



CENG781 Network Security

E. Onur ©2012-∞

Wireless Systems, Networks and Cybersecurity Laboratory
Department of Computer Engineering
Middle East Technical University
Ankara Turkey

March 1, 2022

Lecture 1: Overview of Network Security

Overview of Network Security

Outline

Computer Security
Network Security
Summary

Objectives of the Lecture

At the end of this lecture, you will be able to

- define security attacks, mechanisms and services
- define the network security model
- define CIA triad

Cryptographic Algorithms and Protocols

Four Main Areas

- Symmetric encryption
- Asymmetric encryption
- Data integrity algorithms
- Authentication protocols

Computer and Network Security

Definition of Computer Security

The **protection** afforded to an automated information system in order to attain the applicable objectives of preserving the **integrity, availability** and **confidentiality** of information system resources (includes hardware, software, firmware, information/data, and telecommunications)

Definition of Network and the Internet Security

Measures to **deter, prevent, detect, and correct** security violations that involve the **transmission** of information.

NIST Computer Security Handbook

Objectives of Computer Security

CIA Triad

- Confidentiality
 - Data confidentiality: concealing confidential information
 - Privacy: controlling disclosure of information
- Integrity
- Availability

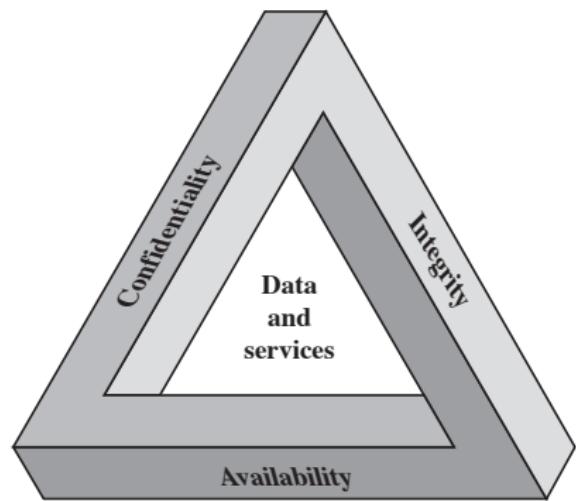


Figure 1.1 The Security Requirements Triad

Some Definitions

Loss of confidentiality

unauthorized disclosure of information

Loss of Integrity

unauthorized modification or destruction of information

Loss of availability

disruption of access to or use of information (or system)

No authenticity

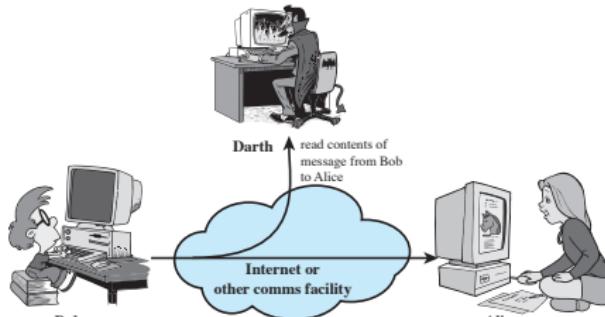
not being genuine, not verified, not trusted, or no confidence in the validity of a transmission (message and/or originator)

Network Security

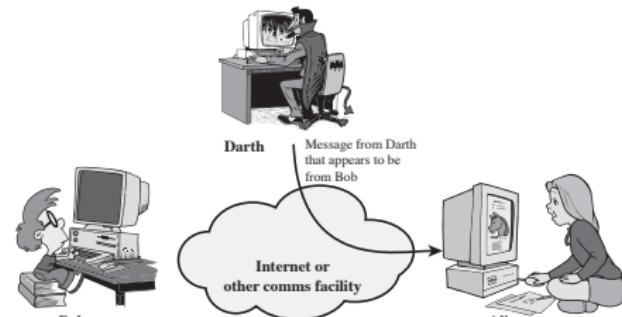
ITU-T X.800 Security Architecture for OSI

- a systematic framework
- **Security attacks:** passive or active
 - Threat: a potential violation of security
 - Attack: an assault on security using a threat
- **Security mechanism:** any process to detect, prevent or recover from attacks
- **Security services:** a service provided by a protocol layer of communicating open systems, which ensures adequate security of the systems or of data transfers

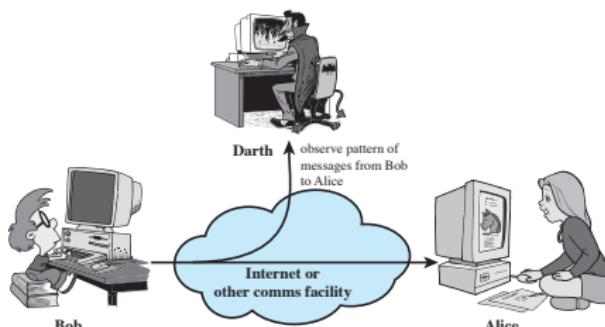
Security Attacks



(a) Release of message contents

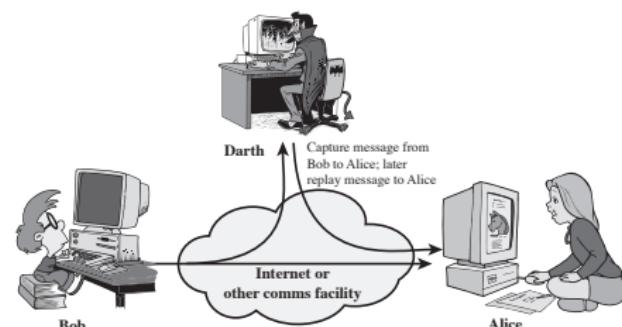


(a) Masquerade



(b) Traffic analysis

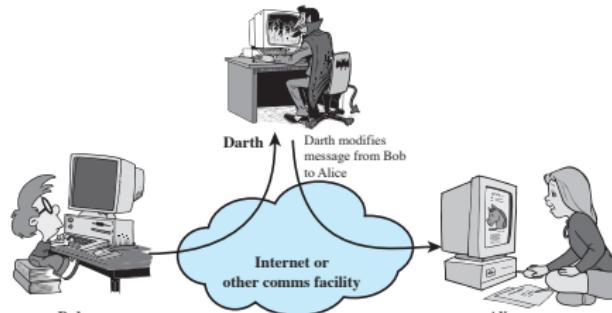
Figure 1.2 Passive attacks.



(b) Replay

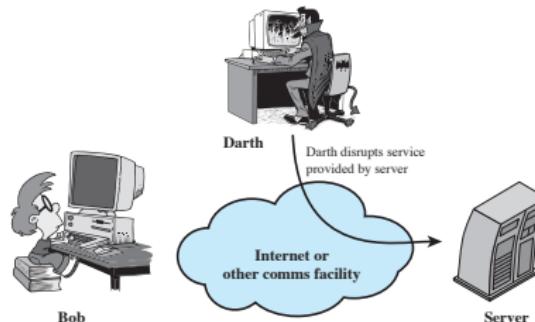
Figure 1.3 Active attacks (page 1 of 2)

Security Attacks



(c) Modification of messages

- Release of message contents
- Traffic analysis
- Masquerade
- Replay
- Modification of messages
- Denial of service
- ...



(d) Denial of service

Figure 1.3 Active Attacks (page 2 of 2)

Security Services

X.800 Security Services

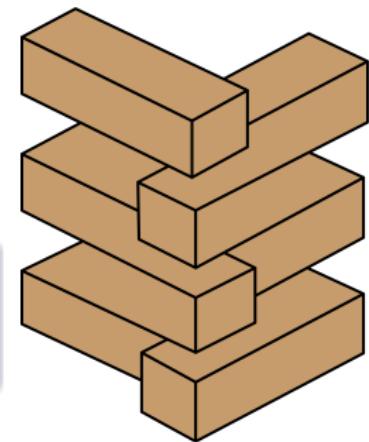
- **Authentication** assurance that communicating entity is the one claimed
- **Access Control** prevention of the unauthorized use of a resource
- **Data Confidentiality** protection of data from unauthorized disclosure
- **Data Integrity** assurance that data received is as sent by an authorized entity
- **Non-Repudiation** protection against denial by one of the parties in a communication
- **Availability** resource accessible/usable

Practice What You Preach

Using 3 levels (low, medium, high) give examples for X.800 security services: authentication, access control, confidentiality, non-repudiation and availability.

Example

Student grade information is an asset whose **confidentiality** is deemed **high**.



Security Mechanisms

Specific Security Mechanisms

encipherment, digital signatures, access controls, data integrity, authentication exchange, traffic padding, routing control, notarization

Pervasive Security Mechanisms

trusted functionality, security labels, event detection, security audit trails, security recovery

Security Services and Mechanisms

Service	Encipher- ment	Digital signature	Access control	Data integrity	Authenti- cation exchange	Traffic padding	Routing control	Notari- zation
Peer entity authentication	Y	Y			Y			
Data origin authentication	Y	Y						
Access control			Y					
Confidentiality	Y						Y	
Traffic flow confidentiality	Y					Y	Y	
Data integrity	Y	Y		Y				
Nonrepudiation		Y		Y				Y
Availability				Y	Y			

Security Services and Attacks

	release of message contents	masquerade	replay	modification of messages	DoS
peer auth.		Y			
data origin auth.		Y			
access control		Y			
confidentiality	Y				
data integrity			Y	Y	
non-repudiation		Y			
availability					Y

Network Security Model

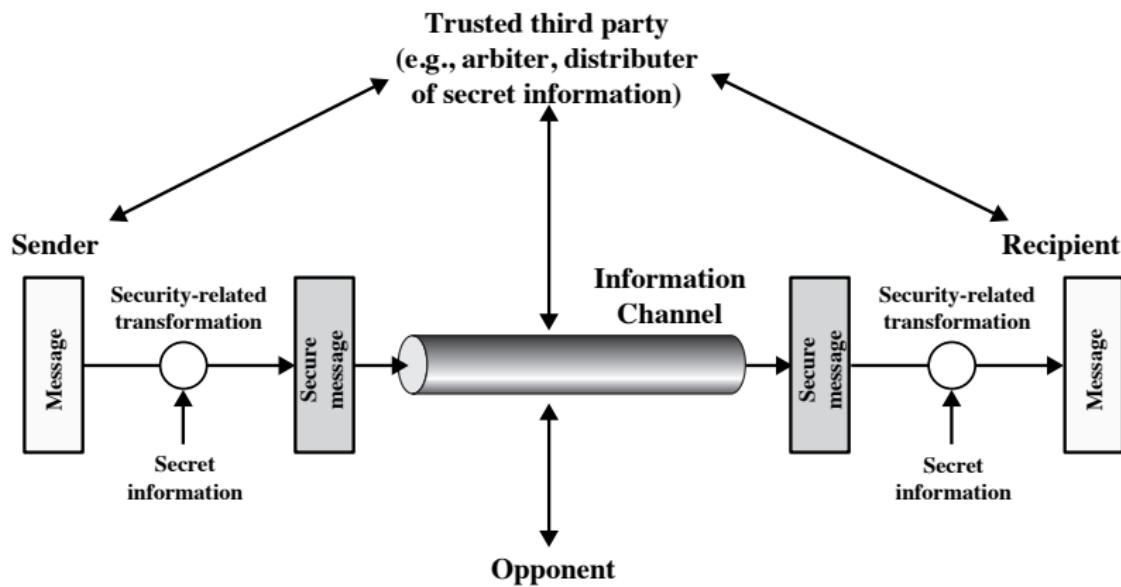


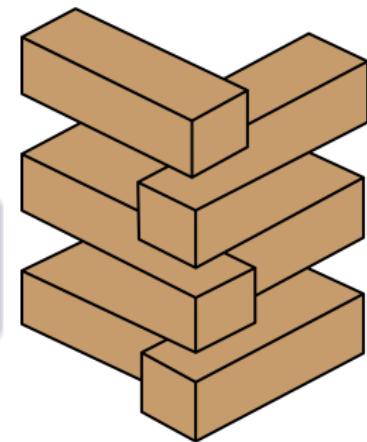
Figure 1.4 Model for Network Security

Practice What You Preach

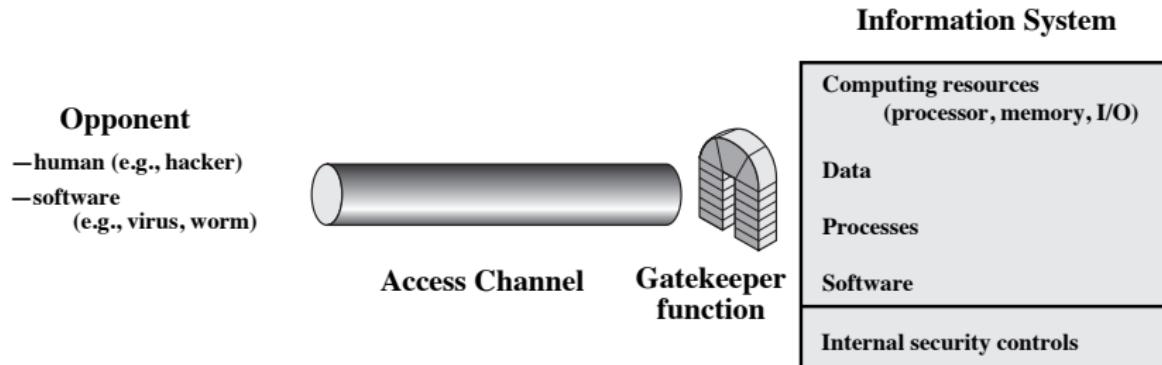
Based on network security model, what are the basic tasks?

Hint

One of the tasks is the distribution of secret information.



Network Access Security Model



Summary

Today, we learned

- CIA Triad
- ISO/OSI X.800 definitions

Read Chapter 1 of the textbook

Lecture 2: Shared-key Encryption Schemes

Shared-key Encryption Schemes

Outline

Introduction to Shared-key Systems
Provable Security
Summary

Objectives of the Lecture

At the end of this lecture, you will be able to

- describe the symmetric (shared-key) encryption schemes

Definitions

Plaintext

Any input message, $m \in \{0, 1\}^*$

Ciphertext

Output of a cipher, $c \in \{0, 1\}^*$

Key

Piece of information only known to principals, $k \in \{0, 1\}^n$ (n -bit)

Shared-key Encryption Schemes

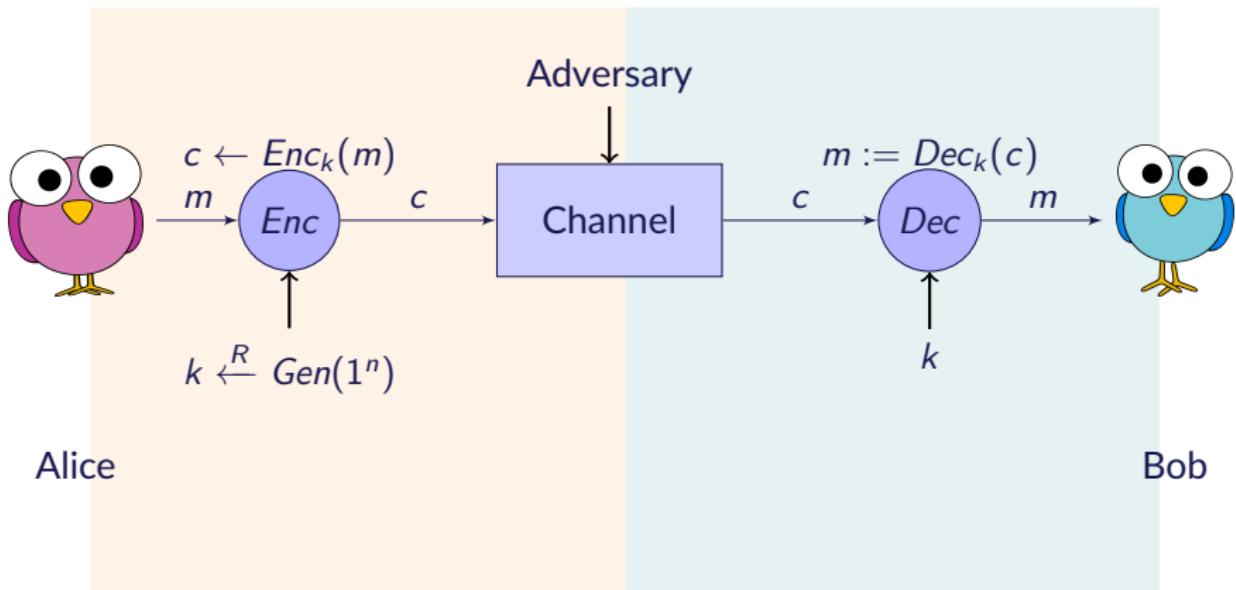


Figure: A simple model for network security employing shared-key encryption.

The Syntax of Shared-key Encryption Scheme

Key generation algorithm, $k \xleftarrow{R} \text{Gen}(1^n)$

probabilistically outputs a key $k \in \mathcal{K}$ according to a distribution determined by the encryption scheme. The security parameter is represented as 1^n (a sequence of n 1's).

Encryption algorithm, $c \leftarrow \text{Enc}_k(m)$

outputs the ciphertext $c \in \mathcal{C}$ by inputting the key k and plaintext m .

Decryption algorithm, $m := \text{Dec}_k(c)$

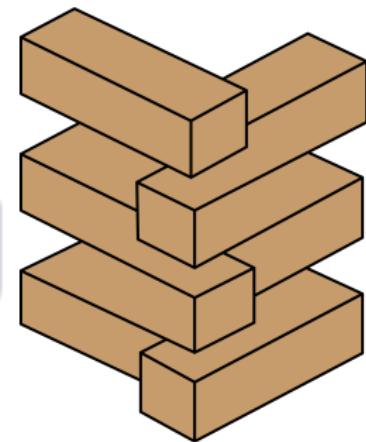
outputs the plaintext $m \in \mathcal{M}$ by inputting the key k and ciphertext c .

Practice What You Preach

POLL: Should the algorithms be secret or public?

Hint

Buy out the designer of the algorithm.



Should the Algorithms Be Secret?

Kerckhoffs' Principles

Only the key should be secret. Gen , Enc , Dec can be public.

The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.

Advantages of Following Kerckhoffs' Principles

- Public scrutiny
- Ethical hackers reveal flaws
- Not subject to reverse engineering
- Standardization

Requirements

For secure use of shared-key schemes

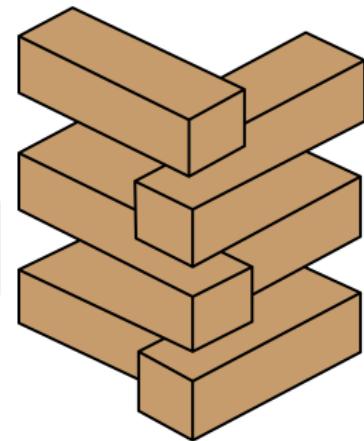
- **Strength requirement:** A strong Enc (random looking ciphertext)
- **Correctness requirement:** $Dec_k(Enc_k(m)) = m$
- **Key distribution requirement:** A secret key known to sender and receiver
- **Sufficient key space principle:** Any secure encryption scheme must have a key space that is not vulnerable to exhaustive search

Practice What You Preach

What does it mean to be strong or secure?

Brainstorming

Security can be defined in different ways.



Principles of Modern Cryptography

- Rigorous and precise definition of security including
 - What is objective and task ?
 - What is a break?
 - What is the power of adversary?
- Assumptions must be clearly defined and be minimal
- Constructions accompanied with thorough proofs

Definition of security

A cryptographic scheme **for a given task** is secure if no **adversary of a specified power** can achieve a **specific break** (katz2007introduction).

Reductionist Approach to Proof of Security

Given that assumption X is true, encryption scheme Y is secure according to the given definition.

Proof of the security of Y

Show how any adversary breaking encryption scheme Y can be used as a tool to violate assumption X .

Example: shared-key

- Shared-key scheme Y generates a uniform random ciphertext
- Prove that the output is not uniform random asymptotically

Example: public-key

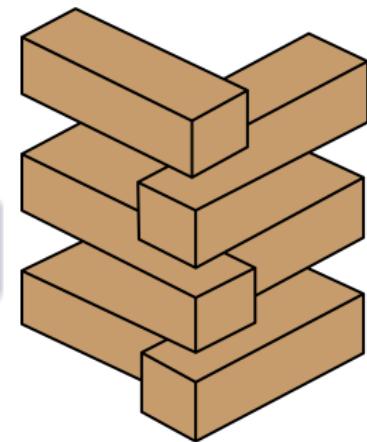
- Public-key system Y relies on the assumption that $P \neq NP$
- Prove that breaking Y requires solving NP-hard problem in polynomial-time which is not possible ($P \neq NP$ not-proven yet).

Practice What You Preach

What is the definition of the adversary's power?

Clarification

Other than computational power of course.



Cryptanalytic Attack Scenarios

Ciphertext-only attack

Passive. Basic type. Adversary only observes ciphertext.

Aim: finding the corresponding plaintext to the ciphertext.

Known-plaintext attack

Passive. Adversary learns several ciphertext/plaintext pairs.

Aim: finding the plaintext of some other ciphertext.

Chosen-plaintext attack

Active. Adversary learns encryptions of his choice of plaintexts.

Aim: finding the plaintext of some other ciphertext.

Chosen-ciphertext attack

Active. Adversary learns decryptions of his choice of ciphertexts.

Aim: finding the plaintext of some other ciphertext.

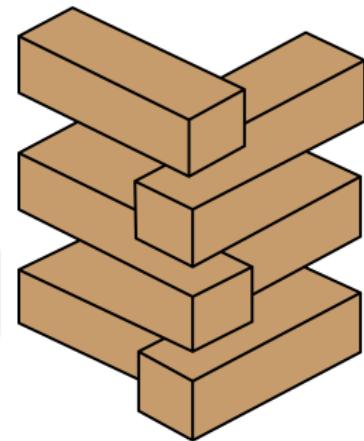
Practice What You Preach

We said encryption schemes rely on assumptions (conditional security).

POLL: Is there unconditional security?

Hint

Rest in Peace Shannon.



Unconditional versus Computational Security

Unconditional security (perfect secrecy)

No matter how much computer power or time is available, the cipher cannot be broken since the ciphertext provides insufficient information to uniquely determine the corresponding plaintext

Computational security

given limited computing resources (eg time needed for calculations is greater than age of universe), the cipher cannot be broken

Basic Blocks of a Cryptographic System

Diffusion and Confusion

Diffusion

The statistical structure of m is dissipated (one bit of m changes, almost all bits in c changes).

Confusion

Make the relationship between m and c complex (non-linear relationship)

Substitution and permutation boxes (and several rounds) satisfy confusion and diffusion properties.

Summary

Today, we learned

- How to define security
- Symmetric (shared-key) encryption schemes

Lecture 3: Shared-key Encryption Schemes: Stream Ciphers

Shared-key Encryption Schemes: Stream Ciphers

Objectives of the Lecture

At the end of this lecture, you will be able to

- Describe what perfect secrecy is
- Analyze how we can get close to perfect secrecy
- Discuss pseudo-random number generation
- Devise a cheating attack on multiple-choice tests

Learning Outcomes

At the end of this lecture, you will be able to

- Explain the concepts of randomness and unpredictability with respect to random numbers.
- Understand the differences among true random number generators, pseudorandom number generators, and pseudorandom functions.
- Present an overview of requirements for pseudorandom number generators.
- Explain how a block cipher can be used to construct a pseudorandom number generator.
- Present an overview of stream ciphers and RC4.
- Explain the significance of skew.

Stream Cipher Definition

Symmetric Cipher

is a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ (keyspace, plaintext space and ciphertext space) is a pair of efficient algorithms (E, D) where

$$E : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$$

and

$$D : \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$$

subject to

$$D_k(E_k(m)) = m$$

$$\forall m \in \mathcal{M}, \forall k \in \mathcal{K}$$

E is randomized and D is deterministic.

Vernam Cipher 1917

- Initial example of a *secure* cipher
- $\mathcal{M} = \mathcal{C} = \{0, 1\}^n$ (plaintext and ciphertext spaces)
- $\mathcal{K} = \{0, 1\}^n$ (keyspace)
- we need keys (at least) as long as the message

Encryption

$$c = \text{Enc}_k(m) = k \oplus m \text{ (bitwise)}$$

Decryption

$$m = \text{Dec}_k(c) = k \oplus c \text{ (bitwise)}$$

Correctness Proof of Vernam Cipher

$$Dec_k(Enc_k(m)) \stackrel{?}{=} m$$

Proof

$$Dec_k(Enc_k(m)) = Dec_k(k \oplus m) = k \oplus (k \oplus m) = (k \oplus k) \oplus m = 0 \oplus m = m$$

Example

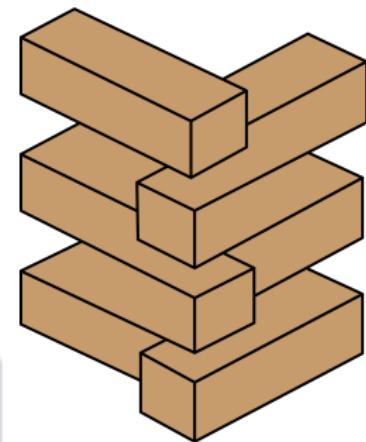
key k	10010110	Alice: Encryption
message m	11111111	uniform randomly selected
ciphertext c	01101001	plaintext to be encrypted
		$c = k \oplus m$
key k	10010110	Bob: Decryption
ciphertext c	01101001	same key used for encryption
message m	11111111	received by Bob
		$m = k \oplus c$

Practice What You Preach

1. Given m and c , can you find key k ? (e.g., known-plaintext attack)
2. Can we encrypt c with the same k again? What do we get?
3. What happens if k starts repeating for long messages?
4. If we are operating in decimal system, what is the decryption operation?

Hint

Is exclusive or (binary addition) \oplus commutative?



Properties of One-time Pad (Vernam Cipher)

- Unconditionally secure (perfect secrecy) against cipher-text only attack
- Perfect secrecy requires $|\mathcal{K}| \geq |\mathcal{M}|$ (key must be at least as long as the message)

Information theoretic security: perfect secrecy

A cipher (Enc, Dec) defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ has **perfect secrecy** if

$\forall m_0, m_1 \in \mathcal{M}$ when $Prob\{Enc_k(m_0) = c\} = Prob\{Enc_k(m_1) = c\}$ where $|m_0| = |m_1|$ and k is derived **uniform randomly** over \mathcal{K} (shannon1949communication).

- Given c , adversary cannot tell whether m_0 or m_1
- Even the most powerful adversary cannot learn anything about plaintext from ciphertext (assuming ciphertext-only attack)

Practice What You Preach

Prove that one-time pad provides perfect secrecy.

Hint

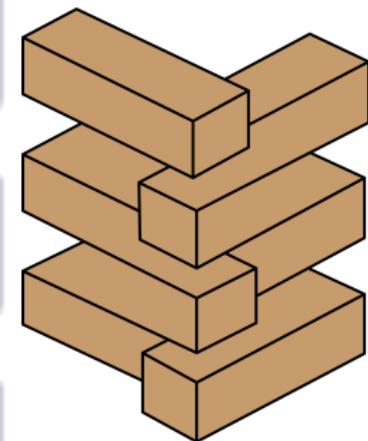
How many keys map m to c and what is the meaning of uniform random k .

Hint

Should probability mass function of c look like uniform random?

Hint

$\forall m \in \mathcal{M}$ and $c \in \mathcal{C}$, $\text{Prob}\{\text{Enc}_k(m) = c\}$ is the number of keys $k \in \mathcal{K}$ such that $\text{Enc}_k(m) = c$ divided by $|\mathcal{K}|$.



Layman's Proof

- m is one-bit and has an arbitrary distribution (e.g., $\text{Prob}\{m = 0\} = 0.1$)
- k is one-bit and uniform random

Proof

m	$\text{Prob}\{m\}$	k	$\text{Prob}\{k\}$	$m \oplus k = c$	$\text{Prob}\{c\}$
0	0.1	0	$\frac{1}{2}$	$0 \oplus 0 = 0$	$0.1 \times \frac{1}{2} = \frac{1}{20}$
0	0.1	1	$\frac{1}{2}$	$0 \oplus 1 = 1$	$0.1 \times \frac{1}{2} = \frac{1}{20}$
1	0.9	0	$\frac{1}{2}$	$1 \oplus 0 = 1$	$0.9 \times \frac{1}{2} = \frac{9}{20}$
1	0.9	1	$\frac{1}{2}$	$1 \oplus 1 = 0$	$0.9 \times \frac{1}{2} = \frac{9}{20}$

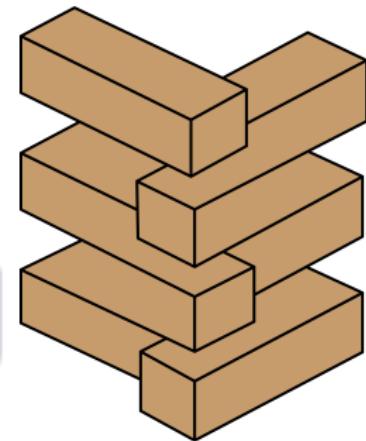
- $\text{Prob}\{c = 0\} = \text{Prob}\{m = 0, k = 0\} + \text{Prob}\{m = 1, k = 1\} = \frac{1}{20} + \frac{9}{20} = \frac{1}{2}$
(First and fourth rows)
- c is also **uniform random**. Each output is **equally likely**.

Practice What You Preach

A multiple-choice exam has 10 questions; each having 5 choices where one choice is the correct answer. The lecturer colludes with a student and they cheat in the exam? How can they cheat in a **perfectly secret** way?

Hint

Vernam cipher



Problems of One-time Pad

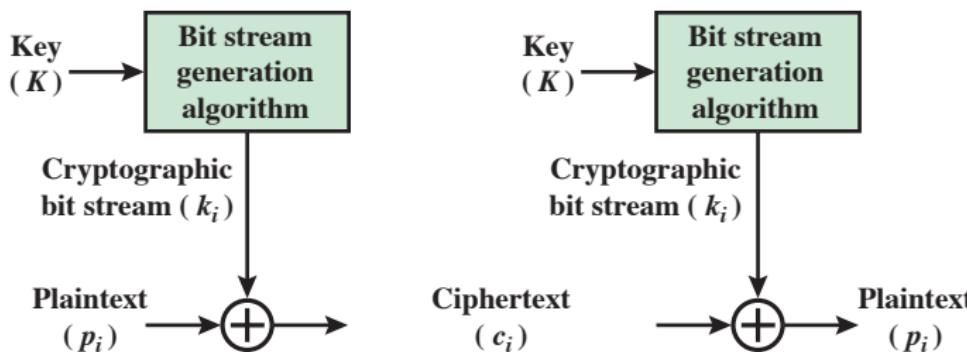
- $|\mathcal{K}| \geq |\mathcal{M}|$ long keys
- another secure channel to distribute the long secret key
- subject to known-plaintext attack

Stream Ciphers

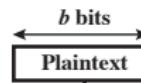
Let's assume that we designed an algorithm which produces the same uniform random output given some input. Alice and Bob employs the same algorithm. This algorithm is basically a pseudo-random number generator.

