Logical Security LLC, and he teaches and develops courses on CISSP, hacking and penetration testing, digital forensics and eDiscovery, CISA, and others. He is currently teaching, consulting, and authoring a book on digital forensics.

About the Technical Editor

Shawn Merdinger is an independent security researcher and consultant. In former corporate lives, he's worked with the Cisco Systems STAT Team and TippingPoint ZDI, as well as non-profit Austin FreeNet. His research in security has led to multiple CVE vulnerabilities, presenting at security conferences, and involvement with VOIPSA, the Voice Over IP Security Association. His current research focuses on medical device security, with more information at the LinkedIn MedSec group.

Dedication

For Dad.

Acknowledgments

I want to thank the Pearson group for their patience with me to get this product out. They are saints!

We Want to Hear from You!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

As an Associate Publisher for Pearson, I welcome your comments. You can email or write me directly to let me know what you did or didn't like about this book—as well as what we can do to make our books better.

Please note that I cannot help you with technical problems related to the topic of this book. We do have a User Services group, however, where I will forward specific technical questions related to the book.

When you write, please be sure to include this book's title and author as well as your name, email address, and phone number. I will carefully review your comments and share them with the author and editors who worked on the book.

Email: feedback@quepublishing.com

Mail: David Dusthimer
Associate Publisher
Pearson
800 East 96th Street
Indianapolis, IN 46240 USA

Reader Services

Visit our website and register this book at informit.com/register for convenient access to any updates, downloads, or errata that might be available for this book.

Introduction

The *CISSP Video Mentor* includes six lesson videos and seven lab videos, as follows:

- Lesson 1, “Cryptography Foundations”
- Lesson 2, “Symmetric Versus Asymmetric Algorithms”
- Lab 2-1: Public and Private Key Generation
- Lab 2-2: PGP Encryption and Signing
- Lab 2-3: Symmetric Encryption
- Lesson 3, “Hashing and Message Authentication Codes”
- Lab 3-1: Hashing and Message Authentication Codes
- Lesson 4, “IPSec and Its Configurations”
- Lab 4-1: IPSec with Windows Vista
- Lesson 5, “OSI Model and Protocols”
- Lab 5-1: Networking and the OSI Model
- Lesson 6, “802.11 Wireless”
- Lab 6-1: Configuring 802.11 WiFi

The lesson videos provide a conceptual understanding of some of the most important topics for the CISSP exams. The lab videos provide a more technical in-depth understanding of these CISSP exam concepts. The goal of this product is to help prepare you for the stringent and most sought-after security certification in the industry. The product also contains this booklet, which is supplemental to the content in the lesson videos. The goal of this booklet is to connect the conceptual items to the more technical topics in this product.

lesson 1

Cryptography Foundations

In this first lesson, you acquire a foundational knowledge necessary for understanding cryptography. We then build on that foundation to explore security services offered by different cryptography methods, cryptosystem components, and encryption and decryption processes. This chapter reinforces what you learn in the Lesson 1 video. Cryptography constitutes an entire domain within the (ISC)^2 Common Body of Knowledge, and it is necessary for you to understand these concepts going into the exam.

The term *cryptography* comes from the Greek *kryptos*, meaning "hidden, secret," and *grapho*, meaning "I write." Even in ancient times, people understood the value of protecting sensitive information by making it unreadable. We've made many advances in cryptography, and today it is considered a science. Cryptography is the practice of storing and sending data in a form that is unreadable by anyone and any process except those for which it is intended. Valuable information is protected by encoding it so that others cannot read or process it. Thus, cryptography is an effective method of protecting sensitive data both at rest and while it is transmitted over untrusted network communication paths.

Cryptography Definitions

There are a lot of terms thrown around in any discussion of cryptography. In order to fully grasp the technology—and pass the CISSP exam—you must understand the following definitions:

Access control—Restricting and controlling access attempts made by subjects and objects.

Algorithm—A set of mathematical and logical rules that dictate how enciphering and deciphering take place in an encryption method.

Cryptanalysis—The practice of breaking cryptosystems.

Cryptography—The science of storing and sending data in a form that is unreadable by anyone and any process except those for which it is intended.

Cryptology—The study of cryptography and cryptanalysis.

Cryptosystem—A system or hardware that carries out the process of encrypting and decrypting data.

Data origin authentication—Also known as system-based authentication, proving a message's source.

Decipher—Transforming ciphertext back into plaintext.

Encipher—Transforming data into ciphertext.

Entity authentication—Proving the identity of the sender of a message.

Key—A secret sequence of bits that dictates encryption and decryption.

Work factor—The time, effort, and resources necessary to break a cryptosystem.

Encryption, Decryption, and Cryptosystems

Information that can be read by humans or processed by a computer is referred to as *plaintext*. For example, a human can read a document in plaintext and a computer can process executable code in plaintext. Through a process called *encryption*, plaintext is transformed into ciphertext. Ciphertext appears to be random and cannot be understood by humans or computer systems. Because ciphertext cannot be deciphered until it is decrypted, the data is protected against unauthorized disclosure. This is particularly important when sending data over untrusted networks. When data is stored on computers, it is usually protected by logical and physical access controls. However, data is in a much vulnerable state when sent over networks that may not be protected.

The process of transforming ciphertext back into readable plaintext is called *decryption*. This whole process of encrypting and decrypting data is carried out by a cryptosystem. A cryptosystem can be a system or a product that is created through hardware components or program code in an application. PGP is an example of one popular cryptosystem available commercially, with GnuPG available as a free implementation of the OpenPGP standard. Cryptosystems can provide a number of services. By transforming plaintext into ciphertext and making it unintelligible by anyone or anything except for authorized recipients, a cryptosystem provides confidentiality. If a cryptosystem can ensure that data has not been modified in an unauthorized manner since it was created, transmitted, or stored, it provides integrity. Authentication services confirm the identity of the user or system that created the information. Authorization involves proving a user or service's identity and then providing that entity with access to the information. And finally nonrepudiation keeps the sender from denying that she sent the message. In order to make the whole encryption/decryption process happen, cryptosystems make use of an encryption algorithm, keys (also called cryptovariables), and various software components and protocols. Let's look at each of these in greater detail.

Encryption Algorithms and Keys

An *encryption algorithm* (also called a cipher) is a set of mathematical rules that dictate how enciphering and deciphering take place. The more complex the mathematics, the more complex the encryption process. However, the internal mechanisms of most algorithms in use today are publicly known. If the algorithm used by a cryptosystem is not a secret, something else must be: the key. The key is a value that comprises a sequence of random bits that is used by the algorithm to encrypt and decrypt data. This value is secret. You can think about it this way: The lock on the front door of your home is not the only one of its kind. The manufacturer has probably sold hundreds of thousands of that same lock. However, that doesn't mean that your neighbor, Bill, who has the same lock, can let himself into your house. Bill has a unique key to his lock, as do you, and every other owner of that lock. The key is the secret that keeps the lock secure.

So if an eavesdropper intercepts an encrypted message, all she can see is ciphertext. She may be able to determine what algorithm is being used and how that algorithm works, but that knowledge alone will not help her decrypt the message. She must have the key.

In encryption, a key is constructed by the algorithm, which contains a keyspace. A *keyspace* is a range of values that can be used to create a key. The larger an algorithm's keyspace, the greater the

number of values that can be used to create different keys. The greater the number of values, the more random the keys. And the more random the keys, the more difficult they are to uncover. Keeping in mind that we are working in binary and that 2^2 equals 4, consider a key length of 2 bits. The keyspace for the associated algorithm would be 4. So four different keys are possible. This small keyspace would not be very difficult to crack because there are few values to choose from. An attacker could easily determine the key value and use it to decrypt ciphertext. However, a large keyspace allows for more possible keys. As long as the encryption algorithm uses the entire keyspace and randomly chooses the values that make up the key, it will be more difficult to decipher.

Note

Cryptanalysis is the science of studying and breaking encryption methods. This includes breaking secret processes, such as key management; compromising authentication schemes; and reverse-engineering algorithms. Cryptanalysis can be carried out by both well-intentioned good guys as well as malicious attackers. Security researchers use cryptanalysis to identify flaws and weaknesses in cryptosystems and encryption methods so that they can be improved upon. This is an important piece of cryptography and cryptology if our methods are ever going to improve. However, when attackers use cryptanalysis to find weaknesses in cryptosystems, it is for the purpose of exploiting them to gain unauthorized access to confidential information.

When we discuss how easily a key can be cracked, we're talking about its strength. Strength can also be discussed in terms of the algorithm if it is not public. In fact, the overall strength of an encryption method comes from a variety of components: the secrecy and length of the key, the algorithm, the initialization vectors, and how they work together in the cryptosystem. The encryption method's strength corresponds to the processing power, resources, and time that must be dedicated to breaking the cryptosystem or cracking the key. When attackers attempt to break cryptosystems, they often execute a brute force attack on the key. This involves exhaustively attempting every combination of possible values until the right combination is uncovered. The difficulty of accomplishing this task is a reflection of the encryption method's strength.

An algorithm's strength also comes from its use of initialization vectors (IVs), random values that help prevent patterns from occurring during the encryption process. When used with keys, IVs ensure that two identical plaintext values (for example, a phrase such as "I love you") encrypted with the same key will not produce the same ciphertext. If IVs are not used, patterns can be detected in the ciphertext, making it easier for an attacker to break the encryption method and the key. The IV, along with the key, provides more randomness to the encryption process, thereby strengthening it. However, although some keys must be kept secret, it is not necessary for an IV to be kept private, and it therefore does not need to be encrypted when transmitted.

As you can see, there are several components that play a role in the strength of an algorithm or cipher. There are two main types of ciphers: block and stream. A stream cipher encrypts individual bits, whereas a block cipher encrypts blocks of bits. Block ciphers are commonly implemented at the software level because they require less processing power. Stream ciphers, on the other hand, require more randomness and processing power, making them more suitable for hardware-level encryption.

