CompTIA Advanced Security Practitioner Certification:

Exam CAS-001

Instructor's Edition

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

CompTIA Advanced Security Practitioner Certification: Exam CAS-001

CEO, Axzo Press:	Ken Wasnock
Vice President, Content and Delivery:	Josh Pincus
Director of Publishing Systems Development:	Dan Quackenbush
Authoring Team:	MAD Security
Authoring Team Leader:	Michael Murray
Developmental Editor:	Andy LaPage
Copyeditors:	Cliff Coryea Dan Quackenbush

COPYRIGHT © 2012 Axzo Press. All rights reserved.

No part of this work may be reproduced, transcribed, or used in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, Web distribution, or information storage and retrieval systems—without the prior written permission of the publisher.

For more information, go to www.axzopress.com.

Trademarks

ILT Series is a trademark of Axzo Press.

Some of the product names and company names used in this book have been used for identification purposes only and may be trademarks or registered trademarks of their respective manufacturers and sellers.

Disclaimer

We reserve the right to revise this publication and make changes from time to time in its content without notice.

ISBN 10: 1-4260-2925-X
ISBN 13: 978-1-4260-2925-7

Printed in the United States of America

1 2 3 4 5 GL 06 05 04 03

PREVIEW

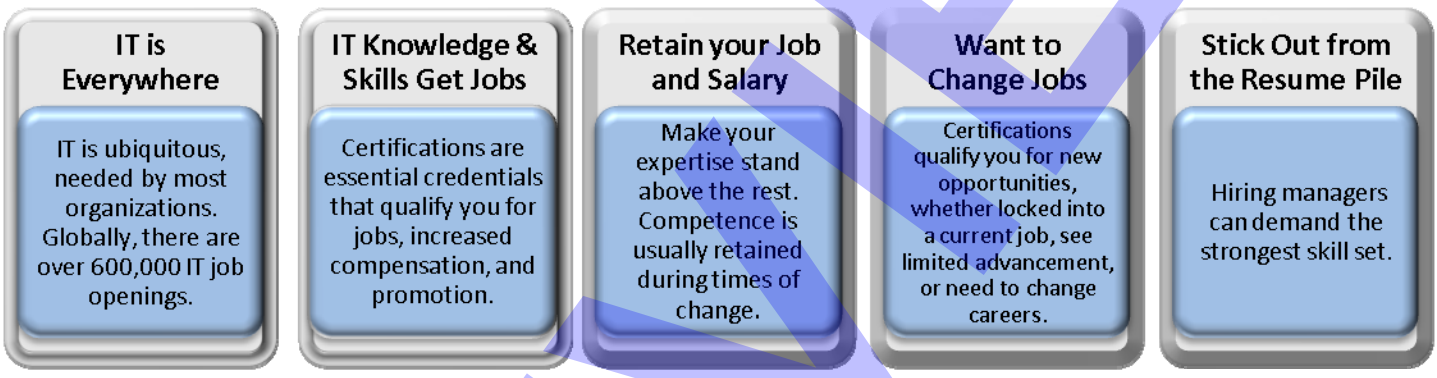
NOT FOR PRINTING OR INSTRUCTIONAL USE

It Pays to Get Certified

In a digital world, digital literacy is an essential survival skill. Certification proves you have the knowledge and skill to solve business problems in virtually any business environment. Certifications are highly-valued credentials that qualify you for jobs, increased compensation, and promotion.

- **Be the First** - CASP is the first mastery level certification available from CompTIA. It expands on the widely recognized path of CompTIA Security+ with other 300,000 certified Security+ professionals.
- **The CASP certified individual** - applies critical thinking and judgment across a broad spectrum of security disciplines to propose and implement solutions that map to enterprise drivers.
- **The cloud is a new frontier** - that requires astute security personnel who understand the security impact of the cloud on network design and risk.

How Certification Helps Your Career



Steps to Getting Certified and Staying Certified	
Review Exam Objectives	Review the certification objectives to make sure you know what is covered in the exam: http://certification.comptia.org/Training/testingcenters/examobjectives.aspx
Practice for the Exam	After you have studied for the certification, take a free assessment and sample test to get an idea what type of questions might be on the exam: http://certification.comptia.org/Training/testingcenters/samplequestions.aspx
Purchase an Exam Voucher	Purchase your exam voucher on the CompTIA Marketplace, which is located at : http://www.comptiastore.com
Take the Test	Select a certification exam provider and schedule a time to take your exam. You can find exam providers at the following link: http://certification.comptia.org/Training/testingcenters.aspx
Stay Certified! Continuing Education	The CompTIA CASP certification is valid for three years from the date of certification. There are a number of ways the certification can be renewed. For more information go to: http://certification.comptia.org/getCertified/stayCertified.aspx

How to obtain more information

- **Visit CompTIA online** - www.comptia.org to learn more about getting CompTIA certified.
- **Contact CompTIA** - call 866-835-8020 ext. 5 or email questions@comptia.org
- **Join the IT Pro Community** – http://itpro.comptia.org to join the IT community to get relevant career information.
- **Connect with us:**

PREVIEW
NOT FOR PRINTING OR INSTRUCTIONAL USE

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Contents

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Introduction

After reading this introduction, you will know how to:

A Use ILT Series manuals in general.

B Use prerequisites, a target student description, course objectives, and a skills inventory to properly set students' expectations for the course.

C Set up a classroom to teach this course.

D Get support for setting up and teaching this course.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Topic A: About the manual

ILT Series philosophy

Our goal is to make you, the instructor, as successful as possible. To that end, our manuals facilitate students' learning by providing structured interaction with the software itself. While we provide text to help you explain difficult concepts, the hands-on activities are the focus of our courses. Leading the students through these activities will teach the skills and concepts effectively.

We believe strongly in the instructor-led class. For many students, having a thinking, feeling instructor in front of them will always be the most comfortable way to learn. Because the students' focus should be on you, our manuals are designed and written to facilitate your interaction with the students, and not to call attention to manuals themselves.

We believe in the basic approach of setting expectations, then teaching, and providing summary and review afterwards. For this reason, lessons begin with objectives and end with summaries. We also provide overall course objectives and a course summary to provide both an introduction to and closure on the entire course.

Our goal is your success. We encourage your feedback in helping us to continually improve our manuals to meet your needs.

Manual components

The manuals contain these major components:

- Table of contents
- Introduction
- Units
- Appendices
- Course summary
- Glossary
- Index

Each element is described below.

Table of contents

The table of contents acts as a learning roadmap for you and the students.

Introduction

The introduction contains information about our training philosophy and our manual components, features, and conventions. It contains target student, prerequisite, objective, and setup information for the specific course. Finally, the introduction contains support information.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Units

Units are the largest structural component of the actual course content. A unit begins with a title page that lists objectives for each major subdivision, or topic, within the unit. Within each topic, conceptual and explanatory information alternates with hands-on activities. Units conclude with a summary comprising one paragraph for each topic, and an independent practice activity that gives students an opportunity to practice the skills they've learned.

The conceptual information takes the form of text paragraphs, exhibits, lists, and tables. The activities are structured in two columns, one telling students what to do, the other providing explanations, descriptions, and graphics. Throughout a unit, instructor notes are found in the left margin.

Appendices

An appendix is similar to a unit in that it contains objectives and conceptual explanations. However, an appendix does not include hands-on activities, a summary, or an independent practice activity.