Stream Cipher

Use algorithmic bit stream generator or pseudo-random number generators (PRNG)



(a) Stream Cipher Using Algorithmic Bit Stream Generator



Examples of Real-life Stream Ciphers

- GSM uses **A5**
- Bluetooth uses **E0**
- WiFi WEP (Wired Equivalent Privacy) uses **RC4**
- WiFi WPA (Wi-Fi Protected Access) uses RC4
- Secure socket layer uses RC4
- Skype uses modified version of RC4

Prerequisites for Pseudorandom Numbers

Randomness

- **Uniform Distribution** : The distribution of bits in the sequence should be uniform; that is, the frequency of occurrence of ones and zeros should be approximately equal. (rukhin2001nist)

Unpredictability

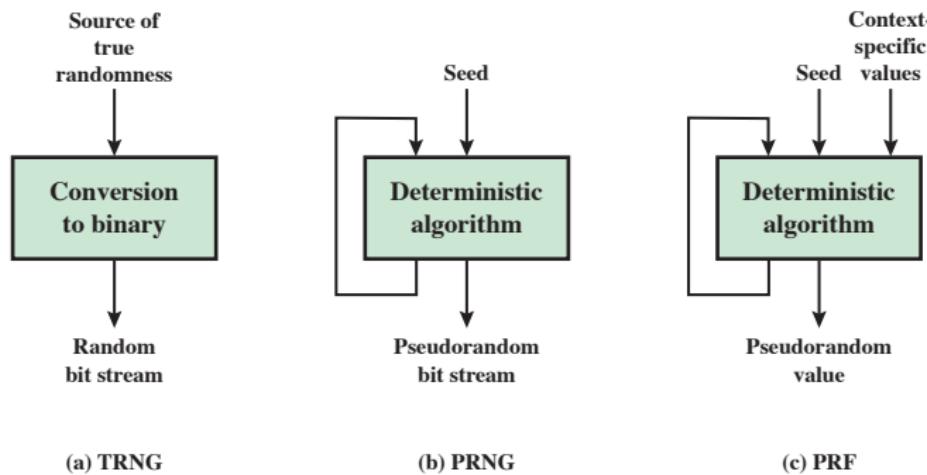
- **Independence**: No one subsequence in the sequence can be inferred from the others.
- With "true" random sequences, each number is statistically independent of other numbers in the sequence and therefore unpredictable

Pseudorandom Numbers

Definition

Cryptographic applications typically make use of **algorithmic techniques** for random number generation. These algorithms are **deterministic** and therefore produce **sequences of numbers that are not statistically random**. However, if the algorithm is good, the resulting sequences will **pass many tests of randomness**. Such numbers are referred to as **pseudorandom numbers**.

Pseudo-random Number Generators (PRNG)



When a PRNG (generator) or PRF (function) is used for a cryptographic application, then **the basic requirement** is that an adversary who does not know the seed is unable to determine the pseudorandom string.

Generator or Function

Pseudorandom number generator (PRNG)

An algorithm that produces an open-ended sequence of bits (e.g., stream ciphers)

Pseudorandom number function (PRF)

An algorithm that produces a fixed-length pseudorandom number (e.g., nonces). Typically, PRF take a seed and a context-specific input).

Both PRNG and PRF has to generate random and unpredictable output

PRNG/PRF Requirements

- **Randomness:**
 - **Uniformity:** $Prob\{0\} = Prob\{1\} = \frac{1}{2}$
 - **Scalability:** Any test should be applicable to any sub-sequence
 - **Consistency:** Should not violate randomness and scalability when different seeds are used.
- **Unpredictability:** no correlation with seed or among bits
 - **forward unpredictability:** seed unknown, next bit cannot be predicted using history
 - **backward unpredictability:** given a sequence, seed should not be determined.

PRNG/PRF Statistical Tests

- **Frequency test:** number of 1's is (almost) equal to number of 0's
- **Runs test:** bounded number of a sequence of the same bit value
- **Maurer's test:** checks compressibility

NIST SP 800-22 defines a suite of statistical tests for PRNG.

PRNG Classification

Purpose-built algorithms (e.g., Linear congruential, BBS, or RC4)

Using other cryptographic algorithms (e.g., block ciphers, hash functions)

Linear Congruential Generator

- The sequence of random numbers $\{X_n\}$ iteratively where $X_{n+1} = (g_a X_n + g_c) \bmod g_m$
- When $g_m, g_a, g_c, X_0 \in \mathbb{Z}$ then $\{X_n\} \in \mathbb{Z}^{g_m}$

Example

$g_a = 1, g_c = 1, g_m = 32, X_0 = 1$ then outputs
 $\{X_0 = 1, X_1 = 2, X_2 = 3, X_3 = 4, X_4 = 5, X_5 = 6, \dots\}$
Looks like random? Predictable?

Example

$g_a = 7, g_c = 0, g_m = 32, X_0 = 1$ then outputs
 $\{X_0 = 1, X_1 = 7, X_2 = 17, X_3 = 23, X_4 = 1, X_5 = 7, \dots\}$
Not predictable but repeats, period 4.

Requirements for PRNG

- Test 1: PRNG should be full period: before starts repeating itself should use all integers up to g_m
- Test 2: The generated sequence should look like as if it is random
- Test 3: The function should be efficiently implementable

Linear Congruential Generator

If values of the parameters are chosen carefully, the output looks like random. When parameters are known, subject to cryptanalysis.

Practice What You Preach

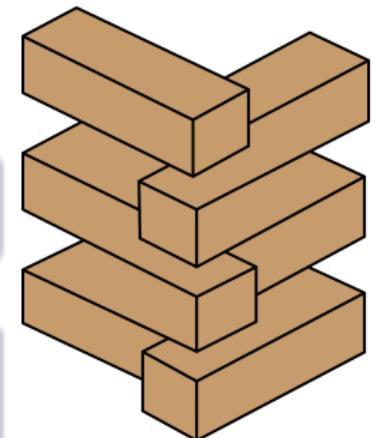
Eve wants to determine the parameters of the linear congruential generator. She has to eavesdrop a number of output values of $\{X_n\}$. How many does she require?

Hint

Linear algebra, $X_{n+1} = g_a X_n + g_c \pmod{g_m}$

Conclusion

Make the output sequence non-reproducible (use some sort of entropy source)



Blum Blum Shub: Cryptographically Secure PR-Bit

- Select primes p and q congruent to 3 modulo 4 ($p \equiv q \equiv 3 \pmod{4}$)
- Example: $p = 7$ and $q = 11$, when divided by 4, remainder is 3.
- Set $n = p \times q$
- Select s relatively prime to n (p, q not factor of s)
- $X_0 = s^2 \pmod{n}$
- $X_i = X_{i-1}^2 \pmod{n}$
- Output $B_i = X_i \pmod{2}$

The BBS is referred to as a **cryptographically secure pseudorandom bit generator** (CSPRNG) since it passes the **next-bit test**.

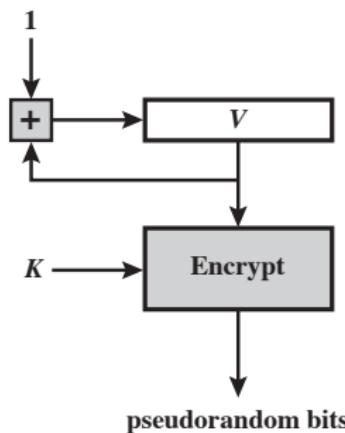
Blum Blum Shub (BBS) is CSPRNG

Next-bit Test for Unpredictability

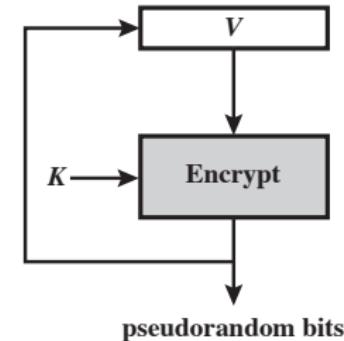
Given the first j bits of the sequence, an efficient adversary cannot predict the $(j + 1)^{\text{st}}$ bit with a probability better than $\frac{1}{2}$

- BBS passes this test and is unpredictable.
- Security of BBS relies on factoring n (p and q hidden, then **difficult**)
- Slow for stream ciphers
- Good for generating random secure seeds

Pseudo-random Number Using a Block Cipher



(a) CTR Mode



(b) OFB Mode

RC4

- Designed in 1987 by Ron Rivest
- Used almost everywhere
- Kept as a trade-secret initially
- September 1994, RC4 algorithms is anonymously posted on Cyberpunks list (Remember Kerckhoff principle).
- Simple, fast, can be implemented on hardware
- Period $> 10^{100}$

RC4

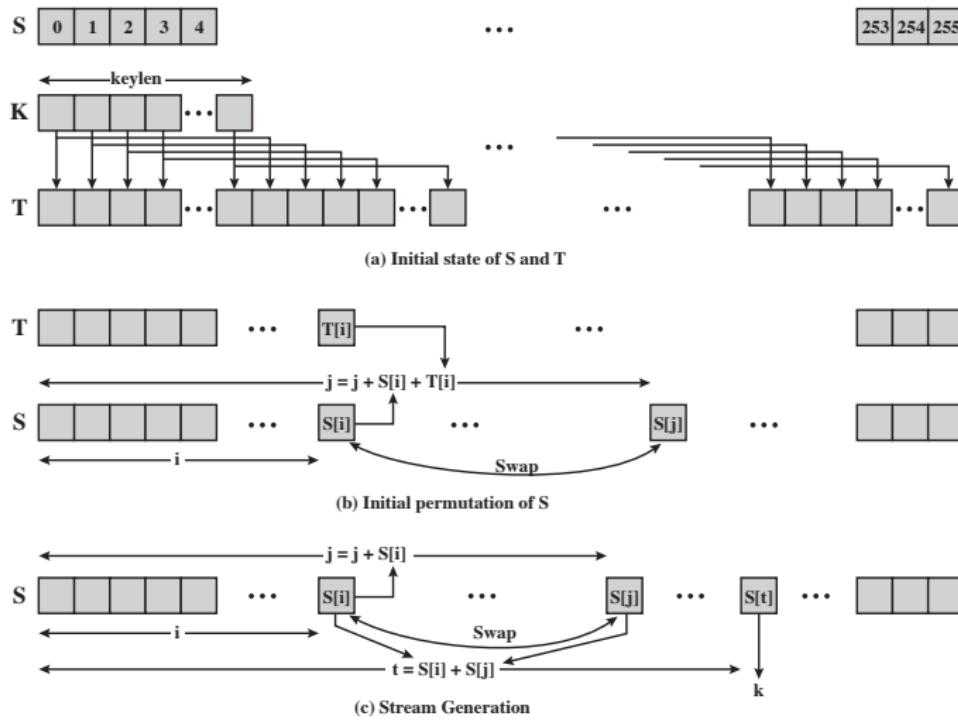


Figure 7.6 RC4

RC4 Initialization

- s : state vector of size 256
- k : key (seed) of length n
- t : temporary vector of size 256

Algorithm 1 RC4 Initialization

```
1: for  $i = 0$  to 255 do  
2:    $s_i = i$   
3:    $t_i = k_{(i \bmod n)}$   
4: end for
```

RC4 Initial Permutation

Algorithm 2 RC4 Initial Permutation

```
1:  $j = 0$ 
2: for  $i = 0$  to 255 do
3:    $j = j + s_i + t_i \bmod 256$ 
4: end for
5: swap  $s_i$  and  $s_j$ 
```

RC4 Stream Generation

Algorithm 3 RC4 Stream Generation

```
1:  $j = 0$   $i = 0$ 
2: while true do
3:    $i = i + 1 \bmod 256$ 
4:    $j = j + s_i \bmod 256$ 
5:   swap  $s_i$  and  $s_j$ 
6:    $t = s_i + s_j \bmod 256$ 
7:   output  $k = s_t$  (byte)
8: end while
```

RC4 Problems

- First byte of keystream is correlated with first 3 bytes of key
- Second byte of keystream is biased toward 0 with prob $\frac{1}{128}$
- First 3-4 bytes of keystream is not uniform random! Reveals information about key
- **aircrack-ptw** can break 104-bit keys in 40000 frames with 0.5 probability, or in 85000 frames with 0.95 probability ([cryptoprint:2007:120](#))

Summary

Today, we learned

- PRNG
- Stream ciphers

Lecture 4: Shared-key Encryption Schemes: Block Ciphers

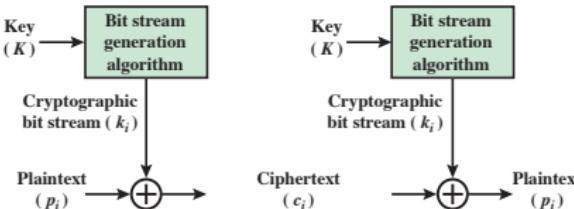
Shared-key Encryption Schemes: Block Ciphers

Objectives of the Lecture

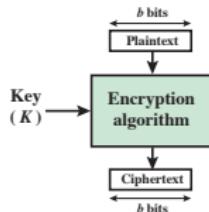
At the end of this lecture, you will be able to

- describe what a secure block cipher is.

Block versus Stream Cipher



(a) Stream Cipher Using Algorithmic Bit Stream Generator



(b) Block Cipher

A block cipher is an encryption/decryption scheme in which a block of plaintext is transformed to a ciphertext block of equal length.

Figure 3.1 Stream Cipher and Block Cipher

G. Vernam and C. Shannon

Vernam Cipher: <http://www.youtube.com/watch?v=8z1XClxqy1M> Claude Shannon: <http://www.youtube.com/watch?v=z7bVw7IMtUg>

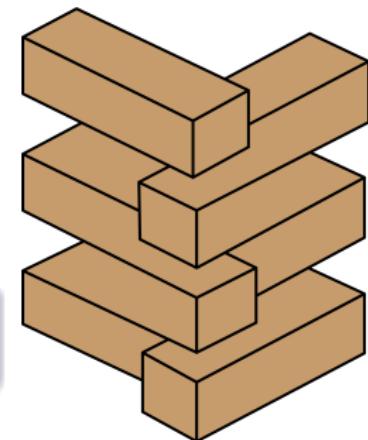
Practice What You Preach

A block cipher operates on a plaintext of n bits to produce a ciphertext of n bits. There are 2^n possible plaintext blocks. Does this mapping have to be reversible (non-singular)?

How many reversible mappings are there?

Hint

Do we have to decrypt?



Block versus Stream Cipher

- If the block size is n bits then for the first plaintext (sequence of n zeros), we have 2^n alternative mappings
- For the second plaintext (all zeros other than the least significant bit), there are $2^n - 1$ alternative mappings
- For the third plaintext, there are $2^n - 2$ and etc.
- Therefore, there are $2^n!$ different reversible mappings.

Reversible mapping	
Plaintext	Ciphertext
00	11
01	10
10	00
11	01

Irreversible mapping	
Plaintext	Ciphertext
00	11
01	10
10	01
11	01

Substitution Cipher

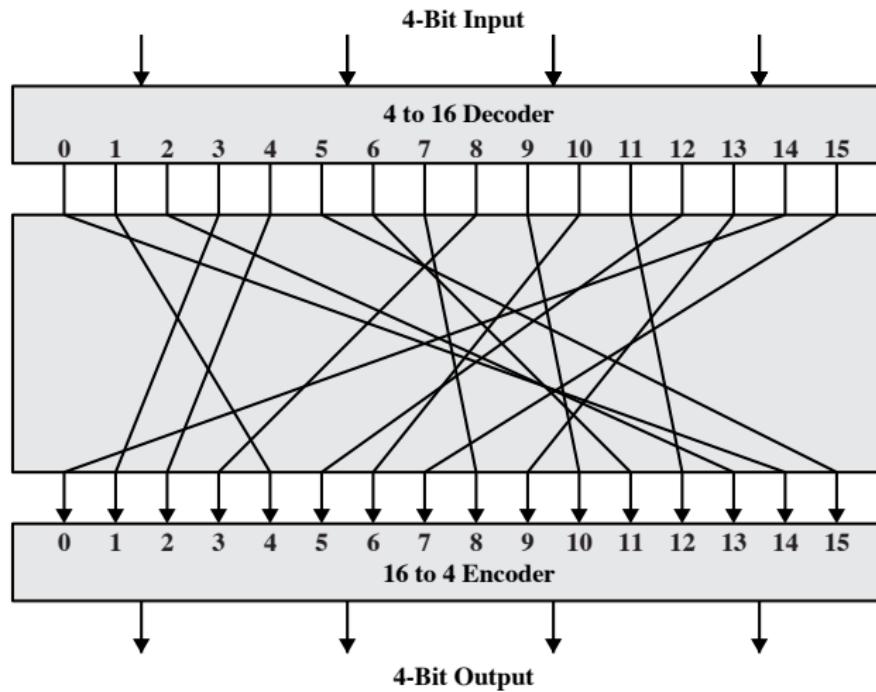


Figure 3.2 General n -bit- n -bit Block Substitution (shown with $n = 4$)

Practice What You Preach

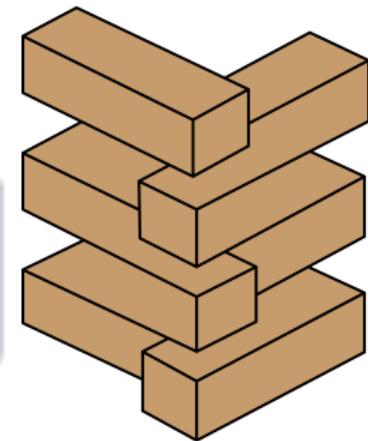
What are the advantages/disadvantages of substitution cipher.

Hint

Ideal block cipher?

Mapping is a table.

The size of the table if $n = 64$ is 10^{21} .



Practice What You Preach

Assume we use a linear mapping in substitution cipher? What would be the problem?

Hint

$$\begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} = (m_0 \ m_1 \dots m_{n-1}) \begin{pmatrix} k_{00} & k_{01} & \dots & k_{0(n-1)} \\ k_{10} & k_{11} & \dots & k_{1(n-1)} \\ \vdots & \vdots & \dots & \vdots \\ k_{(n-1)0} & k_{(n-1)1} & \dots & k_{(n-1)(n-1)} \end{pmatrix}$$

given $k_{ij} \in \{0, 1\}$, $m, c \in \{0, 1\}^n$.

Hill cipher; key storage space requirement in the worst case is n^2 .

Feistel Cipher (feistel1973cryptography)

Cornerstone of block ciphers: Theoretic Reasoning (shannon1949communication)

- **Diffusion:** The statistical structure of m is dissipated. Make the relationship between m and c non-linear.
- **Confusion:** Make the relationship between k and c non-linear.

An arbitrary substitution cipher provides confusion and diffusion, but not practical.

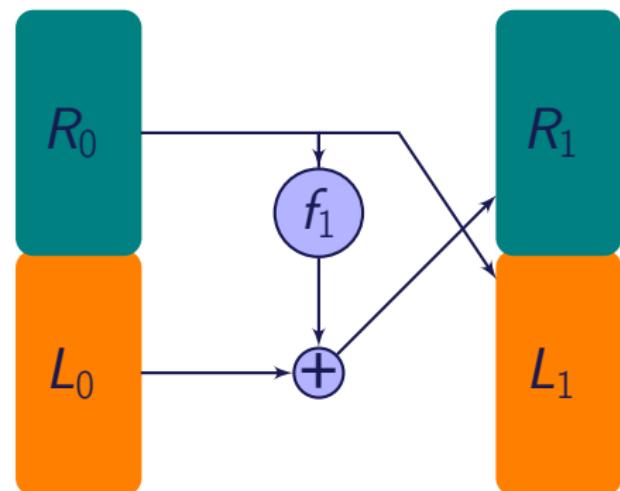
Practical Implementation

- **Substitution:** Each plaintext element is uniquely replaced by a corresponding ciphertext element.
- **Permutation:** A sequence of plaintext elements is replaced by a permutation.

Feistel Cipher: Encryption Round

- R_i and L_i are right and left halves of the plaintext of size $2n$ bits
- $R_i = f_i(R_{i-1}) \oplus L_{i-1}$
- $L_i = R_{i-1}$
- **Permutation:** right-half shifted to next left-half
- **Substitution:** by applying round function f_i

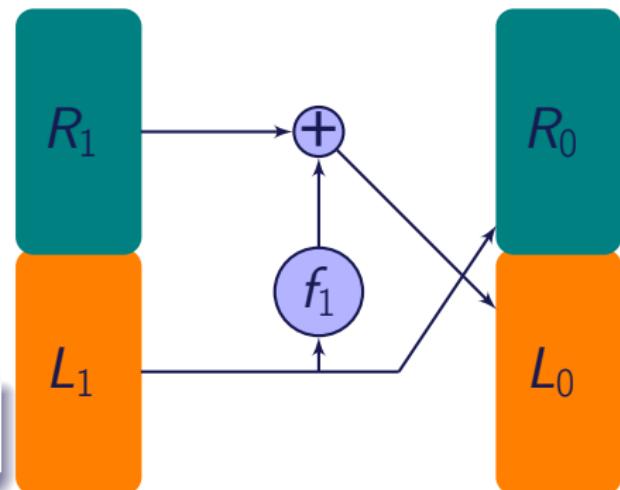
Invertible?



Feistel Cipher: Decryption Round

- R_i and L_i are right and left halves of the plaintext of size $2n$ bits
- $R_{i-1} = L_i$
- $L_{i-1} = f_i(L_i) \oplus R_i$

Yes invertible even if f_i is not.
 f_i can also take keys as input.



Digital Encryption Standard (DES): Overall Structure

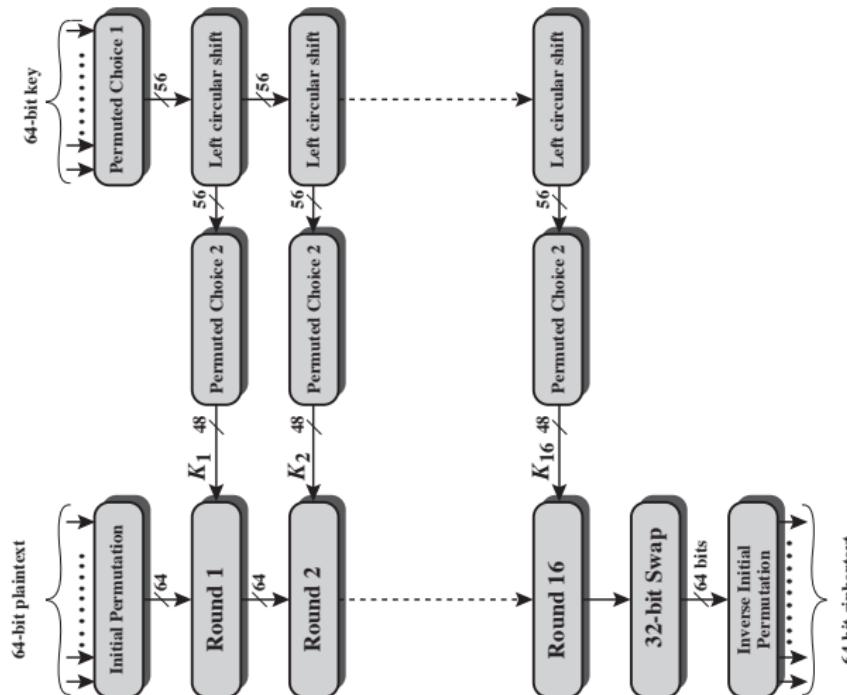


Figure 3.5 General Depiction of DES Encryption Algorithm

Digital Encryption Standard (DES): Overall Structure

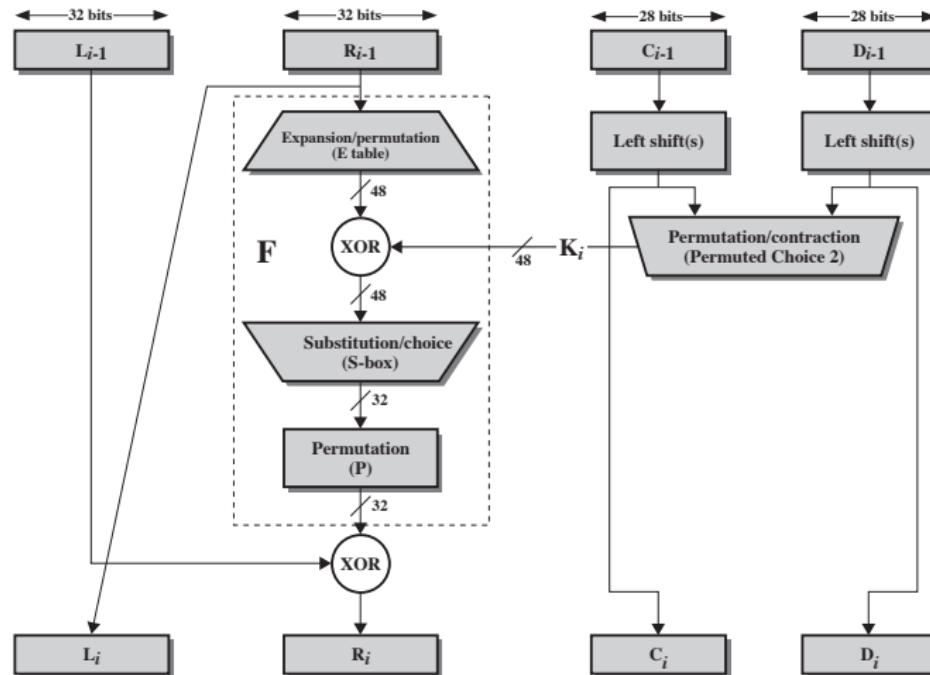


Figure 3.6 Single Round of DES Algorithm

Digital Encryption Standard (DES): Overall Structure

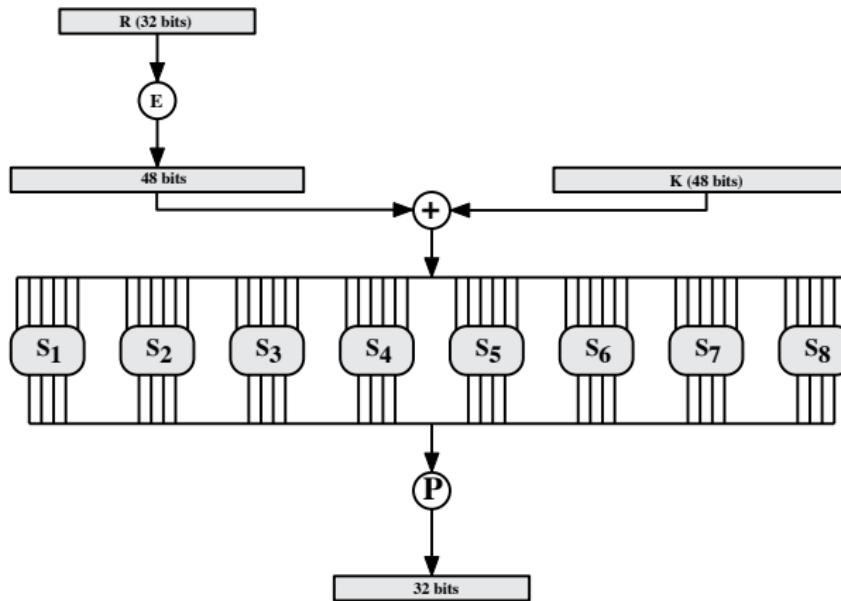


Figure 3.7 Calculation of $F(R, K)$

Considerations When Implementing Feistel Network

- Block size: Larger n , greater diffusion.
- Key size: Larger k , greater confusion.
- Number of rounds: Larger number of rounds, greater security.
- Subkey generation algorithm: Complex algorithm more strong against cryptanalysis.
- Round function: Complex map, greater resistance to cryptanalysis.
- Fast s/w or h/w implementation.
- Ease of analysis: public scrutiny.

Considerations When Implementing Round Function

- Nonlinear: more difficult to approximate the round function by a set of linear equations.
- Strict avalanche criterion (SAC): any output bit j should change with probability $1/2$ when any single input bit i is inverted for all i, j .
- Bit independence criterion (BIC): output bits j and k should change independently when any single input bit i is inverted for all i, j and k .

What is Simplified DES

- developed by Professor Edward Schaefer of Santa Clara University (**Schaefer1996**)
- educational rather than a secure encryption algorithm
- has similar properties and structure to DES with much smaller parameters.

