Shawn's C839v5/D334 All-In-One Study Guide

Applicability: Intro to Cryptography Courses: C839v5 (Exam Code DRO2) and D334 (Exam Code HNO1)

Contents:

1. Chapter Audio and/or Video Lesson Links + Recaps (Chapters 1-10 & 12).
2. Consolidated Algorithms List (Conventional & Light-weight).
3. Final Prep Practice Additional Practice Links – C839v5/D334 Testmoz and Quizlet questions.

Use Chrome for the audio file links

1. Chapter Audio/Video Lessons and Recaps

Chapter 1 Fundamentals Recap

Audio: Chapter1.mp3

- **Encryption** in simplest terms is changing plaintext into ciphertext; **decryption** is the process of changing it back.
- Encryption should be applied to information you want to protect **at rest** as well as **in transit.**
- You can use Aescrypt to **encrypt individual files** and **encrypt full disks** with options such as Bitlocker and FileVault.
- **Mono-alphabetic** code or substitution is where a single mapping from our alphabet to a cipher alphabet is created. Many early cryptosystems used this. **Polyalphabetic** refers to the mapping of our alphabet to a number of cipher alphabets. This approach added a bit of complexity to early cryptosystems.
- **One-time pad** is considered to be unbreakable since it only uses its cipher code once.
- **Pseudo-Random Number Generators** (PRNGs). This method repeats the random numbers after a given time (periodic). They are **fast** and are also **deterministic** and are useful in producing a repeatable set of random numbers.
- **True Random Number Generators** (TRNGs). This method generates a true random number and uses some form of random process. One approach is to monitor the movements of a mouse pointer on a screen or from the pauses between keystrokes. Overall, the method is generally slow, especially if it involves human interaction, but is **non-deterministic and aperiodic**.
- **Frequency Analysis** is a cipher cracking methodology that involves identifying patterns and **variations in the probability** of codes. i.e. a three-letter ciphered text combination spotted at the beginning of a string too often could tip us off that those three letters correlate the letters THE in the English alphabet.
- **Entropy** measures level of **unpredictability;** in encryption relates to the degree of uncertainty of the encryption process.
- Two common binary to characters **encoding methods** are **ASCII** (8-bit values, up to 256 characters) and **UTF-16** (16- bit values, up to 65,536 characters).
- Hardware vs Software encryption. **Hardware encryption is more efficient** than software encryption.
- A **hardware security module (HSM)** is a tamper-evident and intrusion-resistant physical device that safeguards and manages cryptographic keys and provides cryptographic processing.
- A **trusted platform module (TPM)** is a dedicated processor that handles hardware-level encryption; allows the use of full disk encryption on a hard drive in a manner that minimizes the impact on system performance. TPM contains the encryption keys.

Chapter 1 Number Theory Recap

Video: Shawn's Shorts_MOD and Binary Math.mp4

- **Binary Math**: Basic premise is knowing what combination of binary digits will produce a binary "1". AND requires **two 1's** to output a 1; OR requires **at least one 1** to output a 1, and with XOR, there ***must be a mismatch*** (i.e., one 1 and one 0) to output a 1.

With **AND**, both the first and second numbers you are ANDing must be 1 for the answer to be 1. Anything else = 0

```
11110
10100
-----
10100
```

With **OR**, if one or both of the numbers you are ORing is are 1, your answer will be 1. Anything else = 0

```
11110
10100
-----
11110
```

With **XOR** ⊕, one but not both of the numbers you are XORing must be 1 for your answer to be 1. Anything else = 0

```
11110
10100
-----
01010
```

- **Set Theory**:

Players — {mike, fred, bert}
Spectators — {ian, michael, mike}

The main symbols that we use are:

Symbol	**Symbol Name**	**Description**
$\mid$	such that	so that
$A \cap B$	intersection	objects belong to set A **and** set B
$A \cup B$	union	objects belong to set A **or** set B
$A \subseteq B$	subset	subset has fewer elements or equal to the set
$\in$	belongs to	when an object is within a set
$\notin$	does not belong to	when an object is not in a set

Thus $A \cap B$ — {mike} and $A \cup B$ —{mike,fred,bert,ian,michael}.
Then 'mike' $\in$ Players, and 'ian' $\notin$ Players.

- **Modulus Operator Math:** Simply divide the first number by the second and return the remainder. Annotated using the (mod) or (%). **Example:** 5 mod 2 = **1** [*2 goes into 5 a max of twice (4), 5-4 = the remainder which is **1***]
- **Combination vs Permutation**: **combinations** = not concerned with the order//**permutations** = all options considered including sequence.
- In **probability** theory we determine the **likelihood of an event happening**, typically by understanding the chances of how each of the elements involved in an event interact, and the likelihood of them happening. >> **Dependent, Independent, and mutually exclusive**.
- A **prime number** is a value which only has factors of **1 and itself** and used in areas such as key exchange and in public key encryption. Factorizing the result of the multiplication of two large prime numbers takes huge amounts of computational power and time.

Chapter 1 Early Cryptosystems Recap

Early Ciphers

Name	Description
Pigpen	Mono- alphabetic substitution cipher that makes use of mapping plaintext characters to graphical characters rather than to alphabetic ones. i.e. A=(pick a symbol), vs A=(pick a letter). Disadvantage: once the mapping is known, it is difficult to keep the message secret.
Rail Code	Employs a method to scramble text by writing it in a sequence across a number of rails.
BIFID	Makes use of a grid and which maps the letters into numeric values.
Playfair	5 × 5 matrix containing the alphabet less the letter J. Cipher/decipher process consists of a set of rules outlining use of column and row combinations.
Morse Code	Encoding method, rather than a cipher, that works by translating characters into sequences of dots (.) and dashes (-)
Caesar	Mono-alphabetic substitution cipher known as "shift" cipher. Involves plaintext being replaced by a letter some fixed number of positions down the alphabet. i.e., a Caesar Cipher using a shift of +3 would mean a plaintext letter A would result in a ciphertext letter D (a shift of three positions to the right in the alphabet).
Vigenere	Polyalphabetic cipher that involves using a different mapping, based on a keyword, for each character of the cipher. An advantage of this type of cipher is that the same plaintext character is likely to be coded to different mappings, depending on the position of the keyword, making guessing more difficult.
One Time Pad	Cipher code mapping that is used only once. Advantage is it is essentially unbreakable, disadvantage is it takes lots of work as you'd have to generate the pad to be used, each time.
Four-square Cipher	Uses four 5 × 5 matrices arranged in a square, are where each matrix contains 25 letters for encoding and decoding operations.
Enigma Machine	Used a polyalphabetic substitution cipher, which did not repeat within a reasonable time period, along with a secret key. For the cracking of the Enigma cipher, the challenge was thus to determine both the algorithm used and the key. Enigma's main weakness, though, was that none of the plain text letters could be ciphered as itself.

Check Your Knowledge: Chapter 1 Knowledge Check

Chapter 2 Secret Key Recap

Audio: Chapter2.mp3***-and-*** Ride_n_Review_Symmetric_Asymmetric Basics.mp4

- **Secret Key** encryption (symmetric Cryptography) makes use of a **single secret key for both encryption and decryption.**
- Since the same key is used for both encryption and decryption in symmetric cryptography, **a challenge that exists is finding a secure way to share or transport the key** between the entity encrypting and the one decrypting.
- **Diffie-Hellman** is a widely used **key exchange algorithm** used to exchange the secret key in symmetric cryptography.
- **Two types** of symmetric encryption: **Block and Stream**
- Symmetric block encryption involves grouping data into blocks and encrypting the **individual blocks**, and symmetric stream encryption involves encrypting **one bit at a time,** i.e., a synchronous stream
- Symmetric **stream encryption is** often much **faster** than block and can typically be applied in real-time applications.
- With symmetric block encryption, **padding is used to fill blocks to operating size** when the data does not fit perfectly.
- Symmetric block ciphers manage how blocks of data are processed through **block cipher mode** implementations. For instance, one may choose to use the DES block cipher configured with ECB as the mode of operation.
- **Common block cipher modes** covered in this course include ECB, CBC, CFB, OFB, and CTR.
- **CFB, OFB, and CTR** implementations essentially allow the block cipher to **operate like a stream cipher**.
- Secret key ciphers make use of substitution boxes (S-boxes) to perform substitution as part of the encryption process. **S-boxes** take a given input and leverage look-up tables to produce a given output.
- All current **cipher codes are crackable** and a measure of the security of a code is the **amount of time** it would take to break the code based on the **computational power available**. This is often referred to as the work factor. As processing power magnifies, security of current ciphers decreases.
- **Salting is** the process of adding an initialization vector to the ciphering process to change its operation and ensure that the ciphertext does not give the original plaintext when played back.

Chapter 2 Block Cipher Modes Recap

Video: Shawn's Shorts_ Block Cipher Modes.mp4

- ❑ Symmetric block encryption involves first grouping data to be encrypted (typically your plaintext) into blocks of a specific size and then encrypting those blocks.

- ❑ Block cipher modes merely outline **how the blocks will be handled** depending on the implementation selected (i.e., which mode is used). Implementation selection can be based on anything just as type of cipher can. Factors can include security needs or not, processing capacity, organization preference and so on.

- ❑ At minimum, each block of data will be encrypted using the encryption key of the block cipher being used as you will see with ECB. Other variations are configured to incorporate additional components to meet desired security and/or performance.

