CS 493
Secure Software Systems

Ch 3 The Network Environment

Dr. Williams
Central Connecticut State University

Introduction

- Whenever communication is involved in a software system, the information that is communicated immediately becomes a high risk for attack.
- **Communication is the <u>transmission of system information</u> within or outside of the system boundary.**

Objectives

- Basic tools cryptographic tools to accomplish Confidentiality and Integrity
- Terminology in network communications
- The process of communicating on a network
- The possible compromises of network traffic
- The proper application of cryptography to the network environment
- The Open Systems Interconnection (OSI) model of network communication
- Best practices in deciding which security measures to implement on a network

Goals

- Identify threats to network communication.
- Identify proper applications of cryptosystems to the protection of network traffic.
- Identify the risks associated with different layers of connectivity.
- Assess the needs of a message in transit.
- Plan the network communication structure to meet business objectives.

CIA

- **Confidentiality** is the easiest to grasp; you do not want to unintentionally end up on YouTube because that never goes well.
- **Integrity** means you do not want what you say to be twisted or misinterpreted in any way.
- **Authentication,** as it relates to communication, means you know you are talking (or texting) to the person or entity to whom you wish to speak.

CIA

- **Nonrepudiation** means the person or entity to whom you are speaking cannot deny speaking to you and is therefore held accountable for what was said.
- Confidentiality can mostly be provided by the use of **cryptography**.

Introducing Eve (and Trudy)

- If you have done any work in network communications, you are likely familiar with Alice and Bob.
- These are the personifications of A and B, respectively, two nodes that are sending information back and forth.
- Into this happy world of Alice and Bob sending whatever they want, however they want, without risk, comes Eve the Eavesdropper.

- *Note book refers to the "bad guy" as **Eve** which is common for referring to an eavesdropper, but not used for potentially active participants – **Trudy** (for intruder) is the more common security parlance when the intruder could also be potentially active and what I'll use in many of my examples*

Introducing Eve

- Most people like to cast Eve as the villain, but not knowing Alice or Bob personally, we cannot speak to their good intentions or legitimacy, so we will refer to Eve as simply the catalyst for good security.
- A good way to start thinking about secure communication is what you want when you are talking to your friends or family in private.

Eve Unleashed

- The simplest scenario is where Alice sends Bob a message (M) in the clear, or without any encryption. Eve can intercept this message, often without the detection of either party.
- Without any encryption, Eve can read the message and store the information for use at her leisure.

The Science of Secrecy

- Cryptography is a practice that dates back more than 2,000 years; it is the science of encoding a message so that it cannot be read when the message itself is compromised.
- Greeks – simple transposition cipher
- Romans – substitution cipher
- Zimmerman telegram (1917) – code books
- WW II Golden Age of crypto
  - Enigma (German), Purple (Japanese)
  - Modern cyptoanalysis was born…and computing

One time pad

- One time pad encryption like the other ones is symmetric
  - Same key to encode as decode
- Unlike other methods once a message is encrypted/decrypted the cipher should **never** be used again
- Basic idea is use a scheme that any ciphertext message could be decoded to any plaintext
  - No amount of ciphertext whether large or small is going to give you insight (a shortcut) to decoding the message

Applying one time pad

- The application requires the message to be encoded as binary
- Cipher of same length binary is then applied by XORing the two binaries to produce the ciphertext
- Decode is performed by XORing cipher and ciphertext

One-Time Pad: Encryption

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

Encryption: Plaintext $\oplus$ Key = Ciphertext

	h	e	i	l	h	i	t	l	e	r
Plaintext:	001	000	010	100	001	010	111	100	000	101
Key:	111	101	110	101	111	100	000	101	110	000
Ciphertext:	110	101	100	001	110	110	111	001	110	101
	s	r	l	h	s	s	t	h	s	r

One-Time Pad: Decryption

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

Decryption: Ciphertext ⊕ Key = Plaintext

	s	r	l	h	s	s	t	h	s	r
Ciphertext:	110	101	100	001	110	110	111	001	110	101
Key:	111	101	110	101	111	100	000	101	110	000
Plaintext:	001	000	010	100	001	010	111	100	000	101
	h	e	i	l	h	i	t	l	e	r