Summary

Today, we learned

- Feistel cipher
- Simplified DES

Lecture 5: Shared-key Encryption Schemes: Operation Modes

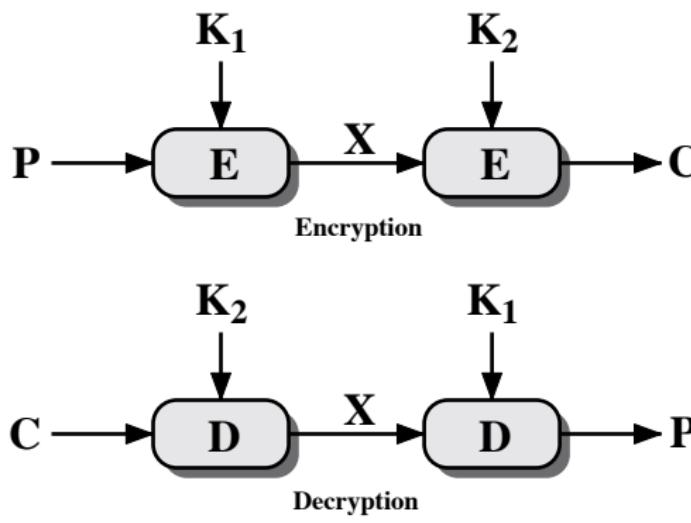
Shared-key Encryption Schemes: Operation Modes

Objectives of the Lecture

At the end of this lecture, you will be able to

- compare contrast different operation modes of block ciphers

Using DES 2 Times



Using a DES 3 Times

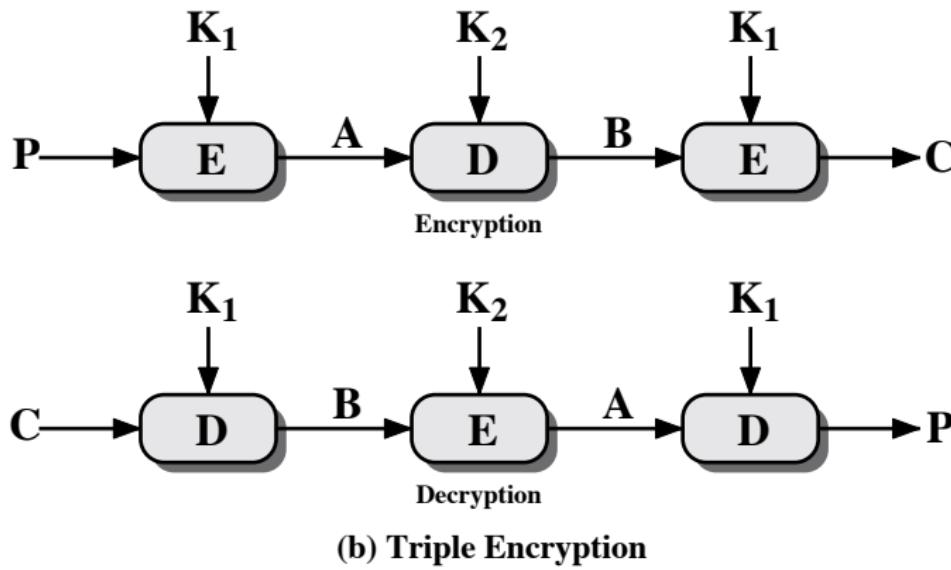
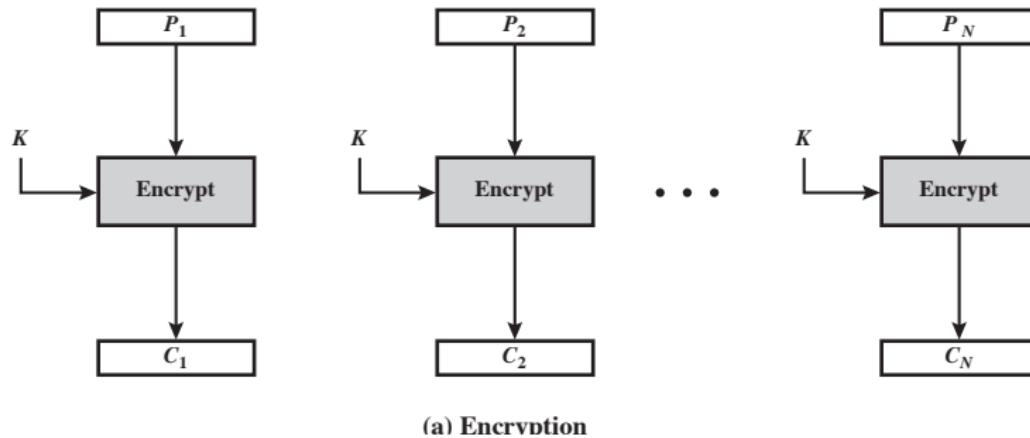


Figure 6.1 Multiple Encryption

Electronic Codebook Model (ECB)



Electronic Codebook Model (ECB)

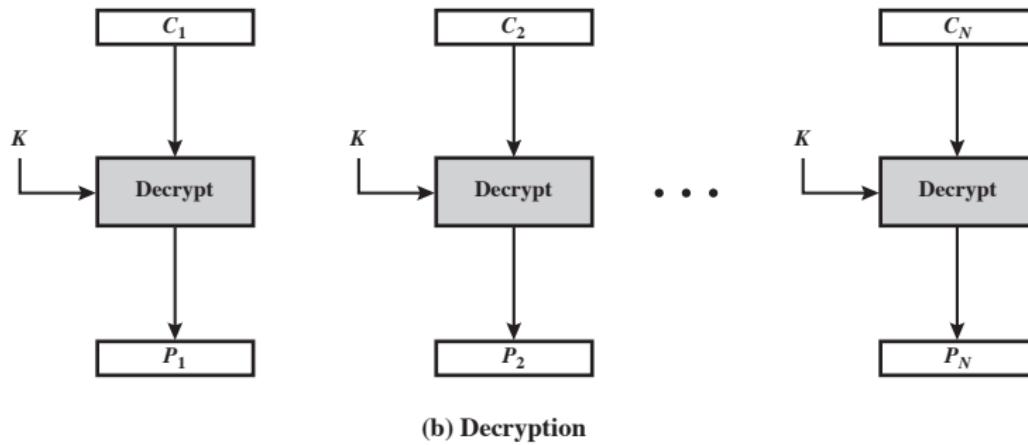
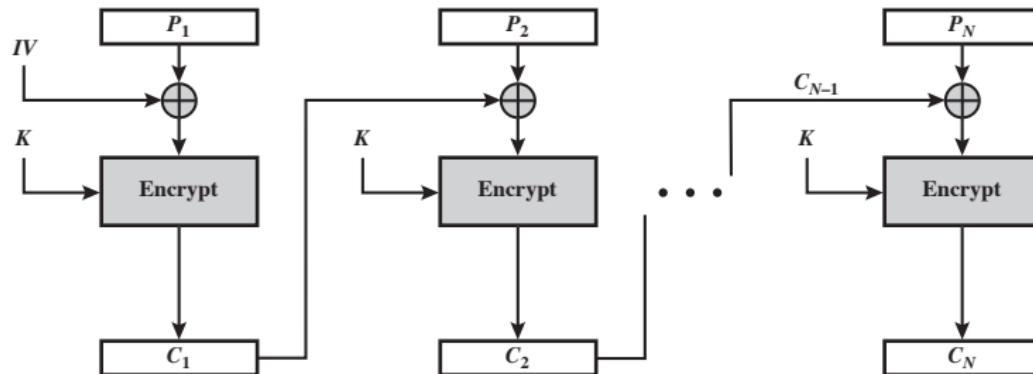


Figure 6.3 Electronic Codebook (ECB) Mode

Electronic Codebook Model (ECB)

- repetitions of blocks
- m_i 's are independently encrypted
- for sending a few blocks of data

Cipher Block Chaining (CBC)



(a) Encryption

Bob can parallelize decryption but Alice can't.

Error can be propagated.

Bob can't decrypt if I swap two blocks but only that two blocks.

No pre-processing

Cipher Block Chaining (CBC)

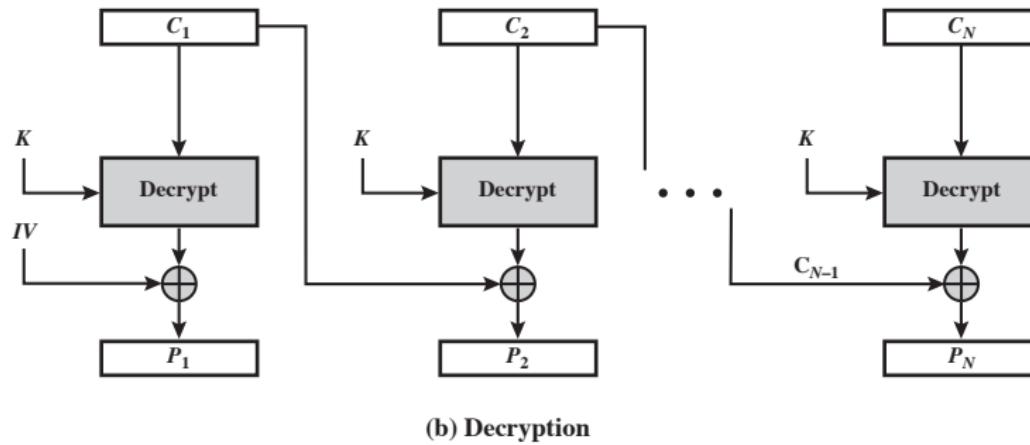
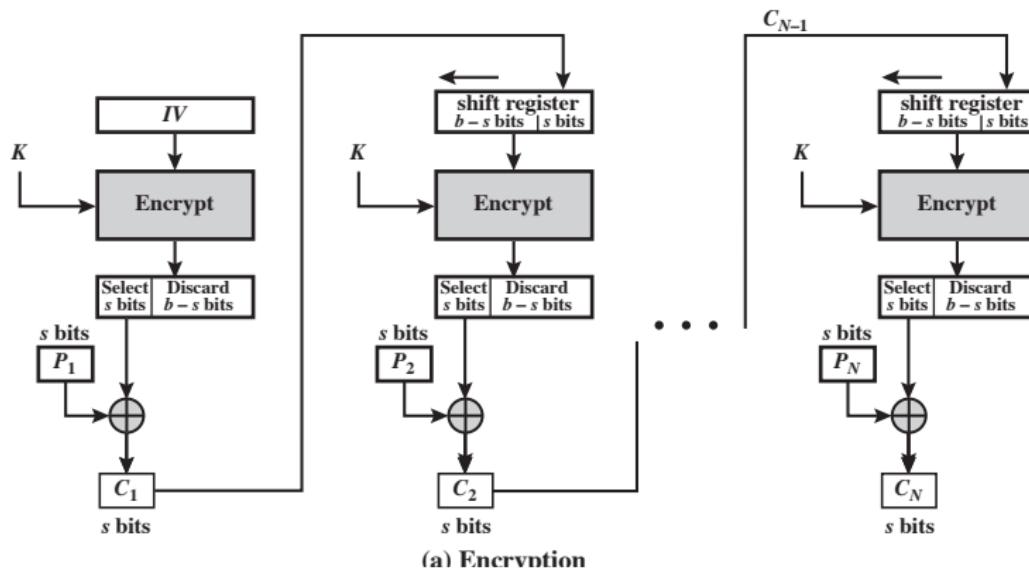


Figure 6.4 Cipher Block Chaining (CBC) Mode

Cipher Block Chaining (CBC)

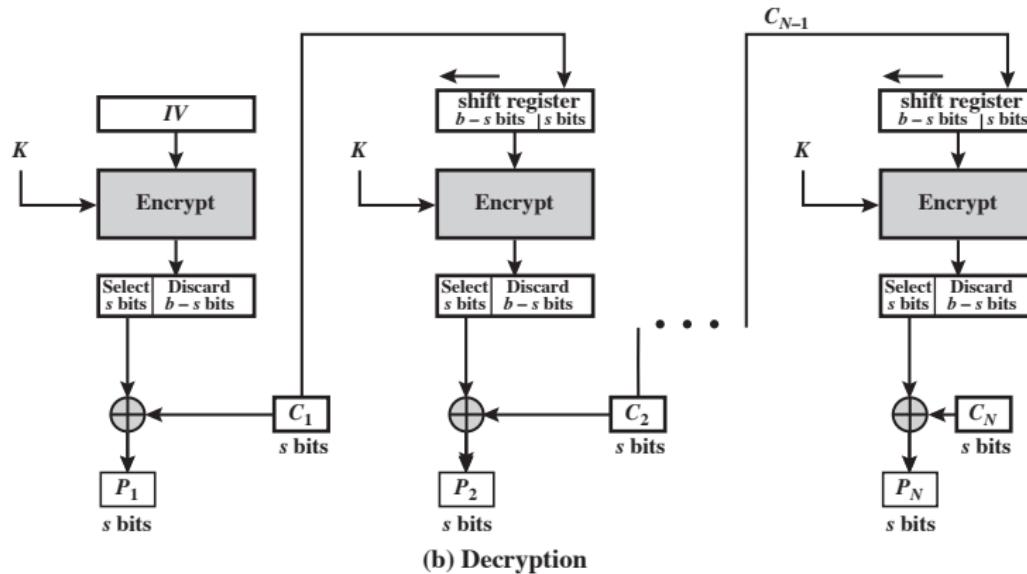
- When last block is not as large as block size of cipher
 - pad either with known non-data value (e.g., nulls)
 - pad last block along with count of pad size
- a ciphertext block depends on all blocks before it
- any change (error) to a block affects all following ciphertext blocks
- need Initialization Vector (IV): fixed or secure

Cipher Feedback Mode (CFB)



(a) Encryption

Cipher Feedback Mode (CFB)



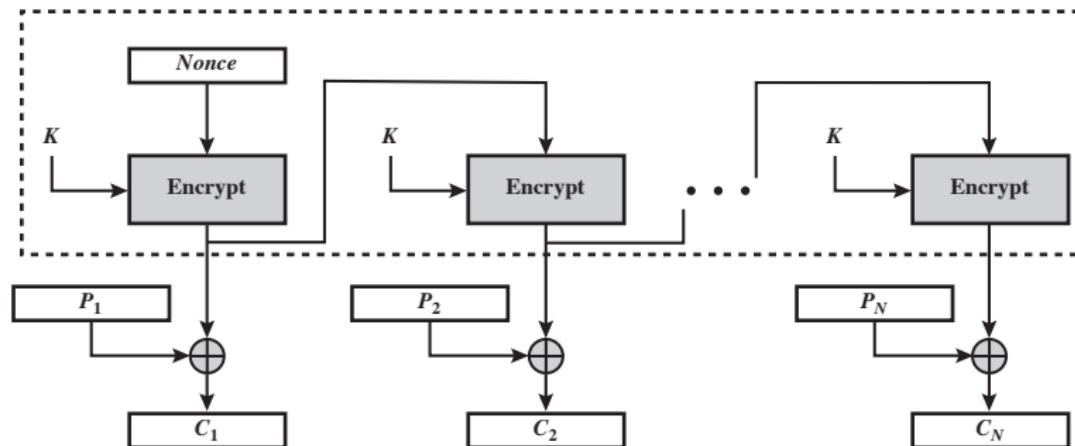
(b) Decryption

Figure 6.5 s-bit Cipher Feedback (CFB) Mode

Cipher Feedback Mode (CFB)

- appropriate for stream data
- block cipher is used **in encryption mode** by principals
- bit (or block) errors propagate for several blocks after the error

Output Feedback Mode (OFB)



(a) Encryption

Output Feedback Mode (OFB)

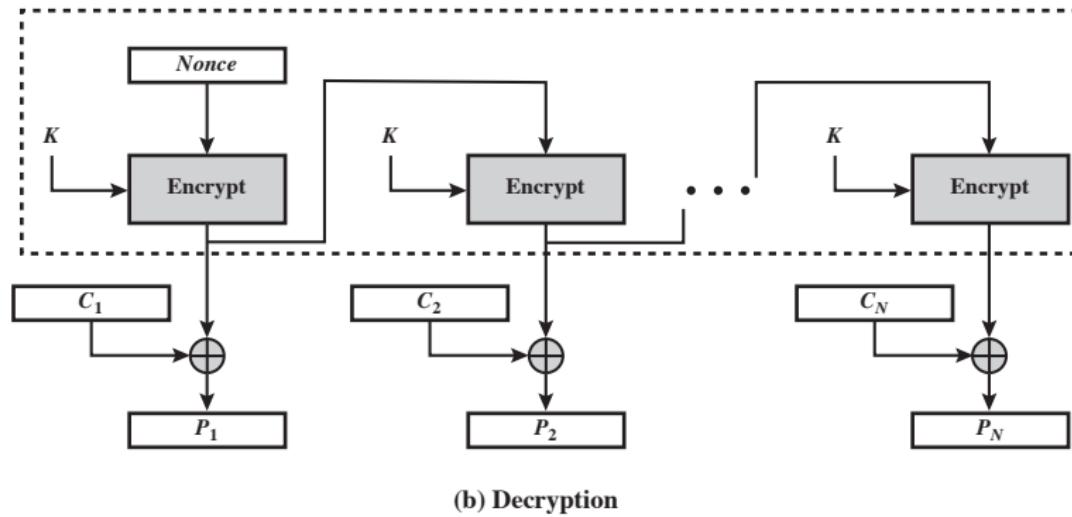
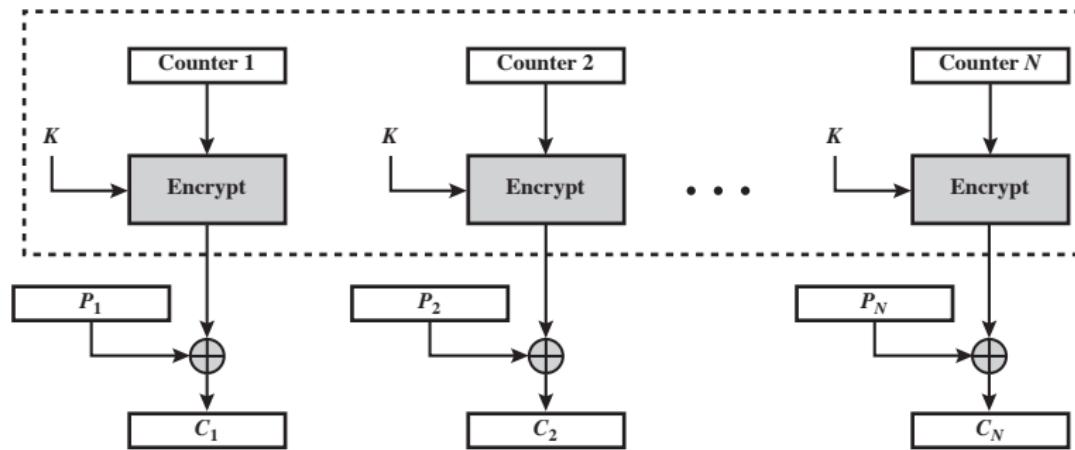


Figure 6.6 Output Feedback (OFB) Mode

Output Feedback Mode (OFB)

- needs a unique IV per use
- **bit errors do not propagate**
- more vulnerable to message stream modification
- synchronized principals
- do not use subsequences, use full block

Counter Mode (CTR)



(a) Encryption

Counter Mode (CTR)

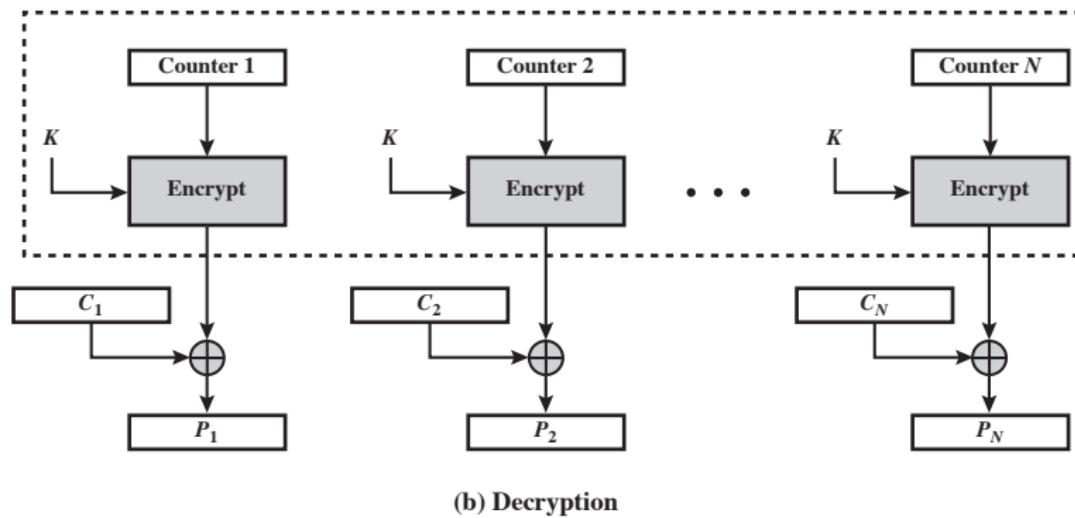


Figure 6.7 Counter (CTR) Mode

Counter Mode (CTR)

- efficient parallel (pipelined) implementation
- pre-processing works
- adequate for high capacity links
- random access
- never reuse key/counter values

Summary

Today, we learned

- Different cipher chaining modes for long stories

Lecture 6: Public Key Schemes: RSA

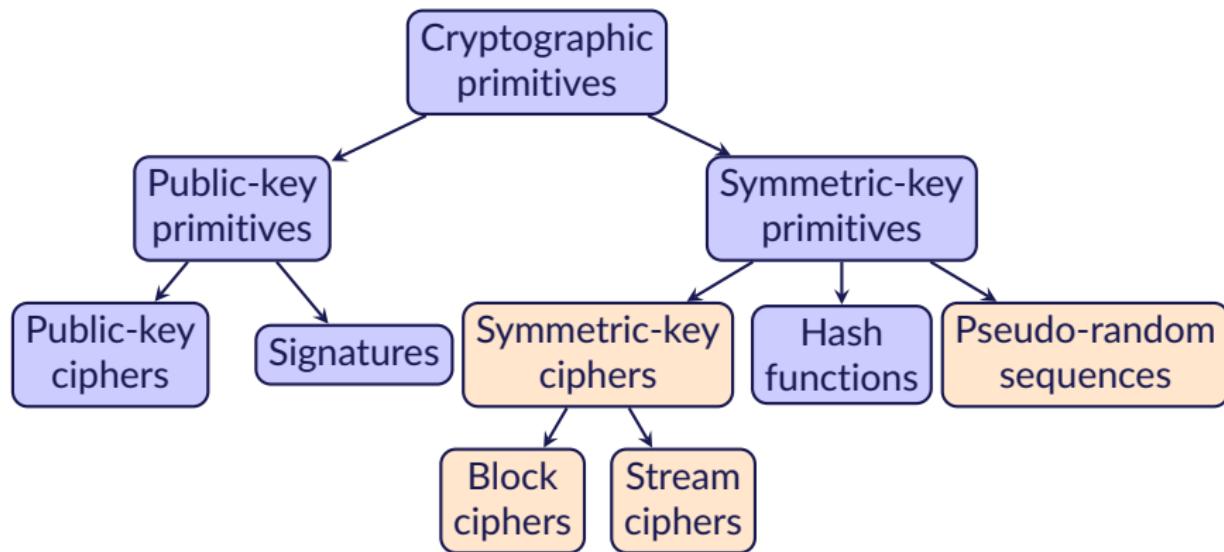
Public Key Schemes: RSA

Objectives of Lecture

At the end of this lecture, you will be able to

- define public key schemes
- share keys over insecure channels

Overview of Cryptographic Primitives



Shared-key Encryption Scheme

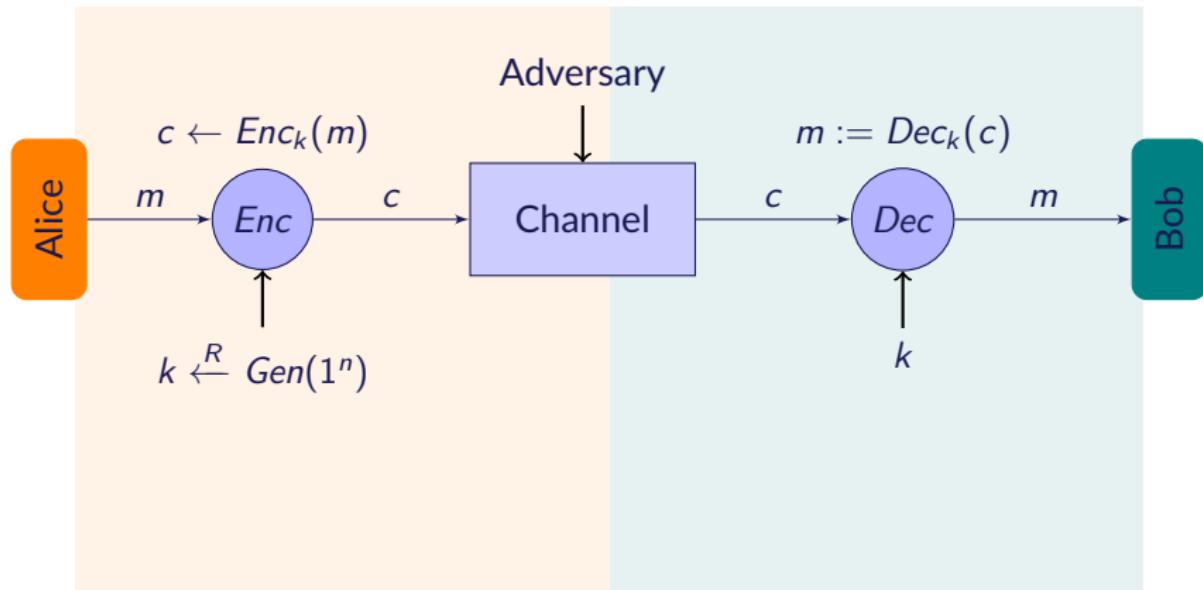


Figure: A simple model for network security employing shared-key encryption.

Public-key Encryption Scheme

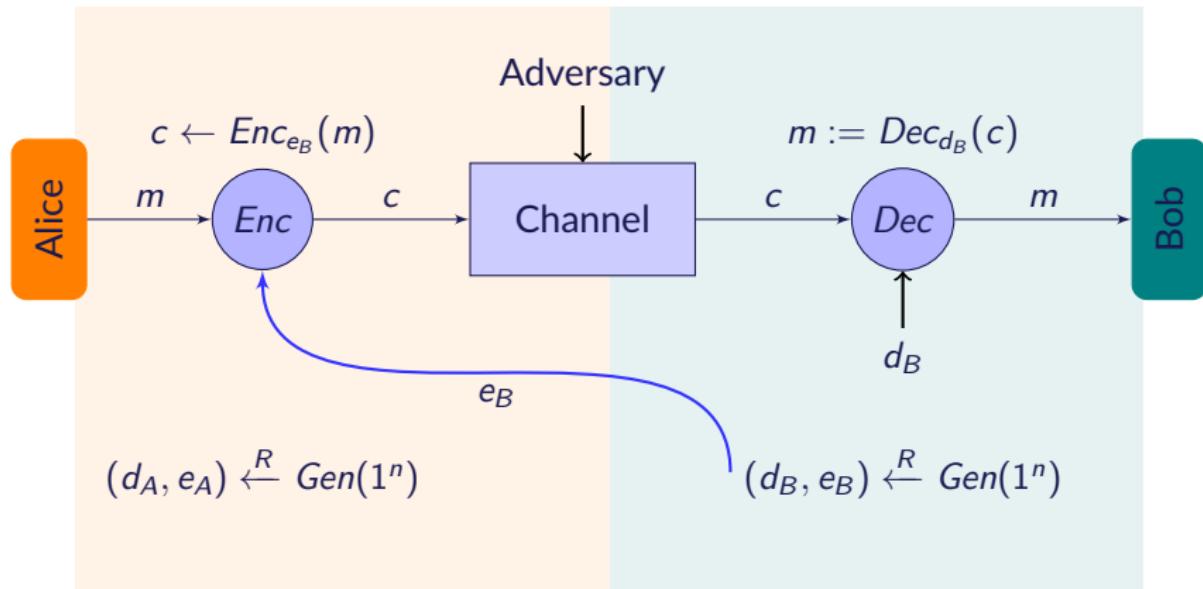


Figure: A simple model for network security employing public-key encryption.

Shared- versus Public-key Schemes Needed to Work

Shared-key	Public-key
The same algorithm with the same key is used for encryption and decryption	One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption
The sender and receiver must share the algorithm and the key	The sender and receiver must each have one of the matched pair of keys (not the same one)

Shared- versus Public-key Schemes Needed for Security

Shared-key	Public-key
The key must be kept secret	One of the two keys must be kept secret
It must be impossible or at least impractical to decipher a message if no other information is available	It must be impossible or at least impractical to decipher a message if no other information is available
Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key	Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key

Applications of Public-key Systems

Algorithm	Secrecy	Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

Public-key Requirements

Public-Key algorithms rely on two keys where it is

- computationally easy
 - to generate the key(s)
 - to encrypt (decrypt) messages when the relevant key is known
- computationally difficult
 - to find decryption key knowing only algorithm and encryption key
 - to decrypt when the relevant key is unknown

RSA by Rivest, Shamir & Adleman of MIT in 1977

- Best known and widely used public-key scheme
- Exponentiation in finite fields
- Encryption/decryption $O((\log(n))^3)$
- Uses large integers (>1024 bits)
- Security relies on factoring large numbers $O(\exp\{\log n \log \log n\})$

RSA Scenario

Alice wants to send a concealed message $m < N$ to Bob

- Bob's public key = (e, N)
- Alice encrypts: $c = m^e \pmod{N}$ using Bob's public key
- Bob's private key = (d, N)
- Bob decrypts: $m = c^d \pmod{N}$ using his private key

Security Assumption

N is the product of two large primes. It is very computationally infeasible to find the primes given N . This is the security assumption.

RSA Key Generation

- Bob generates his public (e, N) - private (d, N) key pair.
- Bob does the following to generate the key pair
 - He selects two primes $p = 17$ and $q = 11$ (very large)
 - He computes $N = pq = 187$
 - He computes $\phi(N) = (p - 1)(q - 1) = (17 - 1)(11 - 1) = 160$ (Euler totient function)
 - He selects $1 < e < \phi(N)$ such that $\gcd(e, \phi(N)) = 1$ randomly ($e = 7$ is relatively prime to $\phi(N) = 160$)
 - He determines d such that $de \equiv 1 \pmod{\phi(N)}$ (i.e., $d = e^{-1} \pmod{\phi(N)}$ e.g., using extended Euclid's Algorithm $d = 23$)
- Public key $(e, N) = (7, 187)$
- Private key $(d, N) = (23, 187)$

RSA Encryption/Decryption Secrecy

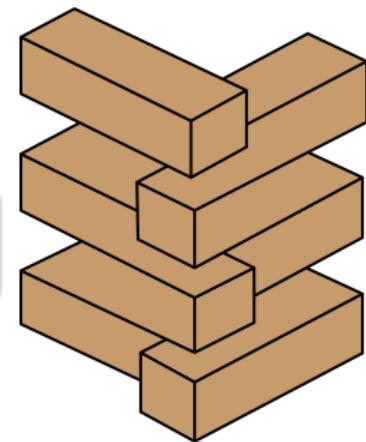
- Alice obtains Bob's public key $(e, N) = (7, 187)$
- Alice wants to encrypt and send message $m = 88$ to Bob
- Alice encrypts $c = m^e \bmod N$ using Bob's public key and produces ciphertext $c = 88^7 \bmod 187 = 11$
- She sends $c = 11$ to Bob
- Only Bob knows his private key $(d, N) = (23, 187)$
- Bob decrypts $m = c^d \bmod N$ and obtains $m = 11^{23} \bmod 187 = 88$

Practice What You Preach

Prove the correctness of RSA

Hint

$x^{\phi(N)} = 1 \pmod{N}$, Euler's Theorem



Correctness of RSA

Let $y = RSA(x)$ represent encryption of $x < N$ using public key (e, N) : $y = x^e \pmod{N}$. Let $x = RSA^{-1}(y) = y^d \pmod{N}$, be the decryption of the ciphertext y using the private key (d, N) where $de = 1 \pmod{\phi(N)}$. Then, $ed = k\phi(N) + 1$ for some $k \in \mathbb{Z}$.

$$\begin{aligned} x &= y^d \pmod{N} = (x^e)^d \pmod{N} = x^{ed} \pmod{N} = x^{k\phi(N)+1} \pmod{N} \\ &= \left(x^{\phi(N)}\right)^k x \pmod{N} = x \pmod{N} = x \end{aligned}$$

Since $x^{\phi(N)} = 1 \pmod{N}$ by Euler's Theorem.

RSA Encryption/Decryption

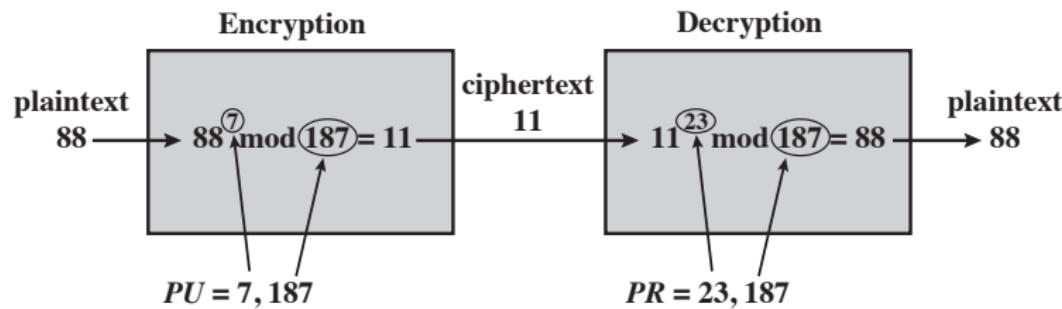


Figure 9.6 Example of RSA Algorithm

RSA Encryption/Decryption

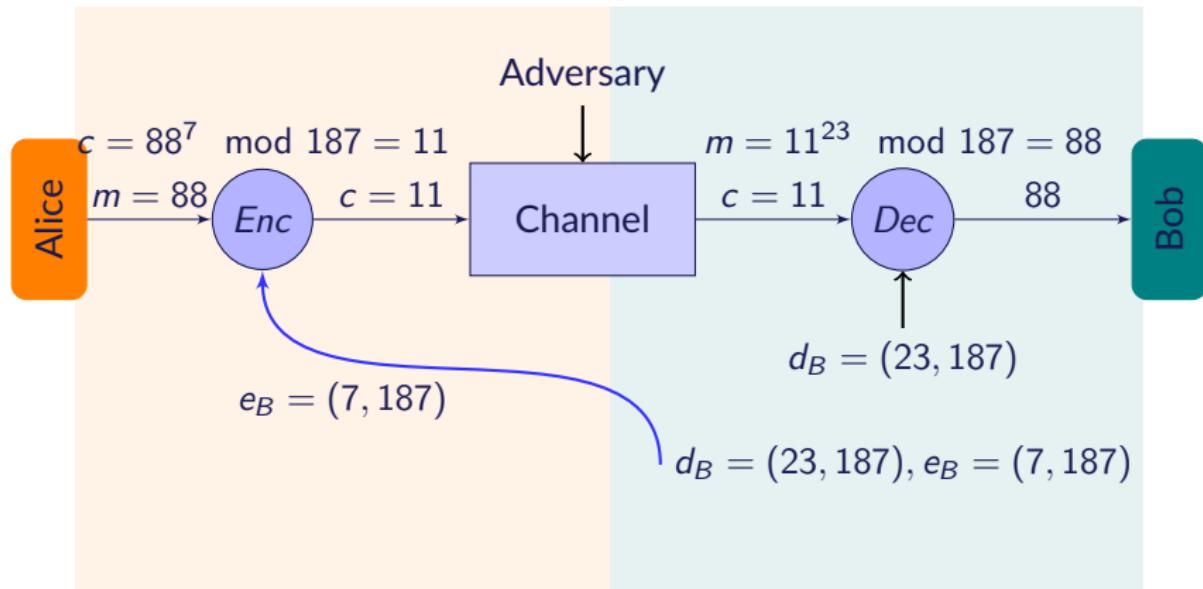


Figure: Example RSA encryption and decryption.