A strong stream cipher is characterized by the following:

- Long portions of bit patterns without repeating patterns within keystream values. The keystream must generate random bits.
- A keystream independent of the key. An attacker should not be able to determine the key value based on the keystream.
- An unpredictable keystream. The keystream must generate statistically unpredictable bits.
- An unbiased keystream. There should be as many 0s as there are 1s in the keystream. Neither should dominate.

Key Management

A cryptosystem's ability to provide confidentiality, integrity, and authentication hinges on the security of the keys. If the keys are compromised, the entire system could be compromised. It won't matter how complicated and complex the algorithm is.

Cryptography is a trust model based on keys. Users must trust one another to protect their own keys. They must trust the server that holds, maintains, and distributes keys. And they must trust the administrator in charge of key management. This is a lot to ask, considering keys can be captured, altered, corrupted, or disclosed to unauthorized individuals in the absence of proper precautions.

It should go without saying, then, that key management is a crucial part of cryptography. Unfortunately, it also tends to be the Achilles heel. Using keys to encrypt messages is just the tip of the key management iceberg. Fortunately, automation can be used to help with the many aspects of key management, which include ensuring the secure distribution of keys to the correct entities, continuously updating keys, ensuring their protection while in transit and while stored on workstations and servers, and ensuring that they are properly created, destroyed, and recovered.

Most keys are distributed today using a protocol via an automated process, relieving individuals from this task. However, that wasn't always the case. At one time, physical cryptographic keys were stored in secured boxes. An escorted courier was responsible for delivering them and, sometimes, distributing them to individual computers. Other times, the courier would deliver the keys to a main server and local administration would then distribute them to individual machines. In some implementations, a master key delivered by the courier would be used to generate unique secret keys for individual users. The fact that the courier was escorted emphasizes the need to transmit keys in a secure manner.

An example of automated key distribution is the Key Distribution Center (KDC) used by Kerberos (Kerberos is an authentication protocol that builds on symmetric key cryptography to enable networked systems to securely authenticate to one another). The KDC stores, distributes, and maintains session and secret keys. When Computer A wants to access a service on Computer B, Computer A submits its request to the KDC. The KDC then creates a session key that is used between the two computers. Automation reduces the potential for errors that can occur when humans are involved with manual processes. However, if the KDC's ticket-granting service (TGS) is compromised, all computers and their services are impacted and possibly compromised as well.

All this effort to securely distribute keys is for naught if the keys are carelessly stored by their recipients. That brings us to the next hurdle in key management: Keys must be securely stored before and after they are distributed to the proper entities. Keys should not be kept just anywhere on a user's desktop but instead should be securely placed within the file system where they can be stored and used in a controlled fashion. A module containing the key, algorithm, configurations, and parameters also needs to be protected. If an attacker gets a hold of any one of these components, she can use them to decrypt, read, and re-encrypt information not intended for her. In fact, anytime a key is stored outside the cryptosystem, it should be in ciphertext. Because many algorithms are publicly understood, it is important that the secret component of the cryptosystem—the key—be protected. If an attacker knows which algorithm a system is using and how that algorithm works, all she needs is the key to compromise the cryptographic system.

Secure storage of the user's key, as well as the other processes associated with key management, should be transparent to the user. They can be integrated into software and operating systems so that they are automated and hidden from the user. Depending on end users to perform these processes only asks for trouble. It adds complexity to the process and increases the possibility of errors being introduced or the processes not getting done at all.

Keys also need to be continually changed. This is one task that when manually managed usually gets neglected. Perhaps it is neglected with other key management tasks, or the network administrator charged with the responsibility is overwhelmed with other duties, or perhaps no one is aware that it should take place. At any rate, changing keys is an important security measure that should not be overlooked. How often should you change keys? That depends on how often a specific key is used and the level of security required from it. The more frequently a key is used, the greater the probability of it being captured and compromised. Conversely, the less frequently a key is used, the less likely it is to be captured and compromised. An independently owned coffee shop might change its cryptographic keys every six weeks, whereas a military unit might change theirs every week. Once you establish a schedule, stick to it and ensure that keys are changed using a secure method.

Because keys can become lost, destroyed, or corrupted, it is important that key backup and recovery are taken into account. Consider, for example, if the CEO encrypts a file outlining details of a merger and then promptly loses the key to decrypt it. He may not be happy to learn that he's lost the only key in existence that can decrypt his data, which is worthless to him now because it's permanently in ciphertext. Backup keys can also be vital if, for example, Sharon decides midway through her maternity leave that she's not coming back and her manager needs to decrypt a number of documents to pick up the pieces, or if Mike from accounting is escorted out of the building for stealing office supplies and reselling them on eBay.

As you can imagine, there are numerous other scenarios that highlight the need for backup copies of keys. But having more than one key also increases the chances that it will be compromised. Companies need to decide if it's worth the risk to back up keys and, if so, how they will be protected. Backup copies of keys should be securely stored but easily accessible when needed. Some cryptosystems feature key recovery options, whereas others require that copies of keys be securely stored elsewhere. Another option is to implement multiparty control for emergency key recovery. Multiparty control requires two or more individuals to recover a key. For example, three individuals may be required to present their authentication information in order to recover a key. To help reduce the potential for abuse, the individuals should come from different departments—it's best to

have someone from management, someone from security, and someone from the IT department. This way, collusion is necessary for fraud to occur.

Overall, automating processes is a more accurate and secure approach to key management than carrying them out manually within the network or security department. However, it might not be the right approach for every company. In deciding how to handle key management, a company should evaluate the overhead associated with key management processes, the level of security required for its cryptographic keys, and cost/benefit issues. If you're lucky, these factors will all point toward automation—or at least to someone other than yourself.

Key and Key Management Rules

There are plenty of dos and don'ts to help guide secure key management and hopefully make the job easier. The following is a summary of these rules:

- Key length should be long enough to provide suitable protection.
- The algorithm should use the full spectrum of the keyspace.
- Keys should be as random as possible.
- Security measures should be taken for the storage and transmission of keys.
- A key's lifespan should reflect its necessary level of protection.
- The more frequently a key is used, the shorter its lifespan.
- When their lifetime comes to an end, keys should be properly destroyed.
- Keys should be backed up or escrowed in case of emergencies.

Summary

In this lesson, you learned about cryptography foundations, the security services offered by different cryptography methods, cryptosystem components, and encryption and decryption processes. It is crucial that you understand these basic concepts before moving on to the next lesson, which delves deeper into the world of algorithms.

lesson 2

Symmetric Versus Asymmetric Algorithms

As you learned in Lesson 1, the two primary pieces of the encryption process are the algorithms and the keys. Remember, an *algorithm* is a group of mathematical equations that dictate how plaintext is turned into ciphertext, whereas a *key* is a string of random values that the algorithm uses to add randomness to the encryption process. In order for two entities to communicate via encryption, they must use the same algorithm. There are two types of cryptographic algorithms: symmetric and asymmetric. In this lesson, you learn the differences between symmetric and asymmetric algorithms, how they operate and the services they provide, their pros and cons, and how they are used together in hybrid systems. These concepts build upon those you learned in Lesson 1 and therefore delve deeper into the $(ISC)^2$ Cryptography domain of the Common Body of Knowledge.

Cryptography Definitions

The concepts behind algorithms are relatively straightforward, but the terminology can be confusing. Be sure you understand the various terms for the CISSP exam, as follows:

Asymmetric algorithm—A class of cryptographic algorithms that uses a public and private key for encryption and decryption.

Digital envelope—The use of asymmetric and symmetric algorithms together to provide a hybrid approach to cryptography.

Open message format—Data encrypted with a private (asymmetric) key, which can be decrypted with the corresponding public key.

Private key—The privately held key used in an asymmetric cryptosystem to encrypt and decrypt data. Only the owner of the private key should have it.

Public key—The publicly available key used in an asymmetric cryptosystem to encrypt and decrypt data.

Public key cryptography—Another name for an asymmetric cryptography.

Secret key—Another name for a symmetric key.

Secure message format—Data encrypted with a public (asymmetric) key, which can only be decrypted with the corresponding private key.

Symmetric algorithm—A class of cryptographic algorithms that uses identical keys for both encryption and decryption.

Symmetric Cryptography

Symmetric algorithms use symmetric, or secret, keys. Each sender and receiver communicating via symmetric cryptography shares a common key that is used for both encryption and decryption. Thus, each key has dual functionality. For example, Bob sends a message encrypted with his key to Susan. Susan uses her copy of the same key to decrypt Bob's message, and then encrypt her reply to him.

In symmetric cryptography, it is necessary for each user to keep her key private. Thus, symmetric keys are also called secret keys. The integrity of the system hinges on the secrecy of the keys. If Bob's key is compromised, the new owner (attacker) of that key can encrypt Bob's messages from Susan and send encrypted messages back to her.

Each user pair in a symmetric cryptosystem requires a unique key, which leads us to one disadvantage of symmetric algorithms: complicated key management. Consider this: Bob requires a unique key for each colleague with whom he wants to exchange encrypted messages. Over the course of a month, Bob communicates with 100 different people. How does Bob keep track of 100 keys and ensure that he uses the correct key for each colleague? We're only talking about Bob here. Let's consider a pool of 100 users who all need to communicate with one another. To determine the number of symmetric keys, use the following calculation:

N(N-1)/2 = Number of keys

The total number of keys for a pool of 100 users is 4,950 keys!

To further complicate matters, the keys must be distributed using an out-of-band method. In other words, the keys cannot be sent to the appropriate users over the same communication channel. For example, a key sent via email can be easily intercepted because it is not protected. So the keys must be distributed via other means. A key can be saved to a thumb drive and either hand delivered or sent via secure message courier to the appropriate users. However, neither method ensures the security of the key. And after the users have received their keys, there is no way to ensure that the correct people have ownership of the keys because symmetric cryptosystems cannot provide authentication or nonrepudiation. Because both the sender and recipient use the same key, there is no way to prove who actually sent a message. Bob could be communicating with Susan—the rightful owner of his corresponding key—or he could be communicating with Hacker Jack, who intercepted Susan's key while it was being delivered to her.