Course summary

This section provides a text summary of the entire course. It is useful for providing closure at the end of the course. The course summary also indicates the next course in this series, if there is one, and lists additional resources students might find useful as they continue to learn about the software.

Glossary

The glossary provides definitions for all of the key terms used in this course.

Index

The index at the end of this manual makes it easy for you and your students to find information about a particular software component, feature, or concept.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Manual conventions

We've tried to keep the number of elements and the types of formatting to a minimum in the manuals. We think this aids in clarity and makes the manuals more classically elegant looking. But there are some conventions and icons you should know about.

Instructor note/icon

Item	Description
Italic text	In conceptual text, indicates a new term or feature.
Bold text	In unit summaries, indicates a key term or concept. In an independent practice activity, indicates an explicit item that you select, choose, or type.
`Code font`	Indicates code or syntax.
`Longer strings of ► code will look ► like this.`	In the hands-on activities, any code that's too long to fit on a single line is divided into segments by one or more continuation characters (►). This code should be entered as a continuous string of text.
Instructor notes.	In the left margin, provide tips, hints, and warnings for the instructor.
Select **bold item**	In the left column of hands-on activities, bold sans-serif text indicates an explicit item that you select, choose, or type.
Keycaps like ↵ ENTER	Indicate a key on the keyboard you must press.
Warning icon.	Warnings prepare instructors for potential classroom management problems.
Tip icon.	Tips give extra information the instructor can share with students.
Setup icon.	Setup notes provide a realistic business context for instructors to share with students, or indicate additional setup steps required for the current activity.
Projector icon.	Projector notes indicate that there is a PowerPoint slide for the adjacent content.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Hands-on activities

The hands-on activities are the most important parts of our manuals. They are divided into two primary columns. The "Here's how" column gives short directions to the students. The "Here's why" column provides explanations, graphics, and clarifications. To the left, instructor notes provide tips, warnings, setups, and other information for the instructor only. Here's a sample:

Do it!

A-1: Creating a commission formula

Take the time to make sure your students understand this worksheet. We'll be here a while.

Here's how	Here's why
1 Open Sales	This is an oversimplified sales compensation worksheet. It shows sales totals, commissions, and incentives for five sales reps.
2 Observe the contents of cell F4	 The commission rate formulas use the name "C_Rate" instead of a value for the commission rate.

For these activities, we have provided a collection of data files designed to help students learn each skill in a real-world business context. As students work through the activities, they will modify and update these files. Of course, students might make a mistake and therefore want to re-key the activity starting from scratch. To make it easy to start over, students will rename each data file at the end of the first activity in which the file is modified. Our convention for renaming files is to add the word "My" to the beginning of the file name. In the above activity, for example, students are using a file called "Sales" for the first time. At the end of this activity, they would save the file as "My sales," thus leaving the "Sales" file unchanged. If students make mistakes, they can start over using the original "Sales" file.

In some activities, however, it might not be practical to rename the data file. Such exceptions are indicated with an instructor note. If students want to retry one of these activities, you will need to provide a fresh copy of the original data file.

PowerPoint presentations

Each unit in this course has an accompanying PowerPoint presentation. These slide shows are designed to support your classroom instruction while providing students with a visual focus. Each presentation begins with a list of unit objectives and ends with a unit summary slide. We strongly recommend that you run these presentations from the instructor's station as you teach this course. A copy of PowerPoint Viewer is included, so it is not necessary to have PowerPoint installed on your computer.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Topic B: Setting student expectations

Properly setting students' expectations is essential to your success. This topic will help you do that by providing:

- Prerequisites for this course
- A description of the target student
- A list of the objectives for the course
- A skills assessment for the course

Course prerequisites

Students taking this course should be familiar with personal computers and the use of a keyboard and a mouse. Furthermore, this course assumes that students have completed the following courses or have equivalent experience:

- CompTIA Certification: *A+ Essentials 220-701*
- CompTIA Certification: *Security+ SY0-301*
- *Visio 2010: Basic*

Target student

Students who are Security+ certified and want to further their knowledge of enterprise security. This course takes students to the next level by focusing on higher level technical skills and the ability to plan and analyze business needs in order to secure the enterprise. Students will also be interested in obtaining the CompTIA Advanced Security Practitioner certification.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Course objectives

You should share these overall course objectives with your students at the beginning of the day. This will give the students an idea about what to expect, and it will help you identify students who might be misplaced. Students are considered misplaced when they lack the prerequisite knowledge or when they already know most of the subject matter to be covered.

After completing this course, students will know how to:

- Distinguish and select appropriate cryptographic tools.
- Distinguish and select virtualized and distributed storage solutions..
- Explain security implications for enterprise storage.
- Integrate and secure disparate network resources.
- Secure host computers.
- Explain application security.
- Conduct security assessments.
- Analyze risks associated with business decisions.
- Implement risk mitigation strategies.
- Explain incident response and recovery procedures.
- Implement organizational security policies and procedures.
- Analyze industry trends and apply them to the enterprise.
- Analyze enterprise security needs.
- Integrate cross-functional teams to help secure the enterprise.
- Explain the affects of organizational change on security.
- Select controls to secure communications and collaboration.
- Explain advanced authentication techniques.
- Secure the technology life cycle.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Skills inventory

Use the following form to gauge students' skill levels entering the class (students have copies in the introductions of their student manuals). For each skill listed, have students rate their familiarity from 1 to 5, with five being the most familiar. Emphasize that this is not a test. Rather, it is intended to provide students with an idea of where they're starting from at the beginning of class. If a student is wholly unfamiliar with all the skills, he or she might not be ready for the class. A student who seems to understand all of the skills, on the other hand, might need to move on to the next course in the series.

Skill	1	2	3	4	5
Creating an MD-5 hash					
Examining an SSL certificate					
Discussing virtualization					
Examining storage solutions					
Designing a secure network					
Discussing host security					
Examining cookie security					
Using Sandboxie					
Examining penetration testing					
Discussing risk					
Examining risk mitigation and controls					
Examining incident response					
Accessing industry-related tweets					
Discussing policies and procedures					
Researching conferences and conventions					
Researching trends					
Examining business disciplines					
Discussing security controls					
Discussing an IT merger					
Exploring mobile device security					
Examining advanced authentication techniques					
Examining technology life cycle security					

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Topic C: Classroom setup

All our courses assume that each student has a personal computer to use during the class. Our hands-on approach to learning requires that they do. This topic gives information on how to set up the classroom to teach this course.

Hardware requirements

You will need one client computer per student. Each client computer should have:

- A keyboard and a mouse
- At least 1 GHz 32-bit or 64-bit processor
- At least 1 GB RAM
- At least 100 GB hard drive with at least 50 GB of available space
- A DVD-ROM drive
- A graphics card that supports DirectX 9 graphics with:
  - WDDM driver
  - 128 MB of graphics memory (minimum)
  - Pixel Shader 2.0 in hardware
  - 32 bits-per-pixel
- SVGA monitor
- Network card
- Mobile device on which students can configure security settings

Software requirements

You will need the following software:

- Windows 7 Professional for client computers
- Firefox
- Microsoft Visio (latest version) or other design software with network design components
- Adobe Acrobat Reader

Network requirements

The following network components and connectivity are also required for this course:

- Internet access, for the following purposes:
  - Completing activities throughout the course
  - Downloading the Student Data files from www.axzopress.com (if necessary)

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Classroom setup instructions

Before you teach this course, you will need to perform the following steps to set up each student computer.