How to Use RSA

$y = m^e \bmod N$ is not secure when m is too short.

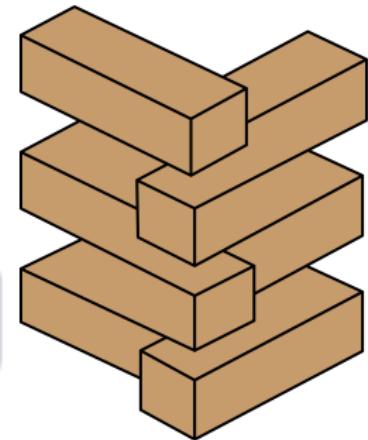
- Determine a symmetric encryption system (E_k, D_k) where $k \in \mathcal{K}$ and \mathcal{K} is the key space.
- Determine a hash function $H : Z_N \rightarrow \mathcal{K}$
- Generate RSA parameters: public key = (e, N) , private key = (d, N) .
- Choose random $x \in Z_N$
- $y \leftarrow RSA(x) = x^e \bmod N, k \leftarrow H(x)$
- Alice encrypts message m and sends to Bob using the generated key k :
Output $(y, c = E_k(m))$
- Bob decrypts the ciphertext by $m = D_{k \leftarrow H(RSA^{-1}(y))}(c)$.

Practice What You Preach

1. Should e or d be small?
2. How would you select e ?
3. What is the minimum value of e ?

Hint

Think about what is public?



RSA Signing Messages

- Alice's private key (d, N)
- Alice wants to **sign** and send a signed message m to Bob
- Alice encrypts $c = m^d \pmod{N}$ using her public key and produces ciphertext c
- She sends m and c to Bob
- Everybody including Bob knows Alice's public key (e, N)
- Bob decrypts $\hat{m} = c^e \pmod{N}$ and checks whether $\hat{m} \stackrel{?}{=} m$

RSA Signing Messages (Origin authentication)

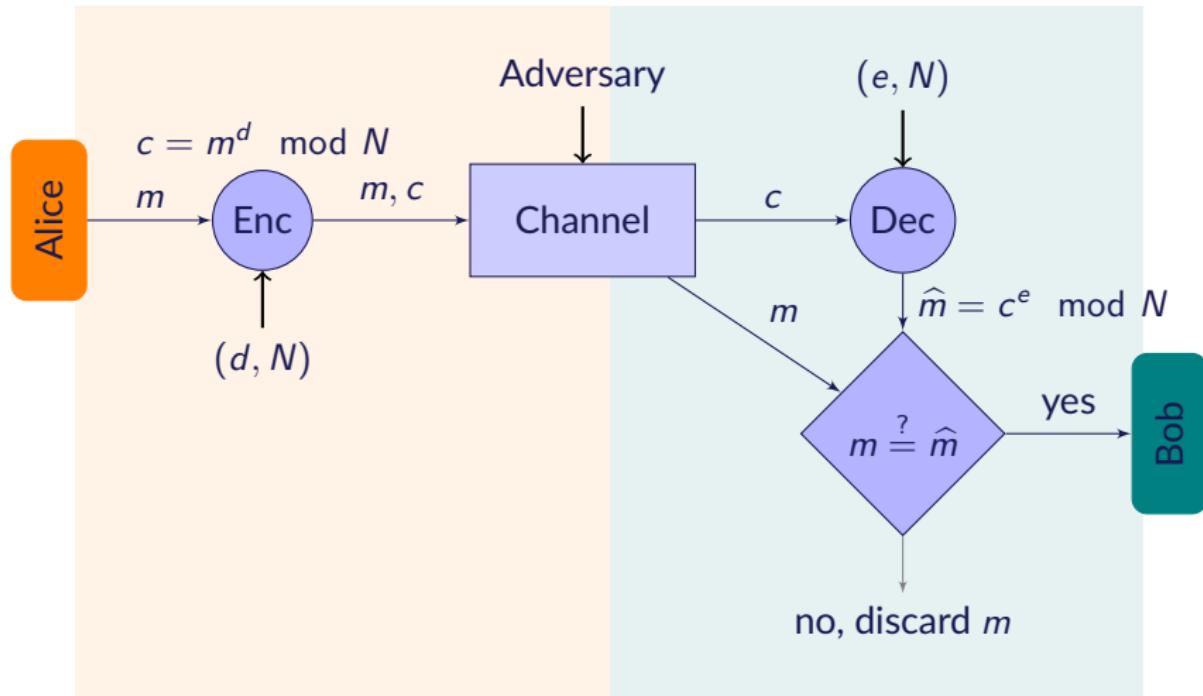


Figure: Signing documents using RSA.

Timing Attacks and Blinding

- Timing attack: Observe the duration of modular exponentiation
- Modular exponentiation example: $x^{11} = \text{xxxxxxxxxx}$ but also $x^{11} = xx^2x^8$
- A simple solution is blinding, $m = c^d \pmod{N}$
 - Generate a secret random $r < n$
 - Compute $c^* = c(r^e) \pmod{N}$
 - $m^* = (c^*)^d \pmod{N}$
 - Compute $m = m^*r^{-1} \pmod{N}$ where $r^{-1} = 1/r$

Summary

Reference: Chapter 9

Today, we learned

- Public-key schemes
- RSA

Lecture 7: Public Key Schemes: Diffie-Hellman Key Exchange

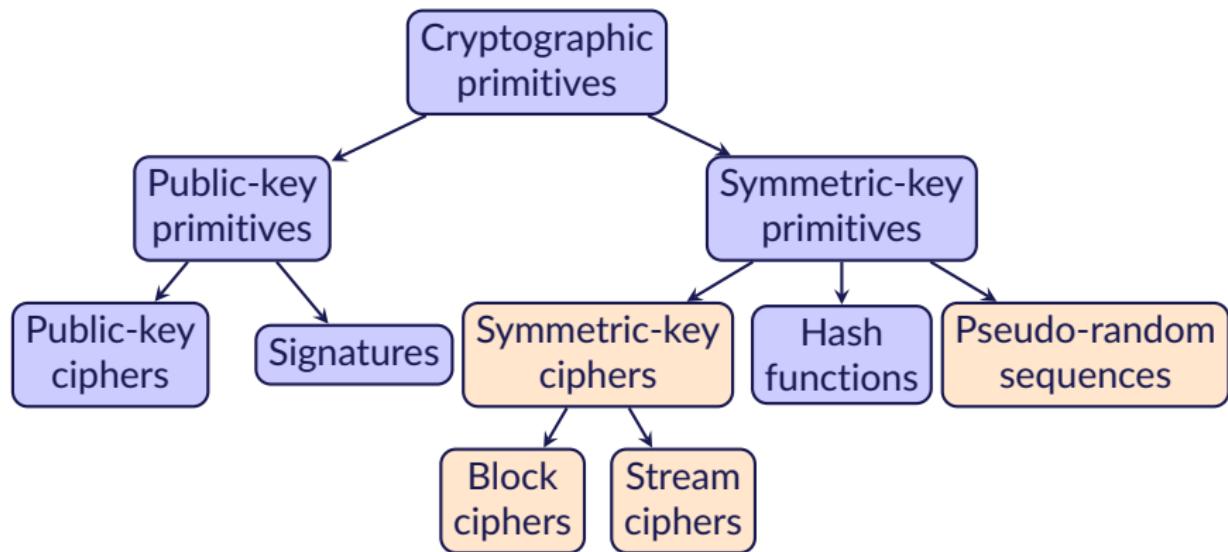
Public Key Schemes: Diffie-Hellman Key Exchange

Objectives of Lecture

At the end of this lecture, you will be able to

- define public key schemes
- share keys over insecure channels

Overview of Cryptographic Primitives



Diffie-Hellman (DH) Key Exchange

Scenario

Alice and Bob do not share a secret. They want to generate a key by communicating over insecure channel.

- by Diffie and Hellman, 1976
- TLS uses DH
- Security assumption: difficulty of computing discrete logarithms

Discrete Logarithm Problem

- A primitive root of a prime p is a if the powers of a generate all integers from 1 to $p - 1$.
- That is if a is a primitive root of p then

$$(a \pmod p), (a^2 \pmod p), \dots, (a^{p-1} \pmod p)$$

are distinct and consistent integers from 1 to $p - 1$

- Then, for any integer b we can find i such that $b = a^i \pmod p$ where $1 \leq i \leq p - 1$
- The exponent i is referred to as discrete logarithm
- Finding $dlog_{a,p}(b)$ is a computationally infeasible problem

Example Primitive Root

Let's say $a = 2$ and $p = 19$.

$$(2^1 \bmod 19 = 2), (2^2 \bmod 19 = 4), \dots, (2^{15} \bmod 19 = 12) \dots$$

Let's say $a = 1$ and $p = 19$

$$(1^1 \bmod 19 = 1), (1^2 \bmod 19 = 1), (1^3 \bmod 19 = 1), \dots$$

2 is a primitive root of 19 (other primitive roots are 3, 10, 13, 14)

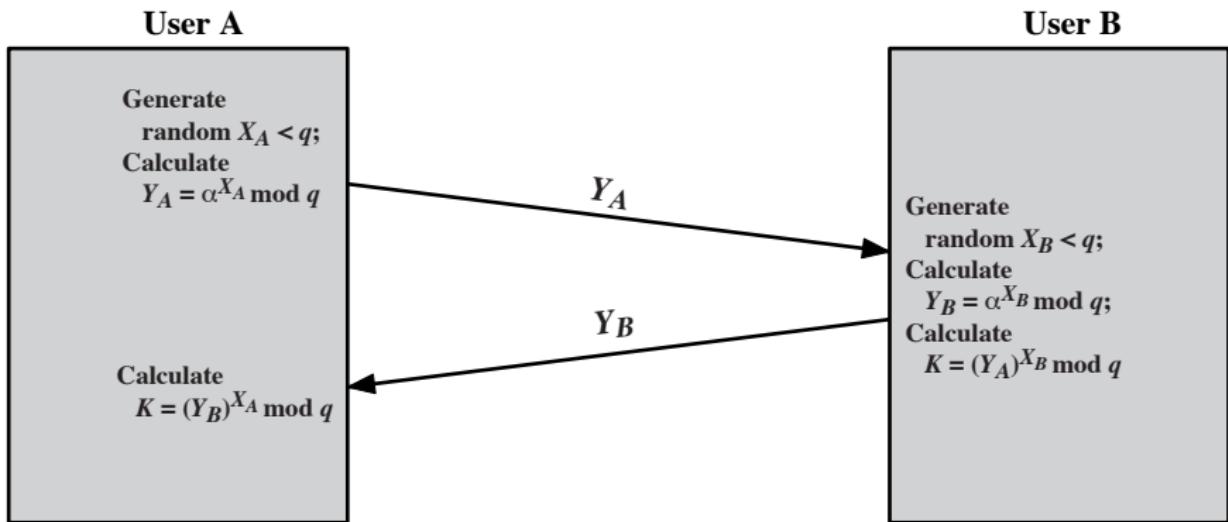
Diffie-Hellman (DH) Key Exchange

- Two public numbers a prime q and α that is a primitive root of q
- Alice selects a random integer $X_A < q$ and computes $Y_A = \alpha^{X_A} \pmod{q}$
- Bob selects a random integer $X_B < q$ and computes $Y_B = \alpha^{X_B} \pmod{q}$
- Each party keeps the X values secret and makes Y public
- Alice computes the key by $k = (Y_B)^{X_A} \pmod{q}$
- Bob computes the **same** key by $k = (Y_A)^{X_B} \pmod{q}$
- Adversary has to solve a discrete log problem that is difficult. Other than X values, other parameters are public.

Diffie-Hellman (DH) Key Exchange Correctness

$$\begin{aligned} k &= (Y_B)^{X_A} \mod q \\ &= (\alpha^{X_B} \mod q)^{X_A} \mod q \\ &= (\alpha^{X_B})^{X_A} \mod q \\ &= \alpha^{X_B X_A} \mod q \\ &= \alpha^{X_A X_B} \mod q \\ &= (\alpha^{X_A})^{X_B} \mod q \\ &= (\alpha^{X_A} \mod q)^{X_B} \mod q \\ &= (Y_A)^{X_B} \mod q \end{aligned}$$

Man in the Middle



Summary

Reference: Chapter 10

Today, we learned

- Diffie Hellman key exchange

Lecture 8: Cryptographic Hash Functions

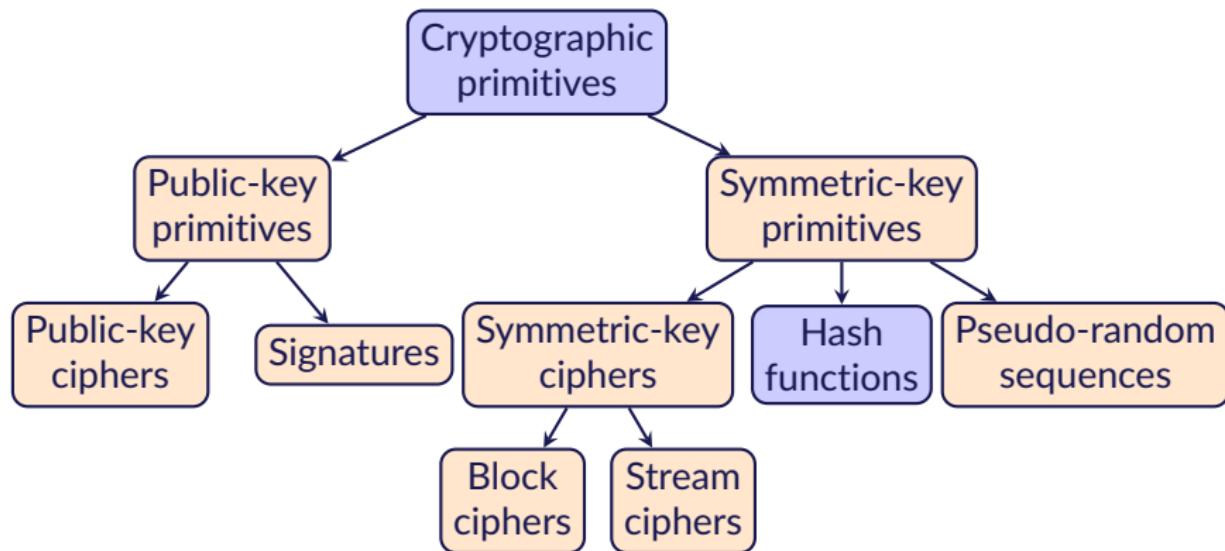
Cryptographic Hash Functions

Objectives of Lecture

At the end of this lecture, you will be able to

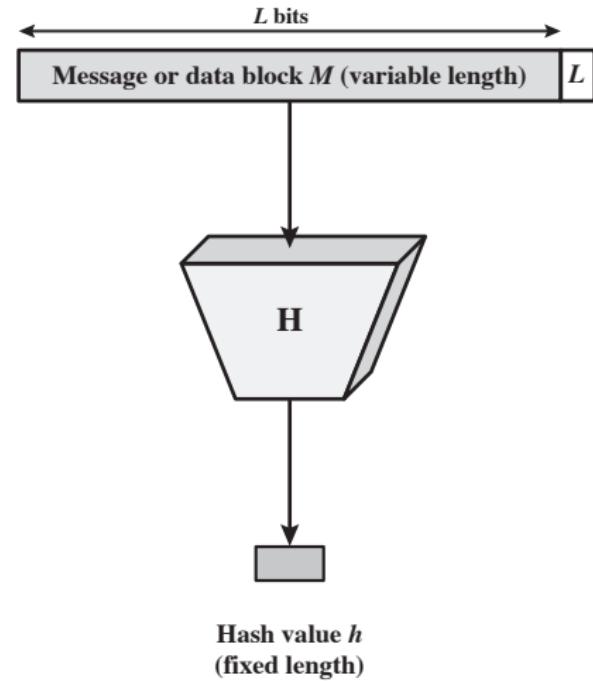
- define hash functions

Overview of Cryptographic Primitives



Hash Functions

- Hash function compresses (condenses) an arbitrary message to a fixed-size digest
- $h = H(m)$

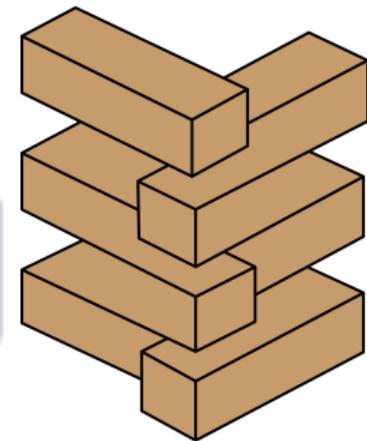


Practice What You Preach

You download software from a web page. How can you be sure that nobody has injected a virus in it?

Hint

Does the software developer provide an MD5 signature?



Applications of Hash Functions

- Integrity check
- Signatures
- One-way password files
- Virus detection
- As a pseudo-random number generator

Integrity Check using Hash (One-way) Functions

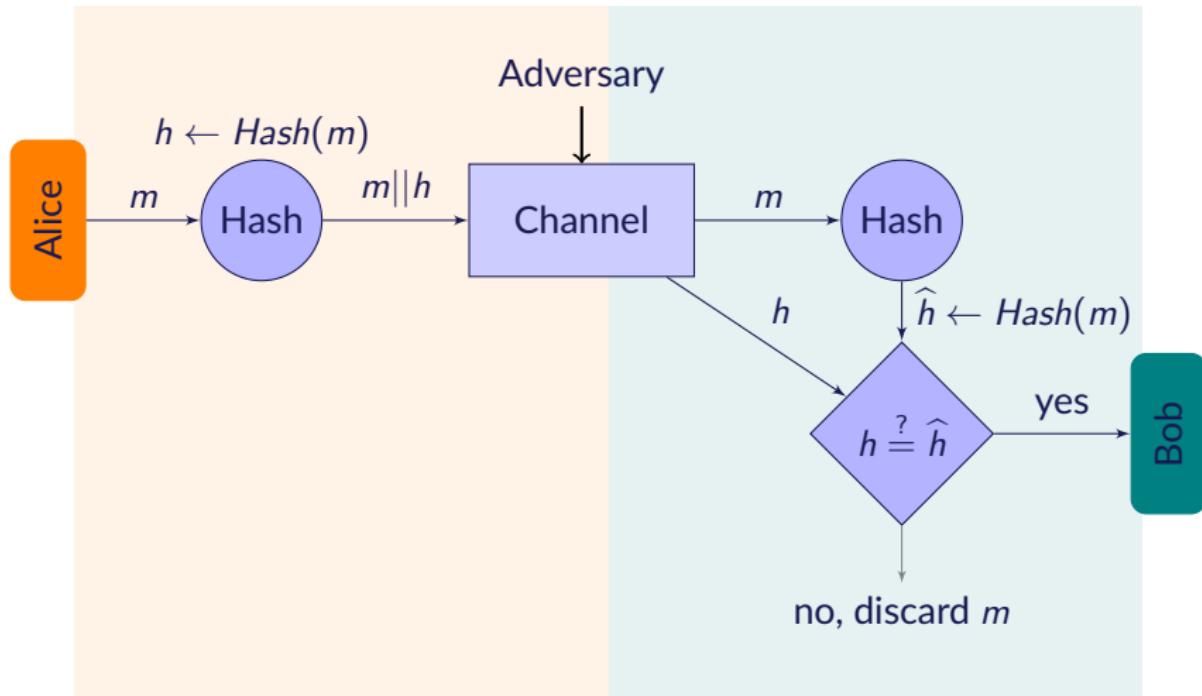
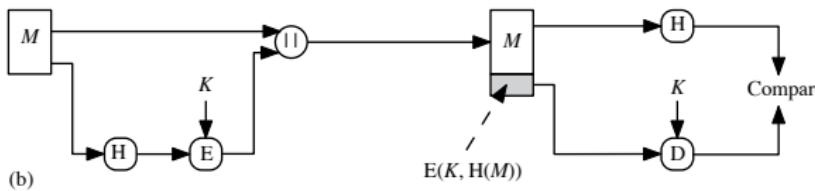
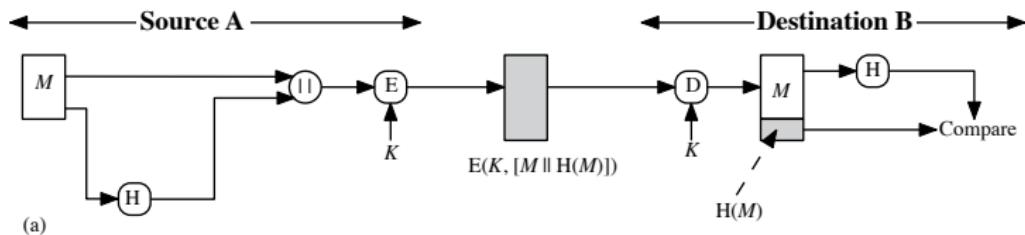


Figure: A simple model for message integrity check.

Simplified Use of Hash Functions



Simplified Use of Hash Functions

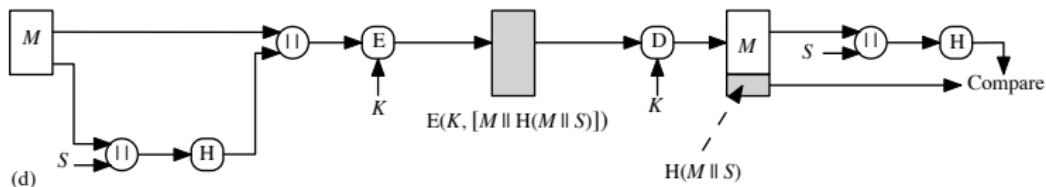
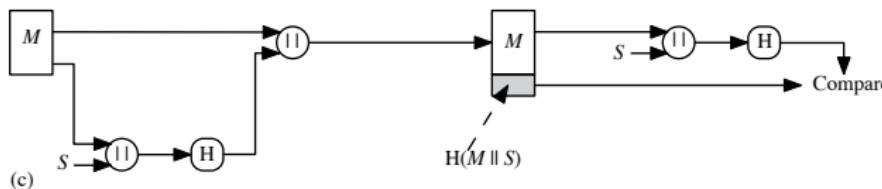
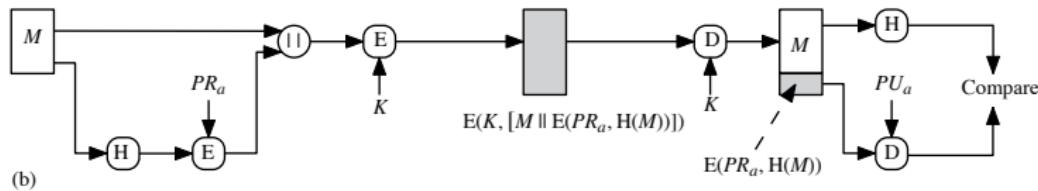
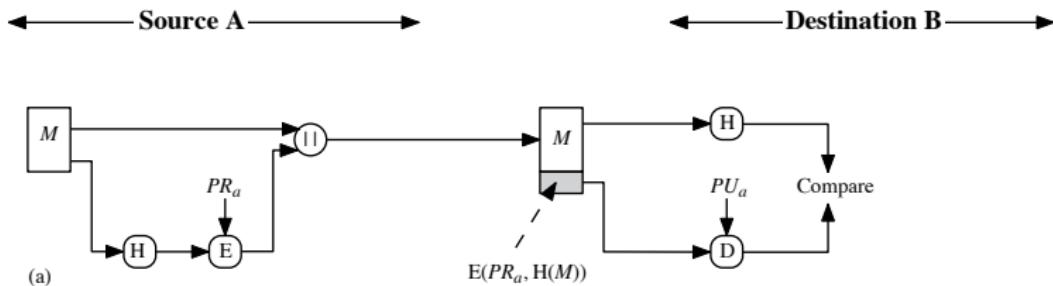


Figure 11.2 Simplified Examples of the Use of a Hash Function for Message Authentication

Simplified Use of Digital Signatures



A Simple Hash Function

- Bitwise \oplus of every block
- $h_i = b_{i1} \oplus b_{i2} \oplus \dots \oplus b_{iM}$
- where b_{ij} is the i th bit of j th block and M is the number of blocks
- works fine for randomized data
- What is the problem?
- Read page 358 in textbook for CBC problem

Hash Function: Definitions

Given

$$h = H(x)$$

then, x is the preimage of h .

H is many-to-one mapping.

A collision occurs when $H(x) = H(y)$ where $x \neq y$.

Question

Assume

- H is a PRF
- input size is $|x| = b$ bits
- hash size is $|h| = n < b$ bits

how many potential preimages are there? What happens when b is arbitrary?
(page 359)

Requirements for a Cryptographic Hash Functions

- 1,2,3 **Variable input size, Fixed output size, Efficiency**
- 4 **Preimage resistance** $2^{|h|}$: Given $h = H(m)$ it must be computationally infeasible to find m given h (one-way property)
- 5 **Second preimage resistance** $2^{|h|}$: Given m_0 it must be computationally infeasible to find $m_1 \neq m_0$ such that $H(m_0) = H(m_1)$ (forgery prevention)
- 6 **Collision resistance** $2^{\frac{|h|}{2}}$: Cannot find (m_0, m_1) pair such that $H(m_0) = H(m_1)$
 - Birthday paradox: Choose rv from $(0, N - 1)$, probability of a repeated element exceeds 0.5 after \sqrt{N} choices
- 7 **Pseudorandomness**: h looks like uniform random

1,2,3,4,5: weak hash function

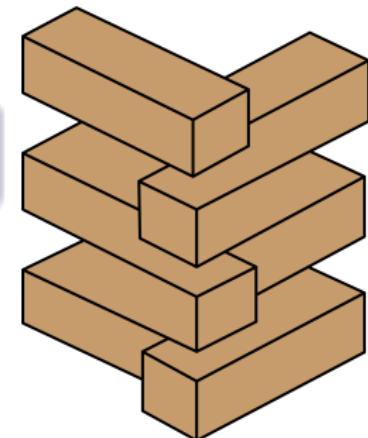
1,2,3,4,5,6: strong hash function.

Practice What You Preach

What is the level of brute-force effort required against preimage attacks, second preimage attacks, and collision resistant attacks?

Hint

How many trials does the attacker have to make?



Practice What You Preach

What is the level of brute-force effort required against preimage attacks, second preimage attacks, and collision resistant attacks?

Hint

How many trials does the attacker have to make?

Answer

Preimage resistant

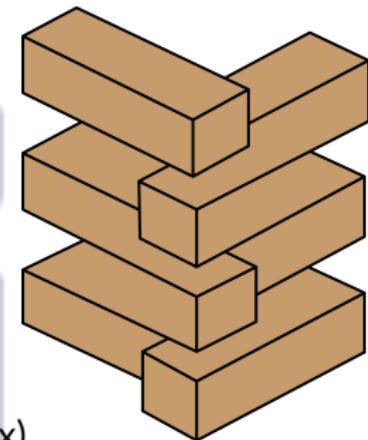
$O(2^m)$

Second preimage resistant

$O(2^m)$

Collision resistant

$O(2^{m/2})$ (Birthday Paradox)



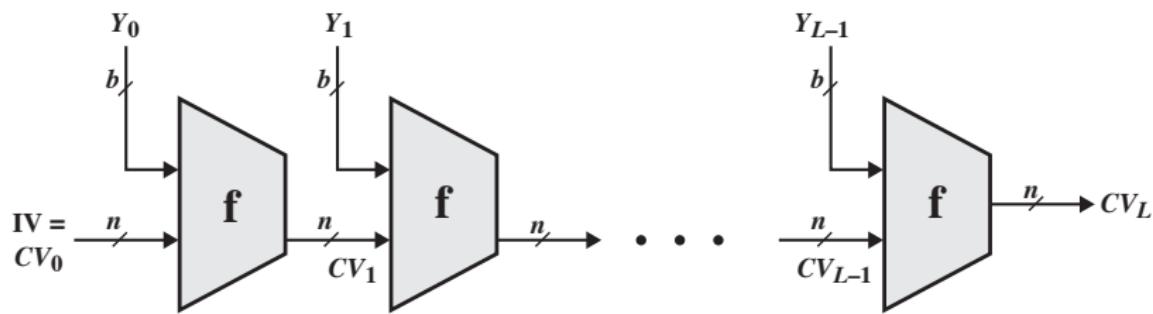
Birthday Paradox

Birthday Paradox

If we choose random variables from a uniform distribution in the range 0 to $N - 1$, then the probability that a repeated element is encountered exceeds 0.5 after \sqrt{N} choices are made.

For an m -bit hash value, if we pick data blocks at random, we can expect to find a collision (two data blocks producing same hash value) within $\sqrt{2^m} = 2^{m/2}$.

General Structure of Secure Hash Functions



IV = Initial value

CV_i = chaining variable

Y_i = i th input block

f = compression algorithm

L = number of input blocks

n = length of hash code

b = length of input block

Example Hash Functions

- MD4, MD5
- SHA: Secure Hash Algorithms (SHA1, SHA-224, SHA-256, SHA-384, SHA-512)

SHA-3: Keccak as of October 2012

<http://keccak.noekeon.org>: This page is dedicated to the cryptographic sponge function family called Keccak, which has been selected by NIST to become the new SHA-3 standard.