Ch2 Block Cipher Modes Basic Components

Block of plaintext

This represents the individual block of plaintext. The size depends on the specific block cipher being used. For instance, if DES is being used, the size of each block will be 64 bits.

This symbol fittingly represents the encryption key, which again is based on the specific cipher being used.

Encryption Function

This represents the encryption or cipher process, i.e., where the key is applied to encipher inputs.

IV

This represents an initialization vector (IV) used to incorporate randomness. With CTR mode you will see an alternative method to achieve this annotated by similar green boxes.

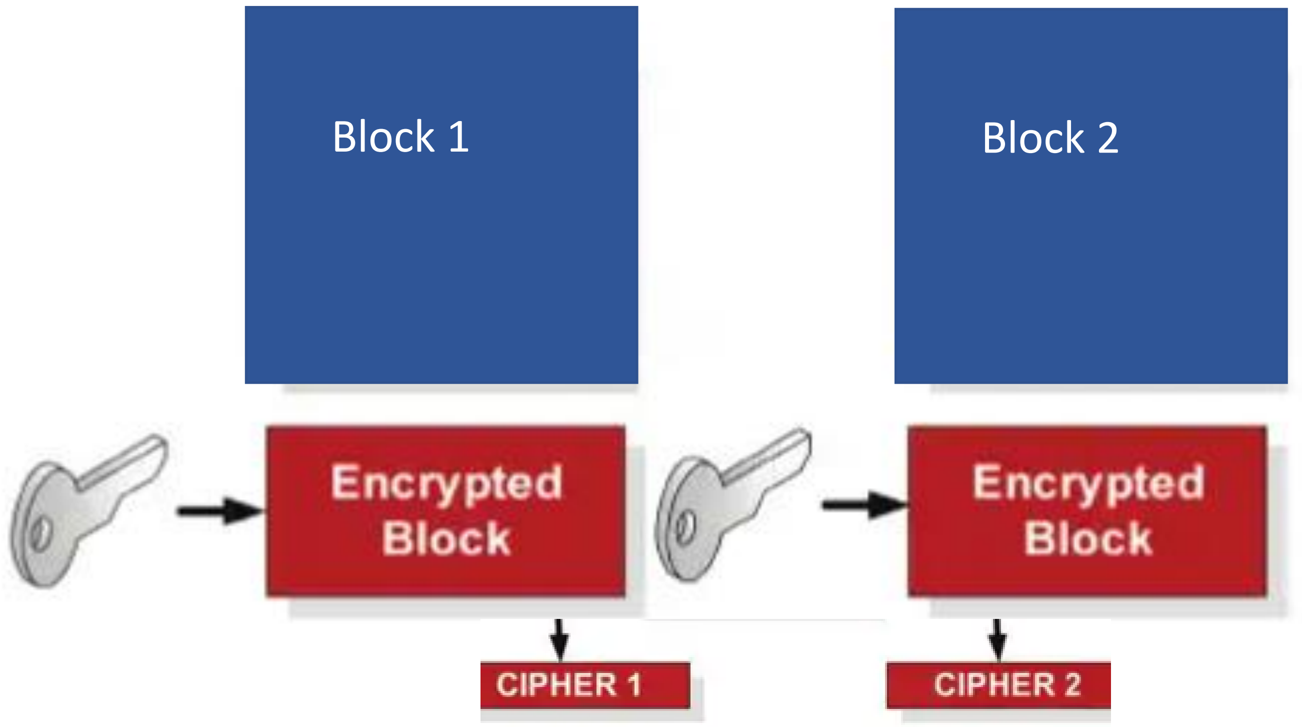
$\oplus$

This symbol annotates the exclusive or (X-OR) function.

Ciphertext

This represents the resultant ciphertext output from a given stage.

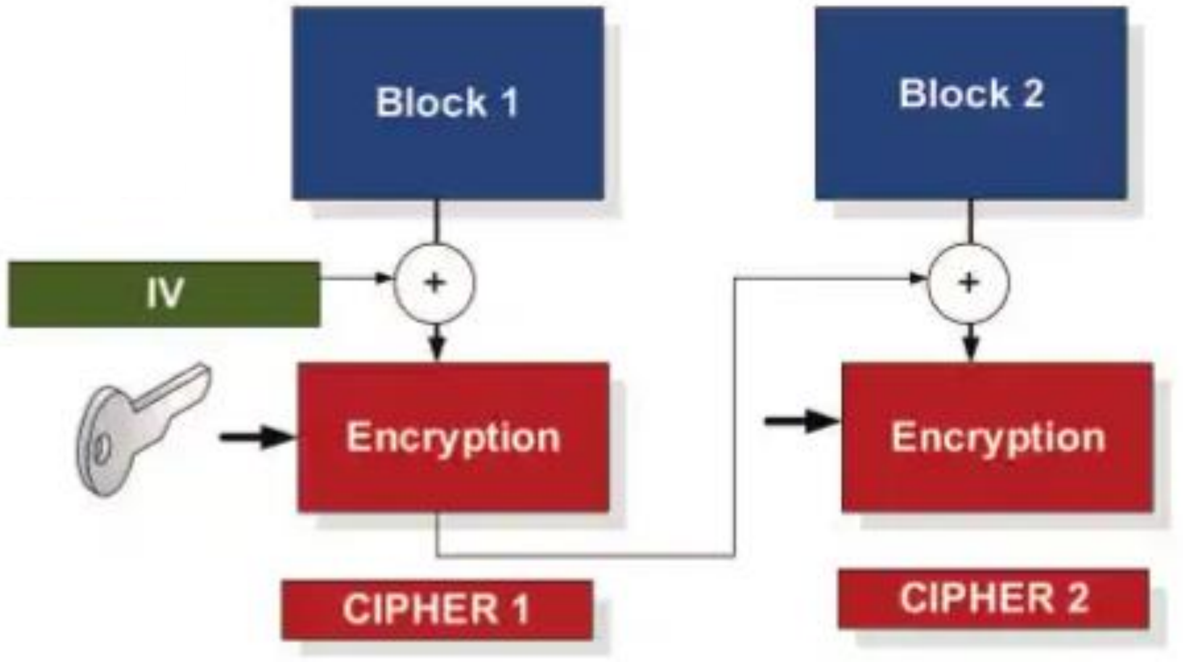
Ch2 Block Cipher Modes



Electronic Code Book (ECB)

- Most basic, **weak**, and unsecure mode.
- Each block is **processed separately**.
- **No Salt or IV** is used and the same key will be used to encrypt each block. This means if a given plaintext is encrypted in ECB and results in a given ciphertext, that **same ciphertext will be output EVERY TIME the same plaintext is encrypted**.

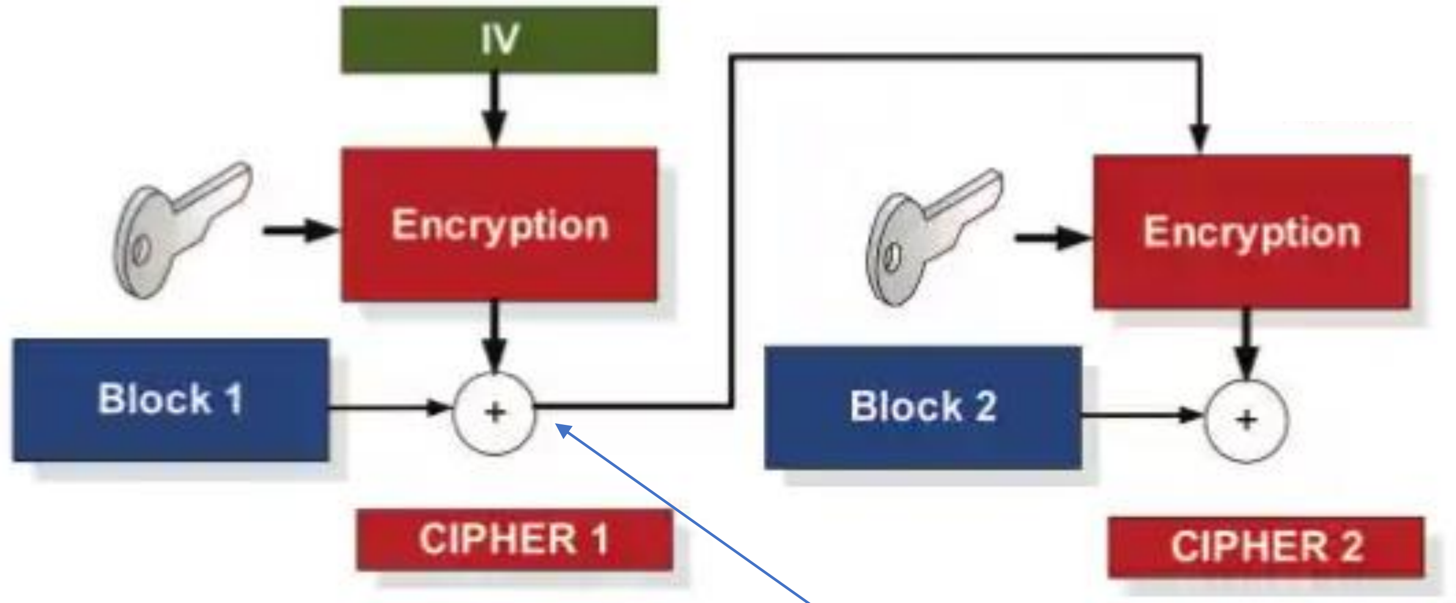
Ch2 Block Cipher Modes



Cipher Block Chaining (CBC)

- Minor **step up from ECB** with **the incorporation of an initialization vector** for the first block.
- Results of encryption from previous block is XOR’d with plaintext of the current block. That result is input into to encryption process of the current block.