You can see that key management is a significant disadvantage to symmetric cryptosystems. However, they do have their benefits. Symmetric algorithms offer enhanced speed and security over asymmetric systems. Symmetric algorithms can efficiently encrypt and decrypt large amounts of data that asymmetric algorithms cannot, because symmetric algorithms use simpler mathematical functions to encrypt and decrypt data. Symmetric algorithms use substitution and transposition (in other words, they scramble) to create ciphertext, which do not require a lot of processing power. However, symmetric algorithms carry out substitution and transposition many times. If a large key size is used, it is also more difficult to uncover data encrypted with a symmetric algorithm than it is with an asymmetric algorithm. For many applications, symmetric key cryptography is the only option.

Pros and Cons of Symmetric Algorithms

The pros of symmetric algorithms include the following:

- They are faster than asymmetric systems and therefore more capable of encrypting large amounts of data.
- It is difficult to uncover encrypted data if using a large key size.

The cons of symmetric algorithms include the following:

- Key management can quickly become overwhelming because each user pair requires a unique key.
- Keys must be securely delivered via an out-of-band method.
- Symmetric cryptosystems do not provide authentication or nonrepudiation.

Note

Examples of symmetric algorithms include the following:

- Data Encryption Standard (DES)
- Triple-DES (3DES)
- Advanced Encryption Standard (AES)
- Blowfish
- IDEA
- RC4, RC5, and RC6

Asymmetric Cryptography

In asymmetric cryptosystems, each entity in a user pair has a different but mathematically-related key. A message encrypted with one key is decrypted with the corresponding key. Each key pair consists of a public key, which can be known or possessed by anyone, and a private key, which must be kept secret and used only by the owner. Asymmetric cryptography is also referred to as public key cryptography.

Public keys are often available to the public via directories and databases of email addresses so that they can be obtained by anyone who wants to exchange secure communications with a particular person. Although public keys are mathematically-related to their private counterparts, it should not be feasible for someone to be able to figure out a private key based on the corresponding public key. However, the private key should be kept just that—private. If the private key is compromised, it should be assumed that all communications involving that key are also compromised.

Here's how it works (follow along carefully because this is where it can get confusing). Linda is using a public key. She encrypts data for John with her private key. Because anybody with her public key can decrypt the data, this is called open message format. It does not ensure confidentiality. However, because Linda's public key will only decrypt data encrypted with her private key, John can rest assured that the message did indeed come from Linda, thus providing authentication.

John can encrypt a reply to Linda using her public key. Linda decrypts the message with her private key. This assures confidentiality and is called a secure message format because only the entity with the corresponding private key can decrypt the message.

Note

In symmetric cryptography, the same key is used to encrypt and decrypt data. That is not the case in asymmetric cryptography. The public and private keys are not the same. Although they can both be used to encrypt and decrypt data, the same key cannot be used to carry out both functions on the same data. A message encrypted with a private key must be decrypted with the corresponding public key, and vice versa.

Note

Asymmetric keys provide authentication; however, symmetric keys do not because the exact same key is used by two entities. The use of one of the secret keys to encrypt a message does not ensure that the message came from a specific entity, thereby providing authentication.

As we explained previously, symmetric algorithms use relatively simple mathematics and are therefore faster than asymmetric algorithms. Asymmetric algorithms use complex mathematics, which requires more processing time. However, depending on the algorithm used, asymmetric algorithms can provide authentication and nonrepudiation where symmetric algorithms cannot. Asymmetric algorithms also offer the benefit of easier key management. A public key is distributed to everyone the entity needs to communicate with. Because it is public, the key can be distributed via email or public directories.

Pros and Cons of Asymmetric Algorithms

The pros of asymmetric algorithms include the following:

- More efficient key distribution than symmetric algorithms.
- Can provide authentication and nonrepudiation.
- Improved scalability over symmetric systems.

The cons of asymmetric algorithms include the following:

- Due to complicated mathematics, asymmetric algorithms work slower than symmetric.
- Mathematically-intensive tasks require more processing power.

Note

Examples of asymmetric algorithms include the following:

- RSA
- Elliptic Curve Cryptosystem (ECC)
- Diffie-Hellman
- El Gamal
- Digital Signature Algorithm (DSA)

Hybrid Encryption Methods

Luckily, asymmetric and symmetric algorithms do not need to be mutually exclusive. They can be used together in hybrid encryption methods to get the best of both: the speed of symmetric algorithms combined with asymmetric's ability to scale, confidentiality services, and streamlined key management.

In a hybrid cryptosystem, asymmetric and symmetric algorithms perform different functions based on their strengths. A symmetric algorithm is used to create keys for encrypting bulk data, whereas an asymmetric algorithm is used to create keys for automated key distribution. Here's how it works: You use a symmetric key to encrypt a message to Mike. In order to decrypt the message, Mike needs a copy of the symmetric key. So you encrypt the key with an asymmetric key. This protects the key from being intercepted by eavesdroppers. Because your message is probably longer than the key, the faster algorithm (symmetric) is used to encrypt the message, whereas the slower algorithm (asymmetric) is used to encrypt the asymmetric (public) key.

Now let's take a closer look at this process. You've encrypted a message with a symmetric key, resulting in ciphertext. Next, you encrypt a copy of the symmetric key needed to decrypt the message with Mike's public key. Doing so ensures that only Mike can decrypt the symmetric key because only he has the corresponding private key. (If you encrypted the symmetric key with your private key, anyone with the corresponding public key could decrypt it.) Mike receives your message. He uses his private key to decrypt the symmetric key and then uses the symmetric key to decrypt your message.

All of this is not to say that you have to manually search for the symmetric and public keys to encrypt your message and symmetric key, respectively, nor does it imply that Mike has to manually apply the decrypted symmetric key to decrypt your message. All of this occurs behind the scenes by software. But you still need to understand the mechanics of it all for the CISSP exam! Moreover, this hybrid approach (or digital envelope, as it is commonly called) is the foundation for cryptography. It is used whenever data and a symmetric key must be securely transmitted.

So, in summary, a hybrid system uses a symmetric algorithm to create a secret key that is used to encrypt bulk data (the message), and an asymmetric (public) key is used to encrypt the secret key.

A Few Pointers about Hybrid Cryptosystems

Keeping asymmetric and symmetric algorithms and digital envelopes straight can be difficult at first. Remember these key points, and you'll be there in no time:

- A symmetric algorithm uses a secret key to perform both encryption and decryption on a single set of data.
- An asymmetric algorithm uses mathematically-related public and private keys to perform encryption and decryption.
- The symmetric (secret) key is used to encrypt the message in a digital envelope.
- The public (asymmetric) key is used to encrypt the symmetric key for secure key distribution in a digital envelope.

Summary

Having completed this lesson, you should have a solid understanding of symmetric and asymmetric algorithms. You should be able to describe how each type of algorithm works and the services they provide. In addition, you should be able to explain the use of secret, public, and private keys in a digital envelope. Just to make sure you understand these concepts, consider the following questions and then check your answers. If you have difficulty with any of them, revisit CISSP Video Mentor Lesson 2:

1. What security service is provided when the sender of a message encrypts data with the recipient's private key?
2. What security service is provided when a sender encrypts a symmetric key with the recipient's public key?
3. What security services are provided when the sender encrypts data with her private key?
4. Why is the symmetric key encrypted with an asymmetric key?
5. Why isn't the message encrypted with an asymmetric key?

Answers:

1. None. Only the recipient should have access to her private key.
2. Confidentiality. Only the corresponding private key can decrypt the symmetric key and only the recipient has the private key.
3. Authenticity and nonrepudiation. The sender cannot deny having sent the message because only she has the private key to encrypt it in the first place (nonrepudiation). Using the corresponding public key authenticates the message.
4. It is encrypted to provide secure key transmission.
5. Asymmetric keys are too slow.

lesson 3

Hashing and Message Authentication Codes

In this third lesson of the *CISSP Video Mentor*, we explore hashing and message authentication codes. Specifically, you learn how hashing algorithms and message authentication code work, the services they provide, and attacks against them. These concepts build upon the previous two lessons; thus, we are still within the cryptography domain of the $(ISC)^2$ Common Body of Knowledge. It is important that you understand these concepts, as they may appear on the exam.

Cryptography Definitions

The concepts behind hashing algorithms are relatively straightforward, but the terminology can be confusing. Be sure you understand the various terms for the CISSP exam, as follows:

Birthday attack—A type of brute-force attack carried out against one-way hash functions to force a collision.

Cipher—Another name for algorithm.

Collision—When a one-way hash function generates the same message digest for two different messages.

Data origin authentication—The weakest form of authentication, which is provided by MAC functions, because it authenticates a key to a system rather than an individual.

Fingerprint—Another name for a hashing value.

Message digest—The hash value generated by a one-way hash.

One-way hash—A function that takes a variable-length string (a message) and produces a fixed-length value called a hash value.

System authentication—Another name for digital origin authentication.

The One-Way Hash

A *one-way hash* is a function that takes a variable-length string (a message) and produces a fixed-length value called a hash value. A one-way hash ensures the integrity of a message by allowing the recipient to verify that it has not been modified in transit. For example, Brad is sending a message filled with office gossip to Alice. He wants to make sure that the message isn't changed in any way while it is transmitted, so he uses a one-way hash to calculate a message digest, which he then appends to the message itself. When Alice receives the email, she uses the same hashing algorithm

that Brad used; if the message digest is the same as that sent with the message, she can rest assured that the message was not altered and that her boss really is dating the CEO. However, if the message digests differ, Alice knows that the message has been altered in an unauthorized manner and it should be discarded. No one wants to spread gossip that isn't true.

Hashing algorithms are publicly known. Their secrecy lies in that they only operate in one direction. They are never used in reverse. The recipient runs the same hashing function in the same direction and compares the results.