One-Time Pad

Double agent claims sender used following "**key**"

	s	r	l	h	s	s	t	h	s	r
Ciphertext:	110	101	100	001	110	110	111	001	110	101
"**key**":	101	111	000	101	111	100	000	101	110	000
"Plaintext":	011	010	100	100	001	010	111	100	000	101
	k	i	l	l	h	i	t	l	e	r

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

One-Time Pad

Or sender is captured and claims the key is…

	s	r	l	h	s	s	t	h	s	r
Ciphertext:	110	101	100	001	110	110	111	001	110	101
"Key":	111	101	000	011	101	110	001	011	101	101
"Plaintext":	001	000	100	010	011	000	110	010	011	000
	h	e	l	i	k	e	s	i	k	e

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

One-Time Pad Summary

- **Provably** secure…
  - Ciphertext provides **no** info about plaintext
  - All plaintexts are equally likely
- …but, only when used correctly
  - Pad must be random, used only once
  - Pad is known only to sender and receiver
  - Thus requires infinite non-repeating, unpredictable stream of bits that must be distributed securely

- Net result while the encryption is perfect, it isn't practical thus the need for mathematics and inherent places for weaknesses we must consider

Claude Shannon

- The founder of Information Theory
- 1949 paper: *[Comm. Thy. of Secrecy Systems]*
- Fundamental concepts
  - **Confusion** — obscure relationship between plaintext and ciphertext
  - **Diffusion** — spread plaintext statistics through the ciphertext
- Proved one-time pad is secure
- Substitution and One-time pad are confusion-only, while double transposition is diffusion-only

Post-WWII History

- Claude Shannon — father of the science of information theory
- Computer revolution — lots of data to protect
- Data Encryption Standard (DES), 70's
- Public Key cryptography, 70's
- CRYPTO conferences, 80's
- Advanced Encryption Standard (AES), 90's
- The crypto genie is out of the bottle…

Taxonomy of Cryptography

- **Symmetric Key**
  - Same key for encryption and decryption
  - Two types: Stream ciphers, Block ciphers
- **Public Key** (or asymmetric crypto)
  - Two keys, one for encryption (public), and one for decryption (private)
  - Also, digital signatures — nothing comparable in symmetric key crypto
- **Hash algorithms**
  - Sometimes viewed as “one way” crypto

Malicious Modifications and Insidious Insertions

- Eavesdropping is generally a passive activity and the participants are unaware of the extra presence.
- But in security we need to be concerned with active attackers in addition to passive listeners…

Confidentiality ≠ integrity?

- Notation: $C = E(P,K)$
- Given plaintext $P_0,P_1,\ldots,P_m,\ldots$
- One way to use a block cipher:

Encrypt	**Decrypt**
$C_0 = E(P_0, K)$	$P_0 = D(C_0, K)$
$C_1 = E(P_1, K)$	$P_1 = D(C_1, K)$
$C_2 = E(P_2, K)$ …	$P_2 = D(C_2, K)$ …

Thus C is our plaintext P, but it can’t be read so it provides confidentiality, but…

Confidentiality ≠ integrity?

- Suppose plaintext is

    `Alice digs Bob. Trudy digs Tom.`
- Assuming 64-bit blocks and 8-bit ASCII:

    P_0 = "`Alice di`", P_1 = "`gs Bob. `",

    P_2 = "`Trudy di`", P_3 = "`gs Tom. `"
- Ciphertext: C_0, C_1, C_2, C_3
- Trudy cuts and pastes: C_0, C_3, C_2, C_1
- Decrypts as

    `Alice digs Tom. Trudy digs Bob.`

Note **confidentiality** was never violated, but Bob has no way of knowing this isn't the actual message from Alice so **no integrity**

Data Integrity

- **Integrity** — detect unauthorized writing (i.e., modification of data)
- Example: Inter-bank fund transfers
  - Confidentiality is nice, but integrity is critical
- Encryption provides **confidentiality** (prevents unauthorized disclosure)
- Encryption alone does **not** provide integrity
  - One-time pad, ECB cut-and-paste, etc.

MAC

- Message Authentication Code (MAC)
  - Used for data **integrity**
  - Integrity **not** the same as confidentiality
- MAC is computed as Cipher Block Chain residue or **CBC residue**
  - That is, compute CBC encryption, only save final ciphertext block, the MAC

Does a MAC work?