1. Install Windows 7 using the manufacturer's installation instructions. Create one account on each computer using the naming scheme Studentxx, where xx is a unique number for each student, such as Student01, Student02, and so on.
2. With flat-panel displays, we recommend using the panel's native resolution for best results. Color depth/quality should be set to High (24 bit) or higher.
3. Install Firefox on each computer using the default installation settings.
4. Install Adobe Acrobat Reader on each computer using the default installation settings.
5. Install Microsoft Visio on each computer using the default installation settings.

Downloading the PowerPoint presentations

If you don't have the disc that came with this manual, you can download the PowerPoint presentations for this course:

1. Connect to www.axzopress.com.
2. Under Downloads, click Instructor-Led Training.
3. Browse the subject categories to locate your course. Then click the course title to display a list of available downloads. (You can also access these downloads through our Catalog listings.)
4. Click the link(s) for downloading the PowerPoint presentations, and follow the instructions that appear on your screen.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Topic D: Support

Your success is our primary concern. If you need help setting up this class or teaching a particular unit, topic, or activity, please don't hesitate to get in touch with us.

Contacting us

Please contact us through our Web site, www.axzopress.com. You will need to provide the name of the course, and be as specific as possible about the kind of help you need.

Instructor's tools

Our Web site provides several instructor's tools for each course, including course outlines and answers to frequently asked questions. To download these files, go to www.axzopress.com. Then, under Downloads, click Instructor-Led Training and browse our subject categories.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Unit 1

Cryptographic tools and techniques

Unit time: 90 minutes

Complete this unit, and you'll know how to:

A Explain how cryptography is used to protect confidentiality, integrity, and authenticity.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Topic A: Cryptography

This topic covers the following CompTIA Advanced Security Practitioner exam objectives.

#	Objective
1.1	**Distinguish which cryptographic tools and techniques are appropriate for a given situation.** • Cryptographic applications and proper implementation • Advanced PKI concepts – Wild card – OCSP vs. CRL – Issuance to entities – Users – Systems – Applications • Implications of cryptographic methods and design – Strength vs. performance vs. feasibility to implement vs. interoperability • Transport encryption • Digital signature • Hashing • Code signing • Non-repudiation • Entropy • Pseudo random number generation • Perfect forward secrecy • Confusion • Diffusion

Cryptography has a long and storied history, with crude forms dating back to the time of the Romans. Today cryptography is used to provide a number of security guarantees on encrypted data whether that data is sent between parties or that data is at rest. These guarantees state that even when faced with active or passive attackers, that the encrypted data will remain confidential; that the receiver can be sure of the sender; and that the transmitted data has not been altered.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

A brief history of cryptography

The ability to reveal sensitive messages and data only to specific people has been sought after for thousands of years. One of the earliest known forms of encryption is the Caesar Cipher, which was invented by Julius Caesar to transmit military messages that would not be readable by enemy spies. While effective for its time, such simple ciphers are trivially defeated by modern attacks. This trend of relatively weak encryption mechanisms continued until World War II when advanced machinery capable of encrypting and decrypting messages was invented. The most notable of these was the German Enigma machine, which was used heavily by the Nazis. The eventual cracking of the Enigma gave the Allies a substantial advantage in the war.

Since World War II, cryptography has branched out from the military space and expanded to all fields of computing and business. This includes substantial developments such as public key cryptography, strong hashing functions, and advances in symmetric encryption. These technologies have been used to provide the foundations for services and applications that we all use, such as encrypted web browsing, e-mail, and file storage.

Cryptography is used to satisfy the requirements of confidentiality, integrity, authenticity, and non-repudiation. Selection of the appropriate algorithms and protocols in specific situations will determine which of these are met and to what certainty they can be guaranteed.

Confidentiality

In the context of cryptography, confidentiality states that only parties who own the data or who are the intended recipients of sensitive communications are able read it. Examples of this include the user of a computer with an encrypted hard drive or the recipients of an encrypted e-mail. It can also mean transactions over a network such as credit card processing or submission of personal information such as a Social Security number.

Integrity

With the confidentiality of data protected, integrity of data is the next desired step in cryptography. While confidentiality guarantees that a message being sent from one person to another cannot be read by others, it does not guarantee that the message was not altered since being sent. Integrity takes this next step to ensure that no tampering of data occurred.

A common example of integrity checking is the use of hashing, discussed later, to verify a file's contents. Downloaded files often are accompanied with their hash so that the end user can verify that the download was successful and was not tampered with. In the field of digital forensics, hashes are used to prove the integrity of collected evidence and that the investigator did not add, delete, or modify evidence.

Authenticity

The next guarantee of cryptography is authenticity, which verifies that the sender of a message is who they claim to be. In situations such as purchasing directives, client/attorney communications, and others where impactful decisions or actions will be taken based on a message, confidentiality and integrity of the message are insufficient. Using technologies discussed later in the unit, such as public key cryptography and digital signatures, receivers of messages can be guaranteed of a message sender's identity and fully trust the message's contents.

PREVIEW
NOT FOR PRINTING OR INSTRUCTIONAL USE

Non-repudiation

Non-repudiation ensures that the party performing a sensitive transaction, such an encrypted e-mail, banking transfer, or online purchases cannot later challenge its authenticity. Protocols and algorithms that provide non-repudiation do so by cryptographically binding the identity of the person to the transaction.

Random numbers

The ability to generate random numbers is at the core of cryptography. As you'll see, the privacy and security guarantees of many cryptographic protocols and algorithms depend on the validity of the random number generation source used to produce keys and nonces.

Entropy

Entropy is the amount of randomness that can be collected by an operating system and provided to applications. This entropy is then used to generate random data for a number of processes, both cryptographic and not. Entropy can come from a number of sources, such as specialized hardware, non-uniform data from the running computer, and even from the end user. If you have ever been asked to type on the keyboard or move the mouse while cryptographic keys are being generated, then you have helped create entropy for a cryptographic process.

Pseudo Random Number Generators (PRNG)

In an ideal world, all random numbers generated would be truly random. Unfortunately, this is not practical as there is very little true randomness in the entire universe, much less in the average computer. To compensate for this, specialized hardware has been developed that attempts to gather data from physical events that show random properties. Examples of these include measuring radioactive particle decay, detecting photons passing through an area in a certain time, measuring different types of "noise," and other discernible data generated from physical events.

While these generators are the best source of random numbers, they are not always practical for the following reasons:

- They can take a long time to gather enough entropy.
- They require integration into software systems.
- They can be cost prohibitive.

Due to the difficulty of generating truly random numbers, the more common scenario is to use pseudo random numbers. These numbers, while not truly random, are sufficiently so to be used during cryptographic processes.

Pseudo random number generators (PRNG) are algorithms that can produce random numbers based on an initial state, called the seed state. The *seed state* is a simply a number that defines what will be the first stage of the number generation. Two of the most known PRNG are the Mersenne Twister and Fortuna. The Mersenne twister is not considered safe for cryptographic purposes, while Fortuna is able to produce random number streams suitable for cryptographic operations.

A PRNG will always produce the same random number sequence when given the same seed state. For this reason, the seed must be truly random and it must be kept secret. If a seed that was used to generate a cryptographic key is comprised, the key can be regenerated on demand by attackers.