The goal of a one-way hash function is to provide a fingerprint (hash value) of the message. Like human fingerprints, the message fingerprint should be unique. No two messages should share the same fingerprint or, in this case, hash value. If they did, patterns would be revealed, making it easier for an attacker to compromise the hash function. Hashing algorithms can take steps to ensure that they do not generate the same hash value for more than one message. These algorithms are described as collision free.

In addition to being collision free, a good hash function should have the following characteristics:

- The hashing function should accept as input the entire message.
- The hash should be a one-way function to prevent disclosure of messages by their values.
- If provided with a message and its hash value, it should be impossible to generate the same hash value for another message.
- The hashing function should be resistant to birthday attacks.

Because one-way hash functions do not use keys, they cannot provide authentication or nonrepudiation. A message that has been run through a one-way hash and has a message digest appended to it can still be intercepted and modified. The eavesdropper can alter the message, recalculate the message digest, append the value to the message, and forward it to the intended recipient. Because the recipient comes up with the same message digest, he or she will assume that message was not altered. To avoid this scenario, the sender can use a message authentication code (MAC).

Message Authentication Code

A MAC function applies a secret key to a variable-length message and generates a value for that message, thereby proving a message's authenticity as well as integrity. There are two basic types of MACs that you need to be aware of for the CISSP exam:

- Hash MAC (HMAC)
- Cipher block chaining MAC (CBC-MAC)

Table 3-1 summarizes the services provided, and the sections that follow cover the two types in more detail.

Table 3-1 Services Provided

Function	Services Provided
Hash	Integrity
HMAC	Integrity and data origin authentication
CBC-MAC	Integrity and data origin authentication

Hash MAC

When using an HMAC, a symmetric key is concatenated to the message. Both the message and the secret key are put through a hash algorithm to create an independent MAC value, which is appended to the message and transmitted to the recipient. If an attacker were to intercept the message and alter it, she would not have the symmetric key to create the MAC value that the recipient will attempt to create.

The steps of the hashing process are as follows:

1. The sender puts the message through a hashing function.
2. A message digest is generated.
3. The message digest is appended to the message.
4. The message is transmitted to the recipient.
5. The recipient puts the message through the hashing function.
6. The recipient generates a message digest.
7. The recipient compares the two message digests. If they are the same, the message has not been modified.

The steps of the HMAC process are as follows:

1. The sender concatenates a secret key to the message.
2. The message and key are put through a hashing algorithm.
3. A MAC value is generated.
4. The MAC value is appended to the message.
5. The message and MAC value are transmitted to the recipient.
6. The recipient concatenates a copy of the same secret key with the message.
7. The recipient puts the message and key through a hashing algorithm.
8. The recipient compares the two MAC values. If they are the same, the message has not been modified.

Note

Don't get confused by the role of the symmetric key in an HMAC function. The key is not being used to encrypt the message, nor is it being sent with the message to the recipient. It is simply attached to the message before it is put through the hashing function. Because the key is not transmitted with the message, it is necessary for the recipient to have the same symmetric key to confirm the MAC value. The HMAC function itself does not offer a means of distributing the key.

Cipher Block Chaining MAC

A CBC-MAC is a technique for generating a MAC from a symmetric cipher. The sender encrypts the message with a symmetric block cipher in cipher block chaining (CBC) mode and the final output is used as the MAC. The MAC is appended to the plaintext version of the message, which is then transmitted to the recipient. The recipient encrypts the plaintext message with the same symmetric block cipher in CBC mode to generate a MAC value, which is compared to the value sent with the message. Unlike HMAC, CBC-MAC does not use a hashing algorithm.

Because a copy of the symmetric key is necessary to generate a MAC value to verify the integrity of the message, an eavesdropper cannot intercept the message, change the data, or replace the MAC value with a new one. No one except the other person with a copy of the symmetric key can generate the MAC value.

Symmetric keys are bound to a computer system or device, unlike private keys, which are bound to individuals. Therefore, the recipient of the message knows that it came from the system (not the user) with the other copy of the symmetric key. This form of authentication is called data origin authentication or system authentication, and it is the weakest form of authentication.

The steps of the CBC-MAC process are as follows:

1. The sender encrypts the message with a symmetric cipher in CBC mode.
2. The last block is used as the MAC.
3. The sender transmits the plaintext message and the MAC to the recipient.
4. The recipient encrypts the message with her copy of the symmetric key to generate a new MAC value.
5. The recipient compares the two MAC values. If they are the same, the message has not been modified.

Types of Hashing Algorithms

One-way hashing functions MD2, MD4, and MD5 are the brain children of Ron Rivest. Each of these creates a 128-bit message digest. MD2 is much slower than its brethren, whereas MD4 is used for high-speed computations in software and is optimized for microprocessors. MD5 provides the highest level of security. Its algorithm and mathematical operations are more complex. Also, a fourth round of operations is performed by MD5 during the hashing functions.

Secure Hashing Algorithm (SHA) was designed by NIST and NSA when a more secure hashing algorithm was needed for U.S. government applications. Although it is similar to MD4, SHA has some extra mathematical computations and generates a 160-bit hash instead of a 128-bit hash. Thus, it is more resistant to brute-force attacks.

SHA was designed to be used with the Digital Signature Standard to create digital signatures. SHA has been improved upon and new versions have been released, including SHA-1 (produces a 160-bit message digest), SHA-256 (produces a 256-bit message digest), SHA-384 (produces a 384-bit message digest), and SHA-512 (produces a 512-bit message digest).

A modification of MD5, HAVAL is a variable-length one-way hash function that processes blocks of 1024 bits—twice the size of those used in MD5. HAVAL was designed to provide added protection against attacks that succeed against MD5.

In 1995, Ross Anderson and Eli Biham developed a hashing algorithm to be used on 64-bit systems that would be faster than MD5 and SHA-1. The result was Tiger, and its message digest is 192 bits. Unlike most hash algorithms that have been built upon the MD4 architecture, Tiger was built upon a unique architecture to protect it against attacks that are successful against other hashing algorithms.

Hash Function Attacks

As explained previously, a good hashing algorithm should not produce the same message digest for more than one message. When an algorithm produces the same message digest for two messages, we refer to this as collision. When executing a birthday attack, an attacker attempts to force a collision.

An Example

Kevin has written his will and has bequeathed more than half of his belongings to his favorite niece, Jane, instead of dividing them up equally among his three children. To ensure that the will is not modified, the will is put through a hashing function and a message digest is created.

Kevin's oldest son, Steve, finds the will in his Dad's desk and decides to make some changes. He begins by making a copy of the message digest. Then he writes up a new will that states that he gets 50 percent of his father's assets, while his siblings split the rest—leaving Jane with nothing. Steve hashes the new will and compares the new message digest with the original, only to find that they don't match. So Steve makes a minor change to the will, hashes it again, and compares the message digests. He continues to do this, making small changes to the will and running it through a hashing algorithm, until he forces a collision and the modified will creates the same message digest as the original.

When Kevin dies, Steve collects 50 percent of his father's assets. Jane objects, but because the message digests match, the siblings claim that no changes have been made to the document.

Note

The output of a hashing algorithm is *n*, and to find a message through a brute force attack that results in a specific hash value would require hashing 2^n random messages. To take this one step further, finding two messages that hash to the same value would require review of only $2^{n/2}$ messages.

Summary

In this lesson, we delved deeper into the world of cryptography to explore one-way hashing algorithms and message authentication codes. At this point, you should have a solid understanding of how one-way hash and MAC functions work, the services they provide, and how birthday attacks are executed. This wraps up the Video Mentor Labs on cryptography. Next, we take a look at IPSec and its configurations.

lesson 4

IPSec and Its Configurations

In this fourth lesson, we explore the protocol IPSec, which has become the de facto tunneling standard for virtual private networks (VPNs). Thus, we begin the lesson with a detailed look at how VPNs work and the technologies used in their creation. In addition to IPSec, you learn about Point-to-Point Tunneling Protocol (PPTP) and Layer 2 Tunneling Protocol (L2TP). Finally, you learn about the inner workings of IPSec, from its main protocols and components to the security services it offers and key management.

Cryptography Definitions

As we delve into the world of VPNs and IPSec, you'll encounter still more cryptography terms and concepts. Here are some of the terms you need to know for this lab and the corresponding portions of the CISSP exam:

Authentication Header (AH)—An authentication protocol used in IPSec that guarantees integrity and data origin authentication of IP packets.

Diffie-Hellman key exchange algorithm—A cryptographic algorithm that allows two entities with no prior knowledge of one another to establish a shared secret key over an insecure communications channel like the Internet.

Encapsulating Security Payload (ESP)—An authentication and encryption protocol used in IPSec that uses cryptographic mechanisms to provide data origin authentication, integrity, and confidentiality.

Integrity check value (ICV)—A fixed-size value computed from a block of data for the purpose of detecting modifications during the data's transmission.

Internet Key Exchange (IKE)—The protocol used to establish a security association in the IPSec protocol.

Internet Protocol Security (IPSec)—A framework for a set of network-layer protocols that provides a method for establishing a channel to securely transmit data between two devices.

Internet Security Association and Key Management Protocol (ISAKMP)—A key exchange architecture used by IPSec that is independent of the type of keying mechanisms used.

Layer 2 Tunneling Protocol (L2TP)—A tunneling protocol that supports VPNs and relies upon an encryption protocol to provide confidentiality.

MS-CHAP—Microsoft's version of the challenge-handshake authentication protocol (CHAP), which is used to authenticate one entity, such as a user or network host, to another entity.

Network address translation (NAT)—The process of changing a packet's network address information while it is in transit from one IP address to another.

OAKLEY—A key-agreement protocol that allows two or more entities to negotiate which keys will be used without the need of a third party.

Point-to-Point Protocol (PPP)—A data link protocol that is used to establish a direct connection between two network nodes.

Point-to-Point Tunneling Protocol (PPTP)—A tunneling protocol that creates a VPN over a PPP connection.

Security association (SA)—The establishment of shared security information between two entities to support secure communication.

Virtual private network (VPN)—A secure, private network that is established over an untrusted network.