- Suppose Alice has 4 plaintext blocks
- Alice computes

  $C_0 = E(IV \oplus P_0, K)$, $C_1 = E(C_0 \oplus P_1, K)$,
  $C_2 = E(C_1 \oplus P_2, K)$, $C_3 = E(C_2 \oplus P_3, K) =$ **MAC**
- Alice sends IV, P_0, P_1, P_2, P_3 and **MAC** to Bob
- Suppose Trudy changes P_1 to X
- Bob computes

  $C_0 = E(IV \oplus P_0, K)$, ***$C_1$*** $= E(C_0 \oplus X, K)$,
  C_2 $= E($***C_1***$\oplus P_2, K)$, ***C_3*** $= E($***C_2***$\oplus P_3, K) =$ ***MAC*** $\neq$ **MAC**
- That is, error propagates into **MAC**
- Trudy can't change ***MAC*** to **MAC** without K

Confidentiality and Integrity

- Encrypt with one key, MAC with another key
- Why not use the same key?
  - Send last encrypted block (MAC) twice?
  - This cannot add any security!
- Using different keys to encrypt and compute MAC works, even if keys are related
  - But, twice as much work as encryption alone
  - Can do a little better — about 1.5 "encryptions"
- Confidentiality and integrity with same work as one encryption is a research topic

Uses for Symmetric Crypto

- Confidentiality
  - Transmitting data over insecure channel
  - Secure storage on insecure media
- Integrity (MAC)
- Authentication protocols
- Anything you can do with a hash function

Public Key Cryptography

- **Asymmetric encryption** Uses **two** separate keys. One key is used to encrypt plain text and a second key is used to decrypt the ciphertext.
  - The encryption key is called a public key.
  - The decryption key is called a private key.
- Based on "trap door one way function"
  - "One way" means easy to compute in one direction, but hard to compute in other direction
  - Example: Given p and q, product N=pq is easy to compute, but given N, it is hard to find p and q
  - "Trap door" used to create key pairs
- Key concept is **private key never needs to be distributed** so avoids that aspect of the key distribution problem

Public Key Cryptography

- Encryption
  - Suppose we **encrypt** M with Bob's public key
  - Bob's private key can **decrypt** to recover M
- Digital Signature
  - **Sign** by "encrypting" with your private key
  - Anyone can **verify** signature by "decrypting" with public key
  - But only you could have signed
  - Like a handwritten signature (only more so...)

Uses for Public Key Crypto

- Confidentiality
  - Transmitting data over insecure channel
  - Secure storage on insecure media
- Authentication (later)
- Digital signature provides **integrity** and **non-repudiation**
  - No non-repudiation with symmetric keys

More demanding Integrity

- A higher and more reliable means of authentication is **nonrepudiation**. This means that the message sent is guaranteed to be from the identified sender and the sender cannot later deny sending it.

Non-non-repudiation

- Alice orders 100 shares of stock from Bob
- Alice computes **MAC** using symmetric key
- Stock drops, Alice claims she did ***not*** order
- Can Bob prove that Alice placed the order?
- **No!** Since Bob also knows the symmetric key, he could have forged message
- **Problem:** Bob knows Alice placed the order, but he can't prove it

Non-repudiation

- Alice orders 100 shares of stock from Bob
- Alice **signs** order with her private key
- Stock drops, Alice claims she did not order
- Can Bob prove that Alice placed the order?
- **Yes!** Only someone with Alice's private key could have signed the order
- This assumes Alice's private key is not stolen (revocation problem)

Encryption Notation

Symmetric cryptography

$C = E(M, K)$ encrypt message with secret key

$M = D(C, K)$ decrypt ciphertext with secret key

Public key cryptography

- **Sign** message M with Alice's **private key:** $[M]_{Alice}$
- **Encrypt** message M with Alice's **public key:** $\{M\}_{Alice}$

Key distribution for public key = PKI

- Distributing a public key in clear is not a security concern, knowing that it is the actual public key for the matching private key for the party you think it is is the security concern
- **Public Key Infrastructure** (**PKI**) – mechanism used to verify a particular public key is actual public key for a particular party
  - Most common is CA approach, but others are used
- **Certificate** contains name of user and user's public key (and possibly other info)
- It is ***signed*** by the issuer, a ***Certificate Authority*** (CA), such as VeriSign