Summary

Reference: Chapter 11

Today, we learned

- Cryptographic Hash functions

Lecture 9: Message Authentication Codes

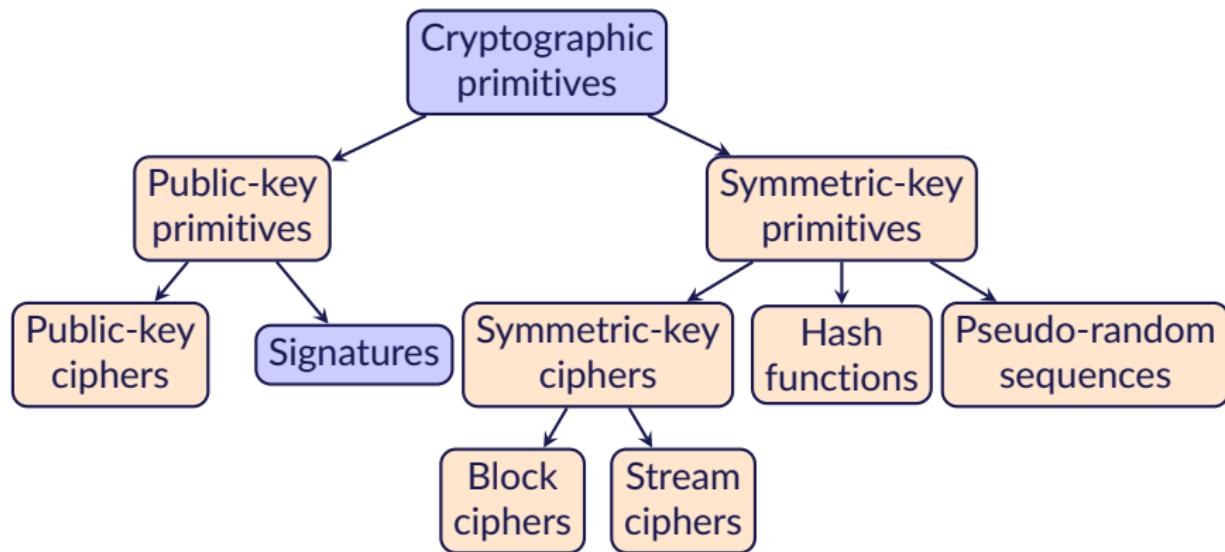
Message Authentication Codes

Objectives of Lecture

At the end of this lecture, you will be able to

- apply message authentication codes in communication
(stallings2001network)

Overview of Cryptographic Primitives



Why Message Authentication?

Network attacks:

- Disclosure, Traffic analysis [CONFIDENTIALITY]
- Masquerade
- Content modification
- Sequence modification
- Timing modification
- Source or destination repudiation [DIGITAL SIGNATURES]

Definition: Authentication

Authentication

Message authentication is a procedure to verify that received messages come from the alleged source (message origin authentication) and have not been altered (message authentication).

Message authentication functions can be:

- Hash functions
- Message encryption
- Message authentication codes

Basic Uses of Message Encryption

- Symmetric encryption: confidentiality and authentication
- Public-key encryption using public key: confidentiality
- Public-key encryption using private key: authentication and signature
- Public-key encryption using private key then public key: confidentiality, authentication, signature

Alternative to Basic Uses of Message Encryption

Message Authentication Codes

- MAC: Message Authentication Codes
- $h = MAC_k(m)$

If the received MAC matches the computed MAC, then

- Message has not been altered
- Message comes from the alleged sender
- Sequence number, timestamp, direction: Assurance of proper sequence (to avoid replay)

Integrity Check using MAC

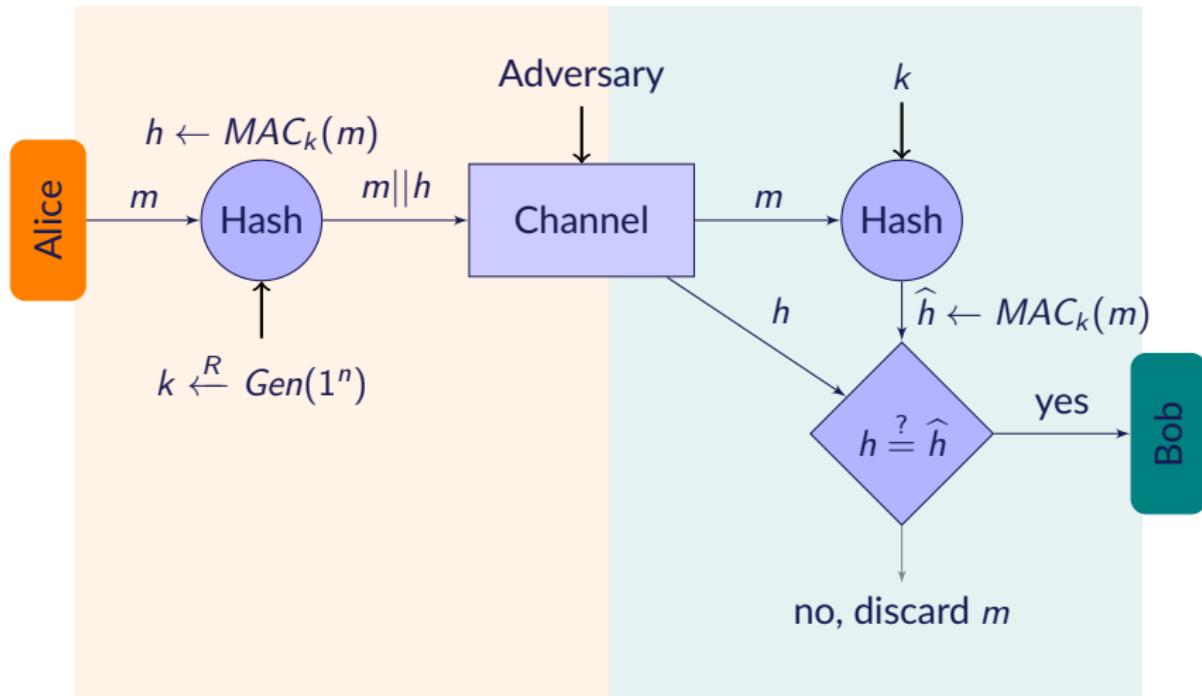
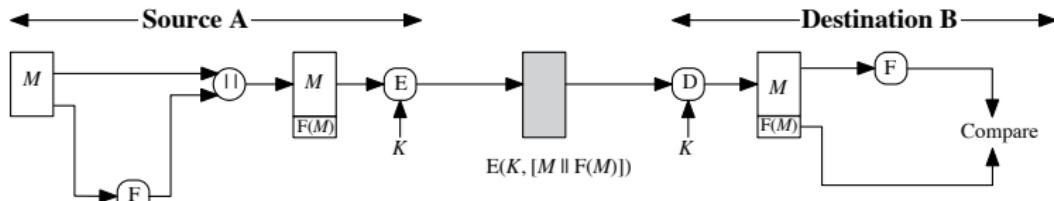
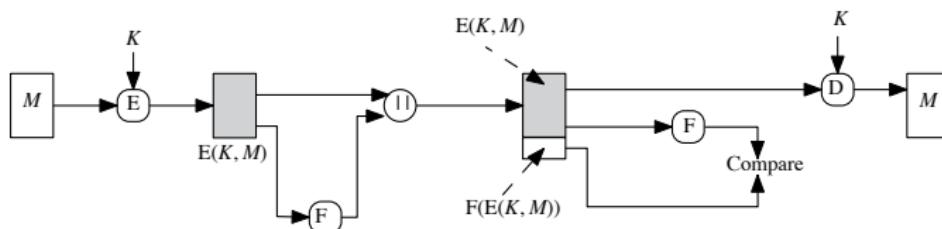


Figure: A simple use of MAC for integrity check.

Internal versus External Error Control



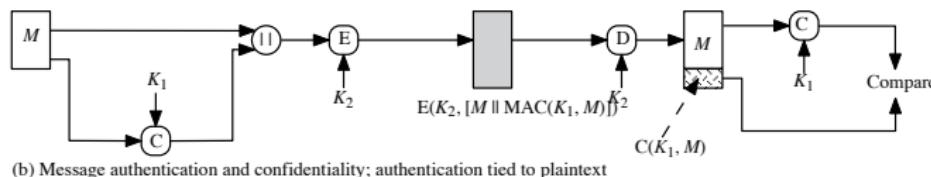
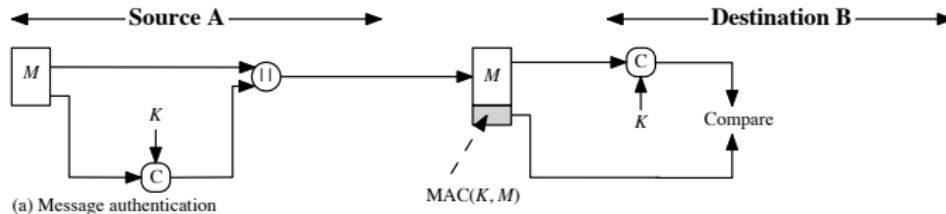
(a) Internal error control



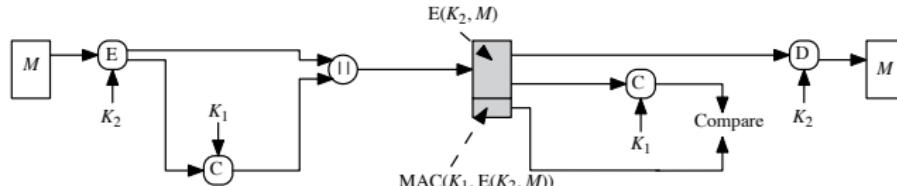
(b) External error control

Figure 12.2 Internal and External Error Control

Simplified Use of MAC



(b) Message authentication and confidentiality; authentication tied to plaintext



(c) Message authentication and confidentiality; authentication tied to ciphertext

Requirements of MAC

- If an opponent observes M and $MAC_k(M)$, it should be computationally infeasible for the opponent to construct a message M' such that $MAC_k(M') = MAC_k(M)$
- $MAC_k(M)$ should be uniformly distributed in the sense that for randomly chosen messages, M and M' , the probability that $MAC_k(M) = MAC_k(M')$ is 2^{-n} , where n is the number of bits in the tag. entropy
- Let M' be equal to some known transformation on M . That is, $M' = f(M)$. For example, f may involve inverting one or more specific bits. In that case, $Prob\{MAC_k(M) = MAC_k(M')\} = 2^{-n}$ little change in message
need to cause high effect on mac

MAC Using Hash Functions HMAC

- Use existing hash functions without modification
- Replaceable hash functions
- No degradation to hash function efficiency
- Use secret keys in a simple way

HMAC

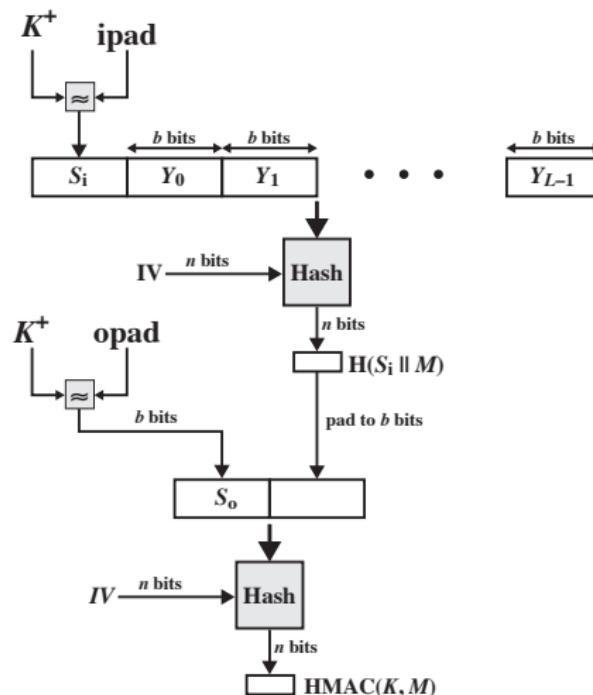


Figure 12.5 HMAC Structure

MAC Using Block Ciphers Data Authentication Algorithm (DAA)

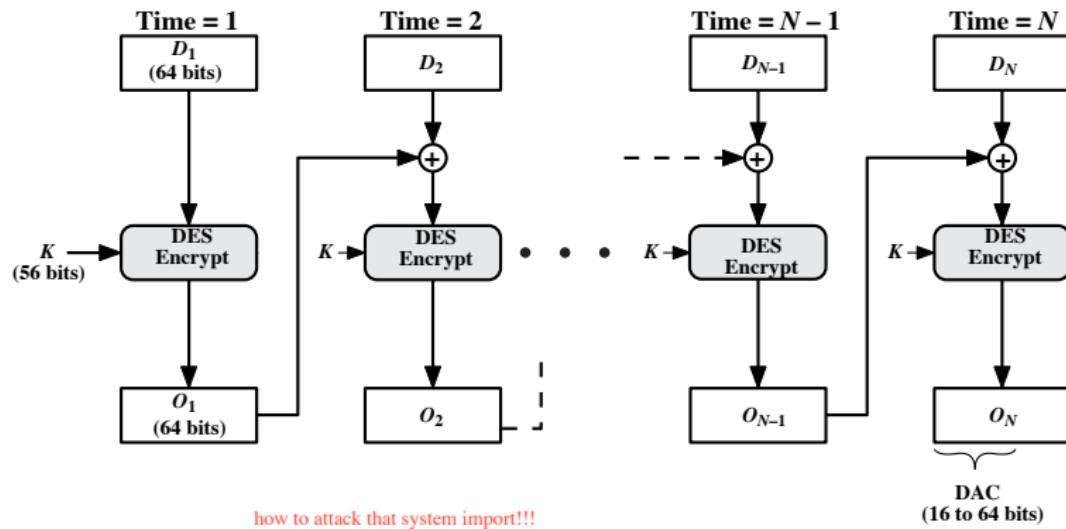


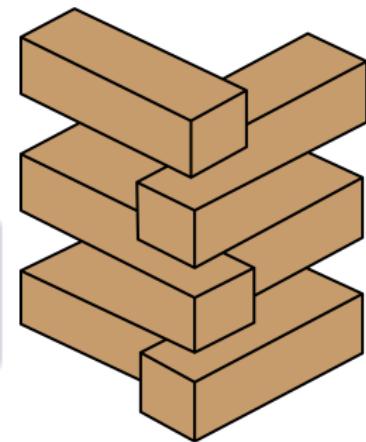
Figure 12.7 Data Authentication Algorithm (FIPS PUB 113)

Practice What You Preach

DAA is CBC where $IV=0$ using DES in FIPS PUB 113 and ANSI X9.17. Assume you know $T = MAC(k, X)$, can you compute the MAC of any other message?

Hint

Think about CBC and \oplus 'ing same message twice

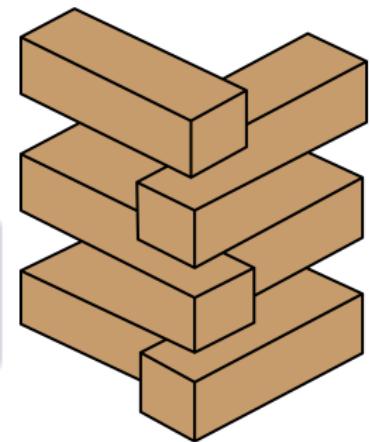


Practice What You Preach

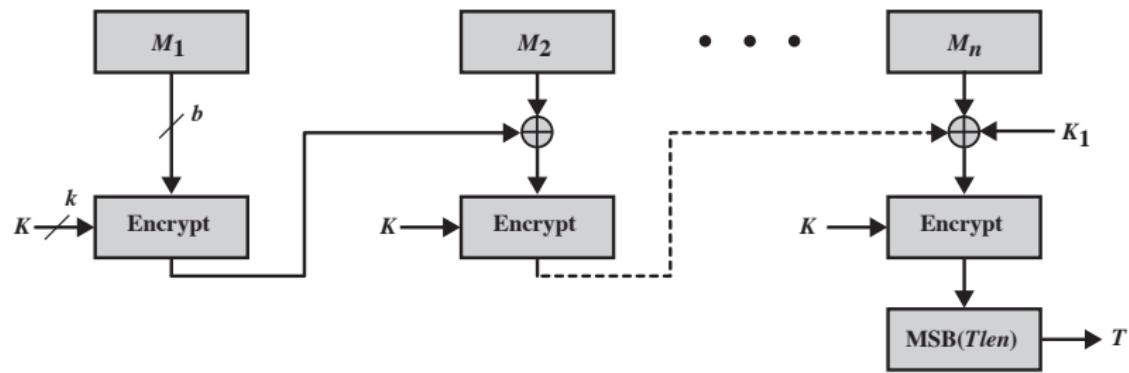
DAA is CBC where $IV=0$ using DES in FIPS PUB 113 and ANSI X9.17. Assume you know $T = MAC(k, X)$, can you compute the MAC of any other message?

Hint

Think about CBC and \oplus 'ing same message twice
 $X||(X \oplus T)$

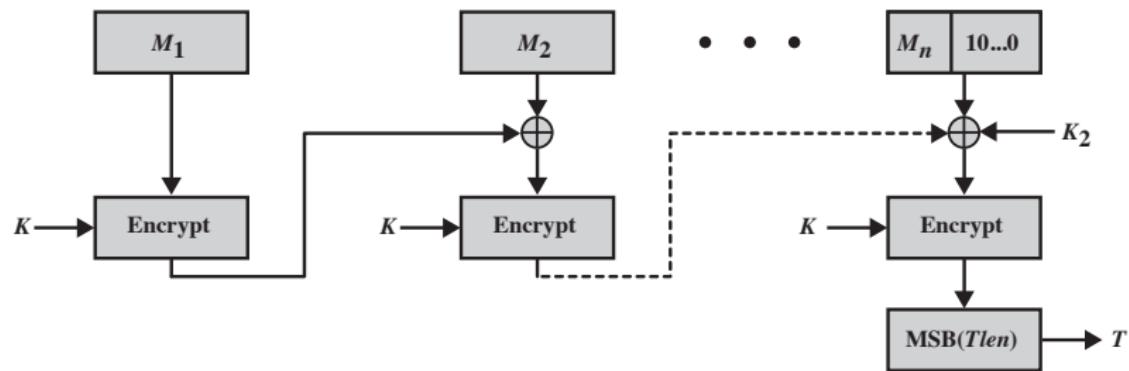


MAC Using Block Ciphers Cipher-based Message Authentication Code (CMAC)



(a) Message length is integer multiple of block size

MAC Using Block Ciphers Cipher-based Message Authentication Code (CMAC)



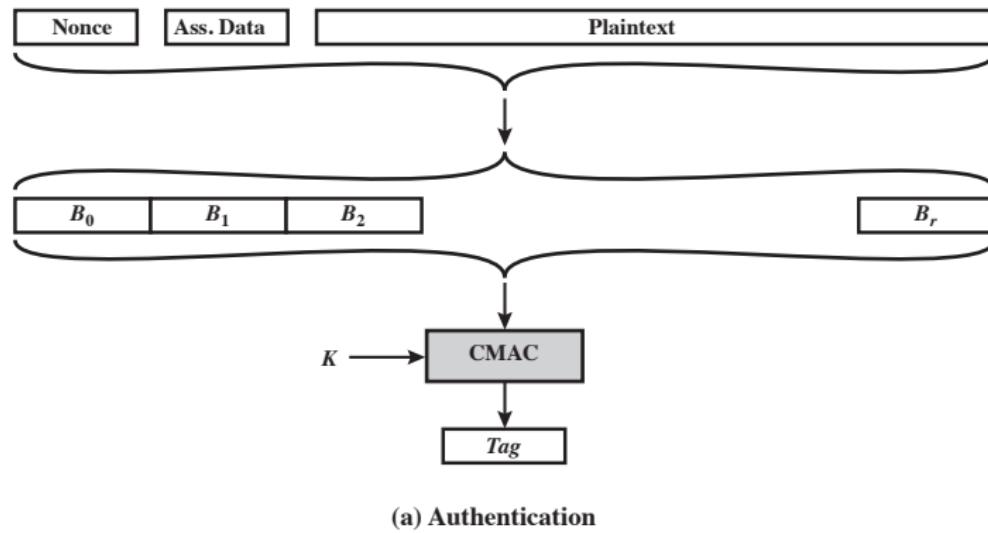
(b) Message length is not integer multiple of block size

Figure 12.8 Cipher-Based Message Authentication Code (CMAC)

Authenticated Encryption Provide Both Confidentiality and Integrity

- **HtE: Hash-then-encrypt:** First compute $h = H(m)$. Then, $Enc_k(m||h)$ (e.g., WEP)
- **MtE: MAC-then-encrypt:** Use two keys. First authenticate m by $h = MAC_{k_1}(m)$. Then, $Enc_{k_2}(m||h)$ (e.g., SSL/TLS)
- **EtM: Encrypt-then-MAC:** Use two keys. First $c = Enc_{k_2}(m)$. Then authenticate $MAC_{k_1}(c)$ (e.g., IPsec).
- **E&M: Encrypt-and-MAC:** Use two keys. $c = Enc_{k_2}(m)$. Authenticate $h = MAC_{k_1}(m)$. (e.g., SSH).

CCM: Counter with Cipher Block Chaining-Message Authentication Code



CCM: Counter with Cipher Block Chaining-Message Authentication Code

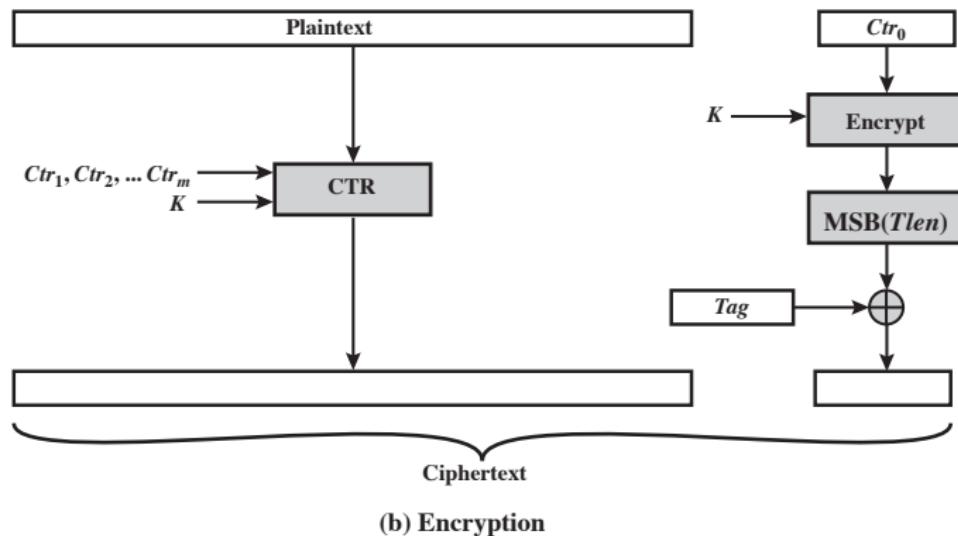
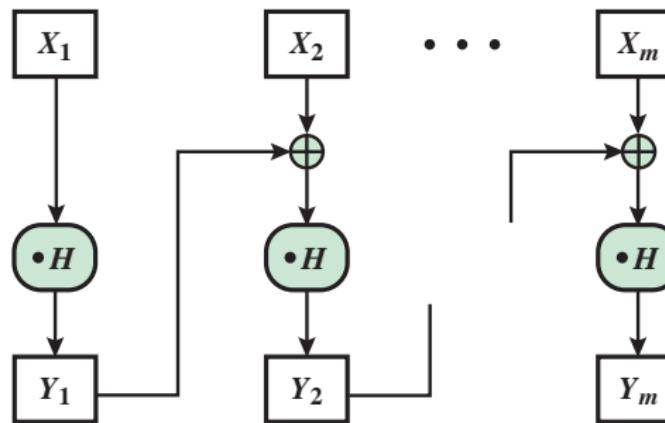


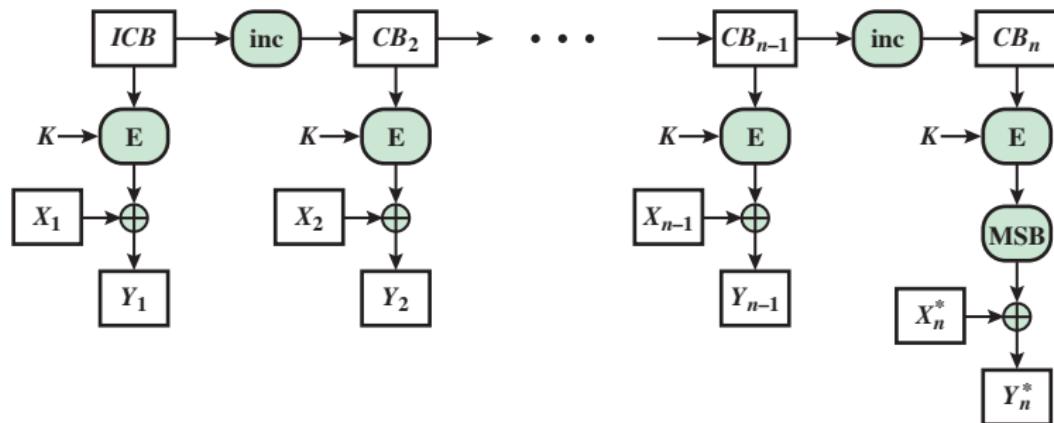
Figure 12.9 Counter with Cipher Block Chaining-Message Authentication Code (CCM)

GCM: Galois/Counter Mode



$$(a) \text{GHASH}_H(X_1 \parallel X_2 \parallel \dots \parallel X_m) = Y_m$$

GCM: Galois/Counter Mode



$$(b) \text{GCTR}_K(ICB, X_1 \parallel X_2 \parallel \dots \parallel X_n^*) = Y_n^*$$

Figure 12.10 GCM Authentication and Encryption Functions

Summary

Reference: Chapter 12

Today, we learned

- Keyed-hash functions (MAC)

Lecture 10: Signatures

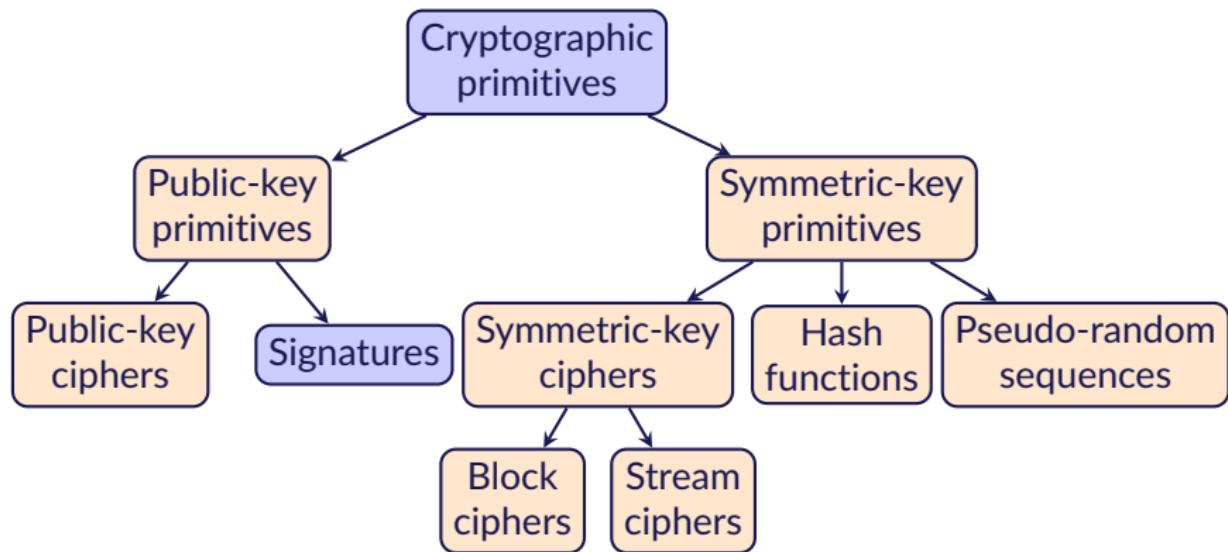
Signatures

Objectives of Lecture

At the end of this lecture, you will be able to

- define signatures

Overview of Cryptographic Primitives



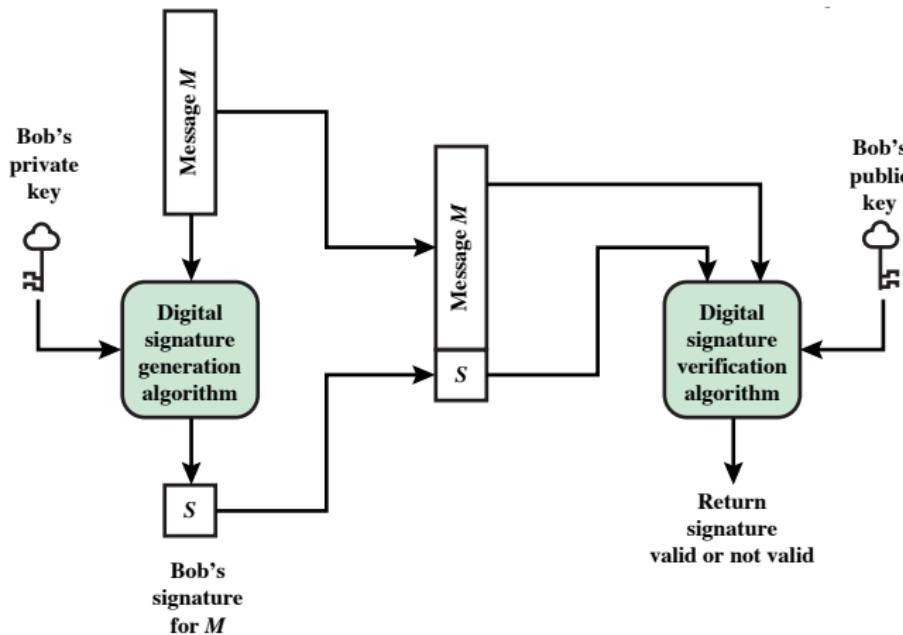
Digital Signatures

- MAC does not address issues of lack of trust
- Digital signatures provide the ability to:
 - verify author, date & time of signature
 - authenticate message contents
 - be verified by third parties to resolve disputes
- Include authentication function with additional capabilities

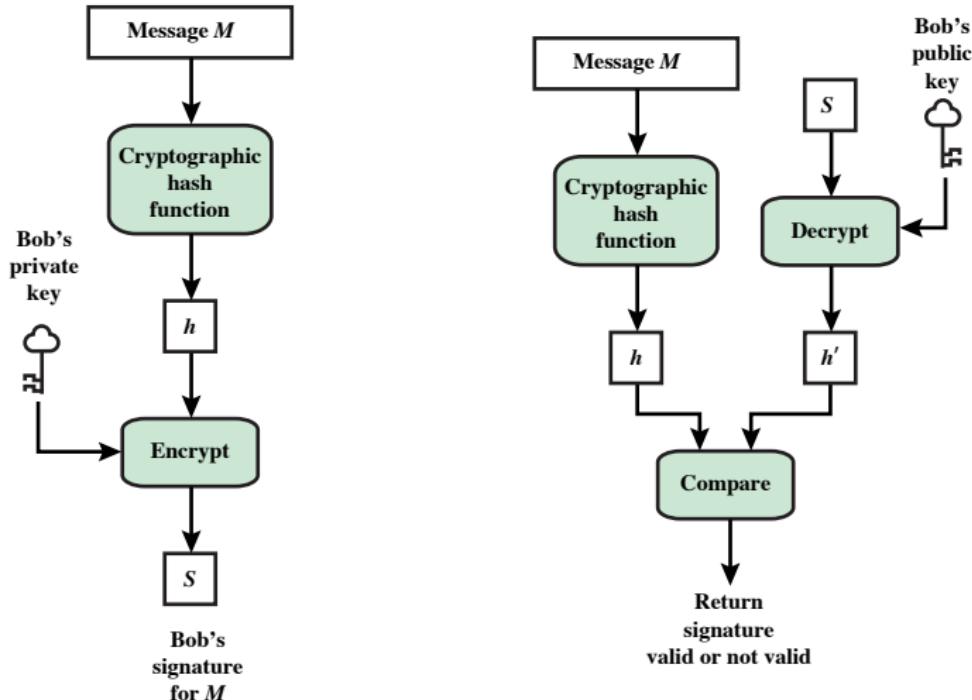
Non-repudiation

- A service that provides proof of the integrity and origin of data.
- An authentication that can be **asserted to be genuine** with high assurance.

Generic Model of Digital Signatures



Elements of Digital Signatures



Properties of a Digital Signature

The digital signature must have the following properties

- It must verify the author and the date and time of the signature.
- It must authenticate the contents at the time of the signature.
- It must be verifiable by third parties, to resolve disputes.