PREVIEW
NOT FOR PRINTING OR INSTRUCTIONAL USE

Cryptographic uses of random numbers

There are two main uses of random numbers in cryptography: nonces and key generation.

Nonces

Replay attacks consist of an active attacker who monitors network traffic and later resends packets that were previously transmitted. If a protocol does not account for these types of attacks then there can be disastrous consequences as actions, such as payment processing, database modifications, and privileged operations, can be re-performed at any point in time.

To prevent these attacks, many protocols include nonces, which are generally pseudo randomly generated numbers, as part of the encrypted content. The server records all nonces that it sees and checks the current value with the previously seen ones before processing a message. Since the nonce should never repeat, this is a very effective defense against replay attacks.

Key generation

Generation of secure cryptographic keys requires that random numbers are available throughout the process. Insecure keys compromise the security of any data encrypted using the key.

Since the security of keys and nonces and all the data they protect rely on the strength of the random number generator used, a bad random number generator can lead to dire consequences.

For example, in 2008, a Debian OpenSSL maintainer erroneously commented out a line of code that was used in the seed creation phase of OpenSSL. The effect of this code removal was that only the process ID of the OpenSSL application was used to seed the PNRG. Since there is a limited number of process IDs on default Linux installlations (roughly 32,000), this meant that a very small number of seeds could be chosen to generate keys for SSL and SSH. Such a small number of keys can lead to trivial decryption of data encrypted with the keys as well as man-in-the-middle attacks against users with an affected key.

Then, in 2012, a research paper was published which showed that a number of TLS public keys used on the Internet were identical. The paper also showed that another large set of keys was factorable. The issue revolved around insufficient entropy in the generation of the keys. The effect of this issue is that attackers can again decrypt data as well as perform man-in-the-middle attacks against people or computers using the affected keys.

PREVIEW
NOT FOR PRINTING OR INSTRUCTIONAL USE

Symmetric encryption

Symmetric encryption uses one cryptographic key to encrypt and decrypt data. This key, often call a shared key, must be kept only between parties who should have access to the protected data as anyone with the key can decrypt the data.

Ciphers

There are two types of ciphers, block ciphers and stream ciphers, that are used to encrypt data. You'll cover both in the following sections. While they use different methods to encrypt data, ciphers in general share the following properties.

- *Confusion* is the process of making the relationship between the ciphertext and the key completely dependent on the key. Without this property, attackers could selectively generate encrypted versions of plaintext messages and then study their relationship. Successful application of this attack could lead to recovery of the cryptographic key.
- *Diffusion* is making the ciphertext change drastically upon changes in the input. This ensures that similar data does not produce similar or repeating information in the resulting hash. Without this property, attackers could selectively determine parts of the message encrypted by the same key.

Block ciphers

Block ciphers perform symmetric encryption on fixed-sized blocks of data. When the data size is larger than the block size, a mode of operation must be used to handle the data.

Electronic Codebook (ECB)

ECB is a naïve mode that simply splits the data into block-sized chunks and each block is then encrypted using the shared key. While providing some security, this algorithm has a severe issue in that blocks of the same data, such as all zeroes or of repeating text, will encrypt to the same value. This is a large cryptographic weakness and makes ECB mode too insecure to use in real world applications.

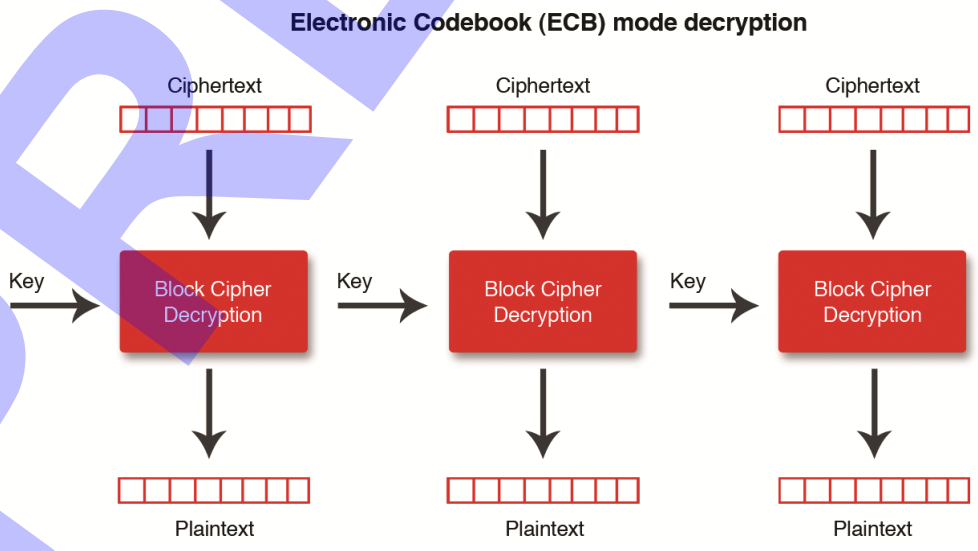


Exhibit 1-1: Electronic Codebook

PREVIEW
NOT FOR PRINTING OR INSTRUCTIONAL USE

Initialization vectors (IV)

The following two modes, which secure the symmetric encryption of data sets larger than the block size, all use initialization vectors (IV). An IV is a randomly generated sequence that is size of a data block.

- Cipher Block Chaining (CBC) is the most popular mode of operation. It works by using an IV as the first block of encrypted data and then XOR'ing each subsequent block of plain text with the previous cipher text. This ensures that blocks of the same data will not encrypt the same value as with ECB. One drawback with CBC is that if the last block is not on a block boundary, it must be padded to fill the block. Padding refers to filling the block with excess data to ensure that it is on a block-sized boundary.

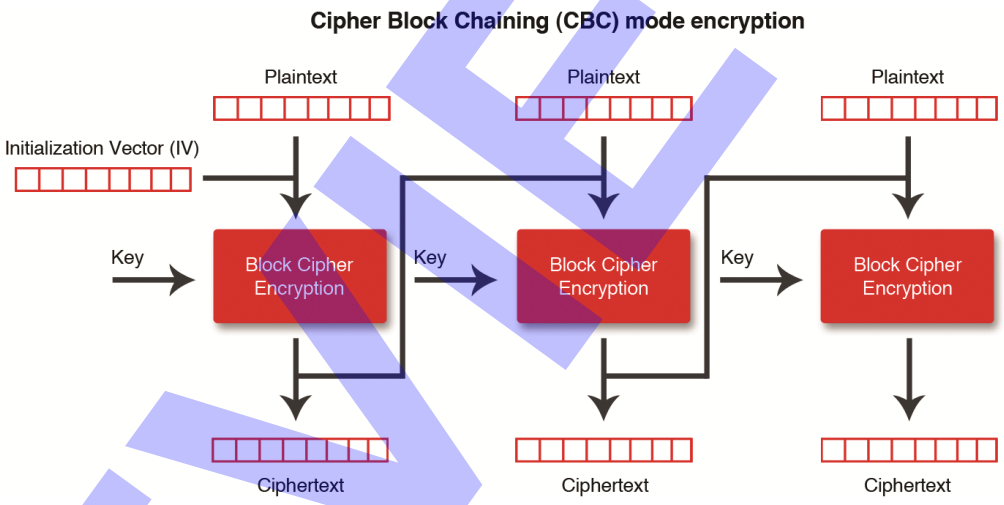


Exhibit 1-2: Cipher Block Chaining

- Cipher Feedback (CFB) mode is very similar in operation to CBC except that CFB does not require padding. This is beneficial in that applications do not need to have specific logic for the last block of data. Not requiring padding also protects against the padding oracle attack that effects CBC implementations. This attack rely on determining if the padding is correct or not and can be used to decrypt encrypted data.