  M = (Alice, Alice's public key), $S = [M]_{CA}$

  Alice's Certificate = (M, S)
- Signature on certificate is verified using CA's public key

  Verify that $M = \{S\}_{CA}$

Certificate Authority

- Certificate authority (CA) is a trusted 3rd party (TTP) — creates and signs certificates
- Verifying signature verifies integrity, identity of **owner of corresponding private key**
  - Does **not** verify the identity of the **sender** of certificate — certificates are public!
- Big problem if CA makes a mistake (a CA once issued Microsoft certificate to someone else)
- A common format for certificates is X.509

Certificate Authorities cont.

- Multiple trusted CAs
- This is approach used in browsers today
- Browser may have 80 or more certificates, just to verify certificates!
- User can decide which CAs to trust

Symmetric Key vs Public Key

- Symmetric key +'s
  - **Speed**
  - No public key infrastructure (PKI) needed
- Public Key +'s
  - **Signatures** (non-repudiation)
  - Avoid key distribution of shared secret
  - No ***shared*** secret (but, private keys…)

Crypto Hash Function

- One way mapping
  - Easy to confirm message matches hash
  - Unlike symmetric/public cryptography original message can not be recovered from hash
- Fast and small
  - Computing hash should be fast – No slower than producing MAC (usually considerably faster)
  - Take any lengthy of message and produce much smaller fingerprint

Hash Function Motivation

- Suppose Alice signs M
  - Alice sends M and $S = [M]_{Alice}$ to Bob
  - Bob verifies that $M = \{S\}_{Alice}$
  - Can Alice just send S?
- If M is big, $[M]_{Alice}$ costly to compute & send
- Suppose instead, Alice signs h(M), where h(M) is much smaller than M
  - Alice sends M and $S = [h(M)]_{Alice}$ to Bob
  - Bob verifies that $h(M) = \{S\}_{Alice}$
- **Smaller and faster**

Crypto Hash Function - requirements

- Crypto hash function $h(x)$ must provide
  - **Compression** — output length is small
  - **Efficiency** — $h(x)$ easy to compute for any x
  - **One-way** — given a value y it is infeasible to find an x such that $h(x) = y$
  - **Weak collision resistance** — given x and $h(x)$, infeasible to find $y \neq x$ such that $h(y) = h(x)$
  - **Strong collision resistance** — infeasible to find ***any*** x and y, with $x \neq y$ such that $h(x) = h(y)$
- Lots of collisions exist, but hard to find ***any***

Eve Unleashed

- Eve can take the message offline and start working to break the encryption scheme; some types of encryption are more difficult than others, so the choice of the cryptosystem to be used is essential in protecting the message.
- The lowest order of attack on a cryptosystem is the use of brute force.

Eve Unleashed

- Passwords should involve similar consideration because both are essential to protecting your software and your information and both represent a single piece of information separating an attacker from what is valuable.
- A **dictionary attack** is the use of common words to form possible keys or passwords; a better variant of this is to hybridize words and numbers.

Eve Unleashed

- Security by obscurity means you rely on an attacker not knowing how the internal mechanism of your security operates as a means of securing the system.
- The strength of your security should be in the key and not in the algorithm chosen; you must assume that an attacker knows how the message is encrypted and how your security schema works even if this is not the case.

Making a Connection

- Network communications follow a structure, so you will often be working within an existing framework to transmit the contents of your message. This framework can limit the size of the message based on the structure of the packet that is being transmitted, and the packet, in turn, is limited by the characteristics of the network on which it will travel.