Virtual Private Networks

A virtual private network (VPN) is a secure, private computer communication channel that is established over a public telecommunication infrastructure like the Internet or other unsecure environments. VPNs are often used by telecommuters and road warriors to securely connect to a corporate network, where they can access email and other network resources.

A VPN uses tunneling protocols and encryption, both of which ensure the confidentiality and integrity of data being transmitted. The protocols used for VPNs are IPSec, Point-to-Point Tunneling Protocol (PPTP), and Layer 2 Tunneling Protocol (L2TP). In order to set up an encrypted tunnel, both ends of the VPN must have the required hardware and software. The encrypted tunnel provides the private link. The encryption protocol encrypts the data, protecting it as it is transmitted through an untrusted network.

In order to use a VPN, users must have the appropriate software loaded on their computers. In many cases, the user makes a connection to an ISP using Point-to-Point Protocol (PPP). In turn, the ISP makes a full connection between the user and destination network or system. PPP is a data link protocol that is used to establish a direct connection between two network nodes; thus, it can form the foundation for the VPN connection. It also encapsulates packets for their transmission over the telecommunication link. Once the connection is made between the user and the destination network, the ISP has completed its responsibilities in the VPN process. The user's software initiates a VPN connection, and the two network nodes go through a handshaking process to establish the type of encryption and components that will be used to encrypt data. After the VPN parameters are negotiated, the user and network can begin communicating through the secure virtual connection.

VPNs can also provide a connection between two routers (in what is called a gateway-to-gateway connection) or between two firewalls that have VPN functionality. A gateway-to-gateway connection requires connecting entities to have the VPN software and the appropriate protocols and encryption mechanisms. The user can access network resources after the VPN connection is made. VPN connections between two firewalls, on the other hand, sit on the outer edge of a network's security domain. A firewall with embedded VPN functionality allows the company to centralize administration for both technologies.

Tunneling Protocols

Now that you have a general understanding of how a VPN is established, let's take a step back and take a closer look at tunneling and tunneling protocols. A VPN may be used to either provide encapsulation or encapsulation and encryption. A tunnel is a virtual path through a network that delivers encapsulated or encapsulated and encrypted packets. Encapsulation allows a payload to be delivered over an incompatible delivery network. For example, if a network that uses NetBIOS Enhanced User Interface (NetBEUI) needs to connect to another network that also uses NetBEUI, which is nonroutable, the packets can be encapsulated within a routing protocol, such as IP, to allow these networks to communicate.

As mentioned, tunnels can also be used to provide both encapsulation and encryption. The encapsulation is still used to send packets through an incompatible network, whereas the encryption provides data confidentiality and integrity.

VPNs create their connection with a tunnel. The three main tunneling protocols used in VPN connections are PPTP, L2TP, and IPSec.

PPTP

Some protocols cannot be transmitted over specific networks because the networks simply do not understand them. For example, the Internet does not understand IPX, NetBEUI, and AppleTalk—among others. So, in order for these frames to cross the Internet, they need assistance. The tunneling protocol provides that help.

Point-to-Point Tunneling Protocol (PPTP) was created by Microsoft to allow remote users to establish a PPP connection to a local ISP, and then create a VPN connection.

Note

The CISSP may use antiquated protocols in their examples, as used in this lesson.

Note

For years, PPTP was the de-facto industry standard tunneling protocol, but it has been replaced by IPSec.

Note

PPTP only works over IP networks. That means that other protocols have to be used to move data over frame relay, X.25, and ATM links. Cisco Systems began developing a protocol called Layer 2 Forwarding (L2F) to tunnel PPP traffic through non-IP networks. However, at IETF's request, Cisco combined its work with PPTP for the sake of interoperability. The result is Layer 2 Tunneling Protocol (L2TP).

L2TP

L2TP provides the similar functionality as that provided by PPTP. Its advantage is that it is not limited to working over IP networks. It can also provide more security when used with IPSec. However, L2TP does not provide encryption or authentication on its own. IPSec must be used to provide those services.

Table 4-1 compares the differences between PPTP and L2TP.

Table 4-1 Differences Between PPTP and L2TP

PPTP	L2TP
Runs only within IP networks.	Runs within networks other than IP, including frame relay and X.25.
Is an encryption protocol.	Is not an encryption protocol; must be used with IPSec for encryption services.
Does not support TACACS+ and RADIUS.	Supports TACACS+ and RADIUS.

IPSec

Internet Protocol Security (IPSec) is a framework for a set of network-layer protocols that provides a method for establishing a channel to securely transmit data between two devices. Those devices can be two hosts (such as end user computers or servers), a pair of security gateways (such as routers or firewalls), or a gateway and a host.

IPSec is mandatory in all standards-compliant implementations of IPv6; however, it has also become a widely-accepted standard for securing IPv4 traffic. Unlike other Internet security systems such as SSL and TLS, IPSec can be used to protect any application traffic transmitted across the Internet. The suite of protocols does not need to be incorporated into the design of the applications, as do SSL and TLS. Furthermore, because IPSec is an open, modular framework, it tends to be more flexible and less expensive than end-to-end and link encryption methods.

As a framework of open standards, IPSec uses two basic security protocols to provide strong encryption: authentication and integrity. The authenticating protocol is Authentication Header (AH). It guarantees connectionless integrity and data origin authentication of IP packets. Encapsulating Security Payload (ESP) is an authentication and encryption protocol. It uses cryptographic mechanisms to provide data origin authentication, integrity, and confidentiality.

As previously mentioned, IPSec can work in one of two modes: tunnel mode or transport mode. IPSec can work in one of two modes: transport mode, in which the payload of the message is protected, and tunnel mode, in which the payload and the routing and header information are protected. ESP in transport mode encrypts the actual message information so that it cannot be sniffed and uncovered by an unauthorized entity. Tunnel mode provides a higher level of protection by also protecting the header and trailer data that an attacker may find useful.

Establishing an IPSec Connection Between Two Computers

The steps to establish an IPSec connection between two computers are as follows:

1. One device requests an IPSec connection with another device.
2. Systems conduct handshakes to determine hashing and cryptographic algorithms, keys, and other parameters.
3. Each device sets up a service association for the connections.
4. Each device authenticates, hashes, encrypts, and decrypts packets sent.

IPSec uses the concept of a security association (SA) as the foundation for its security functions. A SA is the establishment of shared security information between two entities to support secure communication. It is a device's record of the configurations it needs to support an IPSec connection. When two devices complete the handshaking process and have agreed upon the security parameters used to communicate, this information is stored in the SA. This information includes the authentication and encryption keys, algorithms, key lifespan, the source and destination IP addresses, and other parameters.

SAs are directional, meaning that they only work in one direction. So each device requires one SA for outbound traffic and another for inbound traffic for each VPN it communicates with. If a device connects to five other devices, it will have at least ten SAs. The device determines which SA to use based on the security parameter index (SPI). The SPI value, which is in the header of an IPSec packet, dictates which SA to use. Here's how it works: When a device receives a packet via IPSec, it checks the SPI to determine which SA to use. The SA dictates how the packet should be decrypted, how its source should be authenticated, which key to use, and how to reply.

Let's look a little closer at the security services available via IPSec. IPSec can use Message Authentication Code (MAC) to authenticate a packet's sending device. The AH protocol provides authentication and integrity, whereas the ESP protocol provides authentication, integrity, and confidentiality because it also provides encryption functionality. ESP is most commonly used to establish a VPN connection. But why have two protocols that provide overlapping functionality? Why not just use ESP in all instances? The answer has to do with network address translation (NAT). IPSec generates an integrity check value (ICV) over a portion of the packet. The ICV is similar to a MAC value in that it enables the recipient to confirm the integrity of the packet by generating and comparing a value to that sent by the sender. If the values are the same, the recipient can assume that the packet was not altered during transmission. Now, we said that the ICV is generated over a portion of the packet. The AH protocol generates the ICV over the data payload, transport, and network headers. The ESP protocol, on the other hand, generates the ICV over the data payload and transport header only. It does not include the network header. This is important when it comes to NAT.

NAT devices change the IP address of the packets they receive. This means that when a recipient of an IPSec packet authenticated with AH generates an ICV, the result differs from the sender's ICV because a portion of the data that was used to calculate the original ICV has been altered. As a result, the packet is automatically discarded. However, if a packet authenticated using the ESP protocol goes through a NAT device, its ICV is left unchanged because the network header was not used to calculate the ICV. Thus, if the network environment uses NAT, the ESP protocol is the way to go.

As previously mentioned, IPSec is simply a framework. The choice of encryption and authentication algorithms and how keys are exchanged between devices is up to the administrator. The administrator may choose to manually handle key management or automate the process with a key management protocol. Most likely, the administrator will choose the de-facto standard for IPSec: Internet Key Exchange (IKE). IKE is a combination of two other protocols. The first protocol is the Internet Security Association and Key Management Protocol (ISAKMP), which is used to establish SAs and cryptographic keys. Thus, ISAKMP provides the framework for the algorithms, protocols, modes, and keys that are negotiated to establish an IPSec connection. The other protocol used by IKE is the OAKLEY protocol, which carries out the negotiation process. As a key-agreement protocol, OAKLEY allows two or more entities to negotiate which keys will be used without the need of a third party. It also allows authenticated parties to exchange keying material across an insecure connection using the Diffie-Hellman key exchange algorithm, which is a cryptographic protocol that allows two entities with no prior knowledge of one another to establish a shared secret key over an insecure communications channel like the Internet.

Tunneling Protocols At-a-Glance

Point-to-Point Tunneling Protocols (PPTP):

- Used for client/server connectivity.
- Establishes a single point-to-point connection between two systems.
- Works at the data link layer.
- Only transmits through IP networks.

Layer 2 Forwarding (L2F):

- Created by Cisco Systems.
- Merged with PPTP, resulting in L2TP.
- Provides mutual authentication.
- Does not provide encryption.