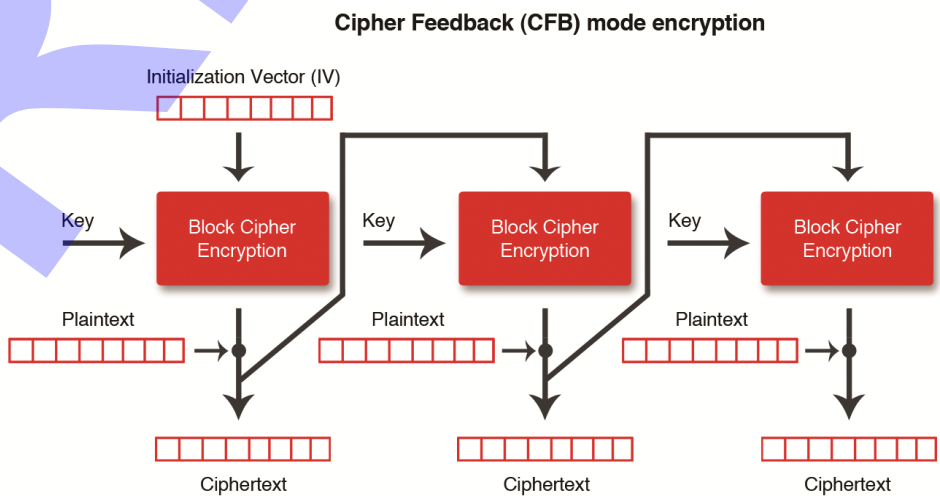


Exhibit 1-3: Cipher Feedback mode

PREVIEW
NOT FOR PRINTING OR INSTRUCTIONAL USE

Counter mode (CTR)

CTR is similar to CFB in that it does not require padding. It is different from CBC and CFB though in that instead of using an IV, it uses a counter to ensure that data blocks of the same content do not encrypt to the same value.

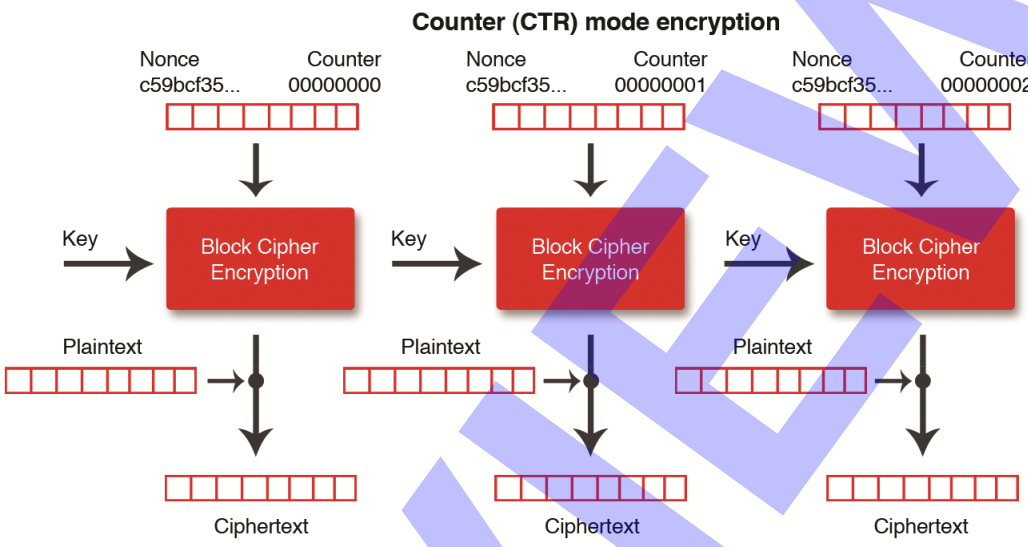


Exhibit 1-4: Counter mode

XTS

XTS mode is used to encrypt data that is randomly accessed, such as hard drives and RAM. It is used by disk encryption projects such as TrueCrypt.

DES

DES is a block cipher that was developed in in the 1970s by a collaboration of the NSA and IBM. It was one of the first widely used ciphers, and was considered secure until the late 1990s when computing power was able to bruteforce its short 56-bit key. It is no longer considered safe to use because of this reason.

Bruteforcing is the act of attempting to discover an encryption key by generating every possible key within the desired key's keyspace. For keys generated with a small keysize, modern computing power can generate all such keys in a reasonable amount of time for successful attacks to be mounted. For this reason, security policies often list a minimum key size that must be used with a specific encryption algorithm in order for protected data to be considered safe.

3DES

3DES, or Triple DES, was then developed as a more secure algorithm that would be resistant to bruteforce attacks. It can operate with key sizes of 56, 112, and 168 bits. 3DES is still considered secure although its popularity has dwindled in the face of AES, which has much better performance.

AES

AES is the most popular block cipher and is used throughout organizations and governments worldwide. It has no usable attacks published against it, and it is very efficient in terms of RAM usage and performance. It can operate with key sizes of 128, 192, and 256 bits.

PREVIEW
NOT FOR PRINTING OR INSTRUCTIONAL USE

Stream ciphers

Unlike block ciphers, which divide the data to be encrypted or decrypted into fixed-sized blocks, *stream ciphers* deal with the entire data set one unit at a time. This approach means that both modes of operation and padding are not applicable to stream ciphers. Stream ciphers work by generating a random sequence that is the same size as the data to be encrypted and then encrypts each plaintext unit against the sequence.

RC4

RC4 is the most used and well-known stream cipher. It is used in algorithms such as WEP, Bittorrent, Microsoft's Remote Desktop Protocol, and can be chosen as the cipher to use in a number of algorithms and protocols that support multiple ciphers. RC4 has a number of security weaknesses though, and protocols relying on RC4 must be aware of these. A prevalent example is WEP, used to secure wireless networks, which was completely broken in the early 2000s due to its misuse of RC4.

Public key cryptography

Explanation

Public key cryptography, also known as asymmetric cryptography, is different from symmetric cryptography in that two keys are used in the encryption and decryption process. This has a number of advantages in terms of usability, accountability, and authenticity.

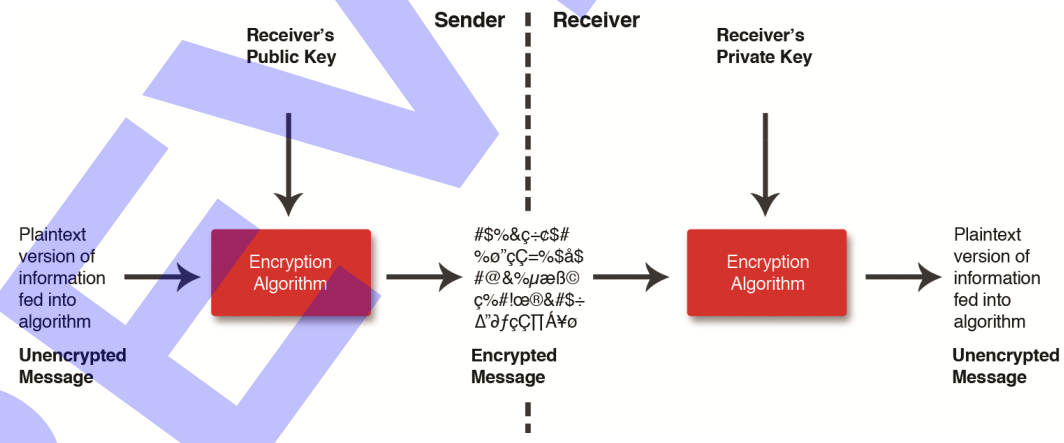


Exhibit 1-5: Public key cryptography

Public and private keys

The two keys of public key cryptography are known as the public and private keys. The private key is kept secret by the key owner, and the public key can be freely published and distributed. These keys work as a pair and both are required for secure, two-way communication.