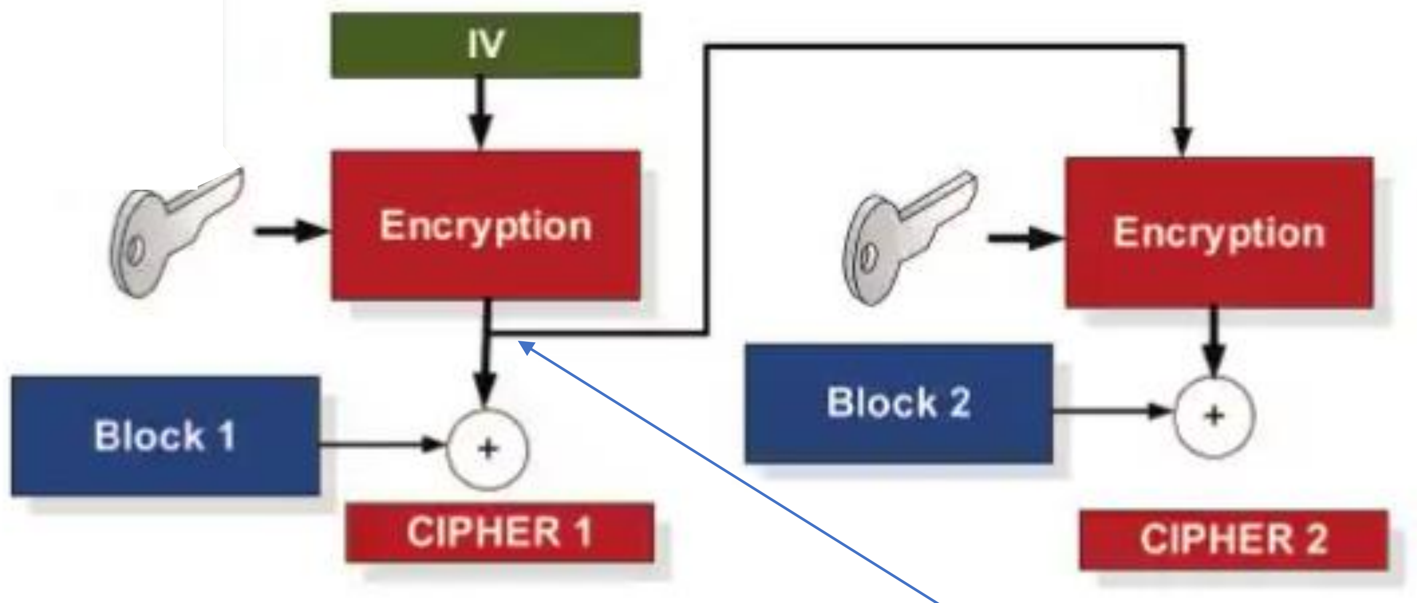
Ch2 Block Cipher Modes



Cipher Feedback (CFB)

- Converts the block cipher into a self-synchronizing **stream cipher.**
- **Current block takes output of the XOR ⊕ process** vs from the cipher stage of the previous block (difference between CFB and OFB).

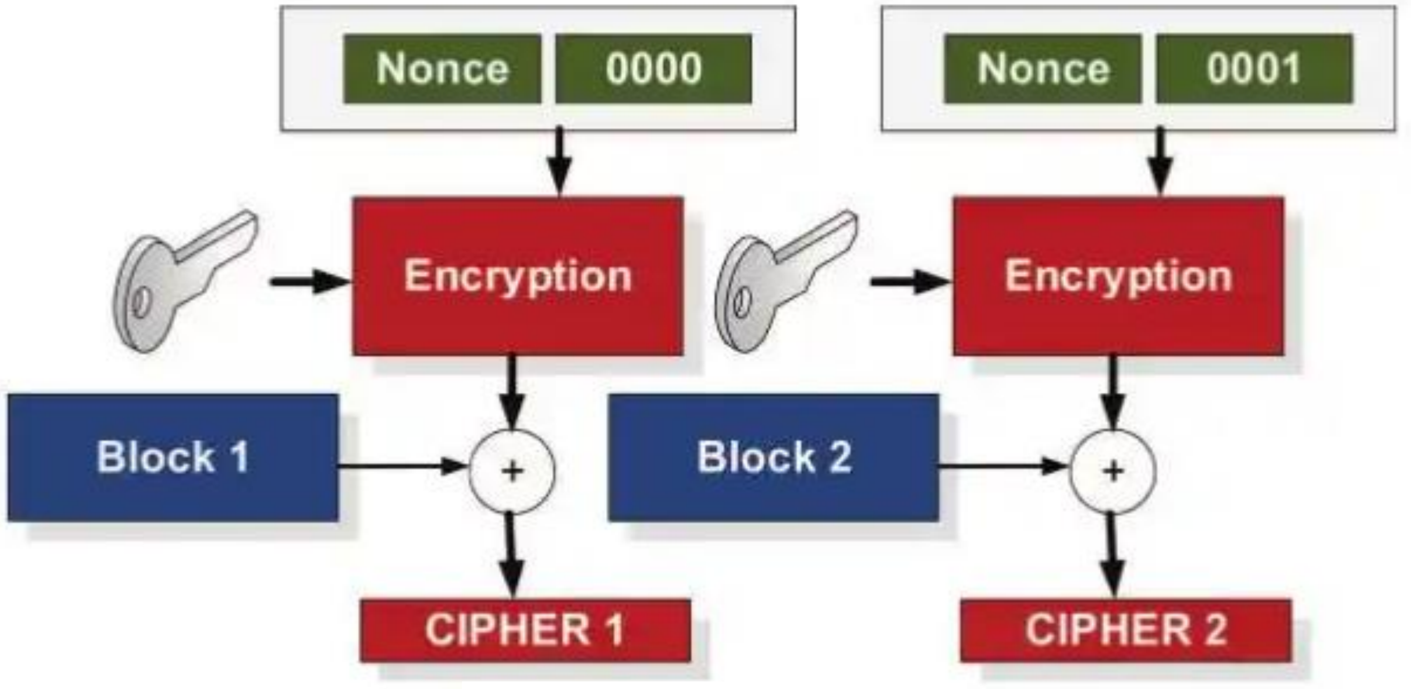
Ch 2 Block Cipher Modes



Output Feedback (OFB)

- Converts the **block cipher to a synchronous stream output**.
- **Current block takes output from the cipher stage** vs from the output of the XOR process of the previous block (difference between CFB and OFB).
- The first stage takes the data blocks and X-ORs it with the encrypted version of the IV value. The output of the first stage encryption is then feed into the next stage, and encrypted, with the output being X-OR'ed with the second block.

Ch 2 Block Cipher Modes



Counter Mode (CTR)

- **Converts the block cipher into a stream cipher**.
- Generates a **counter value** and a **nonce**, and encrypts this, in order to EX-OR with the plain text block.
- The advantage of CTR is that each block is processed independent of the others, facilitating ability to conduct **parallel processing of blocks**. i.e., feedback from other stages to feed into the current one is not required.

Check Your Knowledge: Chapter 2 Knowledge Check

Chapter 3 Hashing Recap

Audio: Chapter3.mp3*-and-* Ride_n_Review_Fundamentals_Hashing Basics.mp4

- **Hashing** describes **one-way** or irreversible encryption used for protecting the **integrity** of data and in authentication applications. Hashing is normally used to either hide the original contents of a message (such as hiding a password), or to check the integrity of data.
- Hashing involves taking a **variable length input** and producing a **fixed length output (message digest).**
- A weakness of one-way hashing is that the **same piece of plaintext** will result in the **same ciphertext** (unless **salt** is applied).
- Important factors with hash signatures are:
  - **Collision**. This is where another match is found, no matter the similarity of the original message. Collision attacks exploit this.
  - **Similar context**. This is where part of the message has some significance to the original and generates the same hash signature. This is defined as a Pre-image attack.
  - **Full context**. This is where an alternative message is created with the same hash signature and has a direct relation to the original message. This is an extension to a Pre-image attack.
- A **collision** occurs when two different input values that produce the same hash signature.
- The Apache-defined APR1 format **addresses the problems of brute forcing an MD5 hash**, and basically iterates the hash value 1,000 times.
- While APR1 has a salted value, the **SHA method has for storing passwords does not have a salted value**. SHA produces a 160-bit signature, thus can contain a larger set of hashed value than MD5, but because there is no salt it can be open to rainbow table attacks, and also brute force. (A **rainbow table** is a collection of precomputed hash values of actual plaintext passwords used for password cracking.)
- **HMAC** is a message authentication code (MAC) that can be used to **verify the integrity and authentication** of the message. It involves hashing the message with a secret key, and thus differs from standard hashing, which is purely a one-way function.
- Passwords which use a hashed value can be cracked as either with rainbow tables or brute force. An improved method of generating passcode is to generate a different one each time based on an initial seed value or based on time.
  - **One Time Passwords (OTP).** This allows a new unique password to be created for each instance, based on an initial seed.
  - **Timed One Time Password (TOTP).** This allows for a new unique passcode to be created for each instance, based on an initial seed and for a given time period.
  - **Hashed One Time Password (HOTP).** This allows a new unique passcode to be created each instance, based on a counter value and an initial seed.