Layer 2 Tunneling Protocol (L2TP):

- The best of L2F and PPTP.
- Establishes a single point-to-point connection between two systems.
- Works at data link layer.
- Transmits over networks other than IP.
- Requires IPSec to provide encryption.

IPSec:

- Handles multiple connections simultaneously.
- Provides authentication, encryption, and integrity.

- Only supports IP networks.
- LAN-to-LAN and user-to-user communication.
- Works at the network layer.
- Provides security on top of IP.
- When working in tunnel mode, the payload and header are protected; when working in transport mode, only the payload is protected.

Summary

Now that you've completed Lesson 4 of the *CISSP Video Mentor*, you should have a thorough understanding of how VPNs and tunneling protocols work. In addition to PPTP and L2TP, you should understand the inner workings of IPSec, including the services it provides, its main protocols, its modes of operation, and key management. If any of these concepts is still unclear, take another look at the video or reread this manual.

lesson 5

OSI Model and Protocols

Lesson 5 of the *CISSP Video Mentor* focuses on the Open Systems Interconnection (OSI) reference model. You'll learn about the model's history, its purpose, and the responsibilities of each of the model's seven layers, as well as the protocols and services that work at each of them, and the devices and technologies working at each layer. The CISSP exam requires you to understand the functionalities that take place at each OSI layer, as well as their corresponding protocols. This knowledge will help you pass the exam, as well as understand the communication process that takes place between computer systems.

Definitions

ISO—An international standard-setting body comprised of representatives from national standards organizations. ISO is a worldwide federation that works to provide international standards.

Open network architecture—A nonproprietary, nonvendor-owned network architecture that easily integrates different technologies and their implementations.

Open Systems Interconnection (OSI)—A reference model for how data is transmitted between two network nodes.

Protocol—A set of rules that govern the syntax, semantics, and synchronization of communication between computer systems over a network.

Open Systems Interconnection Reference Model

Computing systems use protocols to communicate with one another across networks. A network protocol is a set of rules that govern the syntax, semantics, and synchronization of communication between systems. Thus, two different systems can use the same protocol to communicate.

Note

People often make the mistake of thinking that the OSI reference model was introduced at the dawn of the computing age. But that's not the case. It was introduced in 1984, many years after the basic Internet protocols—the Transmission Control Protocol/Internet Protocol (TCP/IP)—had been developed and implemented. TCP/IP uses four layers: application, transport, Internet, and network.

The OSI model divides a networking stack into seven layers: application, presentation, session, transport, network, data link, and physical. The protocols and services that work within each layer are responsible for carrying out certain functionalities that help systems communicate. The goal of the OSI model is to help technology vendors, engineers, developers, and others create products that operate within an open network architecture.

Note

An open network architecture is a nonproprietary, nonvendor-owned network architecture that easily integrates different technologies and their implementations.

Technology vendors, engineers, developers, and others use the OSI reference model as a starting point for developing their own networking frameworks, protocols, and interfaces. Often, they produce functionality that differs from or overlaps with what others have developed. Still, the integration and interoperability of these technologies is easier because they have the OSI model as their foundation.

Computers communicate in two ways. They communicate in a physical sense via electronic signals passed over a wire (or other communication channel types), and they communicate through logical channels. Using encapsulation, the protocols at each layer of the OSI model on one system communicate with the corresponding protocols at the same layer on another system, thereby communicating through logical channels. During encapsulation, each OSI layer adds its own information to the data packet as it is passed down through the protocol's stack. The original message increases in size as it goes through this process. When the message is received by the destination computer, the data packet is "unwrapped" as it moves up through the protocol stack. At each layer, data is extracted before it moves up to the next layer for further unwrapping. Thus, the destination computer learns how to properly interpret and process the packet.

Note

Different types of devices and protocols work at different layers of the OSI model. For example, computers can interpret and process data at each of the seven layers of the OSI model, whereas routers only understand up to the network layer. Bridges understand up to the data link layer, and repeaters understand up to the physical layer.

As previously mentioned, the protocols at each layer fulfill specific responsibilities and carry out specific control functions. They also expect specific data format syntaxes. Each of the OSI layers has a connection point or interface that enables it to communicate with three other layers: the layer above it, the layer below it, and the corresponding layer at the destination computer. Each layer's protocols add the control functions in the form of the packet's headers and trailers. The modularization of these layers and their functionalities enables different technologies, protocols, and services to communicate with one another. As a result, one system can use, for example, an application protocol, a session protocol, and a transport protocol, each developed by a different vendor, to transmit a single message over the network.

Technologies, protocols, and computers that work within the OSI model are referred to as *open technologies*. They can communicate with one another because they use international standard protocols and interfaces. Because the interfaces are very structured but the code within the software layer is undefined, vendors can create modularized plug-ins that integrate seamlessly into the network stack. Thus, technologies from different vendors can interact *and* provide vendor-specific extensions and functions.

Application Layer

There are seven layers within the OSI model. Layer 7, the application layer, works closest to the user, meaning that both it and the user can work more directly with the application network services. However, don't make the mistake of thinking that the actual applications reside here. Rather, the protocols that support the applications work here. Just a few of the services provided by application layer protocols are file transfer, network management, and terminal sessions. When an application needs to transmit data over a network, it sends instructions and the data to the application layer in the OSI model. The application communicates with the protocols using application programming interfaces (APIs). Here, the supporting protocols process and format the data. This is then passed down to the next layer, and so on, until each layer has added the information necessary to send the data over the network. At that point, the data are sent to the destination computer.

Application Layer Protocols

A few of the protocols that operate at layer 7 include the following:

- Simple Mail Transfer Protocol (SMTP)
- Hypertext Transfer Protocol (HTTP)
- Line Printer Daemon (LPD)
- File Transfer Protocol (FTP)
- Telnet
- Trivial File Transfer Protocol (TFTP)

Presentation Layer

Layer 6 of the OSI model is the presentation layer. No protocols work at this layer; only standards and services do. Layer 6 is concerned with the formatting and syntax of the data it receives from the application layer. It translates this data into a standard format that all systems using the OSI model can understand and adds information about what type of file it is and how it should be processed and presented. This allows, for example, a system running Microsoft Word to be able to open and manipulate a text document created in Open Office. The presentation layer on the sender's system converts the file to a standard file type—in this case, American Standard Code for Information Interchange (ASCII), and the recipient's system knows to open these types of documents with Word. Consider, as another example, a graphic created in Adobe Photoshop. If the sender saves the file in a standard format, such as a Graphic Interchange Format (GIF) or Joint Photographic Experts Group (JPEG), the recipient doesn't need to have Photoshop to open the graphic. The presentation layer tells the destination computer how to process and present the file.

Services at the presentation layer are also concerned with data compression and encryption. When a file is compressed and encrypted, the presentation layer includes information that the destination computer needs to properly process and present it to the user, including the type of compression and encryption used, and how to decompress and decrypt the data.

Presentation Layer Standards

A few of the standards that operate at layer 6 include the following:

- American Standard Code for Information Interchange (ASCII)
- Extended Binary-Coded Decimal Interchange Mode (EBCDIC)
- Graphic Interchange Format (GIF)
- Joint Photographic Experts Group (JPEG)
- Motion Picture Experts Group (MPEG)
- Musical Instrument Digital Interface (MIDI)

Session Layer

Sometimes two applications need to communicate or transfer information in a series of requests and responses. The session layer, layer 5, is responsible for setting up, managing, and closing a connection between the two applications. The services that work in layer 5 are commonly used by applications that use remote procedure calls.

The session layer works in three phases: connection establishment, data transfer, and connection release. If necessary, layer 5 also provides session restart and recovery. It also provides general session maintenance and breaks down the connection when communications are finished. All parameters then revert to the original settings. This process, called *dialog management*, is similar to how telephone circuitry works. When someone places a phone call, the telephone network circuitry and protocols establish a connection and maintain a communication path over the telephone line. When the caller hangs up the phone, the network circuitry and protocols disengage the resources used to keep the connection open.

Session Layer Protocols

A few of the protocols that operate at layer 5 include the following:

- NetBIOS
- Network File System (NFS)
- Remote Procedure Call (RPC)
- Structured Query Language (SQL)

Communications between two applications can occur in one of three different modes, as enabled by a session layer protocol. In a simplex communication circuit, all communication flows in one direction. In a half-duplex communication circuit, communication can flow in both directions, but only one application can transmit data at a time. Finally, a full-duplex communication circuit can

be likened to a two-way street. Communication can flow in both directions, and the two applications can transmit data simultaneously.

Transport Layer

While the session layer establishes connections between two applications, the transport layer establishes the logical connection between two computers.

The transport layer receives data from different applications at higher layers. It then assembles this information into a stream. The stream consists of data segments from various applications and is prepared for transmission.

If a connection-oriented protocol is used, this connection between computers is made possible through a handshaking process. During this process, certain parameters are agreed upon, such as the amount of data each computer will send at a time, how the data's integrity will be verified upon receipt, and how it will be determined whether a packet was lost during transmission. With these issues out of the way, the transport layer can provide for more reliable data transfer; error detection, correction, recovery, and flow control; and optimized network services. Thus, the transport layer—layer 4—provides end-to-end data transport services.

Note

It is easy to confuse the session and transport layers because of the similarity of their definitions. The session layer controls application-to-application communications, and the transport layer controls computer-to-computer communications.

Two types of protocols work at the transport layer: connection-oriented and connectionless. Connection-oriented protocols, such as TCP, guarantee packet delivery. So if a packet is lost during transmission, the protocol will recognize this and resend the packet. Connectionless protocols, such as User Datagram Protocol (UDP), are considered best-effort protocols in that they do not guarantee packet delivery. If a packet is lost, it is not recovered.

To further complicate matters, now is a good time to note that some resources assign specific protocols to different layers of the OSI model. For example, you may find some books that place Secure Sockets Layer (SSL) in the session layer and yet others that place it in the transport layer. This happens because the OSI model attempts to draw clean lines between the layers, but some protocols work in more than one layer. SSL, for example, works in the lower portion of the session layer *and* in the transport layer. When it comes to the CISSP exam, $(ISC)^2$ places SSL in the transport layer.