Man in the middle

- Risks of networked communications
  - Passive
    - Eavesdropping
  - Active
    - Manipulation
    - Man in the middle

Diffie-Hellman

- Invented by Williamson (GCHQ) and, independently, by D and H (Stanford)
- A “key exchange” algorithm
  - Used to establish a shared symmetric key
- ***Not*** for encrypting or signing
- Based on **discrete log** problem (NP problem):
  - **Given** g, p, and g^k mod p
  - **Find** k

Diffie-Hellman

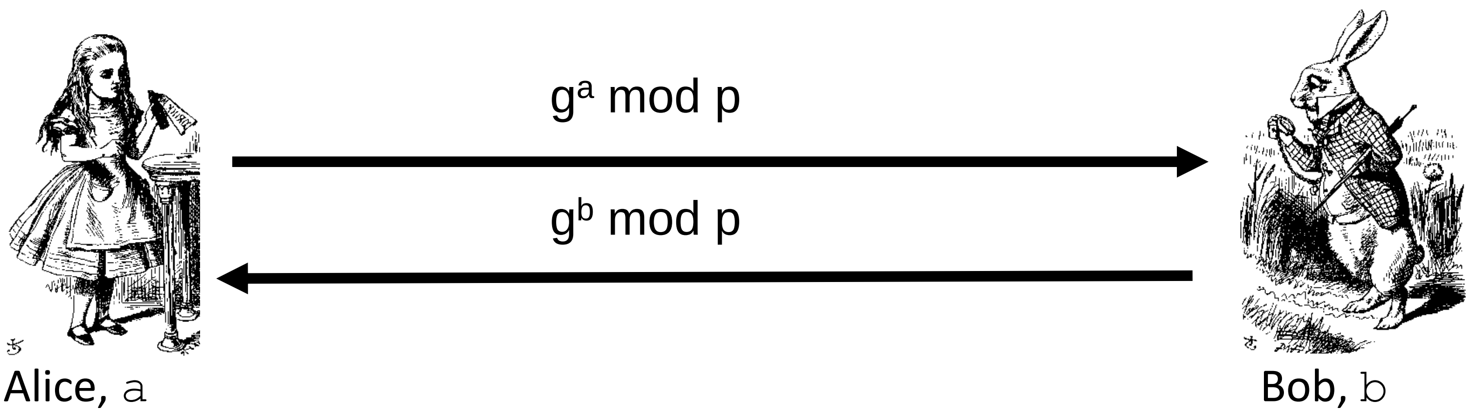
- Let p be prime, let g be a **generator**
  - For any $x \in \{1,2,\ldots,p-1\}$ there is n s.t. $x = g^n \bmod p$
- Alice selects her secret value a
- Bob selects his secret value b
- Alice sends $g^a \bmod p$ to Bob
- Bob sends $g^b \bmod p$ to Alice
- Both compute shared secret, $g^{ab} \bmod p$
- Shared secret can be used as symmetric key

Diffie-Hellman

- Suppose that Bob and Alice use g^{ab} mod p as a symmetric key
- Trudy can see g^a mod p and g^b mod p
  - But… $g^a g^b$ mod p = g^{a+b} mod p $\neq$ g^{ab} mod p
- If Trudy can find a or b, system is broken
- If Trudy can solve **discrete log** problem, she can find a or b

Diffie-Hellman

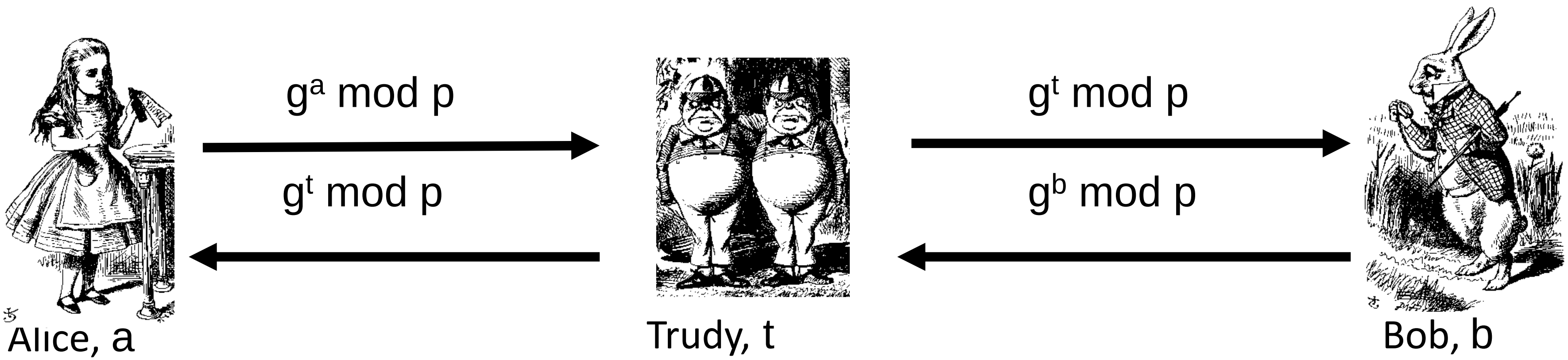
- **Public:** g and p
- **Secret:** Alice's exponent a, Bob's exponent b



- Alice computes $(g^b)^a = g^{ba} = g^{ab}$ mod p
- Bob computes $(g^a)^b = g^{ab}$ mod p
- Use $K = g^{ab}$ mod p as symmetric key

Diffie-Hellman

- Subject to man-in-the-middle (MiM) attack



- Trudy shares secret g^{at} mod p with Alice
- Trudy shares secret g^{bt} mod p with Bob
- Alice and Bob don't know Trudy exists!

Man in the middle

- This is one example of how an active intruder can cause havok
- How to prevent MiM attack?
  - Protocols and eliminating assumptions
- You **MUST** be aware of MiM attack

Proximity is Not Security

- Networking could potentially be considered the root of all cybercrime.
- **Networking**, if you are not familiar with the term, is allowing one machine to communicate with another.
- The safest system in the world is one that is turned off, poured in concrete, and buried in a vault, but how useful would that be?

- *Example early Bluetooth protocol*

Making the Connection

- A **protocol** is a set structure for a message that allows network hardware to determine what information is being sent and what to expect; a protocol can include a single pattern for all communication or multiple patterns for continued communication between parties.
- Dating from 1978, the International Organization for Standardization (ISO) began defining what has evolved into the **Open Systems Interconnection (OSI) model.**

OSI Model

OSI Model			
Layer		Protocol data unit (PDU)	Function[3]
Host layers	7. Application	Data	**High-level APIs**, including resource sharing, remote file access
	6. Presentation		Translation of data between a networking service and an application; including **character encoding, data compression and encryption/decryption**
	5. Session		Managing communication sessions, i.e. continuous exchange of information in the form of **multiple back-and-forth transmissions between two nodes**
	4. Transport	Segment (TCP) / Datagram (UDP)	Reliable **transmission of data segments between points on a network**, including segmentation, acknowledgement and multiplexing
Media layers	3. Network	Packet	Structuring and managing a multi-node network, including **addressing, routing and traffic control**
	2. Data link	Frame	Reliable transmission of data frames between **two nodes connected by a physical layer**
	1. Physical	Bit	Transmission and reception of **raw bit streams over a physical medium**

OSI Model

- The physical layer (layer 1) is primarily concerned with connecting the machine to the medium of transmission.
- The data link layer (layer 2) starts to provide identifying characteristics to each machine on the network such as a physical address of a device, which is a Media Access Control (MAC) address, and the ability to detect errors in the physical layer.

OSI Model

- Routing between machines is provided at the network layer (layer 3).
- The transport layer (layer 4) is where you find the most common network transmission protocols. This layer is responsible for providing end-to-end transfer of data, detecting errors in transmission, and retransmitting data if necessary.

OSI Model

- The session layer (layer 5) sits above the transport layer and is tasked with organizing connections between a network node and a remote entity or service.
- The presentation layer (layer 6) above the session layer translates between application and network formats; this layer is primarily concerned with the representation of data and any possible structure of the data for use in the application layer.

OSI Model

- The highest level of the OSI model is the application layer (layer 7). This is where the software is directly involved in directing network communications.

Roll Up the Welcome Mat

- One of the highest risks to a system that is connected to a network is uninvited traffic. The more complex a system becomes, the more likely it is that there is a proverbial backdoor that is not locked.
- A good housekeeping rule in development is to make sure that the module or object that opens a session also closes it.

Summary

- Communication is a vital component of any modern software system.
- There are techniques available to secure communications through confidentiality, integrity, nonrepudiation, authentication, and message freshness.
- Mitigation techniques need to be used for communication of any sensitive information housed within the system.

Security example
Alice's Online Bank

- Alice opens Alice's Online Bank (AOB)
- What are Alice's security concerns from a network perspective?
  - Consider ones in own environment
  - Consider ones interoperating with another bank
  - Consider ones interacting with customer
    - Web
    - Mobile app