Requirements of Digital Signatures

- The signature must be a bit pattern that depends on the message being signed.
- The signature must use some information unique to the sender to prevent both forgery and denial.
- It must be relatively easy to produce the digital signature.
- It must be relatively easy to recognize and verify the digital signature.
- It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.
- It must be practical to retain a copy of the digital signature in storage.

Examples of Digital Signatures

- Elgamal signature scheme
- Schnorr digital signature scheme
- Digital signature standard (DSS)

Summary

Reference: Chapters 11, 12

Today, we learned

- Hash and keyed-hash functions (MAC)
- Signatures

Lecture 11: Key Management and Distribution

Key Management and Distribution

Objectives of Lecture

At the end of this lecture, you will be able to

- distribute keys securely in a network and discuss centralized versus decentralized key distribution protocols

Key Distribution

- **symmetric schemes** require both parties to share a common secret key
- **public key schemes** require parties to acquire valid public keys
- Problem: how to securely distribute keys, while protecting them from others
- Frequent key updates desirable

Key Distribution Alternatives

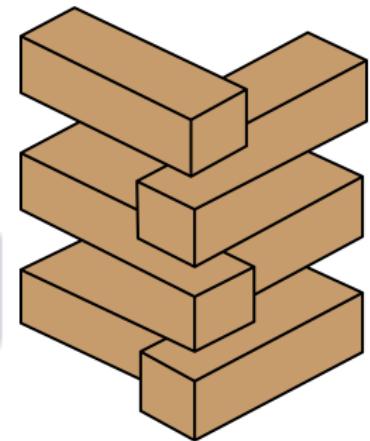
Given parties Alice and Bob have various key distribution alternatives:

- Manual delivery (good for link encryption)
 - Alice can select key and **physically deliver** to Bob
 - **Third party** can select & deliver key to Alice and Bob
- Both link and end-to-end (e2e) encryption
 - If Alice and Bob have communicated **previously** can use previous key to encrypt a new key
- End-to-end (e2e) encryption
 - If Alice and Bob have secure communications with a **third party** Carol, Carol can relay key between Alice and Bob

Practice What You Preach

There are N nodes in a network employing IPv4, each pair of nodes require a secret key. How many are keys are there?

Solution

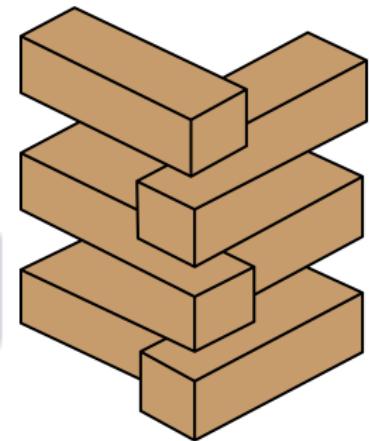


Practice What You Preach

There are N nodes in a network employing IPv4, each pair of nodes require a secret key. How many are keys are there?

Solution

$$\frac{N(N-1)}{2}$$



Key Distribution Scenario: Centralized

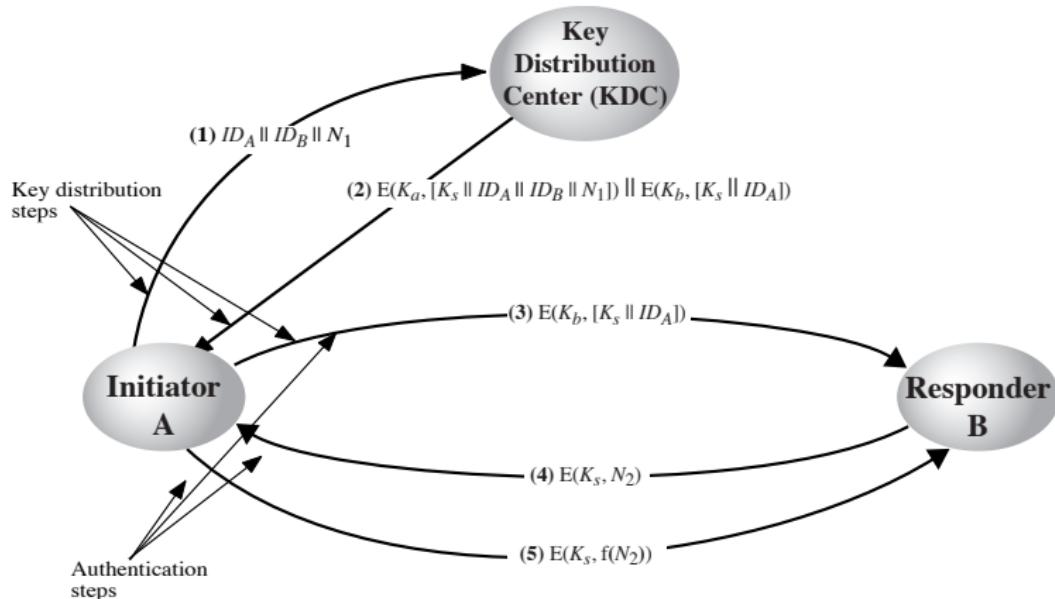


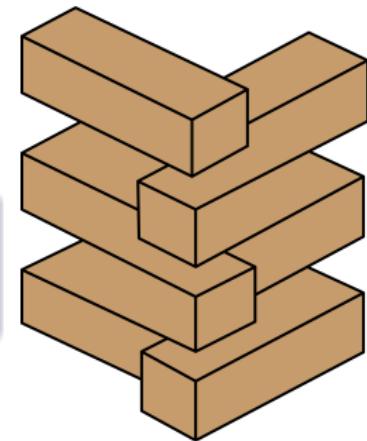
Figure 14.3 Key Distribution Scenario

Practice What You Preach

Do you see any problems in centralized key distribution scenario.

Solution

Needham-Schroeder Protocol

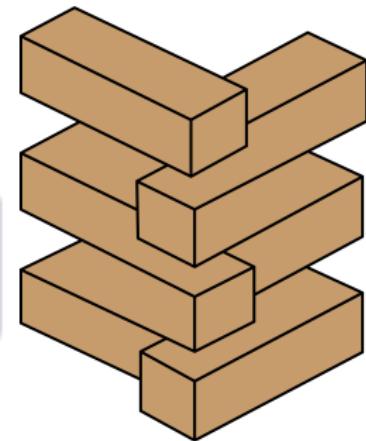


Practice What You Preach

Do you see any problems in centralized key distribution scenario.

Solution

Needham-Schroeder Protocol
Replay (3)



Key Distribution Scenario: Decentralized

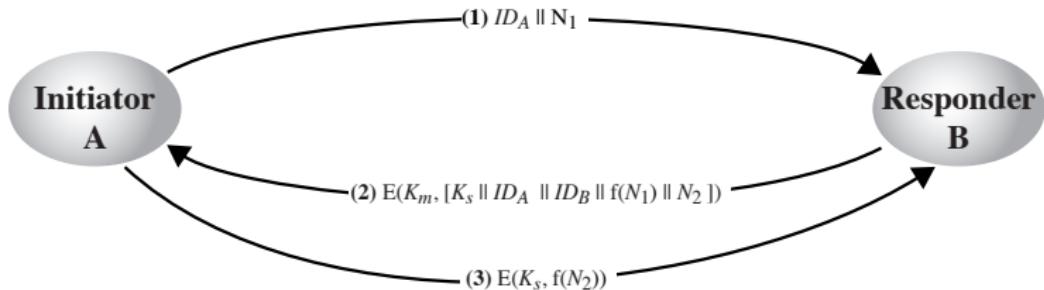


Figure 14.5 Decentralized Key Distribution

Control Vectors

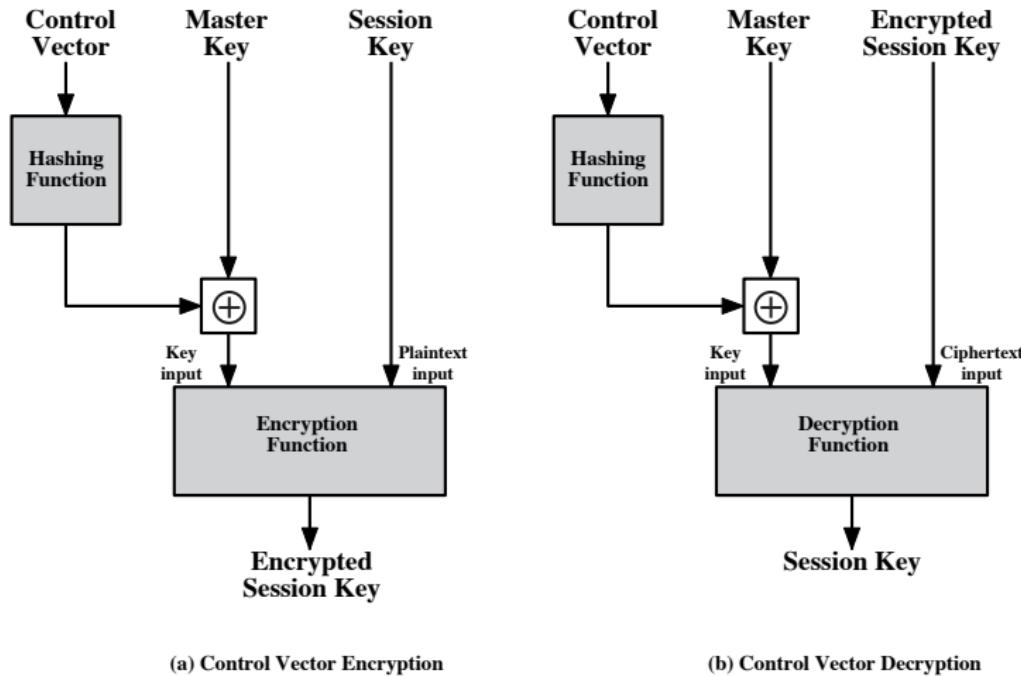


Figure 14.6 Control Vector Encryption and Decryption

Simple Key Distribution using Public-key Schemes

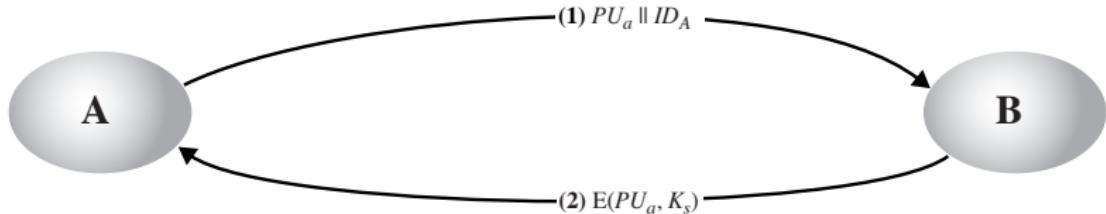


Figure 14.7 Simple Use of Public-Key Encryption to Establish a Session Key

Key Distribution with Confidentiality and Authentication

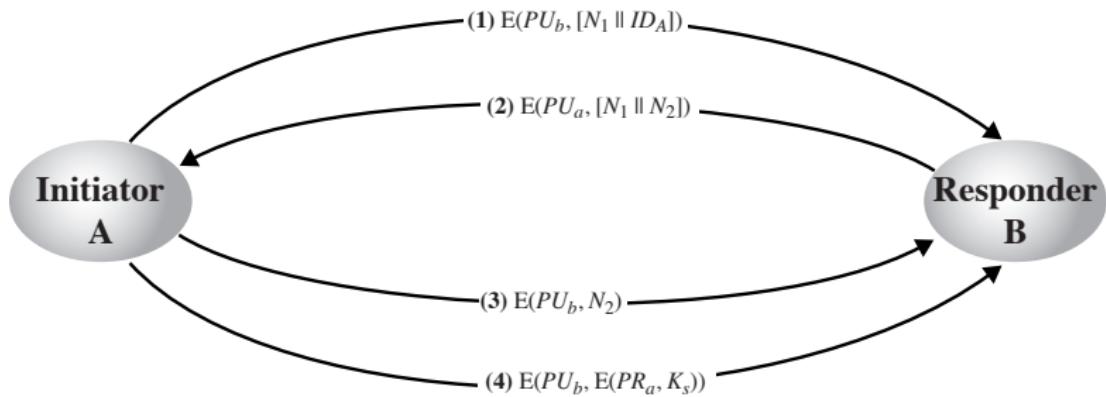


Figure 14.8 Public-Key Distribution of Secret Keys

Distribution of the Public Keys

- Public announcement
- Publicly available directory
- **Public-key Authority**
- **Public-key Certificates**

Using Public-key Authority for Distribution

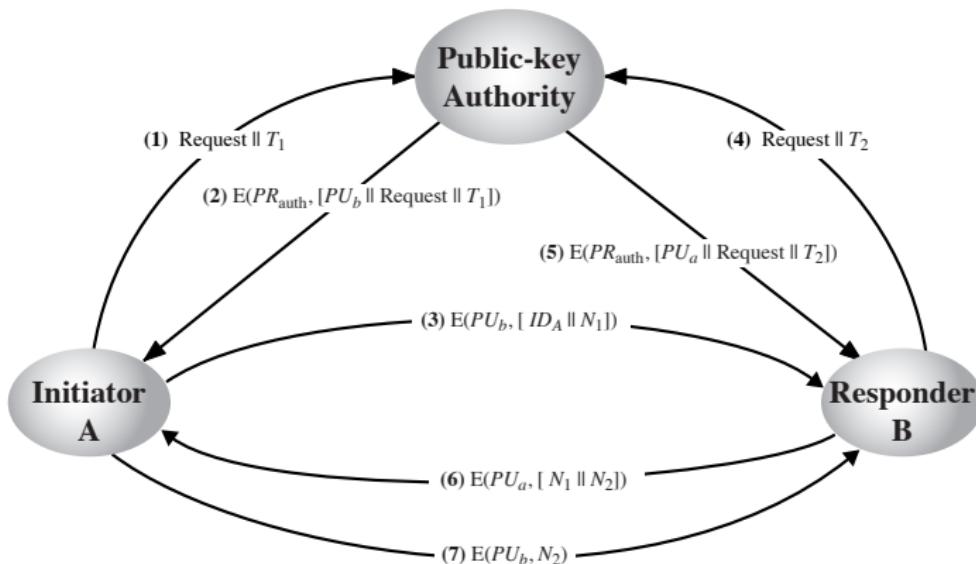
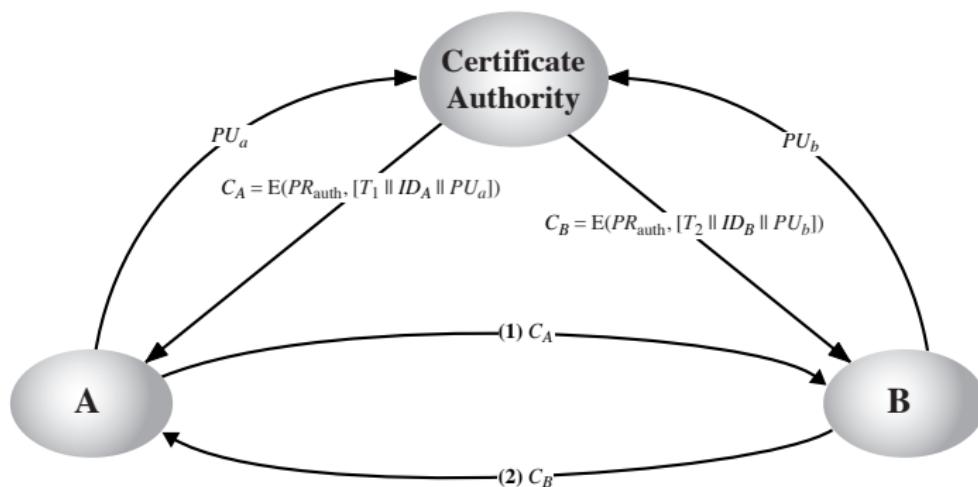
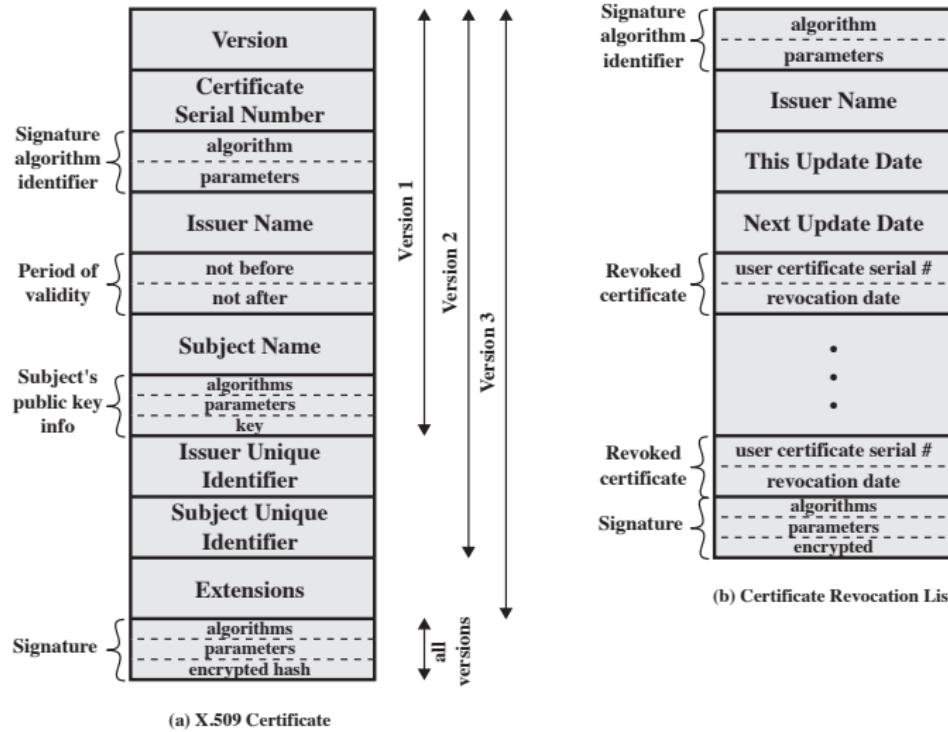


Figure 14.11 Public-Key Distribution Scenario

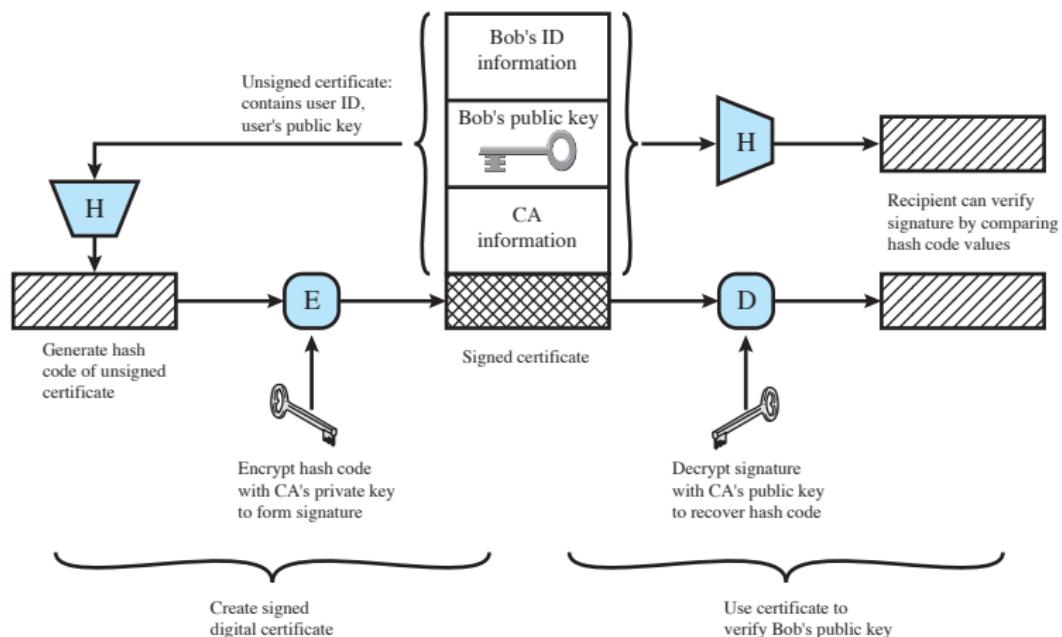
Using Certificate Authority for Distribution



ITU-T X.509 Certificates



Public-key Certificate Use: ITU-T X.509 Certificates



Public-key Infrastructure: PKI

IETF RFC 2822 Definition of PKI

The set of h/w, s/w, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates based on public-key schemes.

IETF X.509 Public key infrastructure (PKIX)

Summary

Reference: Chapter 14

Today, we learned

- Public-key Infrastructure

Lecture 12: User Authentication

User Authentication

Objectives of Lecture

At the end of this lecture, you will be able to

- define authentication protocols

Authentication: Identification and Verification

The process of verifying an identity claimed by or for a system entity. An authentication process consists of two steps:

Identification step

Presenting an identifier to the security system. (Identifiers should be assigned carefully, because authenticated identities are the basis for other security services, such as access control service.)

Verification step

Presenting or generating authentication information that corroborates the binding between the entity and the identifier.

Means of Authentication

- **Something the individual knows:** Examples include a password, a personal identification number (PIN), or answers to a prearranged set of questions.
- **Something the individual possesses:** Examples include cryptographic keys, electronic keycards, smart cards, and physical keys. This type of authenticator is referred to as a token.
- **Something the individual is (static biometrics):** Examples include recognition by fingerprint, retina, and face.
- **Something the individual does (dynamic biometrics):** Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm.

Replay Attacks

where a valid signed message is copied and later resent

- simple replay
- repetition that can be logged
- repetition that cannot be detected
- backward replay without modification

countermeasures include

- use of sequence numbers (generally impractical)
- timestamps (needs synchronized clocks)
- (do not forget) direction (e.g., LTE)
- challenge/response (using unique nonce)

Challenge Response Authentication

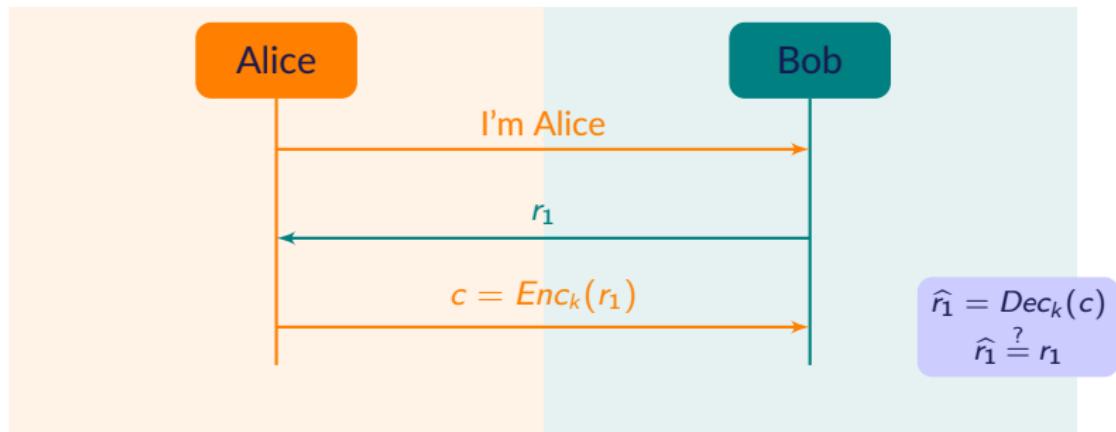


Figure: Challenge response authentication

Efficient Mutual Challenge Response Authentication

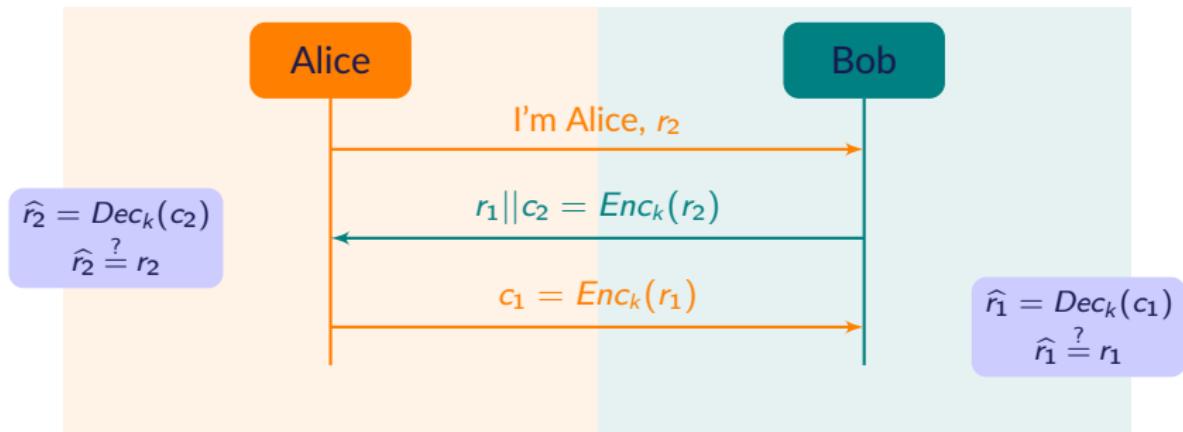
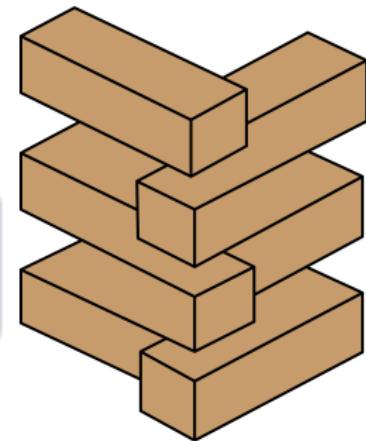


Figure: Efficient mutual challenge response authentication

Practice What You Preach

Any problems in efficient mutual challenge-response authentication?

Solution



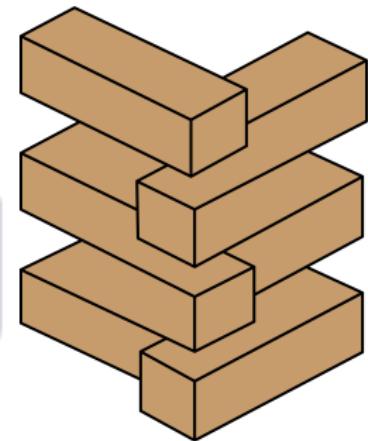
Practice What You Preach

Any problems in efficient mutual challenge-response authentication?

Solution

Reflection attack

Alice reflects r_1 to Bob.



Kerberos

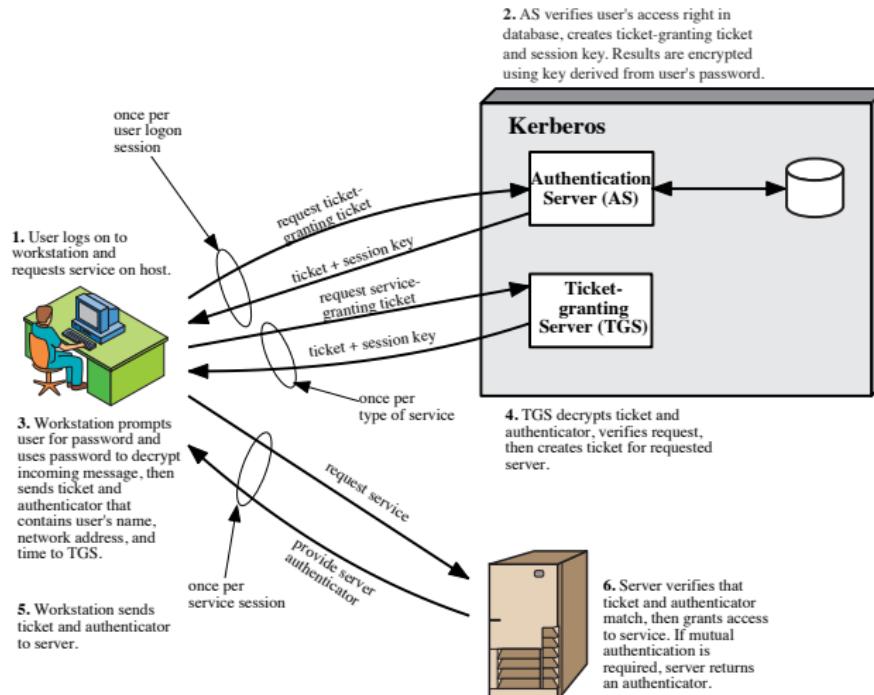


Figure 15.1 Overview of Kerberos

Summary

Reference: Chapter 15

Today, we learned

- Authenticate not only for accounting

Lecture 13: Transport Layer Security

Transport Layer Security

Objectives of Lecture

At the end of this lecture, you will be able to

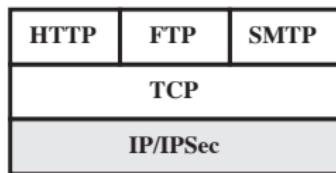
- define secure socket layer and transport layer security

Web Security

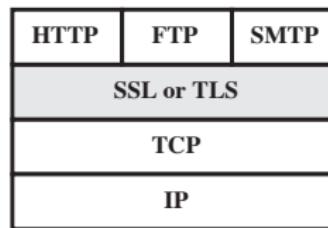
Many threats

- integrity (modification of data, trojans, etc.)
- confidentiality (eavesdropping, theft, etc.)
- denial of service (DoS) (flooding bogus messages etc)
- authentication (impersonation, forgery)

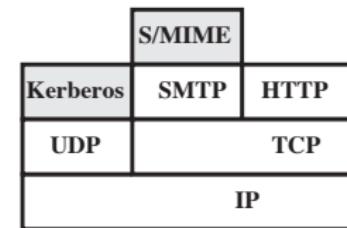
Securing TCP/IP Stack



(a) Network Level



(b) Transport Level



(c) Application Level

Secure Socket Layer (SSL)

SSL is designed to make use of TCP to provide a reliable end-to-end secure service (Netscape)

Connection

A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session.