Check Your Knowledge: Chapter 3 Knowledge Check

Chapter 4 Public Key Recap

Audio: Chapter4.mp3 ***-and-*** Ride_n_Review_Symmetric_Asymmetric Basics.mp4

- **Public Key** encryption (asymmetric cryptography) **makes use of a key pair** (one public, one private) to perform encryption and decryption. If a given key in a key pair is used for encryption, only the opposite key in that key pair can perform the reverse decryption.
- Only the **public key** should be distributed or **shared** (exchanged via digital signature, posted on a site etc.).
- Public-key encryption is an excellent method of keeping data secure, but it is often **too slow for real-time communications.**
- Two main applications of public key encryption are **identity checking** and **key protection**.
- **RSA** leverages the fact that products of **large prime numbers** are difficult to factorize as basis of its encryption.
- With **homomorphic encryption**, we can **perform mathematical operations** on ciphered values i.e., **before decryption.**
- **RSA** has a **heavy overhead** on processor loading and is **not well suited for embedded systems** (as the power drain can be high, along with heavy requirements for processing and memory).
- An improved solution over RSA is **Elliptic Curve** which is often used in **key exchange methods** (such as with Elliptic Curve Diffie Hellman – ECDH) and for the creation of **digital signatures** (Elliptic Curve Digital Signature Algorithm – ECDSA).
- The main **advantages of Elliptic Curve** methods are:
  - Much smaller keys. The prime number P is normally only 160 bits, and much smaller than in RSA. This considerably speeds up the encryption process.
  - Creation of the curves are more difficult than generating prime numbers, which makes it more difficult to crack than RSA.
  - They can be used to factorize values, such as finding the prime number factors within RSA.
- **Bitcoins use Elliptic Curve cryptography** with 32-byte private keys (which is a random number) and 64-byte public keys, on a secp256k1 curve.
- **El Gamal** is a public key method that is used in both **encryption and digital signing**. It is used in many applications and uses **discrete logarithms.**
- **Cramer-Shoup** is a public key encryption method that is an extension of El Gamal but adds a one-way hashing method which protects against an adaptive chosen ciphertext attack.
- The **Paillier** cryptosystem supports **homomorphic** encryption.

Ch 4 Message sending using public key cryptography without Hashing Recap

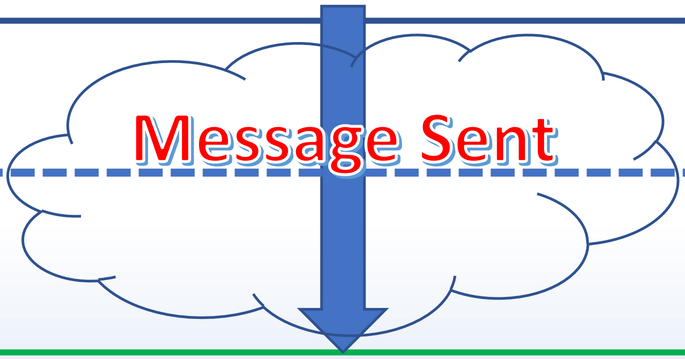
Audio: Shawn's Shorts_ Key Pairs.mp4

*****Learn these steps and know them like the back of your hand*****

The 4 basic steps to send a digitally signed message using public key cryptography include:

Sender

- **Step 1**: [Sender signs] Sender signs message with sender's own private key
- **Step 2**: [Sender Encrypts] Sender encrypts message with receiver's public key



Receiver

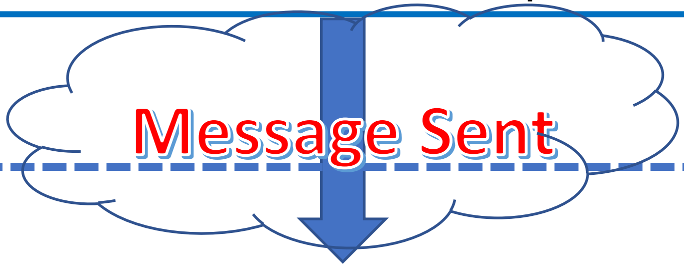
- **Step 3**: [Receiver Decrypts] Receiver decrypts message with receiver's own private key
- **Step 4**: [Receiver verifies] Receiver verifies message with sender's public key

Ch 4 Message sending using public key cryptography w/ Hashing Recap

The 5 basic steps to send a confidential, authenticated message that incorporates hashing ensure integrity are as follows:

Sender

- **Step 1:** [Sender signs] Sender produces a hash of the message and encrypts it using sender's private key [this is considered “signing” when hashing is incorporated in the process]
- **Step 2:** [Sender Encrypts] Sender packages up [original message + encrypted hash] and encrypts both with the receiver's public key



Receiver

- **Step 3:** [Receiver Decrypts] Receiver decrypts packaged up [original message + encrypted hash] with receiver's private key
- **Step 4:** [Receiver verifies] Receiver decrypts the encrypted hash with sender's public key
- **Step 5:** [Receiver verifies] Receiver computes hash of the original message and compares to the now decrypted original hash to ensure they match.

Check Your Knowledge: Chapter 4 Knowledge Check

Chapter 5 Symmetric Key Exchange Recap

Audio: Chapter5.mp3 **Video**: Shawn's Shorts Symmetric Key Exchange.mp4

- The **major problem** of secret-key encryption is **how to pass the key** between the entity encrypting and the entity decrypting.
- The **two main methods for key exchange** in symmetric cryptography is to (1) use a **key exchange algorithm** (such as Diffie-Hellman) or (2) encrypt the key with the recipient's public key, pass it to the other side and then allow the recipient use their private key to decrypt it i.e., **via public key encryption**.
- An important concept within key exchange is the usage of **forward secrecy**, which means that a compromise of the long-term keys will not compromise any previous session keys.
- With **ephemeral key** methods, **a different key is used for each connection**, and, again, the leakage of any long-term key would not cause all the associated session keys to be breached.
- **Diffie-Hellman** is a widely used **key exchange algorithm** used to exchange the secret key in symmetric cryptography.
- A **weakness** discovered in Diffie Hellman is that it is fairly easy to precompute values for two popular Diffie-Hellman parameters (and which use the DHE_EXPORT cipher set).
- The **DHE_EXPORT Downgrade attack** involves forcing the key negotiation process to default to 512-bit prime numbers. For this the client only offers DHE_EXPORT for the key negotiation, and the server, if it is setup for this, will accept it. The **precomputation of 512-bit keys** with g values of 2 and 5 (which are common) **are within a reasonable time limits**.
- **Methods to combat DHE_EXPORT Downgrade attacks** on Diffie Hellman include: (1) Disabling Export Cipher Suites, (2) Using (Ephemeral) Elliptic-Curve Diffie-Hellman (ECDHE), (3) Use a strong group.
- Diffie Hellman has **three groups** (bases): Group 1, Group 3 or Group 5, which vary in the size of the prime number used.
- **Diffie-Hellman methods** have been used extensively to create a shared secret key but **suffers from man-in-the-middle attacks**, where an attacker sits in-between and passes the values back and forward and negotiates two keys: one between a sender and the attacker, and the other between the receiver and the attacker. An improved method is to use **public key encryption**.
- The strength of Diffie-Hellman relates to the **size of the prime number bases** which are used in the key exchange.

Check Your Knowledge: Chapter 5 Knowledge Check

Chapter 6 Digital Certificates Recap

Video: Chapter6_Vid

- **Common Certificate Applications**: Server authentication, Client authentication, Code signing, Email signing, Time stamping, IP security, Windows hardware driver verification, Smart card logon, Document signing, Public key transport.
- **Common Certificate Types**: IKE, PKCS #7, PKCS #10, RSA signatures, X.509v3.
- **Common uses of Common Pubic-Key Cryptography Standards (PKCS)**:
  - **PKCS #5** - Used for password-based encryption.
  - **PKCS #7** - Used to sign and/or encrypt messages for PKI.
  - **PKCS #10** - A standard format used for requesting digital certificates from certificate authorities.
  - **PKCS #12** - Used to bundle a private key with its X.509 certificate or to bundle all the members of a chain of trust.
- **Example Use: Passing PKI public keys in a verifiable way**. When a digital certificate is created, (whether self-generated/signed or by a trusted well-known Certificate Authority (CA) such as Verisign or Entrust, the certificate will contain the public key of the certificate owner. So, generating and securely sharing a certificate that can be validated by a trusted source is a viable option for public key transport in PKI.
- **4 Basic steps for obtaining a digital certificate signed by a trusted Certificate Authority (CA):**
  - **Step 1**: Requester generates a key-pair (one public, one private). Public key is provided to the CA.
  - **Step 2**: Requester creates and submits a Certificate Signing Request (CSR), along with requester's public key to the CA.
  - **Step 3**: CA generates the digital certificate for the requester.
  - **Step 4**: CA signs the requester's digital certificate with the CA's own private key, and issues certificate to requester.
- **Two major encoding schemes for X.509 certificates**: PEM (Base64 ASCII text) format, and DER (binary) format
- **Common X.509 Certificate file types**: ***.cer*** (used with both PEM and DER formats), others - ***.crt, .pem, .key*** (common with PEM formats) and ***.der*** (common with DER formats).
- With **end-to-end authentication**, the user authenticates themselves to the end service, and with **intermediate authentication**, only part of the conversation between the entities is authenticated.