Transport Layer Protocols

A few of the protocols that operate at layer 4 include the following:

- Secure Sockets Layer (SSL)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

Network Layer

The network layer, layer 3, is primarily responsible for internetworking service, addressing and routing. Layer 3 protocols prepare the packet so that it can be transmitted across the network. Networks offer many routes to the same destination, but not every route is created equal. Routing protocols at this layer are responsible for creating a routing table that maps the best path for the packet to take. The protocols add information to the packet's header based on the routing table to allow it to be properly addressed and routed to be able to travel to the next system in its path. The protocols then route the packet to its destination. Although this routing information helps the packet arrive at its destination, it does not guarantee its delivery. The protocols at the transport layer are responsible for identifying delivery problems and resending packets accordingly.

Network Layer Protocols

A few of the protocols that operate at layer 3 include the following:

- Border Gateway Protocol (BGP)
- Internet Control Message Protocol (ICMP)
- Internet Group Management Protocol (IGMP)
- Internet Protocol (IP)
- Open Shortest Path First (OSPF)
- Routing Information Protocol (RIP)

Data Link Layer

Next on our journey down the protocol stack is the data link layer, layer 2. The protocols at the data link layer are responsible for converting data into LAN or WAN frames for transmission, converting messages into bits and defining how a computer accesses a network.

As a packet progresses through the protocol stack, its outer format changes at each layer. It is at the data link layer that the data is formatted so that it can be properly transmitted at the physical layer. This is necessary because LAN and WAN technologies use a variety of protocols, network interface cards (NICs), cables, and transmission methods, and each has its own data format structure. They also interpret electricity voltages differently. Layer 2 protocols know how to properly format the data frame for transmission over Token Ring, Ethernet, ATM, or Fiber Distributed Data Interface (FDDI) networks. For example, the protocols ensure that the header is the correct length, flags are in the appropriate place within the packet, and the trailer information is correctly placed within specific fields.

Data Link Layer Protocols

A few of the protocols that operate at layer 2 include the following:

- Fiber Distributed Data Interface (FDDI)
- Integrated Services Digital Network (ISDN)
- Layer 2 Forwarding (L2F)
- Layer 2 Tunneling Protocol (L2TP)
- Point-to-Point Protocol (PPP)
- Serial Line Internet Protocol (SLIP)

The data link layer is divided into the Logical Link Control (LLC) and Media Access Control (MAC) sublayers. The LLC communicates with the protocol in the network layer, whereas the MAC communicates with the physical layer. The LLC is defined in the 802.2 specification. The IEEE MAC specifications differ depending on the network. Ethernet is 802.3, Token Ring is 802.5, and wireless LAN is 802.11.

Note

IEEE standards such as 802.3 and 802.5 refer to the protocols that work at the MAC sublayer of the data link layer of the protocol stack.

The data link layer is also responsible for converting information into bits, which the physical layer then converts into electrical signals. As mentioned previously, LAN and WAN technologies use different types of cables and communication channel types. They also have defined electronic signaling and bit patterns, so a signal of .5 volts may represent a 0 for one technology and a 1 for another. Network cards bridge the data link and physical layers. When the packet reaches the network card driver at the data link layer, the bits are encoded according to the network technology being used (Ethernet, Token Ring, and so on). The physical layer then translates the bits into electrical coding and electricity state transitions and puts them on the wire for transmission.

Physical Layer

And so we come to layer 1 of the OSI model. The physical layer works with the data link layer technologies and is responsible for converting bits into electrical signals and controlling the physical aspects of data transmission, such as optical, electrical, wireless, and mechanical requirements. Different LAN and WAN technologies use different signal and voltage schemes. For example, data sent through dial-up software, through a modem, and onto a telephone line differs from that sent through a NIC and onto an unshielded twisted pair (UTP) cable for LAN communication. The data format, electrical signals, and control functionalities differ greatly between the two. To accommodate these differences, the physical layer has mechanisms that control synchronization, data rates, line noise, and medium access. No protocols work at layer 1. Specifications like the timing of voltage changes, voltage levels, and the physical connectors for electrical, optical, and mechanical transmission, as well as standard interfaces, are found at this layer.

Physical Layer Standard Interfaces

A few of the standard interfaces that operate at layer 1 include the following:

- EIA/TIA-232 and EIA/TIA-449
- High-Speed Serial Interface (HSSI)
- X.21

Summary

As you reach the conclusion of Lesson 5, you should be able to recite in order the seven layers of the OSI model (a handy mnemonic device for remembering the order of the seven layers is “All People Seem To Need Data Processing”), explain the functions that take place at each layer, and understand the services and protocols that work at each layer. You will be required to know these concepts for the CISSP exam. In the next and final lesson, we explore IEEE 802.11 standards.

lesson 6

802.11 Wireless

Wireless communications extend well beyond home wireless networks. Television transmissions, mobile phones, satellite transmissions, spying surveillance—and even the remote control for your DVD player—use broadband wireless signals that occupy the same frequency bands as microwaves, radars, and ham radios. In this lesson of the *CISSP Video Mentor*, we're going to focus on how IEEE 802.11 wireless technologies are used in LANs. You'll learn about the different IEEE 802 standards; the different components that comprise an 802.11 network; the differences between IEEE 802.11, 802.11b, 802.11a, and 802.11n; and how 802.15 and 802.16 differ from 802.11.

802.11 Definitions

Bluejacking—An attack in which an unsolicited message is sent to a Bluetooth-enabled device.

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)—A method used by systems to access the network wire and avoid collisions by signaling an intent to send data before actually doing so.

Carrier Sense Multiple Access with Collision Detection (CSMA/CD)—A method used by systems to avoid collisions while accessing the network wire by monitoring transmission activity to determine the best time to send data.

Channel—A particular frequency within a specific frequency band.

Institute of Electrical and Electronics Engineers—An international non-profit organization that supports the advancement of electricity-related technology, including wireless.

Wired Equivalent Privacy (WEP)—A wireless protocol that enables encrypted data transfers.

Wireless Communications

Wireless communication is the transmission of signals via airwaves through air and space. The airwaves alter the airwaves. Wireless communication isn't anything new. In fact, we've been communicating wirelessly since before wires! Two people conversing is an example of wireless communication. Their vocal chords alter the airwaves, which are signals that travel without wires.

Signals can be measured in frequency and amplitudes. *Amplitude* is the magnitude of change in each signal, whereas the *frequency* indicates how much data can be carried and how far. The higher the frequency, the more data the signal can carry, but over a shorter distance—otherwise, the signal becomes increasingly susceptible to atmospheric interference.

Wireless devices share an allotted radio frequency spectrum, which is finite. This means that it cannot grow to accommodate an increase in devices. This is similar to Ethernet in that all computers on a segment share the same medium, and in order to avoid a collision, only one computer can send data at a time. Ethernet networks use CSMA/CD (collision detection) technology to help prevent collisions, whereas wireless networks use CSMA/CA (collision avoidance). A wireless device sends out a broadcast to indicate that it is going to send data. Other devices on the shared spectrum receive this broadcast and wait to send their own data, thereby reducing or preventing collisions.

Wireless LAN Components

Access points (APs) are devices that enable wireless communication devices like a laptop to connect to a wireless network using a wireless standard (such as Wi-Fi or Bluetooth). In a wireless LAN (WLAN), APs are transceivers. They are placed in fixed locations where they work as communication beacons. When a wireless device sends data across the network, the wireless NIC modulates the data onto radio frequency signals that are understood by the AP. In turn, the wireless NIC receives signals from the AP and converts them into a digital format that the device can understand.

Access points can be set up to work in one of two modes: infrastructure and standalone. In infrastructure mode, an AP is connected to an Ethernet cable, thereby connecting the wired and wireless worlds. This setup extends the existing wired network while enabling wireless devices to access resources on the wired network. This is referred to as an infrastructure WLAN. When working in infrastructure mode, the AP and wireless clients form a group, which is called a Basic Service Set (BSS). When an AP is not connected to a wired network, it acts as a wireless hub and is said to be in standalone mode.

In order for a wireless device and AP to communicate, they must use the same frequency or channel. The AP is configured to work on a specific frequency to which the wireless device must tune itself.

The devices on a WLAN must also be configured with the proper Service Set ID (SSID). The SSID is simply a name given to a BSS. SSIDs can be used to segment hosts onto different WLANs, just as wired systems are segmented on a network based on the need to access different resources or based on different business functions or varying levels of trust. The SSID is also used to authenticate wireless devices to the AP. In order for the wireless device to communicate with the AP, it must have a valid SSID value. However, this does not mean that the SSID is a reliable security mechanism. Many APs broadcast their SSIDs, making it easy for an eavesdropper to sniff the data and use it to access the network.

There are two ways that a wireless device can authenticate to an AP. Open system authentication (OSA) does not involve a cryptographic key, whereas shared key authentication (SKA) does involve a cryptographic key. Often, a device authenticating using OSA needs only to provide a correct SSID value to the AP to gain access to network resources. Depending on the product and how it's configured, access can also be restricted based on specific MAC addresses. Either way, all communications are transmitted in cleartext because no encryption is involved. This method is not very secure because an eavesdropper only has to sniff the network to acquire the SSID value and

the necessary steps of authentication and then carry them out himself to authenticate to the AP and access network resources.

SKA, on the other hand, is based on the Wired Equivalent Privacy Protocol (WEP). Authentication of wireless devices is based on possession of an encryption key, rather than just a valid SSID value. And the use of WEP enables data to be encrypted. When the AP is configured to use SKA, it sends the wireless device a random value. The value is encrypted with the wireless device's cryptographic key and returned to the AP. The AP then decrypts the value, and if it matches the original value, the wireless device is authenticated.