Session

An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

SSL Protocol Stack

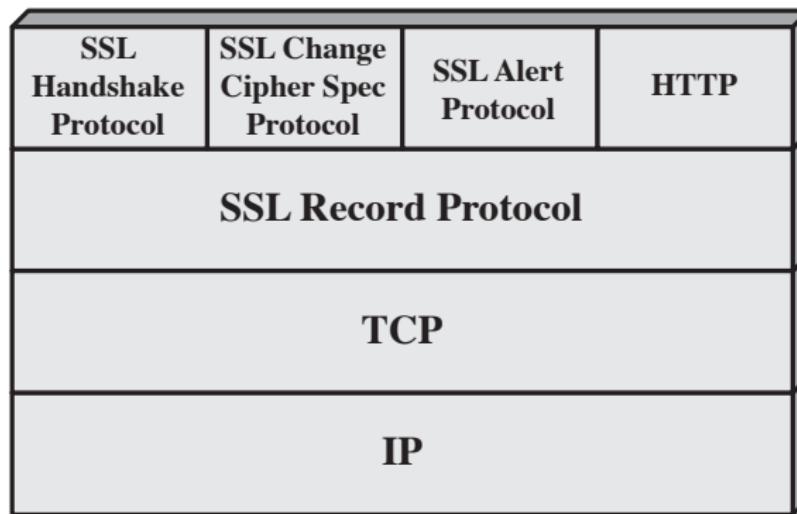


Figure 16.2 SSL Protocol Stack

SSL Record Protocol

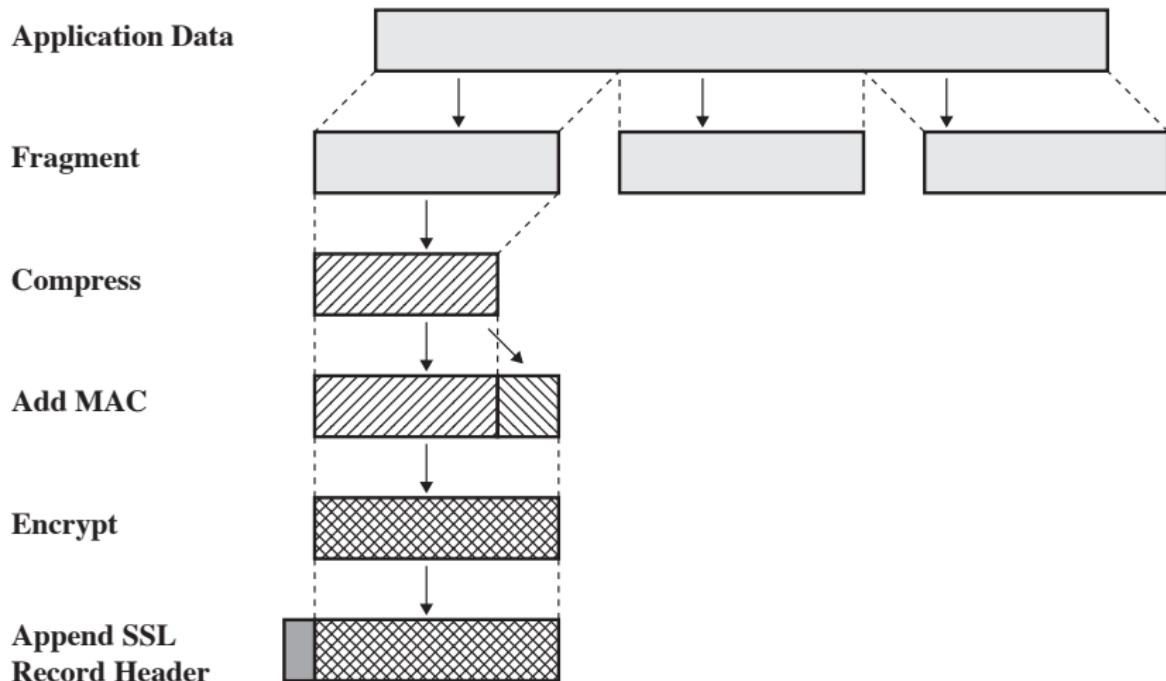
Confidentiality

The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads.

Message Integrity

The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC).

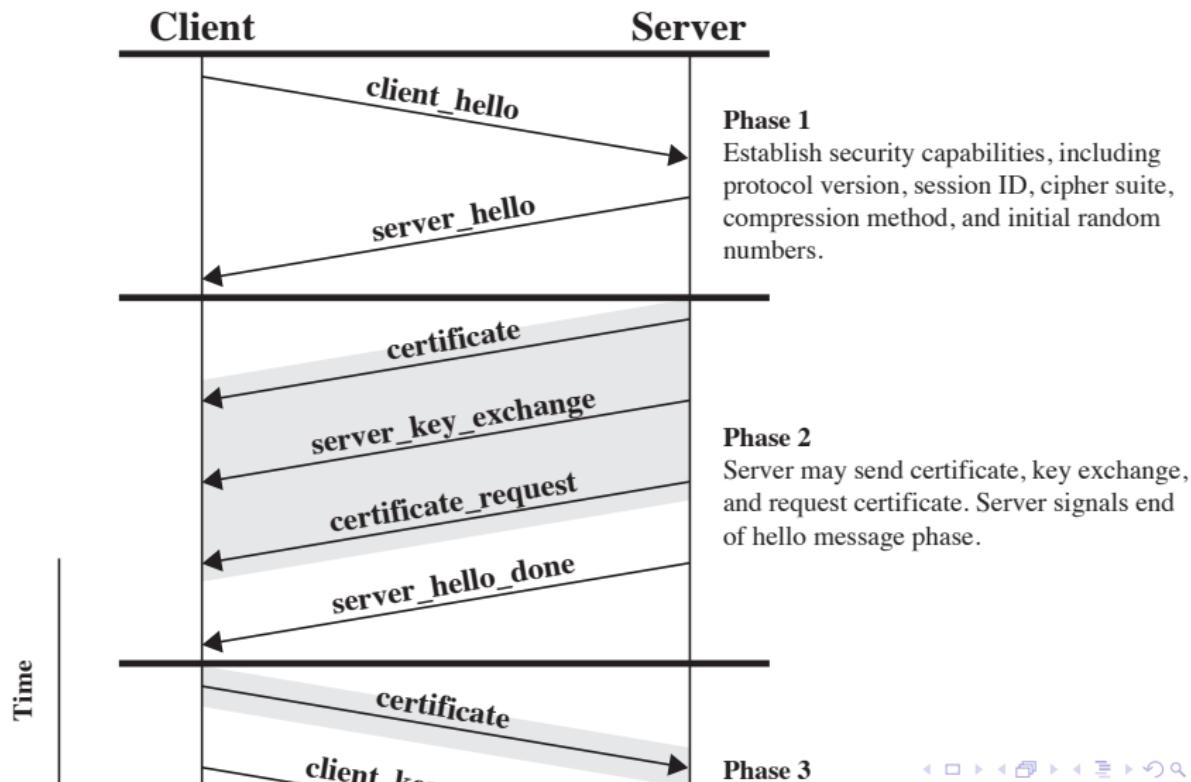
SSL Record Protocol



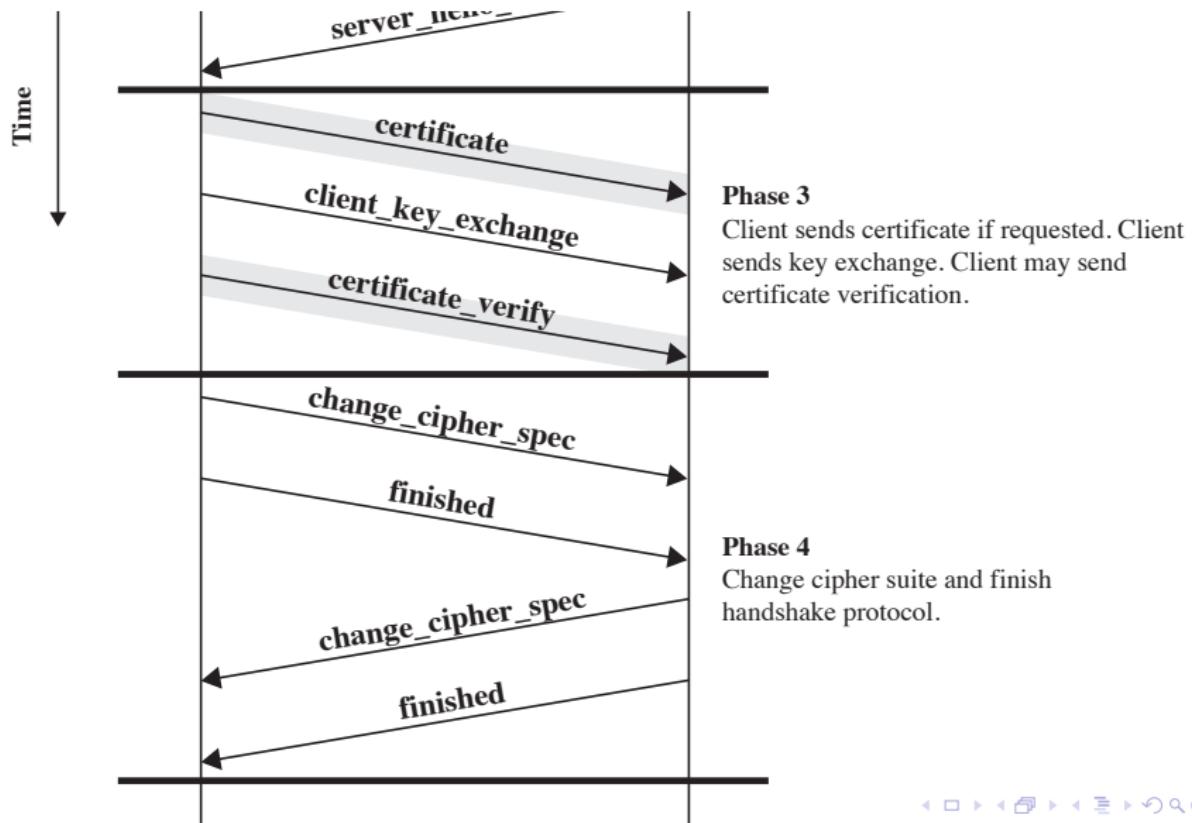
SSL Handshake Protocol: Phases

- **Establish security capabilities:** to initiate a logical connection and to establish the security capabilities associated with connection. Client initiates by *client hello*
- **Server authentication and key exchange:** Server sends its X.509 certificate.
- **Client authentication and key exchange:** Upon receipt of the *server done* message, client verifies server certificate (if required). If server requests the client certificate, client sends its certificate.
- **Finish:** completes setting up secure connection

SSL Handshake Protocol: Phase 1 and 2



SSL Handshake Protocol: Phase 3 and 4



Transport Layer Security

- TLS is the IETF's Internet Standard version of SSL.
- Version number: major:3, minor:3
- TLS uses HMAC
- A little more differences in padding

Summary

Reference: Chapter 16

Today, we learned

- TLS, SSL

Lecture 14: WiFi Security

WiFi Security

Objectives of Lecture

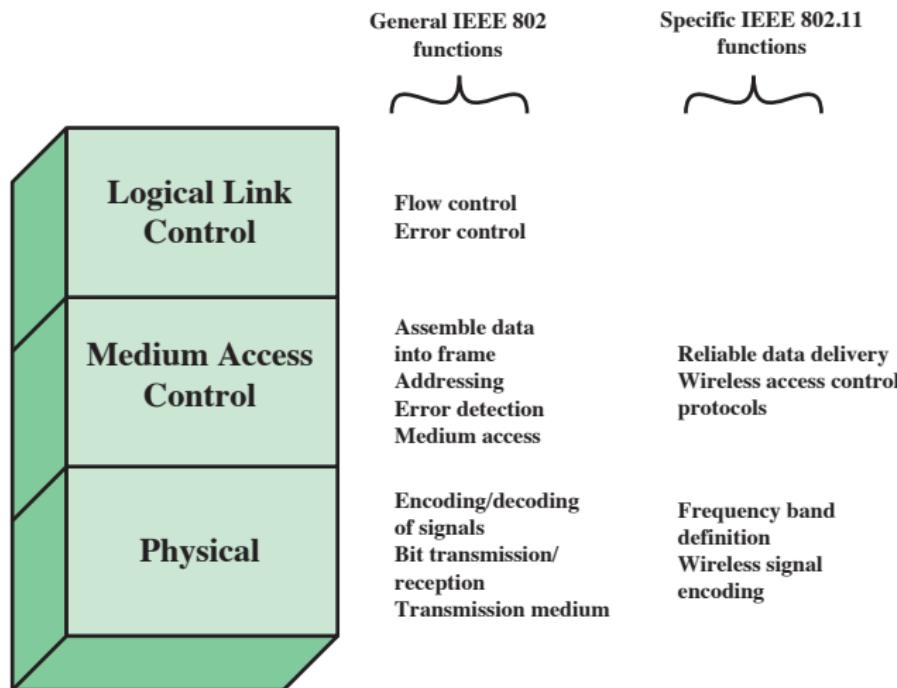
At the end of this lecture, you will be able to

- define the security architecture of IEEE 802.11

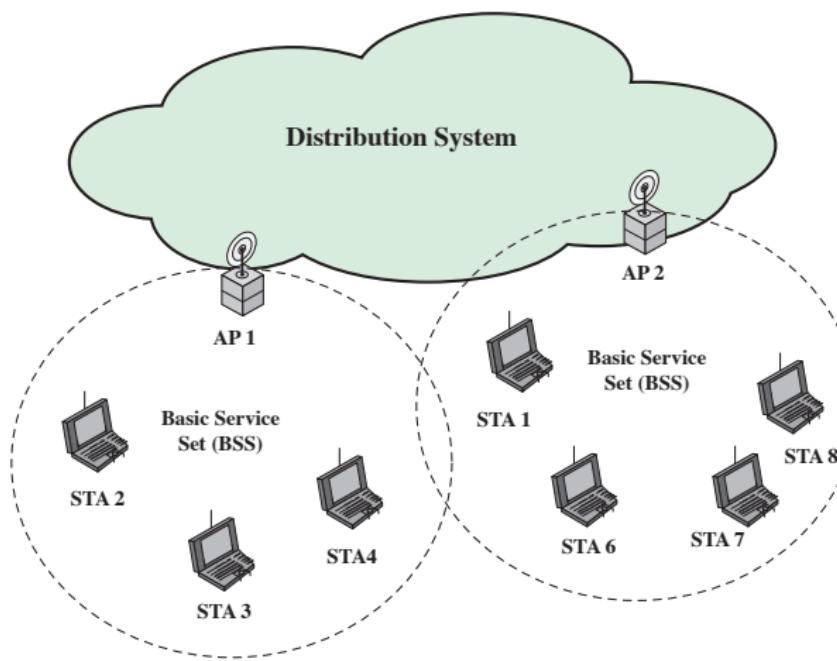
IEEE 802.11 Overview

- IEEE 802.11 is a standard for wireless LANs.
- Inter-operable standards compliant implementations are referred to as Wi-Fi
- IEEE 802.11i specifies security standards for IEEE 802.11 LANs, including authentication, data integrity, data confidentiality, and key management.
- Inter-operable implementations are also referred to as Wi-Fi Protected Access (WPA).

IEEE 802.11 Protocol Architecture



IEEE 802.11 Network Components



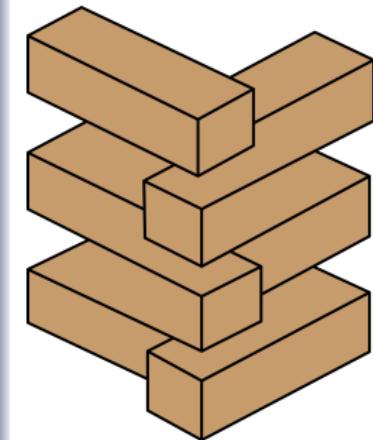
IEEE 802.11 Services

Service	Provider	Used to support
Association	Distribution system	MSDU delivery
Authentication	Station	LAN access and security
Deauthentication	Station	LAN access and security
Disassociation	Distribution system	MSDU delivery
Distribution	Distribution system	MSDU delivery
Integration	Distribution system	MSDU delivery
MSDU delivery	Station	MSDU delivery
Privacy	Station	LAN access and security
Reassociation	Distribution system	MSDU delivery

Practice What You Preach

What are the differences in wireless networks compared to wired networks in terms of security?

Brainstorm

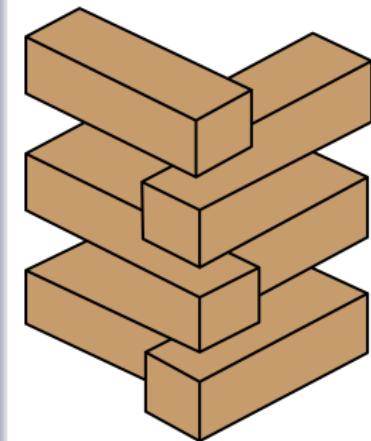


Practice What You Preach

What are the differences in wireless networks compared to wired networks in terms of security?

Brainstorm

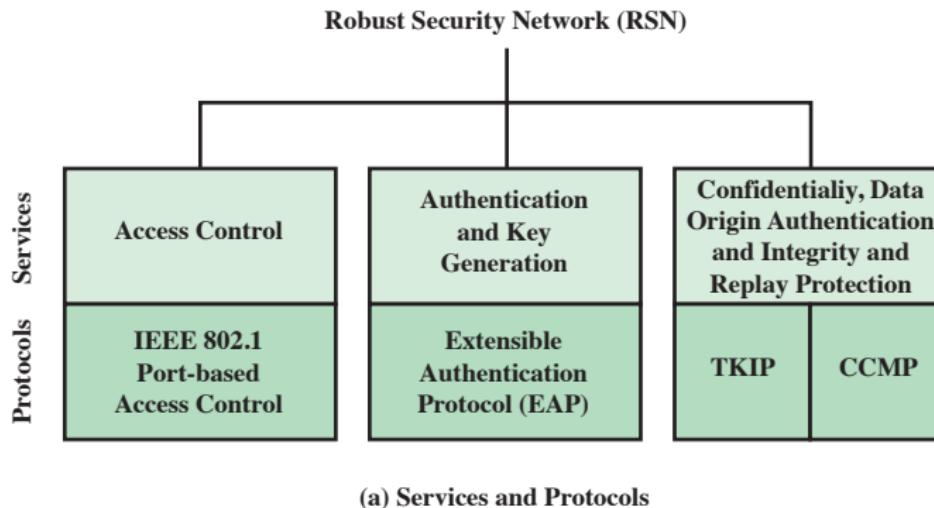
- All of threats in wired networks: Masquerade, eavesdropping, authorization violation,...
- Anybody can listen to the traffic
- Anybody within range can access the network
- Anybody can inject traffic
- Anybody can interfere
- Range is affected by propagation conditions, geography
- Range depends also on receiver (sensitivity, antenna gain, etc.)
- Mobility?



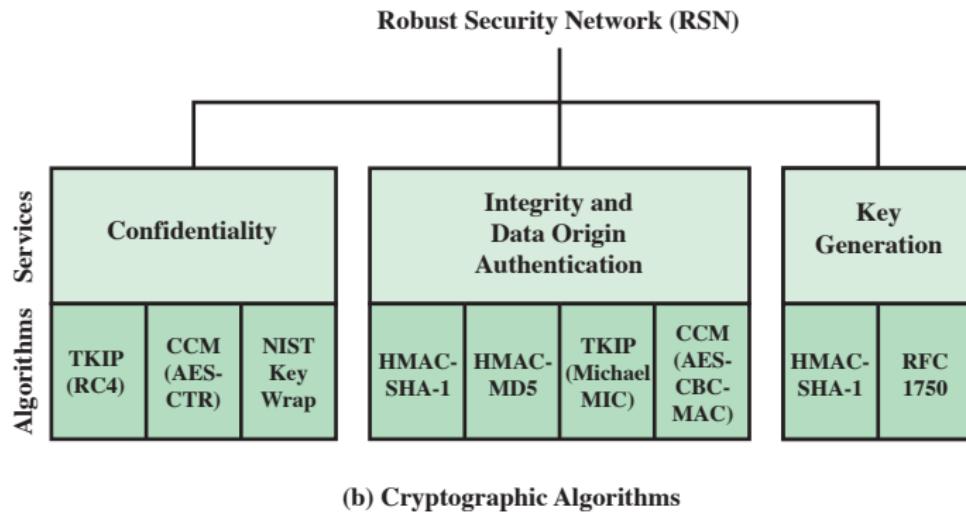
IEEE 802.11 Security Overview

- Original 802.11 specification had security features: Wired Equivalent Privacy (WEP) algorithm
- 802.11i task group developed capabilities to address WLAN security issues
 - Wi-Fi Alliance Wi-Fi Protected Access (WPA)
 - Final 802.11i Robust Security Network (RSN)

802.11i RSN Services and Protocols



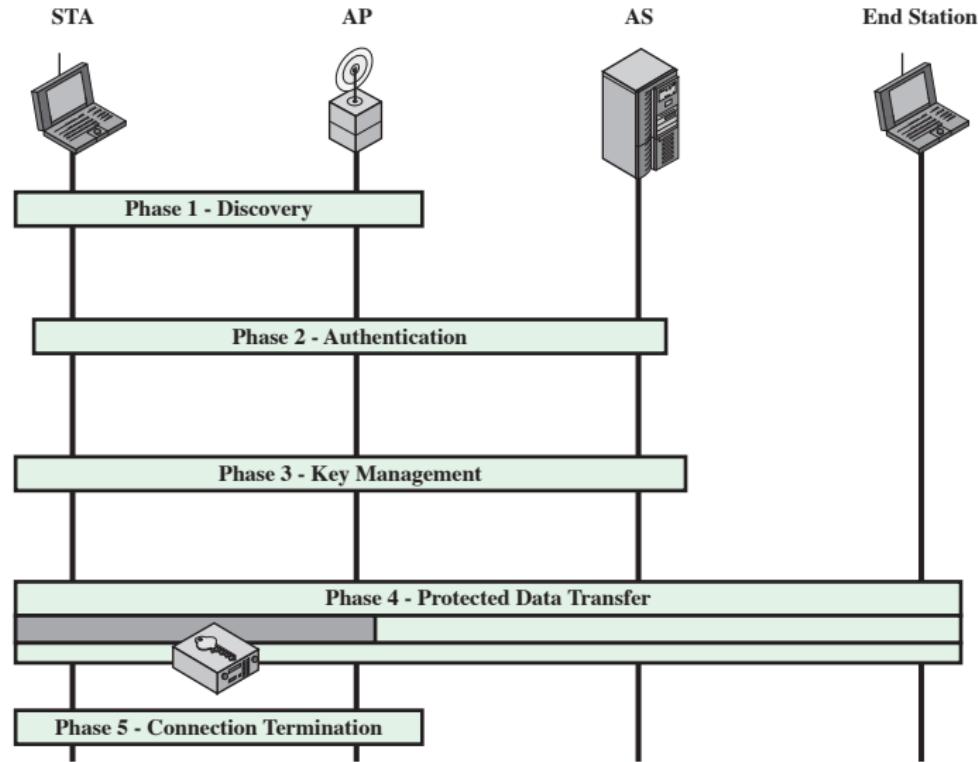
802.11i RSN Services and Protocols



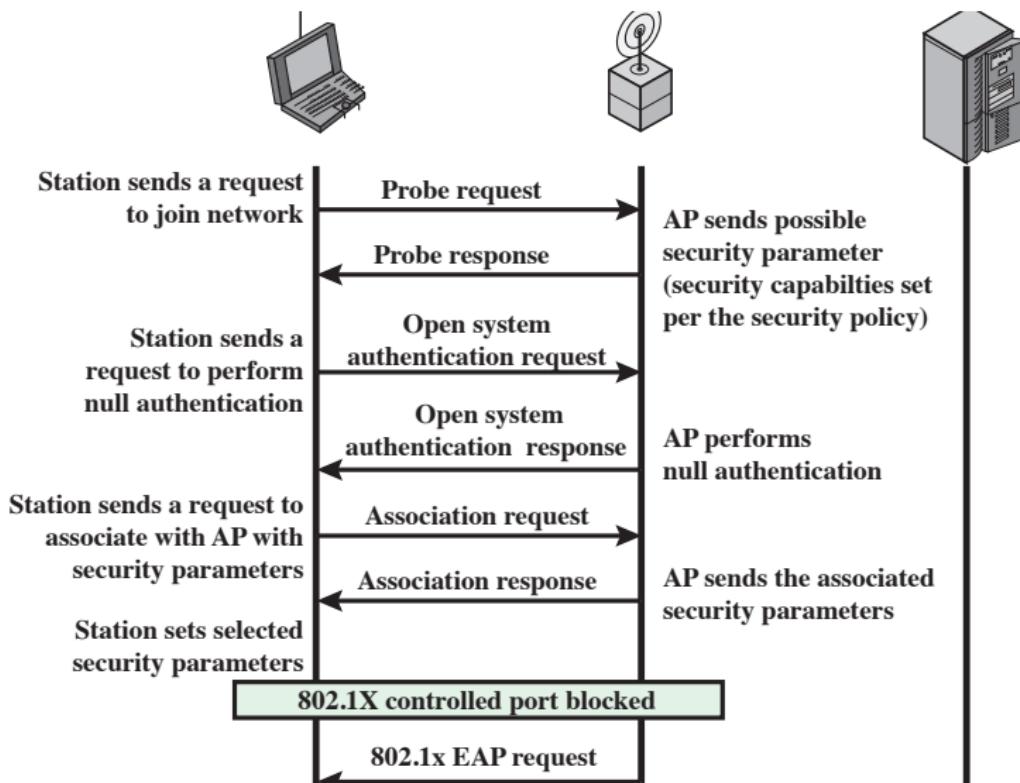
CBC-MAC = Cipher Block Block Chaining Message Authentication Code (MAC)
CCM = Counter Mode with Cipher Block Chaining Message Authentication Code
CCMP = Counter Mode with Cipher Block Chaining MAC Protocol
TKIP = Temporal Key Integrity Protocol

Figure 17.4 Elements of IEEE 802.11i

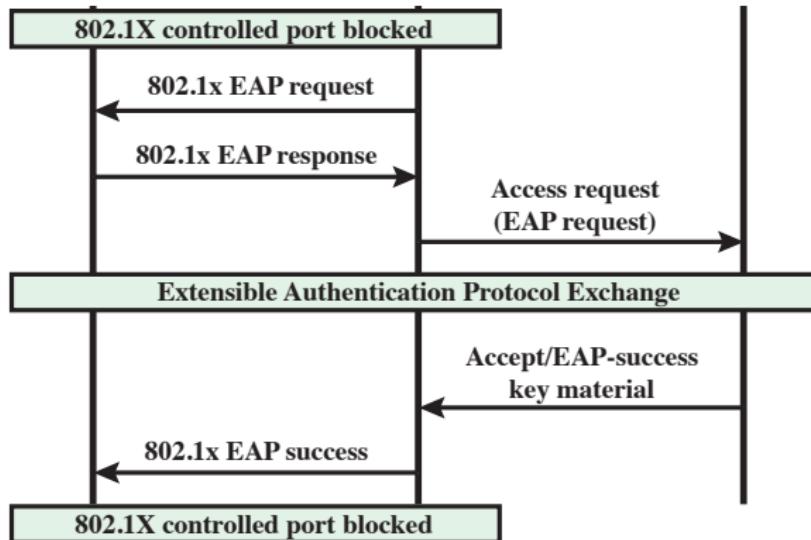
802.11i RSN Phases of Operation



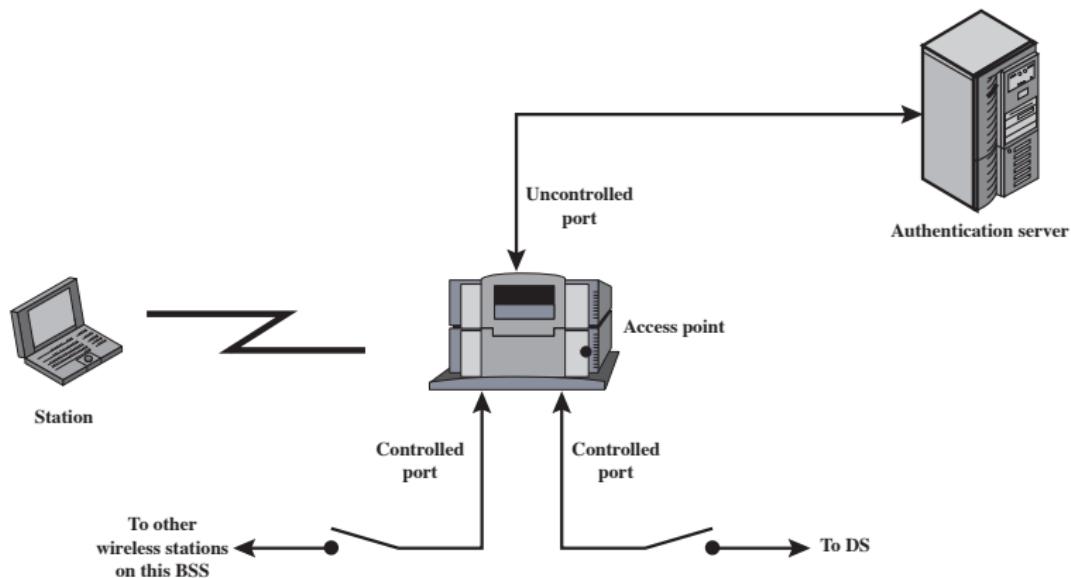
802.11i RSN Discovery and Authentication Phases



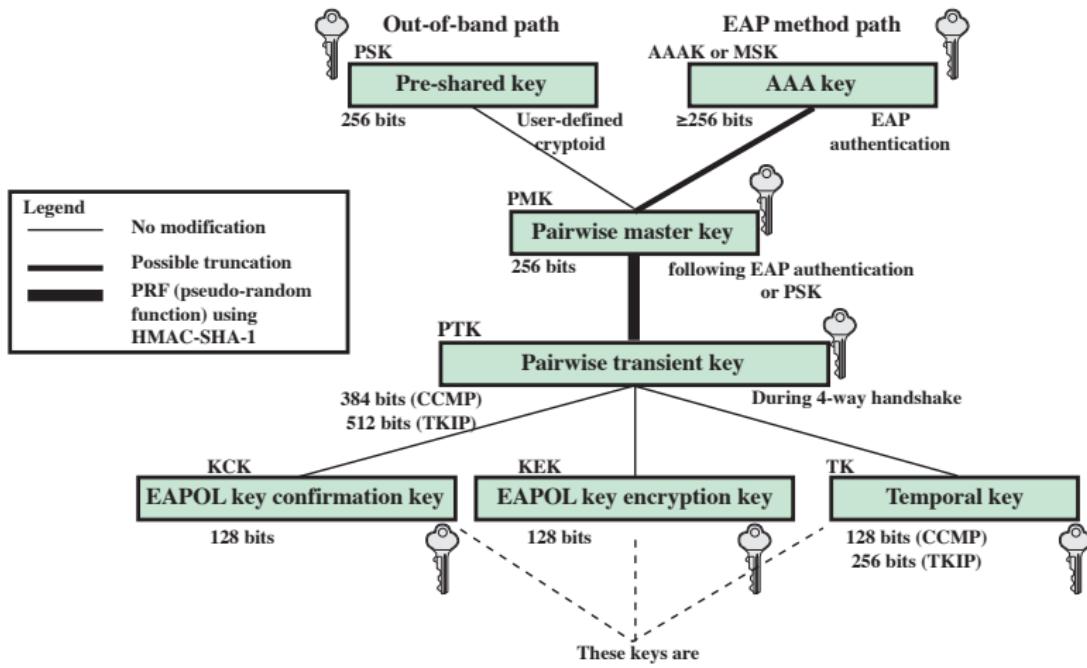
802.11i RSN Discovery and Authentication Phases