Ch 6 Certificate Management Recap

- **The main stages of key/certificate management include**:
  - **Initialization.** This includes registration, **key pair generation**, certificate **creation** and certificate/key **distribution**, certificate **dissemination**, and key backup.
  - **Issued.** This includes certificate **retrieval**, certificate **validation**, key recovery and key update.
  - **Cancellation.** This includes certificate **expiration**, certificate **revocation**, key history and key archiving.
- Certificates receive a **period of validity** designation (timeframe the cert is valid and should be trusted) at creation via a start and end date or expiration date.
- There are also instances where a certificate needs to **moved to an invalid/untrusted state (revoked) prior to the original expiration date**. Some reasons warranting having a cert revoked include but are not limited to:
  - Issuing CA was compromised
  - The cert itself was compromised
  - Certificate affiliation has changed
  - Certificate has been updated or superseded
- RFC 5280 thus defines "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", and includes two main states for revocation:
  - **Revoked.** This is where a certificate has been revoked, and cannot be reversed, and often occurs when a certificate is defined as having its private key breached.
  - **Hold.** In this case the certificate's trust level is on hold and can be reversed at some time in the future. It could relate to a private key being thought to be compromised, but where an investigation has show that it has not been breached.
- The **CRL must be published by the CA who originally generated the targeted certificates** and is only valid for a given amount of time (which is typically less than 24 hours).
- **CRLs are signed by the CA**, in order that they can be validated, and thus signed by the private key of the CA, and then checked against its public key (which is stored in a root certificate folder or preinstalled within a Web browser).
- An alterative to CRL is to use **Online Certificate Status Protocol (OCSP),** an online service used to check the validity of a certificate.

Check Your Knowledge: Chapter 6 Knowledge Check

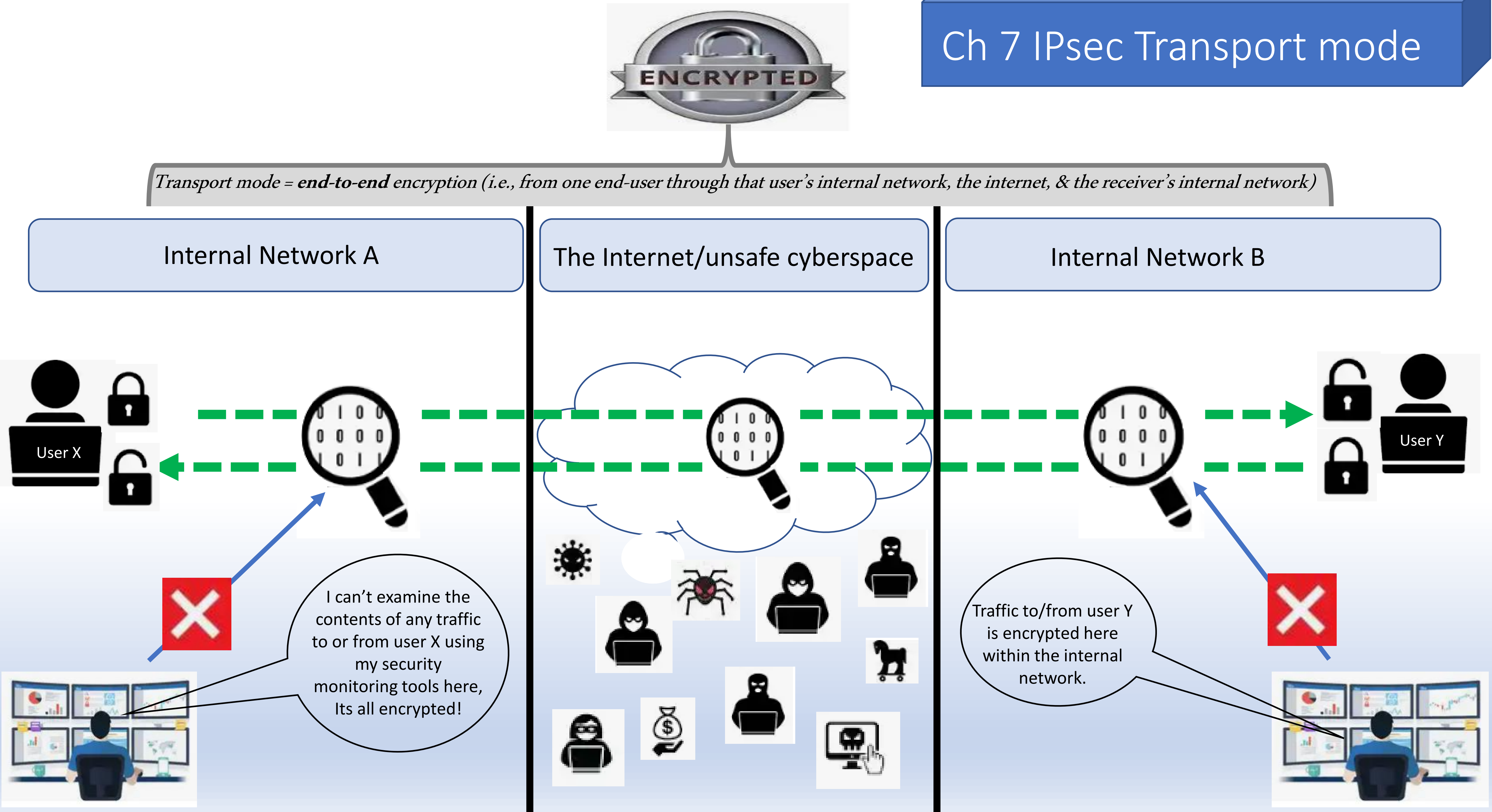
Chapter 7 Tunneling Recap

Audio: Chapter7.mp3

- The most common tunneling protocols are PPTP (Point-to-point Tunneling Protocol), L2TP (Layer 2 Tunneling Protocol) and IPSec.
- One of the greatest **flaws of SSL v2** was the usage of "export-grade ciphersuites" – which were created to comply with US Export regulations, and which made sure that the **keys were crackable**. This included a small key size, such as using a 40-bit session key for a connection. Additional **SSL risks** include DROWN, POODLE, and FREAK.
- With **SSL/TLS, the tunnel is created with a symmetric key** method (such as with RC4 or AES), and then a signature is created with a defined hashing method (such as SHA-1 or MD5).
- With a Virtual Private Network (**VPN**) tunnel we aim to create a connection from a host machine to a trusted network, and which is **tunneled through a public network**.
- The IPsec protocol includes two mechanisms which can be used separately or together. They are **ESP (Encapsulated Security Protocol)** and **AH (Authentication Header).** View the AH/ESP Transport vs Tunnel slides below to see how each works in either mode.
- There are two main phases in **setting up an IPSec connection:**
  - First phase defines the Internet Key Exchange, where the hashing method, and encryption and key exchange methods are defined.
  - The second phase defines the policies to be used for the tunnel.
- IPSec handshake takes place on **UDP port 500 for key exchange**. The protocol number will be **50 for ESP** and **51 if AH is being used**.
- **IPSec Modes**:
  - With IPsec **transport mode**, we have end-to-end tunnelling, where the encryption scope spans across of the network,
  - Within the **tunnel mode,** the connection is tunneled over a public network, but the network traffic is unprotected on either side of the connection. This mode allows for the inspection of network packets on either side.
- **Onion routing** involves using subscriber computers to route data packets over the internet, instead of using publicly available routers.
- With the **Tor** network, the routing is done using computers of volunteers around the world to route the traffic around the Internet, and within each hop the chances to trace the original source significantly reduces.

Ch 7 IPsec Transport mode

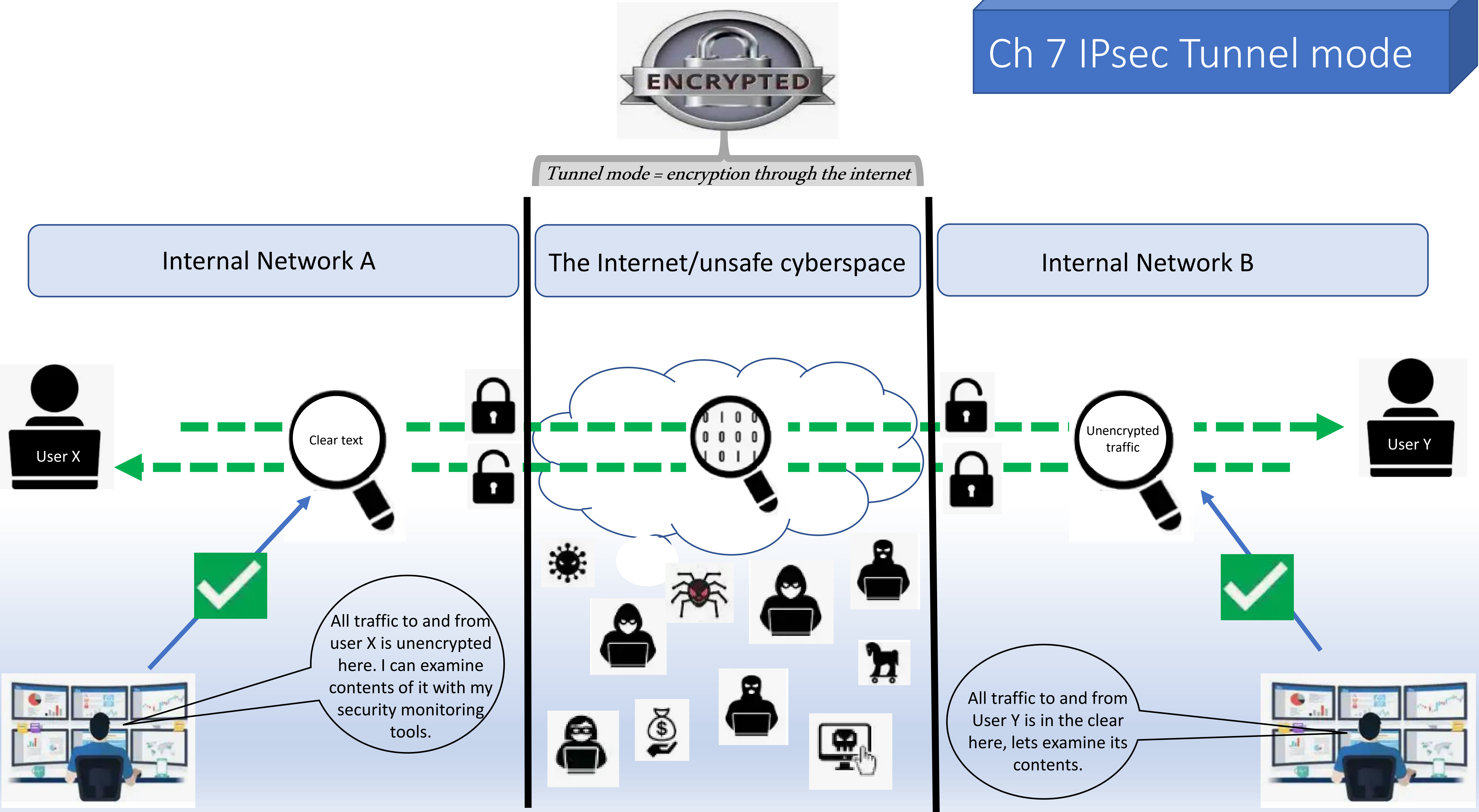
*Transport mode = **end-to-end** encryption (i.e., from one end-user through that user's internal network, the internet, & the receiver's internal network)*