We will now go through the process of Alice and Bob having a private conversation using public key cryptography.

Exchanging the key

Before any communications can be performed, both Bob and Alice must either give each other their public keys or publish them in an online accessible location. Secure exchange of keys will be discussed later, and is a required condition for secure use of public key cryptography. If an attacker can substitute his own public key for either party, then he can effectively perform a man-in-the-middle attack.

PREVIEW
NOT FOR PRINTING OR INSTRUCTIONAL USE

Sending a message

To send Bob an encrypted message, Alice encrypts it with Bob's public key. Upon delivery, Bob uses his private key to decrypt the message. This process ensures confidentiality as Alice has Bob's public key and Bob has his own private key to decrypt the message. When sending the message, Alice can be assured that only Bob can read her message.

Receiving the reply

To reply to the message, Bob encrypts his response with Alice's public key. Upon delivery, she can decrypt the message using her private key.

Note that while only Bob or Alice can read the encrypted messages, they have no guarantees of who the sender actually was (authenticity) or that the original message was not tampered with in transit (integrity). Digital signatures, discussed later, will explain how these two guarantees can be satisfied.

RSA

RSA is one of the most popular algorithms used for public key cryptography and is very secure. It was developed by computer scientists Rivest, Shamir, and Adleman (RSA) in 1978. It relies on the inherent difficulty of prime factorization for the protection of a user's public and private key pair. In this system, the public key is the product of two very large prime numbers. The private key, the two prime numbers, must then be kept secret as in other public key cryptographic schemes. RSA is used by government agencies, large corporations, in a number of hardware devices and software applications to provide public key cryptography capabilities.

Diffie Hellman

Diffie Hellman was the first key exchange protocol. It allows for two parties without any prior knowledge of each other to exchange keys securely over an untrusted communication medium. However, because the protocol is not authenticated, it is susceptible to a number of attacks, including man-in-the-middle. By adding authentication to Diffie Hellman, it becomes a very secure protocol and is used as the foundation for a number of other cryptographic protocols.

Hashing

Hashes can be considered "digital fingerprints" of a unique sequence of data. When using a good hash function, no two inputs should ever produce the same output (hash). This property can be used in a number of situations to prove that data has not been tampered with and that it is authentic.

Hash functions

A hash function, also known as a one-way function, is able to take in an arbitrary length input and produce a constant size output. To be used in a cryptographic setting, a hash function must satisfy a number of requirements:

- Any change in input should greatly change the resulting output hash.
- Given the hash, the original plain-text message should not be computable.
- No two inputs should generate the same output (collisions).

Hashing plays an integral role in a number of areas related to computer security. Understanding when to use hashing and how to verify hashes against datasets is a very important skill.

PREVIEW
NOT FOR PRINTING OR INSTRUCTIONAL USE

Passwords, credit cards, PII

Websites that deal with sensitive information, such as passwords, credit cards, and social security numbers, must take great care of that data. Long-term storage of the raw data would provide an unacceptable security risk and is banned by many regulations and laws. Instead, the common practice is for websites to receive such sensitive input from the user, hash it, discard the input, and the store the hash in a database. This ensures that the relevant data is still linked to the user but not immediately accessible to malicious attackers who access the database.

When a user makes subsequent visits to the website and needs to be authenticated, the website can then accept the given password, hash it, and compare to the contents in the database. Assuming a good hash function is used, this provides the same level of security as if the plain text password was stored in the database for comparison.

Digital forensics

Digital forensics, which is the field involved with the acquisition and examination of digital evidence, makes extensive use of hashing to prove that evidence was not tampered with during investigation. A very common example is for a forensics investigator to take a bit-for-bit copy of a hard-drive, known as an image, and then compute the hash of the image. Assuming best practices are followed, the copy of the original evidence will then never be tampered with and the hash can be later used in court if the opposing side questions the findings. Since hash functions are deterministic, a non-altered evidence source will later produce the same hash and the evidence will remain suitable for a legal setting.

Malware analysis and virus detection

For quick detection of previously seen malware, an analyst can build a database that maps hashes of previous samples to information collected about the samples. When a new sample comes in, a quick hashing of it, followed by a lookup in the database can save a tremendous amount of analysis time.

Hash collisions

A *collision* is when a hash function generates the same output for two or more different inputs. This breaks the entire trust model around hashes as the integrity of the message can no longer be guaranteed.

Using authentication with hashed passwords as an example, if a collision is found in the function used to generate the hashes, then multiple passwords can be used to log into an account. This matter is even worse in digital forensics as investigators can no longer prove that they did not tamper with evidence. Depending on the legal setting in which this takes places, it could invalidate entire investigations.

MAC

Message authentication codes (MAC) are hashes sent along with an encrypted message to ensure its integrity and authenticity during symmetric encryption based communications. Each MAC is computed with the shared key as well as a hash value of the message being sent. Together, this provides integrity of the message, as the receiver can verify the hash of the message, as well as authenticity, as the shared key used to generate the MAC is only known to trusted parties.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

HMAC

HMACs serve the same purpose as MACs, but use cryptographically sound hashing functions throughout the process. Common examples include MD5, SHA-1, and variations of SHA-2.

Digital signatures

A digital signature is used to provide integrity, authenticity, and non-repudiation of messages during public key cryptography based communications. In this scheme, a message to be transmitted is hashed, and the hash is then encrypted using the sender's private key. To verify the integrity of the message, the receiver decrypts the message using the sender's public key and computes its hash using the same hashing algorithm as the sender.

The digital signature is then decrypted and the two hashes are compared. If they match then the integrity of the message is verified. This process also ensures authenticity as only the message sender's private key can be used to encrypt the message and his public key must be used to decrypt the message. Non-repudiation is also guaranteed as the digital signature is encrypted by the sender's private key.

MD5

MD5 was published in 1992 and was the most popular hash function until the mid-2000s when a number of attacks were published against it. Though still in use in non-critical environments, the function is a considered broken and should not be used in new systems and existing systems should transition away from it.

SHA-1

SHA-1 was published in 1995 and experienced wide ranging adoption in a number of fields. Only theoretical attacks exist against SHA-1, but it has still been superseded by the SHA-2 functions.

SHA-2

SHA-2 was published in 2001 and refers to a number of functions with differing digest lengths (224, 256, 384, 512). SHA-2 is currently implemented in several widely used network protocols and adoption is expected to rise as the U.S government has chosen it as the replacement for SHA-1.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Do it!

A-1: Creating an MD5 hash

As websites can change over time, verify that this activity works correctly before class.