IEEE 802.1X Access Control Approach

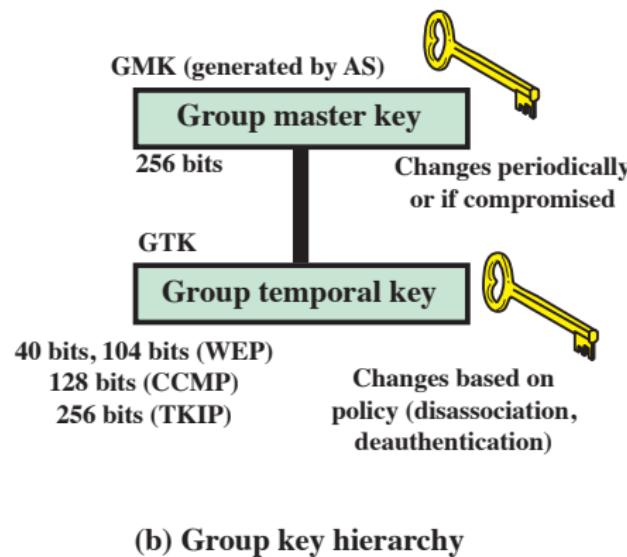


802.11i Key Management Phase

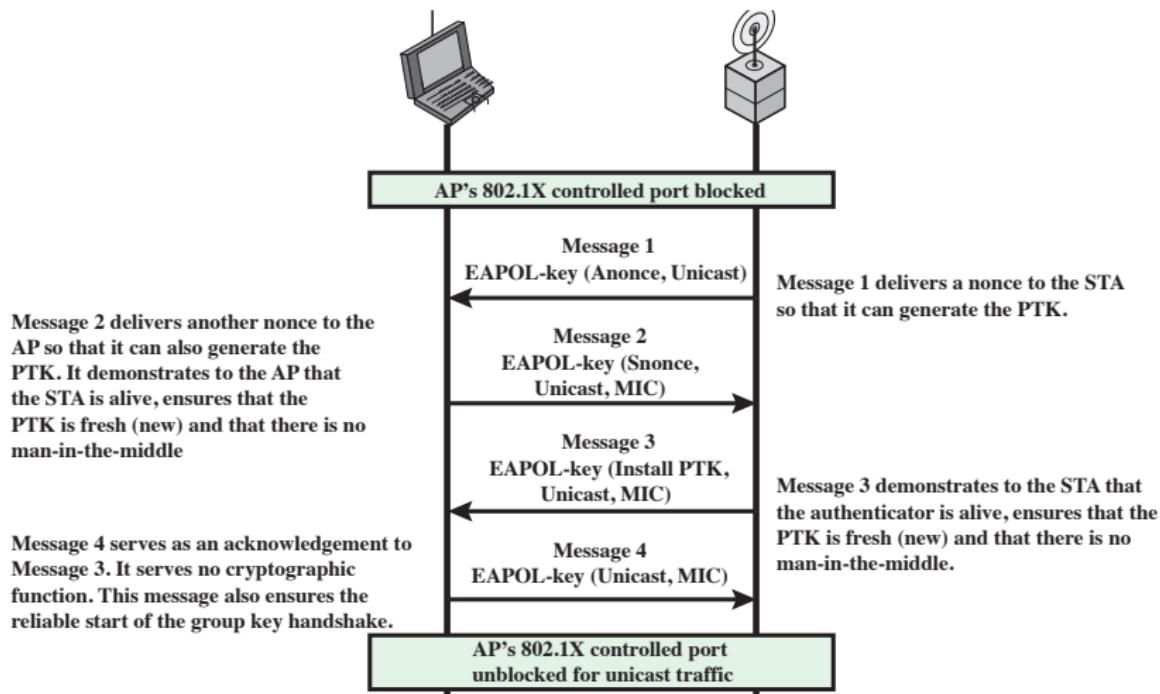


(a) Pairwise key hierarchy

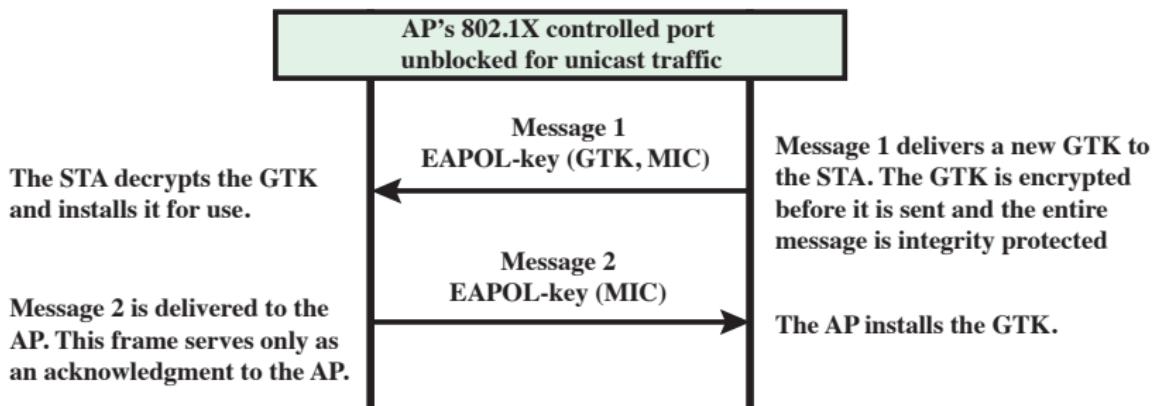
802.11i Key Management Phase



802.11i Key Management Phase



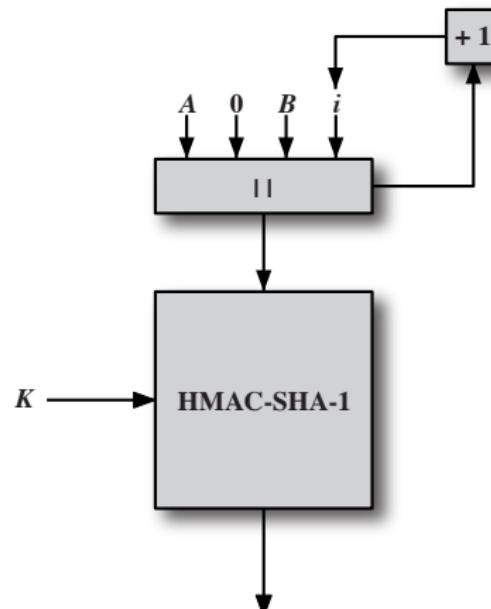
802.11i Key Management Phase



802.11i Protected Data Transfer Phase

- Temporal Key Integrity Protocol (**TKIP**)
 - Software changes only to older WEP
 - Adds 64-bit Michael message integrity code (MIC)
 - Encrypts MPDU plus MIC value using RC4
- Counter Mode-CBC MAC Protocol (**CCMP**)
 - Cipher block chaining message authentication code (CBC-MAC) for integrity
 - CRT block cipher mode of operation

EEE 802.11i Pseudorandom Function



$$R = \text{HMAC-SHA-1}(K, A \parallel 0 \parallel B \parallel i)$$

Summary

Reference: Chapter 17 (Excluding WAP)

Today, we learned

- 802.11 Security

Lecture 15: IPSEC

IPSEC

Objectives of Lecture

At the end of this lecture, you will be able to

- define the security architecture of IPSEC

IP Security

- General IP Security mechanisms: authentication, confidentiality and key management
- Applicable to use over LANs, across public & private WANs, & for the Internet
- Services: Access control, Connectionless integrity, Data origin authentication, Rejection of replayed packets, Confidentiality, Limited traffic flow confidentiality

IP Security Use Cases

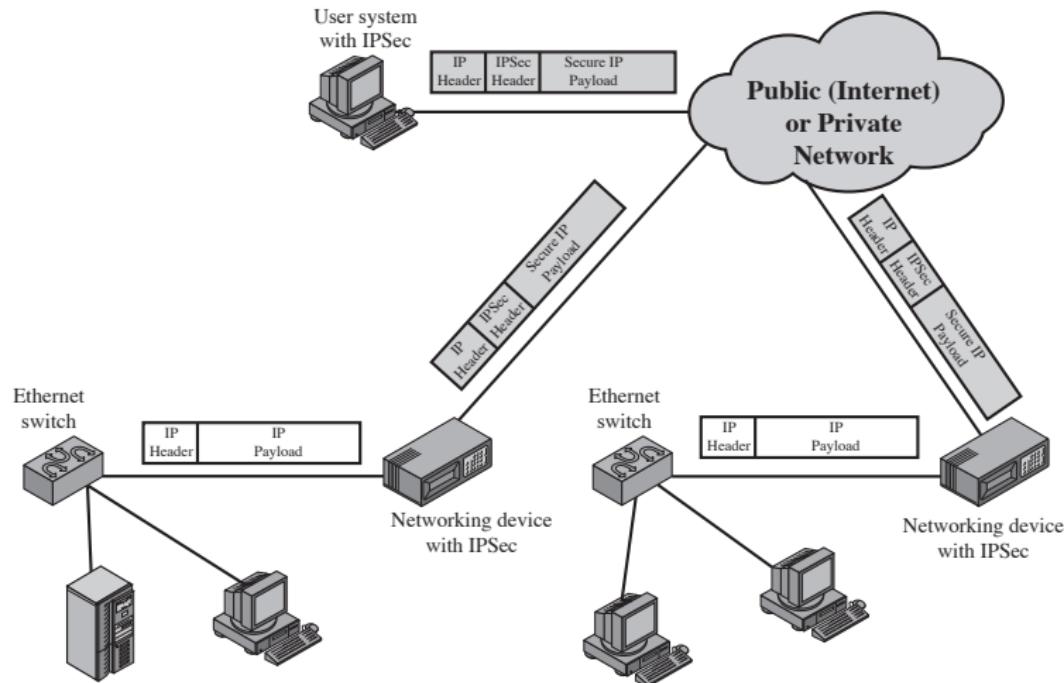


Figure 19.1 An IP Security Scenario

IP Security Benefits

- in a firewall/router provides strong security to all traffic crossing the perimeter
- in a firewall/router is resistant to bypass
- is below transport layer, hence transparent to applications
- can be transparent to end users
- can provide security for individual users
- secures routing architecture

IP Security Soup

- Architecture: RFC4301 Security Architecture for Internet Protocol
- Authentication Header (AH): RFC4302 IP Authentication Header
- Encapsulating Security Payload (ESP): RFC4303 IP Encapsulating Security Payload (ESP)
- Internet Key Exchange (IKE): RFC4306 Internet Key Exchange (IKEv2) Protocol

IP Security Modes

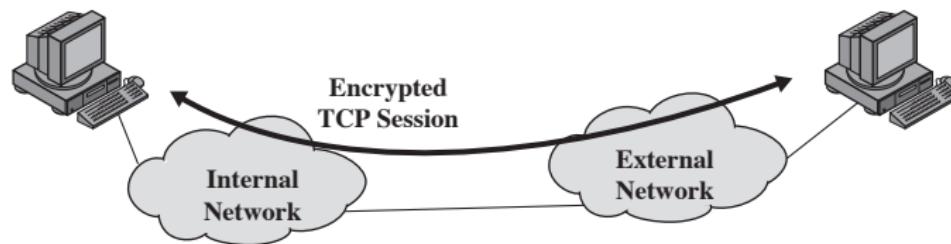
Transport Mode

- to encrypt & optionally authenticate IP data
- can do traffic analysis but is efficient
- good for ESP host to host traffic

Tunnel Mode

- encrypts entire IP packet
- add new header for next hop
- no routers on way can examine inner IP header
- good for VPNs, gateway to gateway security

Transport Level Security



(a) Transport-level security

A Virtual Private Network via Tunnel Mode

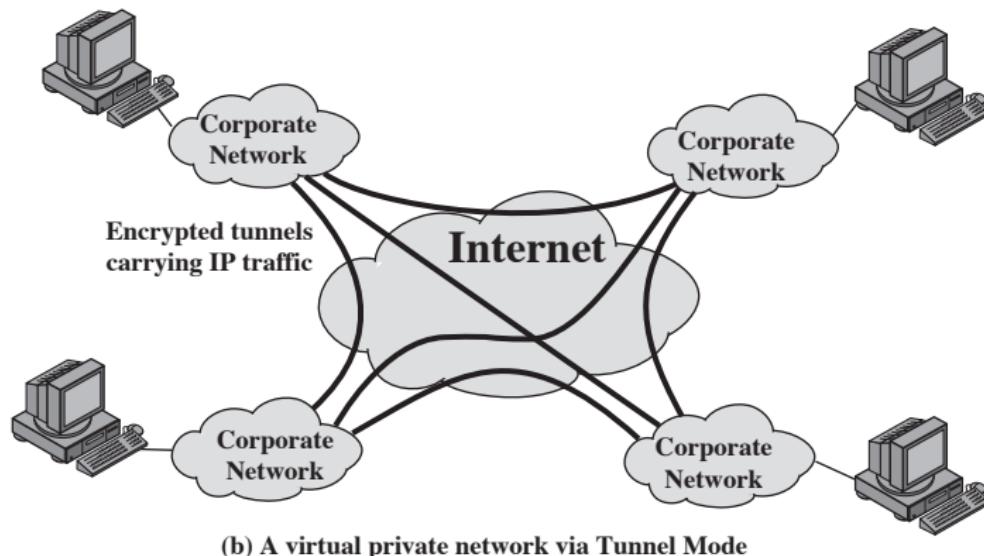
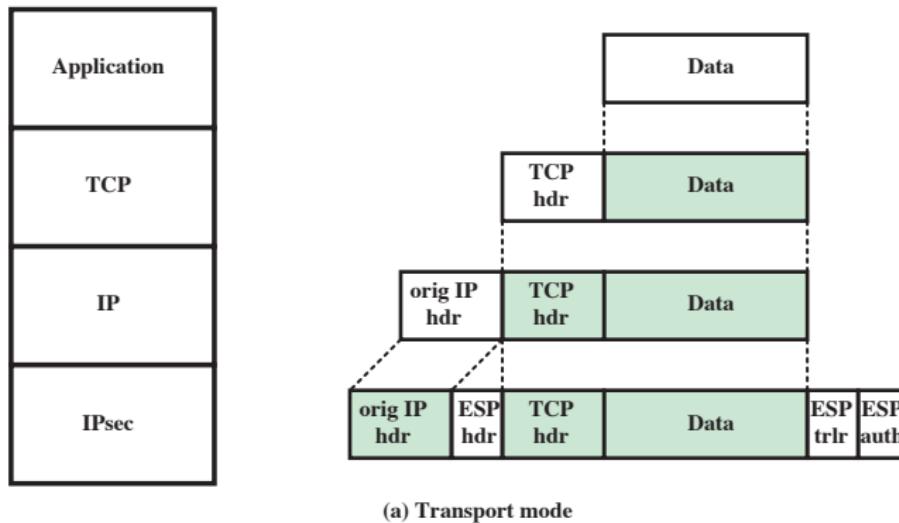
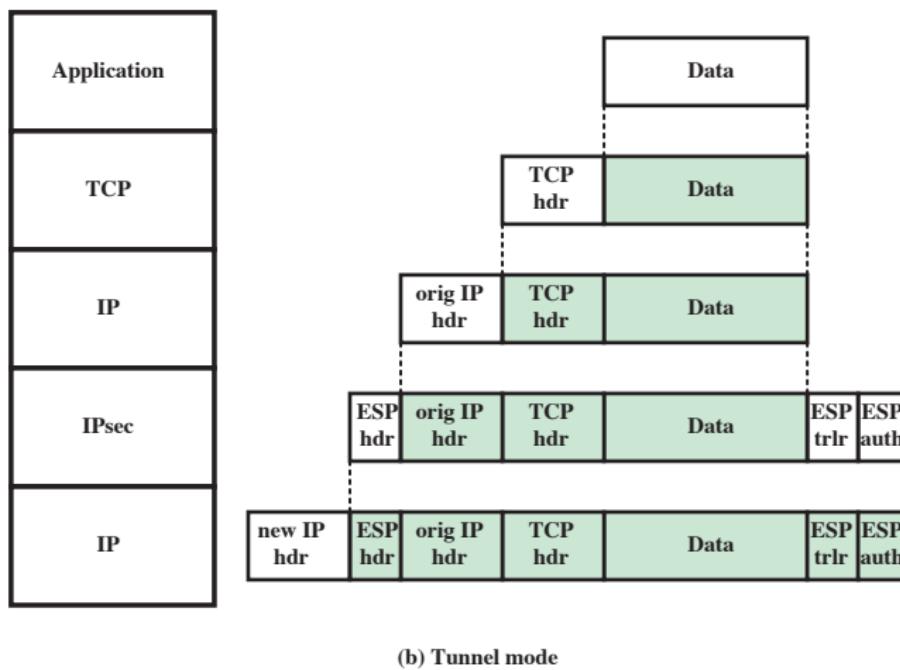


Figure 19.7 Transport-Mode vs. Tunnel-Mode Encryption

Transport Mode



Tunnel Mode



Security Association (SA)

- A one-way relationship between sender & receiver that affords security for traffic flow:
 - Security Parameters Index (SPI)
 - IP Destination Address
 - Security Protocol Identifier
- Has a number of other parameters: seq no, AH & EH info, lifetime etc
- Have a database of Security Associations

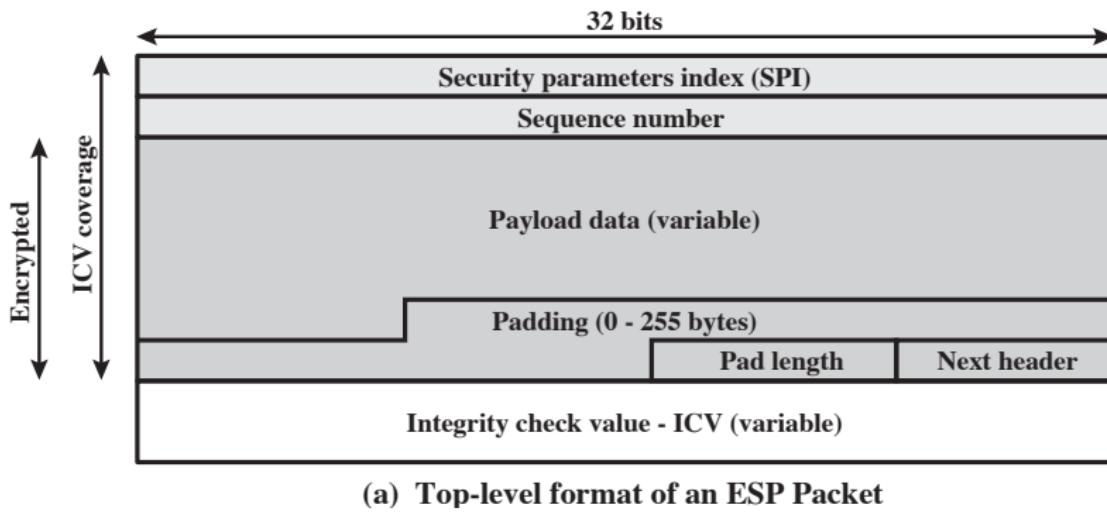
Security Policy Database

- Match subset of IP traffic to relevant SA
- Use selectors to filter outgoing traffic to map
- Based on: local & remote IP addresses, next layer protocol, name, local & remote ports

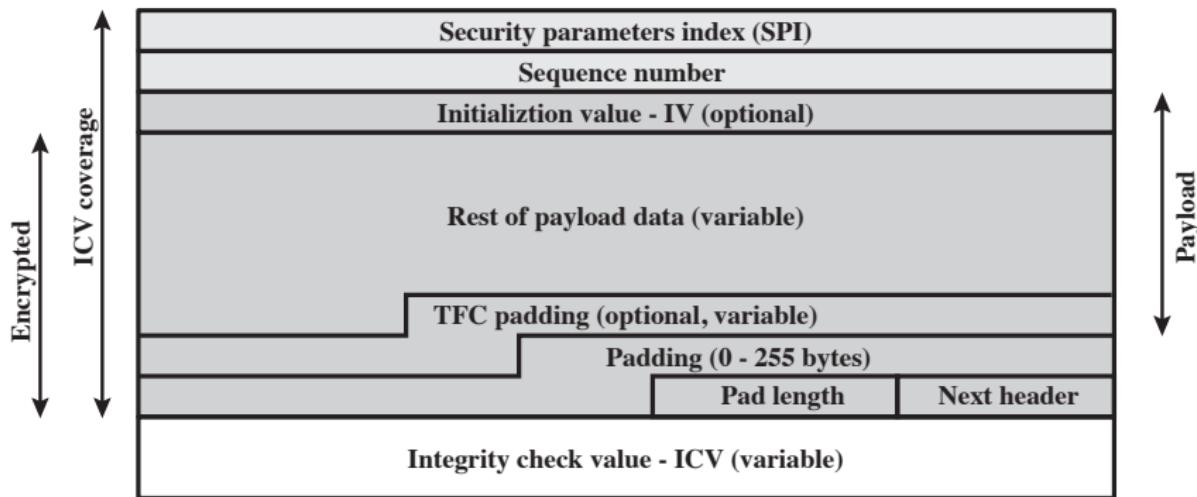
Encapsulating Security Payload (ESP)

- Provides message content confidentiality, data origin authentication, connectionless integrity, an anti-replay service, limited traffic flow confidentiality
- Services depend on options selected when establish Security Association (SA), net location can use a variety of encryption & authentication algorithms

Encapsulating Security Payload (ESP)



Encapsulating Security Payload (ESP)

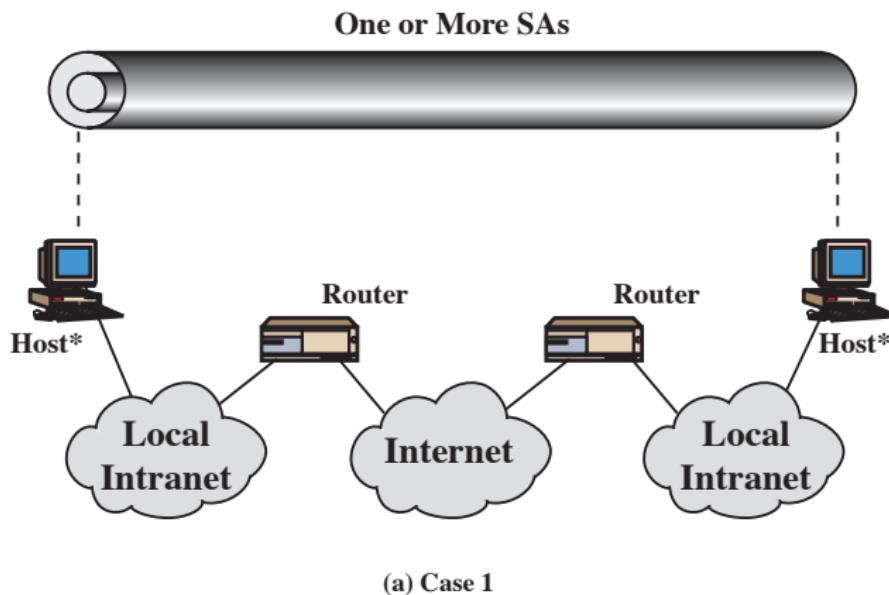


(b) Substructure of payload data

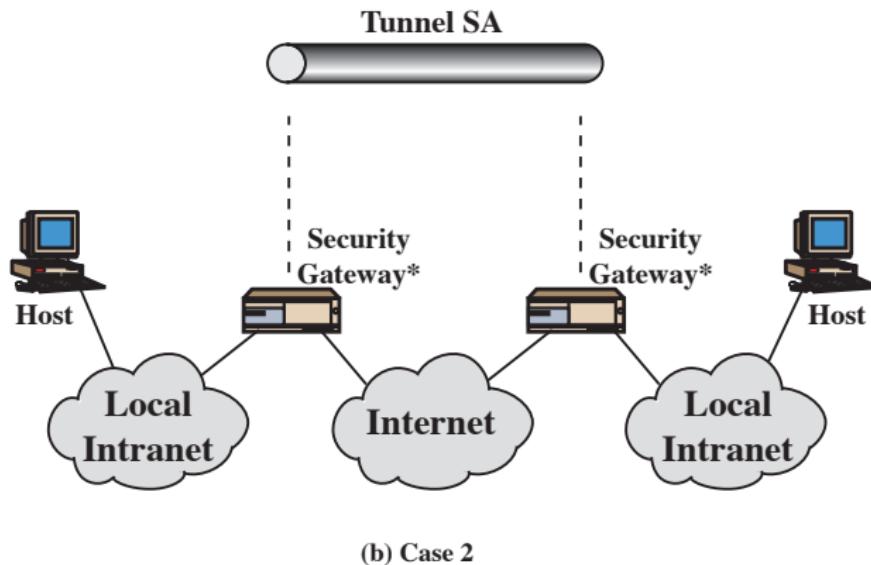
Anti-replay Service

- replay is when attacker resends a copy of an authenticated packet
- use sequence number to thwart this attack
- sender initializes sequence number to 0 when a new SA is established
- increment for each packet
- must not exceed limit of $2^{32} - 1$
- receiver then accepts packets with seq no within window of $(N - W + 1)$

Combining Security Associations

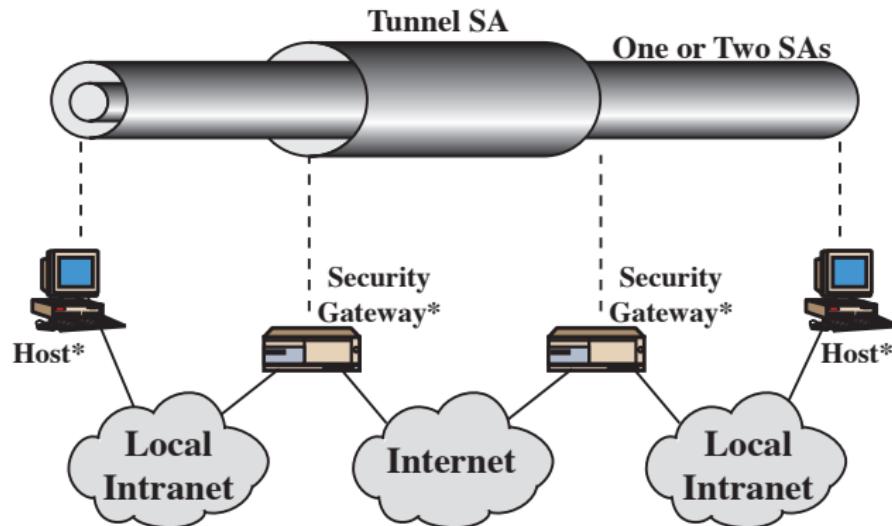


Combining Security Associations



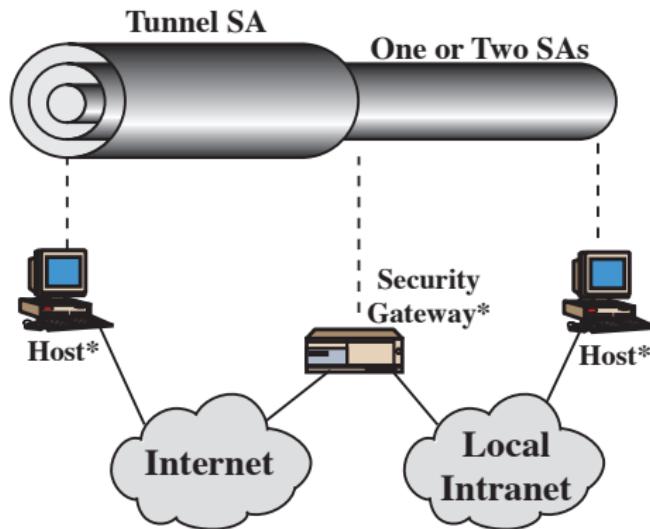
* = implements IPsec

Combining Security Associations



(c) Case 3

Combining Security Associations

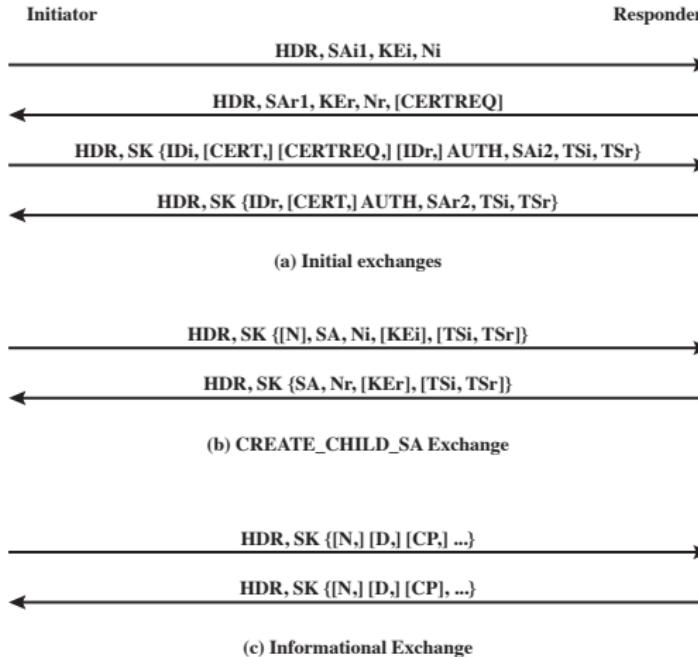


(d) Case 4

Key Management

- Typical: 2 pairs of keys
- **OAKLEY**
 - Based on Diffie-Hellman
 - No info on parties, man-in-middle attack, cost
 - Adds cookies, groups, nonces, DH key exchange with authentication
- **ISAKMP: Internet Security Association and Key Management Protocol**
 - Defines procedures and packet formats to establish, negotiate, modify, & delete SAs
 - Independent of key exchange protocol, encryption alg, & authentication method
 - IKEv2 no longer uses Oakley & ISAKMP terms, but basic functionality is same

Internet Key Exchange IKEv2



HDR = IKE header

SAx1 = offered and chosen algorithms, DH group

KEx = Diffie-Hellman public key

Nx = nonces

CERTREQ = Certificate request

IDx = identity

CERT = certificate

SK {...} = MAC and encrypt

AUTH = Authentication

SAx2 = algorithms, parameters for IPsec SA

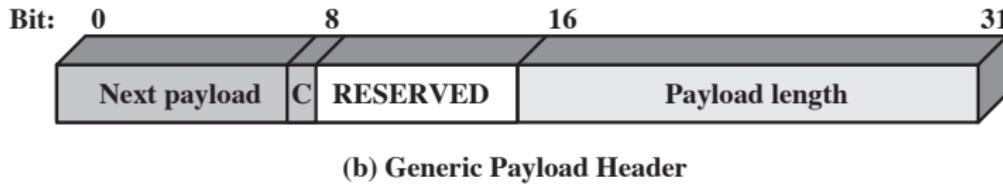
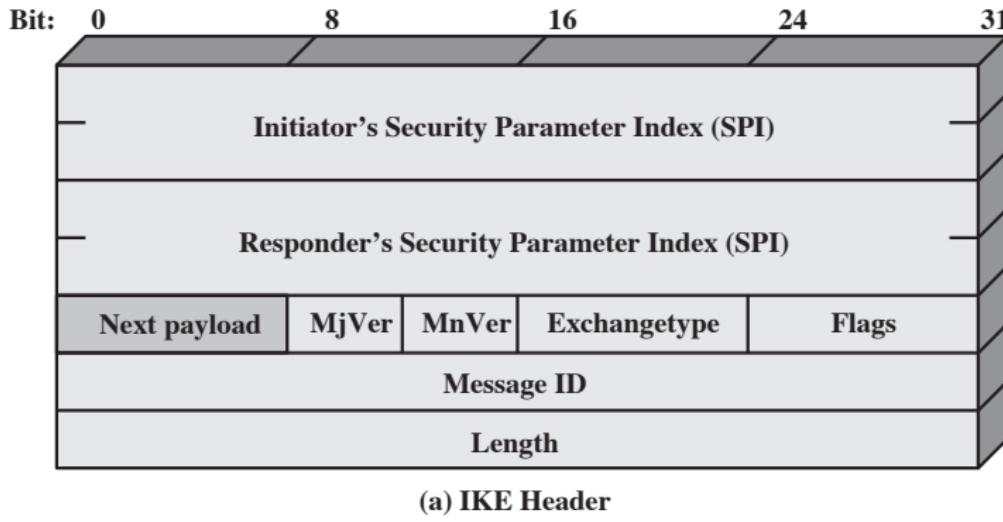
TSx = traffic selectors for IPsec SA

N = Notify

D = Delete

CP = Configuration

Internet Key Exchange IKEv2



Summary

Reference: Chapter 19

Today, we learned

- IPSec

References I