I can't examine the contents of any traffic to or from user X using my security monitoring tools here, Its all encrypted!

Traffic to/from user Y is encrypted here within the internal network.

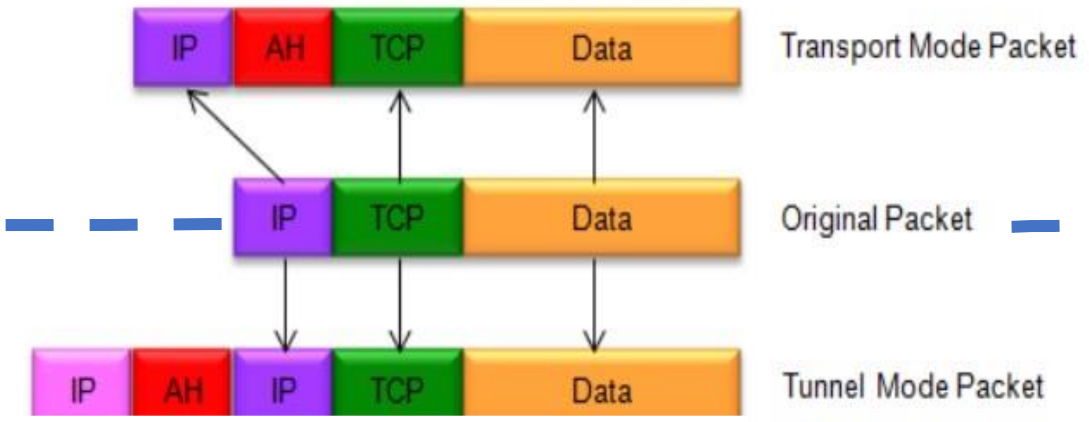
Ch 7 IPsec Tunnel mode


ENCRYPTED
Tunnel mode = encryption through the internet
Internal Network A
The Internet/unsafe cyberspace
Internal Network B
User X
Clear text
Unencrypted traffic
User Y
All traffic to and from user X is unencrypted here. I can examine contents of it with my security monitoring tools.
All traffic to and from User Y is in the clear here, lets examine its contents.

Authentication Header (AH) Transport vs Tunnel Mode

Transport Mode with AH

- Original IP header is added for routing
- AH header is inserted between the IP header and the transport-layer header providing protection for the transport-layer protocol.



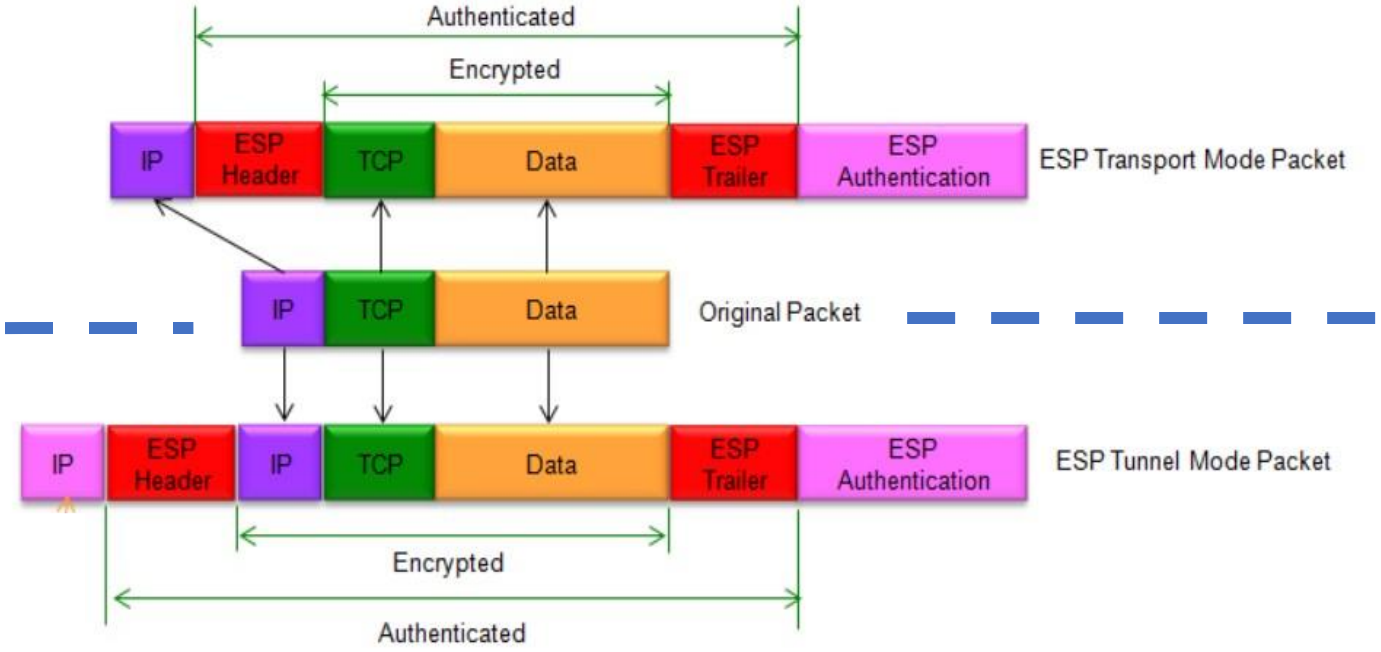
Tunnel Mode with AH

- New IP header is added for routing.
- The entire original IP packet is encapsulated in new IP packet, with the AH header inserted between the new IP header and the original IP header, providing protection for the entire IP packet.

Encapsulating Security Payload (ESP) Transport vs Tunnel Mode

Transport Mode with ESP

- Original IP packet payload is encrypted.
- The Original IP header, ESP headers and trailers and ESP authentication trailer added for routing.



Tunnel Mode with ESP

- Original IP packet is encrypted.
- A New IP header, ESP headers and trailers and ESP authentication trailer added for routing.

Check Your Knowledge: Chapter 7 Knowledge Check

Chapter 8 Crypto Cracking Recap

Audio: Chapter8.mp3

- In terms of a **backdoor** in cryptography, the two main methods which could be used are:
  - **Key escrow**. This is where a copy of the encryption key is kept in escrow so that it can be used by a government agent.
  - A **NOBUS** ('nobody but us') backdoor. This is where it is mathematically possible for government agents to crack the encryption, but no-one else can.
- There are several methods that an intruder can use to crack a cipher, including:
  - **Exhaustive search.** Where the intruder uses brute force to decrypt the ciphertext and tries every possible key.
  - **Known plaintext attack**. Where the intruder knows part of the ciphertext and the corresponding plaintext. The known ciphertext and plaintext can then be used to decrypt the rest of the ciphertext.
  - **Man-in-the-middle.** Where the intruder is hidden between two parties and impersonates each of them to the other.
  - **Chosen-ciphertext.** Where the intruder sends a message to the target, this is then encrypted with the target's private key and the intruder then analyses the encrypted message. For example, an intruder may send an e-mail to the encryption file server and the intruder spies on the delivered message.
  - **Active attack**. Where the intruder inserts or modifies messages.
  - **The replay system.** Where the intruder takes a legitimate message and sends it into the network at some future time.
  - **Cut-and-paste.** Where the intruder mixes parts of two different encrypted messages and is able to create a new message. This message is likely to make no sense but may trick the receiver into doing something that helps the intruder.
  - **Time resetting.** Some encryption schemes use the time of the computer to create the key. Resetting this time or determining the time that the message was created can give some useful information to the intruder.
  - **Time attack.** This involves determining the amount of time that a user takes to decrypt the message; from this the key could be found.
- **AES** has proven to be free from major vulnerabilities, but **poor implementation of the encryption** method leaves it susceptible to attacks such as: Brute force, use of Non-Random Numbers, and copy-and-paste.
- RSA suffers from several weaknesses and is susceptible to numerous attacks and cracking methods.

Check Your Knowledge: Chapter 8 Knowledge Check

Chapter 9 Light-weight Cryptography Recap

Audio: Chapter9.mp3***-and-*** Ride_n_Reveiw_Lightweight Cryptography.mp4

- Most **conventional** cryptosystems require too much **processing powe**r; too much **physical space**; and consume too much **battery power** for implementation in IoT, embedded SYSTEMS, and RFID.
- **Light-weight cryptography essentially** consists of cryptosystems able to support IoT, embedded systems, RFID etc. (i.e. provide cryptographic functions but require less processing power, physical space, and battery power than conventional cryptosystems).
- NIST outlines the device spectrum as:
  - **Conventional cryptography**. Servers and Desktops. Tablets and smart phones.
  - **Light-weight cryptography.** Embedded Systems. RFID and Sensor Networks.
- **Quantum computers** have fast multiplication circuits, and thus can be used to perform multiplications and search a range of prime numbers at a speed which would break most existing RSA implementations.
- A **Merkle tree** is a tree that defines each non-leaf node with a value or a label and contains a hash of its children. This builds a hash trees and is used to provide a verification of large-scale data structures.
- **Lattice-based cryptography** uses asymmetric cryptographic primitives based on lattices. It has been known about for several decades, and is now being investigated because of its quantum robustness, whereas many of the existing public key methods such as RSA and Diffie-Hellman cryptosystems can be broken with quantum computers
- Be sure to familiarize yourself with the light-weight cryptosystems discussed in the text on the cryptosystems drill slides.

Check Your Knowledge: Chapter 9 Knowledge Check

Chapter 10 Blockchain and Crypto-currency Recap

Videos: Chapter10_Vid

- One of the most popular crypto-currencies is **Bitcoin** (BTC). A key focus is for the crypto-currency to protect against someone spending money that they do not have, so **Bitcoin uses Blockchain**.
- **Blockchain** is a publicly available **ledger of transactions** that allows the Bitcoin network to know the number of bitcoins that a given user has in their account. Can be public or private.
- A blockchain **mining process** where a new block of transactions is added to the blockchain and transactions within the block are considered to be processed occurs **every 10 minutes or so.**
- Conventional currencies usually have a central bank that creates money and then controls its supply. The **Bitcoin currency is instead created when users mine** for it, using their computers to perform complex calculations through special software.
- Bitcoin transactions will be captured by **miners** who will compile a list of the latest transactions. If valid, the transaction is then recorded within a mining process, where mining nodes gather new transactions and compute a hash of the new block, and which should also contain the hash of the previous block, and then build a transaction log. Once complete, this becomes **part of the official Blockchain in the network**, and the miners reach a consensus on the current Blockchain. **Miners receive rewards** for successful mining efforts.
- The crypto currency **Ethereum** was built on the Bitcoin/Blockchain concept but included the concept of **smart contracts**.
- Within Ethereum applications we define the concept of **gas.** This is basically the unit that is used to measure the amount of work that is required to perform a single Keccak-256 hash.
- **Smart contracts** are programs stored on a blockchain that run when predetermined conditions are met; they typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without an intermediary's involvement or time loss.
- Along with creating a new currency (**Ether),** the main contribution of Ethereum is to create the concept of **peer-to-peer smart contracts** which enables users to create their own contracts, and which will be strictly abided to.
- Although Bitcoin, Ethereum, and Hyperledger all use blockchain, **Ethereum, and Hyperledger are considered "common" options for implementing blockchain** for this course. **Hyperledger and Ethereum offer the smart contracts feature.**

Check Your Knowledge: Chapter 10 Knowledge Check

Chapter 12 Wireless Recap

Audio: Chapter12.mp3

- Encryption schemes commonly used with Wi-Fi include: 40-bit RC4 (WEP), TKIP with 128-bit RC4 (WPA – Wi-Fi Protected Access), or 128-bit AES-CCMP (WPA-2).
- Because of export restrictions, the size of the original key in RC4 was limited to **40 bits** and was then increased to **128 bits**.
- Overall **WEP is weak** from a number of viewpoints:
  - Small value of IV (24 bits). This meant that it repeated within a reasonable time, and the key could then be attacked.
  - Construction of keys made it susceptible to the weak key attacks (FMS attack).
  - Lack of protection against message replay. There was no protection against cipher streams being played back over the network.
  - Lack of message tampering identification. The method did not support the detection of message tampering.
  - Directly used a master key. The method had no way of updating the keys.
- After WEP, there was a strong need to fix the problems, but to keep compatibility, thus **WPA supported TKIP**, and which increased the **IV value to 48 bits** (rather than 24 bits)
- **WPA addressed the weaknesses of WEP**, and without requiring significant hardware changes, and focused on two main methods: WPA-PSK and WPA Enterprise.
- **WPA-2** (IEEE 802.11i-2004) advanced the WPA standard, by keeping compatibility with WPA, but adding **AES-CCMP** (AES-Counter Mode CBCMAC Protocol), which is a block encryption method. Again, it supported two modes: Personal (with a pre-shared key) and Enterprise.
- With **WPA Enterprise,** as with WPA enterprise, no pre-shared key is used, and it also **includes a MIC** (Message Integrity Check). The MIC mainly guards against the bit flipping attacks identified within WEP.
- Mobile phone networks/GSM typically uses **the A5/1 or A5/2 stream encryption method**, but almost on its first day of operation it has been a target for crackers, and the source code to crack A5/2 was released within one month of being made public.
- The **A5/3** encryption system – known as KASUMI – the Japanese word for “mist” – is the upgrade to A5/1 and uses a block cipher. A5/1 is designed to be used for the GSM network, whereas **A5/3 is for 3GPP**, and is based on the MISTY1 cipher. 128-bit key
- While WEP contains many weaknesses, a properly defined **stream cipher can be much faster than block ciphers**, as they just have to create a key stream from an IV (also known as a nonce value) and a key. Google proposed **ChaCha20** – named as it has 20 rounds – as an alternative to AES to be used with TLS connections.

Ch 12 Wireless Recap, cont.

	WEP	WPA	WPA2
✓ **Encryption Method**	RC4	TKIP + RC4	AES-CCMP
✓ **Key Size**	40-bit	128-bit	128-bit
✓ **Cipher Type**	Symmetric Stream	Symmetric Stream	Symmetric Block
✓ **IV Size**	24-bit	48-bit	48-bit

Check Your Knowledge: Chapter 12 Knowledge Check

2. Algorithms (Conventional & Light-weight)

Audio: Ride_n_Review_Algorithms Drill.mp4

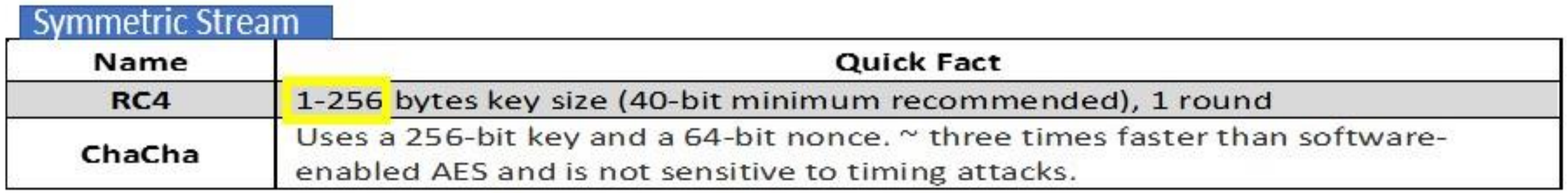
Check Your Knowledge: Algorithms Knowledge Check ***-and-*** Algorithms Drill (use in flashcard mode)

Conventional Cryptosystems Recap

Symmetric Block			
Name	Block Size (In Bits)	Key Size (In Bits)	Rounds
DES	64	56	16
3DES	64	112	48
AES	128	128, 192, OR 256	10, 12, or 14
IDEA	64	128	>17
Skipjack	64	80	32
Blowfish	64	32-448 (common = 128, 192, or 256)	16
Twofish	128	1-256 (common = 128, 192, or 256)	16
Camellia	128	128, 192, OR 256	18 or 24
RC5	32, 64, or 128	0-2048	0-255
RC2	64	1-128 (suggested minimum = 40)	18
RC6	Variable (common = 128)	Variable (common = 128, 192 or 256)	20
XTEA	64	128	Variable (64 suggested)

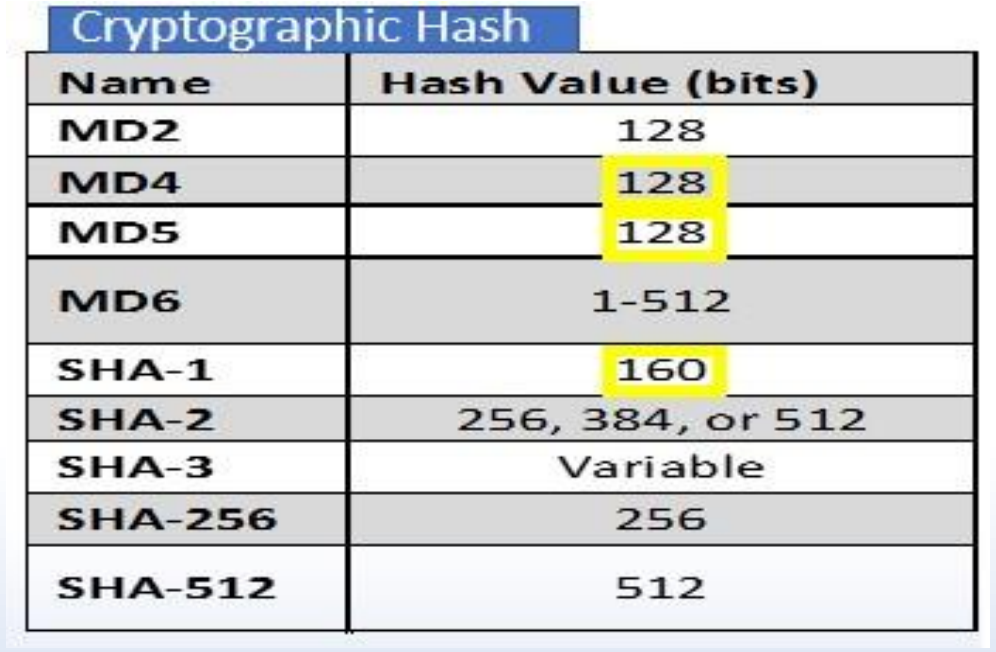
Highlighted figures in the chart are those referenced in the textbook for his course and should be considered testable!

Conventional Cryptosystems Recap, cont.



Symmetric Stream

Name	Quick Fact
RC4	1-256 bytes key size (40-bit minimum recommended), 1 round
ChaCha	Uses a 256-bit key and a 64-bit nonce. ~ three times faster than software-enabled AES and is not sensitive to timing attacks.



Cryptographic Hash

Name	Hash Value (bits)
MD2	128
MD4	128
MD5	128
MD6	1-512
SHA-1	160
SHA-2	256, 384, or 512
SHA-3	Variable
SHA-256	256
SHA-512	512

Highlighted figures in the chart are those referenced in the textbook for his course and should be considered testable!

Conventional Cryptosystems Recap, cont.

Asymmetric	
Name	Quick Fact
RSA	Partially homomorphic crypto system that leverages prime number characteristics, 1024-4096 bits variable key size, 1 round.
ECC	Improved solution over RSA often used with key exchange methods as well as with DSA in creating digital signatures.
EL Gamal	Used in both encryption and digital signing.
DSA	Federal Information Processing Standard (FIPS 186) for digital signatures, based on the mathematical concept of modular exponentiation and the discrete logarithm problem.

Key Exchange	
Name	Quick Fact
Diffie-Hellman	Provides a method for key exchange using a one-way function.

Light-weight Cryptosystems Recap

Light-weight Symmetric Block				
Name	**Block Size (In Bits)**	**Key Size (In Bits)**	**Rounds**	**Key attributes for light-weight suitability**
PRESENT	64	80 or 128	32	Relatively small key and block sizes. Uses an SPN (substitution permutation network) method. **One of the 1st considered as an AES replacement for use in light-weight implementations.**
XTEA	64	128	Variable (64 recommended)	Relatively small key and block sizes and variable rounds setting. Operates with just a just a few lines of code. **Fast speed**.
RC5	32, 64, or 128	0-2048	Variable 0-255	Variable block size, key size, and rounds. Can be optimized to IoT devices. **Conventional method suitable for light-weight implementations**.
SIMON	32, 48, 64, 96, 128	64, 72, 96, 128, 144, 192 or 256	Variable (32 or 44 common)	Variable block sizes key sizes, and rounds. **Optimized for hardware implementations**.
SPECK	32, 48, 64, 96 or 128	Variable	Variable (32 or 44 common)	Variable block sizes key sizes, and rounds. **Optimized for software implementations**.
CLEFIA	128	128, 192 and 256	18, 22, OR 26	Variable key size and rounds.

Highlighted figures in the chart are those referenced in the textbook for his course and should be considered testable!

Light-weight Cryptosystems Recap, cont.

Light-weight Symmetric Stream		
Name	**Key Size (In Bits)**	**Initialization Vector (IV) Size (In Bits)**
Rabbit	128	64
Mickey v2	80	Variable up to 80
Trivium	80	80
Grain	80	64
Enocoro	128	64

Light-weight Hashing		
Name	**Hash Value (In Bits)**	**Key attributes for light-weight suitability**
PHOTON	80, 128, 160, 224 or 256-bit	Small memory footprint and have a target an input of just 256 characters (whereas typical hash functions support up to 264 bits). Different size hash values available.
SPONGENT	88, 128, 160, 224 or 256	Small memory footprint and have a target an input of just 256 characters (whereas typical hash functions support up to 264 bits). Different size hash values available.
Lesamnta-LW	256	Small memory footprint and have a target an input of just 256 characters (whereas typical hash functions support up to 264 bits). **Fast (five times faster than SHA-256).** Primary target implementation is 8-bit CPUs for **short message hashing**.
Quark	64 or 112	Small memory footprint and have a target an input of just 256 characters (whereas typical hash functions support up to 264 bits). **Can be used for hashing and in stream encryption**.

Highlighted figures in the chart are those referenced in the text for his course and should be considered testable!

Light-weight Cryptosystems Recap, cont.

Light-weight Signing		
Name	**Key Size (In Bits)**	**Description**
Chaskey	128	Light-weight cryptography method for signing messages (MAC). Relatively undemanding hardware implementation (only ~3,333 gates required at 1MHz clock rate) making it suitable for IoT implementation.

Light-weight Public-Key	
Name	**Description**
Elli	Light-Weight public key solution. Short for "Elliptic Light". Uses Elliptic Curves along with a Diffie-Hellman related handshake between the RFID tag and the RFID reader in **RFID implementations**.

Shawn's C839v5/D334 Algorithms mnemonics

Symmetric Block Cipher List:

"**3 d**ogs **a**nd **two fish** had an **idea** to **r**ow **256** miles by sea,
but instead **skip**ped over to **Camelia**'s e**x**-husband's house for **tea**."

3 - Three (3DES)
D - Dogs (DES)
A - and (AES)
T - two fish had an (Twofish)
I - idea to (IDEA)
R - row 256 miles (RC2/RC5/RC6)
B - but instead (Blowfish)
S - skipped over to (Skipjack)
C - Camellia's (Camelia)
X - ex-husband's house for tea. (XTEA)

64-bit Block size ciphers (non-variable):

"**3** *block*-headed **d**ogs **r**uined **b**oth **x**-rays **i**n **s**urgery in *64* seconds!"

3 Three (3DES) + block for "block cipher"
D dogs (DES)
R ruined both (as in 2) (RC2)
B both (Blowfish)
X x-rays (XTEA)
I in (IDEA)
S surgery (Skipjack) + 64 for "64 bit"

128-bit Symmetric Block Cipher Keys (non-variable):

"Both ***EA***'s have ***128's"*** (say *1-2-a's*) | IDEA key size is 128 bits, XTEA key size is 128 bits

128-bit block sizes (non-variable):

"Camelia's 128-year-old CAT":

C - **C**amelia
A - **A**ES
T - **T**wofish

Symmetric Stream Cipher List

"**R**ay and **C**indy danced **4** hours doing the **Cha Cha** dance on live ***stream***."

Ray and Cindy 4 (RC4)
Cha Cha (ChaCha)

DES vs 3DES:

"DES vs 3DES is not quit how it sounds, 3DES' keys are only *double*, but does have *3 times* the rounds."

DES - 56 bit key, 3DES - 112 bit key (**double 56**) |DES - 16 rounds, 3DES - 48 rounds (**triple 16**)

Asymmetric List:

"Asymmetric Antlers on a *deer."*

D - **D**SA & **D**iffie-Hellman (*Key exchange using asymmetric method*)
E - **E**CC
E - **El** Gamal
R – **R**SA

3. Final Preparation

- ❑ Be sure to view the 4 videos located on the Learning Resources page of your course Material. Be sure to review the course Textbook as needed.
- ❑ Now work to master…
  - ✓ Your Course Preassesment
  - ✓ Shawn's Testmoz Knowledge Checks: Shawn's Testmoz Quiz Bank
  - ✓ Shawn's Algorithms Drill Quizlet Deck: Algorithms Drill (use in flashcard mode)
  - ✓ Shawn's Additional Practice Quizlet Deck Shawn's C839v5/D334 Quizlet Flashcards (use in flashcard mode)

Audio/Video List

Audio: Chapter1.mp3

Video: Shawn's Shorts_MOD and Binary Math.mp4

Audio: Chapter2.mp3***-and-*** Ride_n_Review_Symmetric_Asymmetric Basics.mp4

Video: Shawn's Shorts_Block Cipher Modes.mp4

Audio: Chapter3.mp3***-and-*** Ride_n_Review_Fundamentals_Hashing Basics.mp4

Audio: Chapter4.mp3 ***-and-*** Ride_n_Review_Symmetric_Asymmetric Basics.mp4

Audio: Shawn's Shorts_Key Pairs.mp4

Audio: Chapter5.mp3

Video: Shawn's Shorts Symmetric Key Exchange.mp4

Video: Chapter6_Vid

Audio: Chapter7.mp3

Audio: Chapter8.mp3

Audio: Chapter9.mp3***-and-*** Ride_n_Reveiw_Lightweight Cryptography.mp4

Videos: Chapter10_Vid

Audio: Chapter12.mp3

Audio: Ride_n_Review_Algorithms Drill.mp4