Here's how	Here's why
1 On the Windows host computer, open Firefox and access **md5generator.net**	There are command line and GUI utilities that can create MD5 hashes, but for this activity you will use an online generator.
2 In the Input String field, enter **Hello World** and click **Generate MD5 Hash**	This will calculate the md5 hash of the "Hello World" string and report it to you.
Copy the output into a Notepad window	Alternatively, you can open a new tab for the next step.
3 Generate a hash for **hello World**	The only difference is that the first letter is not capitalized.
Compare the second output with the first.	

4 Now that you've seen how two very similar inputs generated completely separate outputs, what requirements of a secure hash function can we say are met for the two sample inputs?

The first requirement of a good hash function: Any change in input should greatly change the resulting output hash.

5 A person new to the digital forensics field wants to hash an evidence drive, save it in secure storage, and then later prove it was not tampered with by re-hashing it. The procedure he plans is this:

1. Acquire the hard drive from his client.
2. Attach the hard drive to a computer in his forensics lab.
3. Boot the computer from the hard drive.
4. After logging in, use a program to calculate the hash of the running disk.
5. Save the hash to the chain of custody form.
6. Store the drive in a safe until needed, and then repeat steps 1-5.

Are there flaws in his procedure? If so, describe the flaws and why his evidence will be invalid for legal settings.

A good answer should recognize that the process listed in the question will change portions of the drive which will invalidate the hash and make it never repeatable. This will subsequently make it unusable in legal settings. Flaws include:

- ***Booting the computer from the hard drive, because this will change the data on it, which means it will change the hash from when it was given to the investigator by his client.***
- ***Logging in and using the computer will further change evidence.***
- ***Hashing the running system, which is pointless because data will change as the hash is being calculated.***
- ***The final issue is steps 5 and 6, because the saved hash is both useless and will never be repeatable because the computer will be rebooted, logged into, and hashed while it's turned on.***

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Protecting data at rest

Data at rest refers to information that does not leave the device on which it is stored. Protecting this data requires a different set of algorithms, protocols, and policies than those of data that traverses computer networks.

Full disk encryption

The most common example of data at rest is encrypted hard drives. A large majority of data stored on a hard drive is never transferred over the network, but there still a number of risks associated with it. When dealing with laptops, cell phones, and other mobile devices, the risk of theft, "forgetting it at the hotel," and other situations involving loss of physical possession of the device are very real.

Without proper encryption, anyone who gains access to the unencrypted device can also access the data. Security measures, such as passwords and PINs, may prevent a casual attacker from accessing data, but nothing short of full disk encryption will completely protect a device's contents. Proper use of full disk encryption, including selection of a strong passphrase, ensures that attackers who steal a device or its hard drive cannot access the plaintext contents.

Full disk encryption is generally done using a block cipher algorithm. All major full-disk encryption vendors use the XTS mode of operation due to its ability to deal with randomly stored and accessed data such as is on a hard drive.

File encryption

A similar concept to full disk encryption is encryption of individual files or folders. Programs such as TrueCrypt allow you to create encrypted containers that can be transferred and stored as regular files. This provides ultimate flexibility as the container can be opened and decrypted on any computer with the cryptographic software installed.

Code signing

Code signing is a protection mechanism that ensures a user that the application or executable that they have acquired is legitimate. Code signing works by the developer digitally signing the executable and then the user's operating system checking its integrity and authenticity before execution.

Microsoft Windows application and driver signing is the most well-known code signing implementation. It requires that code to be loaded onto the system be digitally signed by a known entity that is trusted by the operating system. Only after this verification can the software be loaded onto end-users' computers and servers.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Transport encryption

Transport encryption deals with the secure delivery of data between parties. It provides the ability to communicate over untrusted mediums while still guaranteeing confidentiality, integrity, authenticity, and non-repudiation. We have already discussed the basics behind these guarantees, and we'll now look the protocols that build on them.

Secure network protocols must defend against a wide range of active and passive attacks.

- Passive attacks are when a party can monitor a communication in an attempt to glean information from it.
- Active attackers are more dangerous in that they can intercept, modify, add, and remove packets from a network stream and between communicating parties. In order to be secure against both attack types, network protocols must use a range of cryptographic techniques to ensure full security.
- A man-in-the-middle attack (MITM) is when an active attacker silently tampers with communications between parties. If an attacker can substitute his own key for one of the communicating parties, then he can successfully decrypt the messages of one participant. If he can substitute for both keys, then he can decrypt (and tamper with) the entire conversation. Neither participant in the conversation will notice the attack as the attacker can successfully relay messages between parties.

  A common example of such an attack is rogue web servers or proxy servers with self-signed SSL certificates that users are tricked into trusting them. All web traffic can then be successfully decrypted by the attacker.

SSL

Secure Sockets Layer (SSL) was developed by Netscape as a protocol that supports sending of sensitive data across untrusted networks. It provides integrity and confidentiality of data and it can also provide authenticity depending on how the client and server are configured. SSL uses a mix of cryptographic techniques to provide its security guarantees.

- Authenticity — Public key cryptography is used to authenticate the server and client. There is usually also a certificate authority, discussed extensively in the PKI topic, which is used during this step to verify the public key of the server to the client.
- Integrity — SSL uses a MAC to provide integrity of the data sent. This prevents tampering of the communications.
- Confidentiality — Symmetric cryptography is used with a negotiated shared key to encrypt and decrypt messages between the client and server. This provides confidentiality as the shared key is only known to the client and server as they have the public and private keys necessary to encrypt and decrypt the channel on which the shared key was agreed.

TLS

Transport layer security (TLS) is the successor to SSL. It provides the same base functionality as SSL as well as newer enhancements to make it more flexible and secure. It is supported by many well-known protocols and is used to secure major Internet traffic such as web browsing and e-mail.

PREVIEW
NOT FOR PRINTING OR INSTRUCTIONAL USE

Public key infrastructure

Public Key Infrastructure (PKI) is the set of components necessary for two previously unknown parties to exchange public keys over an untrusted network. Secure use of public key cryptography requires that public keys can be exchanged between parties without tampering by a third party. In the case of the Internet and other non-private networks, this is impossible to do without a trusted third party, such as a PKI.

In addition to digital certificates, a complete PKI solution includes the use of three components:

- A certificate authority (CA)
- A registration authority (RA)
- A method to revoke certificates

The entire PKI infrastructure requires that digital certificates be generated and used by the servers and clients throughout the PKI network.

Digital certificates

Digital certificates are an integral component of PKI. They are used to tie a public key to the identity of the certificate owner, whether that is an individual, organization, or a specific website domain. The X.509 standard dictates the format of digital certificates. To prove the identity of an entity, it requires a number of entries in the certificate such as the certificate issuer, expiration date, owner, and the owner's public key.

Certificate Authority (CA)

The certificate authority (CA) is responsible for the generation and publication of certificates to be used within the PKI. To act as a trusted third party, the CA signs the generated certificate using its private key. The CA can then be used to verify the authenticity of an entity's public key as well as to tie the public key to the entity's identity.

Registration Authority (RA)

A registration authority (RA) acts as an intermediary between PKI clients and the CA. Its role is to receive requests from the client, validate them, and, if validated, send the request to the CA. The CA then sends the response to the RA who forwards it to the client. The RA has no real power within the PKI and simply acts to help scale the infrastructure by handling processing for the CA.

Certificate revocation methods

To ensure that certificates that can no longer be trusted, such as those that were generated incorrectly or that have been compromised, are not used anymore, PKI relies on certificate revocation. This ensures that sensitive transactions cannot be compromised through the use of untrusted certificates.

Certificate Revocation List (CRL)

A CRL is a simply a list of certificates that have been revoked. PKI clients request the current CRL list from the CA and then cache it until the next CRL list update. PKI operations then check a certificate against the CRL before performing processing with it.

PREVIEW
NOT FOR PRINTING OR INSTRUCTIONAL USE

Online Certificate Status Protocol (OCSP)

OCSP differs from CRLs in that, instead of downloading a list of revoked certificates, the PKI client queries the CA about the revocation status of a particular certificate. This method has a number of advantages and disadvantages over CRLs. The first advantage is that certificates that are revoked will then be immediately unused by OCSP clients. With CRL, the revoked certificate may still be used until the cache refresh time. A large disadvantage of OCSP is that the client must always be online in order to query the CA. There are also privacy issues with OCSP as the CA knows every certificate that a particular client is using. In the case of HTTPS, this can leak information about a person's browsing history and habits.

The PKI process

To illustrate the PKI process, we will walk through the usage of the HTTPS PKI by our fictional outlanderspices.com website. Because our website sells goods online, it is required to encrypt all of the sensitive transaction information between the client (a web browser) and the web server.

Obtaining an SSL certificate

To start, the owners of outlanderspices.com must buy a SSL certificate from a trusted CA. The two most popular SSL CAs are Verisign and GoDaddy. After deciding on a CA, the company must decide if it wants to purchase a certificate for individual subdomains or if it wants to purchase a wild card certificate. A wild card certificate, in this case *.outlanderspices.com, will secure transactions across any subdomain that is currently in use or later created by the webmaster. This has advantages in both costs and administration as there is only one certificate to manage. There are down sides though in that compromise of the certificates key will compromise all subdomains on the website.

Managing the purchased certificate

After purchasing the certificate, it must be then placed onto the production web servers. This encompasses downloading the generated files from the chosen CA and configuring the web servers to use them. There must also be a process in place to rotate the key before it expires or customers will be not secured when using the website.

A customer visits from the website

Upon visiting the website, the customer's browser will receive the server's certificate. It can then verify it against the CA that signed and issued the certificate to ensure that it is valid and for the correct domain. Once this process has taken place, the browser and the user are ensured that a secure channel has been established with the web server and that sensitive information can be safely submitted.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Perfect forward secrecy (PFS)

Perfect forward secrecy (PFS) is the notion that compromise of a key used during a particular session should not affect previously encrypted data. This is a very desirable trait, as long term keys, such as public/private key pairs, shared keys, and others, expose all data ever encrypted with them to risk if a key is compromised.

In cryptographic systems without PFS, data that was previously captured by an attacker can be vulnerable to decryption at any point in the future if the long term keys used during the communication are compromised.

Two real-world examples of this include:

- Decryption of previous HTTPS traffic through compromise of a web server.
- Decryption of previously sent and received e-mails through compromise of a user's desktop or laptop.

Obtaining PFS requires that long term keys are only used to derive ephemeral (per-session) keys and that the same ephemeral key is never used twice to generate other keys. By never using the same key twice, an attacker who compromises a session key or long term key can only decrypt one piece of information, such as one packet of a network stream or one conversation.

PFS is a standard function of the following protocols:

- SSH
- Off-the-Record Messaging (OTR)

PFS is optionally supported by the following protocols:

- IPSec
- TLS

Do it!

A-2: Examining an SSL certificate

Here's how	Here's why
1 In Firefox, navigate to **www.google.com**	
2 Next to the address bar, click the google.com bar	
In the dialog box, click **More Information**	
3 Click **View Certificate**	
Observe the General tab and the Details tab	
4 What is the Subject Public Key Algorithm used in the certificate? Who issued the certificate?	***Answers will vary depending on the certificate issuer.***

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

Unit summary: Cryptographic tools and techniques

Topic A

In this topic, you learned about **cryptography**, which includes the techniques, **algorithms**, and **protocols** that allow for secure communication between parties. To ensure the security of communications, cryptography makes a number of guarantees about its protection of the conversation, including **integrity**, **non-repudiation**, **authenticity**, and **confidentiality**. Combined, they ensure the communicating parties that the other person is who they say they are, that the message is legitimate, and that no other parties can **eavesdrop** on the conversation.

Review questions

1 True or false? Entropy is easy to gather as truly random data exists throughout RAM.

False

2 True or false? A good encryption key algorithm can remain secure even in the face of a badly generated random number.

False

3 A certificate authority (CA) can be used for which of the following purposes?

A To validate good-standing certificates

B To validate revoked certificates

C To answer queries from a registration authority (RA)

D To man-in-the-middle communications using keys registered with it

4 A hash generated from a secure hash function can be used for which of the following purposes?

A To recover the contents of the hashed data

B To prove that data was not tampered with since creation of the hash

C To provide a timeline of alterations to the data

D To prove who generated the hash

5 True or false? MD5 is a secure hash function.

False

6 Which of the following is used to prevent replay attacks?

A Digital signatures

B Digital certificates

C Hashing

D Nonces

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE

7 An attacker has compromised the private key of Alice. Assuming the attacker then monitors the network traffic between Alice's and Bob's next conversation, what type(s) of data can he uncover?

A Cleartext contents of messages Bob sends

B Cleartext contents of messages Alice sends

C Bob's private key

Independent practice activity

The questions for this unit will require answering questions about a number of scenarios where cryptography can be used in an enterprise.

1 An application processes marketing information related to the business's customers. Example data includes favorite types of cars, which sporting events the customer enjoys attending, and clothes preference. Does this data need to be encrypted before being stored? Why or why not?

This data does not need to be encrypted as it does not contain personally identifiable information (PII). Secondly, this type of information would be useless in its hashed form because it needs to be readable in its cleartext form in order to provide targeted marketing.

2 An online web application is adding an authentication module that will require the e-mail address and password of each user. Does this data need to be encrypted before being stored? If so, please specify what types of encryption seems appropriate.

At the very least the password needs to be encrypted, and if the company never uses the customer's e-mail address for communications, then it also needs to be encrypted. Both the password and e-mail addresses are sensitive customer information.

The encryption algorithm used should be a strong hashing function. The encrypted data should be stored in the application's database, and when a user attempts to later log-in, the given credentials should be hashed and then compared to the information in the database.

3 A client wants to communicate with his attorney using encryption. He wants to ensure that each message he sends can only be read by the attorney, and that each response is definitively from his attorney. What encryption technologies can the client use to meet his requirements?

The client should use public-key cryptography with his attorney. Public-key cryptography will allow for the exchange of public keys over any medium and will provide authenticity, non-repudiation, and confidentiality. Digital signatures can be used to provide integrity.

Shared-key cryptography would not provide as much security in this situation because it does not provide non-repudiation, which is very critical when sensitive information is being shared between parties.

4 How does the use of a trusted-third party, such as a certificate authority, prevent man-in-the-middle attacks?

A trusted-third party prevents man-in-the-middle attacks between two communicating parties by providing a trusted reference to keys and other information used during the communication process. For a client to ensure that it has a secure communication with the server, it first requests the server's certificate. It then validates this certificate with the certificate authority (CA). If an attacker has substituted his own certificate for the server's legitimate one, then the given certificate will not match what the CA says. This is an immediate red flag to the client not to proceed.

PREVIEW

NOT FOR PRINTING OR INSTRUCTIONAL USE