As you can see, SKA offers a greater degree of security than OSA. However, many APs are shipped with WEP disabled, and individual users must be familiar with how to set up the device accordingly. As a result, it is not unusual for WEP to remain disabled. There is then no privacy, and the AP broadcasts its location and identity, making it possible for nearby users or intruders to "share" the network. Even when it is enabled, WEP is not a cure-all. Many weaknesses have been uncovered in WEP. We'll explain how these weaknesses are being addressed later.

Wireless Standards

The Institute of Electrical and Electronics Engineers (IEEE) is an international non-profit organization that supports the advancement of electricity-related technology, including wireless. One means of doing so is through the development of standards, which enable vendors to create different products that will interoperate. IEEE developed the first WLAN standard, 802.11, in 1997. It provided a 1- to 2-Mbps transfer rate and worked in the 2.4GHz frequency range. Because this frequency range is unlicensed by the FCC, companies and users do not have to pay to use it.

The 802.11 standard addresses the following:

- How wireless devices and APs communicate
- Specifications for wireless device and AP interfaces
- How signal transmissions should occur
- How authentication, association, and security should be implemented

Since 1997, numerous standards based on 802.11 have been developed. These are indicated by a lowercase letter (802.11a, 802.11b, and so on). These versions are based on the work done by task groups created by IEEE to improve specific areas within wireless communication. The letters indicate the order in which the specifications were proposed and accepted.

802.11b

Just to confuse matters, 802.11b was the first extension to 802.11. Although 802.11a was conceived and approved first, its release was delayed due to its technical complexity. So we're starting with 802.11b. Like 802.11, it works in the 2.4GHz frequency range. However, its data transfer rate is faster—up to 11Mbps. It is also backward-compatible with 802.11.

802.11a

Unlike 802.11b, 802.11a works in the 5GHz frequency band and is therefore not backward compatible with 802.11 or 802.11b. It is possible, however, to find products that work with both 802.11a and 802.11b implementations. They either need to be configured appropriately or may be able to configure themselves based on sensing the technologies in use.

There are two primary advantages to using 802.11a: speed and frequency. It provides data transfer rates up to 54Mbps, and it works in a less-crowded frequency. Several devices, such as microwaves, cordless phones, and baby monitors, already work in the 2.4GHz frequency. Because there can be contention for access and inadequate service, the frequency band is referred to as a "dirty" frequency. This can be avoided by moving to the 5GHz frequency band.

The higher frequency also serves as a disadvantage for 802.11a because it does not provide the same range as 802.11b. In order to obtain maximum speeds, client devices must be within 25 feet of the AP. In addition, other countries have not necessarily allocated the 5GHz frequency range for WLAN use. As a result, an 802.11a product that works in the U.S. may not necessarily work outside the country.

802.11e

This working group has provided QoS and proper support of multimedia traffic. Multimedia and other types of time-sensitive applications have a lower tolerance for delays in data transmission. QoS provides the capability to prioritize traffic, and affords guaranteed delivery. This specification and its capabilities may open the door to allow many different types of data to be transmitted over wireless connections.

802.11f

When a user moves around in a WLAN, her wireless device often needs to communicate with different APs. An AP can cover only a certain distance, and as the user moves out of the range of the first AP, another AP needs to pick up and maintain her signal to ensure that she does not lose network connectivity. This is referred to as roaming, and for this to happen seamlessly, the APs need to communicate with each other. If the second AP must take over this user's communication, it will need to be assured that this user has been properly authenticated and must know the necessary settings for this user's connection. This means the first AP would need to be able to convey this information to the second AP. The conveying of this information between the different APs during roaming is what 802.11f deals with. It outlines how these data can be properly shared.

802.11g

The 802.11g standard provides for higher data transfer rates than previous standards—up to 54Mbps. This is basically a speed extension for current 802.11b products. If a product meets the specifications of 802.11b, its data transfer rates are up to 11Mbps, and if a product is based on 802.11g, that new product can be backward-compatible with older equipment but work at a much higher transfer rate.

So do we go with 802.11g or with 802.11a? They both provide higher bandwidth. 802.11g is backward-compatible with 802.11b, so that is a good thing if you already have a current infrastructure. But 802.11g still works in the 2.4GHz range, which is continually getting more crowded. 802.11a works in the 5GHz band and may be a better bet if you use other devices in that frequency range. However, working at higher frequency means that a device's signal cannot cover as wide a range. Your decision will also come down to what standard wins out in the standards war. Most likely, one or the other standard will eventually be ignored by the market, so you will not have to worry about making this decision. Only time will tell which one will be the keeper.

802.11h

As mentioned previously, the 5GHz range that 802.11a works in is not necessarily supported for wireless communications outside of the U.S. Thus, 802.11h was developed as an extension to 802.11a to address European regulations and allow the implementation of products working in the 5GHz range in European countries and now many others. The standard also provides Dynamic Frequency Selection and Transmit Power Control to the 802.11a MAC to help prevent interference with satellites and radar that also use the 5GHz frequency band.

802.11i

A long list of security flaws, which have put many organizations at risk, has been documented in 802.11. Because each of the standards we've gone over so far is based on the same security model, they too are characterized by the same weaknesses. These include a lack of user authentication, as well as mutual authentication between the AP and the wireless device, and a flawed encryption protocol, WEP. We discussed in a previous lesson the importance of randomness in an encryption scheme. WEP uses the RC4 encryption algorithm, and its key and initialization vectors do not provide enough randomness, enabling intruders to easily break encrypted wireless traffic with free tools available on the Web.

Note

Wireless networks that solely depend on WEP for security are vulnerable to a number of attacks. Some of these threats include the following:

- The unauthorized modification of data during transmission
- The implementation of rogue APs that users can mistakenly authenticate to and communicate with
- Sniffing of wireless traffic by eavesdroppers
- Easy decryption of wireless traffic by unauthorized persons

An IEEE task group was created to address the security issues prevalent in 802.11, and the result was 802.11i. The new standard uses specific protocols, technologies, and algorithms to provide a higher level of protection than that offered by WEP. To begin with, 802.11i uses Extensible Authentication Protocol (EAP) to enforce user authentication and mutual authentication. Message Integrity Code (MIC) was also integrated to detect unauthorized changes to bits during their transmission.

In terms of encryption, 802.11i uses two different approaches that provide for a higher level of protection. Temporal Key Integrity Protocol (TKIP) is integrated into 802.11i to work with WEP. TKIP generates random values and feeds them to WEP during the key generation process. This extra randomness increases the complexity of the keys, thus making them more difficult to uncover. TKIP is backward-compatible for many WLAN devices. In addition, the IEEE working group made TKIP available via firmware or software updates to prevent customers from having to purchase new equipment to acquire this protection.

Note

Many vendors have attempted to overcome 802.11's security flaws by developing their own security technologies. However, these have only served as bandages that often lead to other issues regarding interoperability.

The newer standard also includes the Advanced Encryption Standard (AES) algorithm for use in new WLAN implementations. The AES algorithm can be used in counter mode with CBC-MAC (CCM), which is referred to as the CCM protocol (CCMP). CCMP differs from TKIP in that key management and message integrity is handled by a single component built around AES using a 128-bit key and a 128-bit block. CCMP is intended to replace TKIP; however, AES requires more processing power than RC4 and it is not backward-compatible with current WLAN products. Therefore, it is recommended for new WLAN implementations.

802.11n

The World Wide Spectrum Efficiency (WWiSE) has submitted a proposal for a new standard, 802.11n, to replace current Wi-Fi technologies. 802.11n uses a concept called Multiple Input, Multiple Output (MIMO) to increase its throughput to 100Mbps. MIMO requires the use of two receive and two transmit antennas to broadcast in parallel over a 20MHz channel. 802.11n works at the 5GHz frequency range, making it backward-compatible with 802.11a and 802.11g.

802.16

So far, we've covered WLAN standards. With 802.16, we venture into new territory: the Metropolitan Area Network (MAN). As a MAN wireless standard, 802.16 allows wireless traffic to cover larger geographical areas. You've probably heard of this technology by its more frequently used name: broadband wireless access.

802.15

This standard deals with a much smaller geographical network, which is referred to as a Wireless Personal Area Network (WPAN). This technology allows for connectivity to take place among local devices, such as a computer communicating with a PDA, a cellular phone communicating with a computer, or a headset communicating with another device.

Bluetooth Wireless

The Bluetooth wireless technology is actually a portion of the 802.15 standard. It has a 1- to 3-Mbps transfer rate and works in a range of approximately ten meters. If you have a cell phone and a PDA that are both Bluetooth-enabled, and both have calendar functionality, you could have them update each other without any need to connect them physically. If you added some information to your cell phone contacts list and task list, for example, you could just place the phone close to your PDA. The PDA would sense that the other device was nearby, and it would then attempt to set up a network connection with it. Once the connection is made, synchronization between the two devices would take place, and the PDA would add the new contact and task list data. Bluetooth works in the frequency range of other 802.11 devices (2.4GHz), and plans are currently underway to make various technologies interoperable.

Real security risks exist when transferring unprotected data via Bluetooth in a public area, because any device within a certain range can capture this type of data transfer.

Another attack that Bluetooth is vulnerable to is referred to as *Bluejacking*. In this attack, someone sends an unsolicited message to a device that is Bluetooth-enabled. Bluejackers look for a receiving device (phone, PDA, laptop) and then send a message to it. Often, the Bluejacker is trying to send someone else their business card, which will be added to the victim's contact list in their address book. The countermeasure is to put the Bluetooth-enabled device into nondiscoverable mode so others cannot identify this device in the first place. If you receive some type of message this way, just look around you. Bluetooth only works within a ten-meter distance, so it is coming from someone close by.

Summary

In this lesson, we've gone over a number of wireless standards, some of them still in the development phase. Not all of them will catch on. Time will tell which ones are accepted by vendors and customers. In the meantime, you need to be familiar with all of them for the CISSP exam. At this point, you should understand the differences between the various 802.11 standards, the purpose of 802.15 and 802.16, the components that make up a typical 802.11 network, and the security improvements built into 802.11n. If you have problems with any of these areas, revisit the *Video Mentor* or the appropriate section in this manual.

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes