Core Concepts

• Have a thorough understanding of Confidentiality, Integrity, and Availability (C.I.A.).
Confidentiality: only authorized people or systems can access the data or resource
Integrity: assurance that the information is authentic and complete

- Data integrity: the assurance that data received is exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay)
- Origin integrity: the source of data is trustworthy
- System Integrity: the assurance that the system performs intended function without inadvertent or deliberate unauthorized manipulation of the system

Availability: people have the ability to use the information or resource desired
– Example: An attacker intercepts a secret message and modifies it.
Which of the above was compromised? How can cryptography solve this?
This example is a breach of integrity since the message was modified. One possible solution is adding a digital signature to the message.

– Example: What is the difference between data integrity and origin integrity?
Data integrity is about if the data was modified while the origin integrity is assuring that the data came from the right source of data.

• What is the anti-C.I.A. triad? i.e., Disclosure, Alteration, Destruction (D.A.D) triad.
DAD triad is the opposite of the CIA triad.
D isclosure
A lteration
D estruction

• What is the difference between active vs. passive attacks?
Active attacks modify data while passive attacks do not. Passive attacks are harder to detect compared to active attacks.

• What three main aspects of security does ITU-T X.800 consider? I.e., Security Attacks, Security Mechanisms, and Security Services.
ITU - T X.800 is the security architecture for OSI. The main aspects of security it considers are security attacks, security mechanisms, and security services.

• Know all attacks, services, and mechanisms mandated by the ITU-T X.800 (a) security architecture for OSI)
– Example: An attacker intercepts and modifies a message. Was this an active or a passive attack?
This is an example of an active attack since there is data being modified.

• Why are passive attacks so difficult to detect?

Passive attacks are difficult to detect since they do not involve any alteration of data and there is an emphasis on prevention rather than detection. There is no evidence left behind.

• Know all types of active and passive attacks discussed in class.
Passive Attack: release of message contents (eavesdropping on or monitoring of transmissions), traffic analysis (may not be able to extract the information (e.g., because it's encrypted), but might still be able to observe the pattern of these massages)
Active Attack: masquerade (one entity pretends to be a different entity), replay (capture the data unit and transmit to the receiver later to produce an unauthorized effect, modification of messages (some portion of a legitimate message is altered, or messages are delayed or reordered), denial of service (prevents or inhibits the normal use or management of communications facilities)

• How do we protect against traffic analysis?
One way we can protect against traffic analysis is by using traffic padding. Constantly send data so the attacker cannot read anything about what is being sent.

• What are the three main elements of network security (from the point of view of cryptography)?
The three main elements of network security are designing a suitable algorithm for the security transformation, generating the secret information (keys) used by the algorithm, and developing methods to distribute and share the secret information.

Network Security Concepts

• What is the difference between the Internet and the Web?
The Internet is an infrastructure used to deliver data between end systems while the Web is just one protocol that runs on top of the Internet infrastructure.

• What is a protocol?
A protocol is a set of rules that governs communications. A protocol defines format, order of messages sent and received among network entities, and how the receiver should respond.

• Why use a layered protocol model?
We use a layered protocol model since getting data between one application to another is a complex multistep process. Each of these steps require different protocols. We organize these protocols based on their responsibilities.

• Know all layers of the TCP/IP model and the function of each layer.
Application: Provides a means for the user to access information on the network through an application (etc. FTP, HTTP, SMTP)
Transport: Converts messages into TCP segments or User Datagram Protocol (UDP)
Network: routes datagrams from source to destination (routers operate at this layer)
Data Link: provides the functional and procedural means to transfer data between network entities (Bridges and link-layer switches operate)

Physical: encodes and transmits raw data over network communications media (e.g., optical fiber)

– Example: My email program encrypts all data before sending it out. This is an example of application layer security.

• What is the difference between TCP and UDP protocols?
TCP is a reliable connection-oriented protocol while UDP is an unreliable, connectionless protocol.

• Does the TCP/IP model address the security concerns?
The TCP/IP model does not address the security concerns.

• Data encryption often takes place at the presentation layer of the OSI model. Where would it take place in the TCP/IP model?
Data encryption would take place in the application layer in the TCP/IP model.

Data Classification

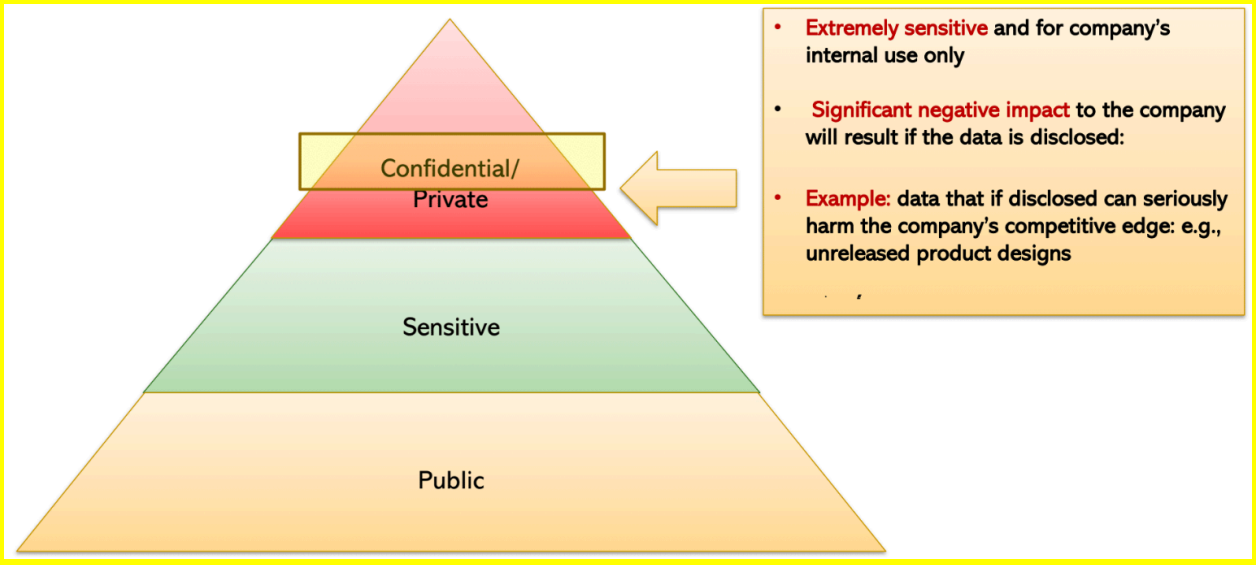
• What is the primary objective of data classification? How does it relate to cryptography?
The primary objective of data classification is to organize data and protect it based on its need for secrecy, sensitivity, or confidentiality. This relates to cryptography since based on how you classify the data, you may want to protect the data differently.

- Why classify data?
  - Demonstrate organization's commitment to protecting valuable assets
  - Helps identify most critical assets
  - Justifies the use of a particular mechanism for protecting the data
  - Required by many security compliance measures
  - Helps establish the guidelines for who may access the data, authorized use, and parameters for declassification or destruction
- Data is classified based on its usefulness, lifetime, maturity, national security implications etc

• What are the steps to properly classifying data?
Identify the custodian of the data, and define their responsibilities
Define criteria for classifying and labeling the data
Classify the resource

Document any exceptions to the classification policy that were discovered, and integrate them into evaluation criteria
Select the security controls that will be applied to each classification level to provide the necessary level of protection
Specify the procedures for declassifying resources and the procedures for transferring custody of a resource to an external entity
Create an enterprise - wide awareness program to instruct all personnel about the classification system.

• What is the difference between how the data is classified in military/government sector vs in the commercial/private sector?



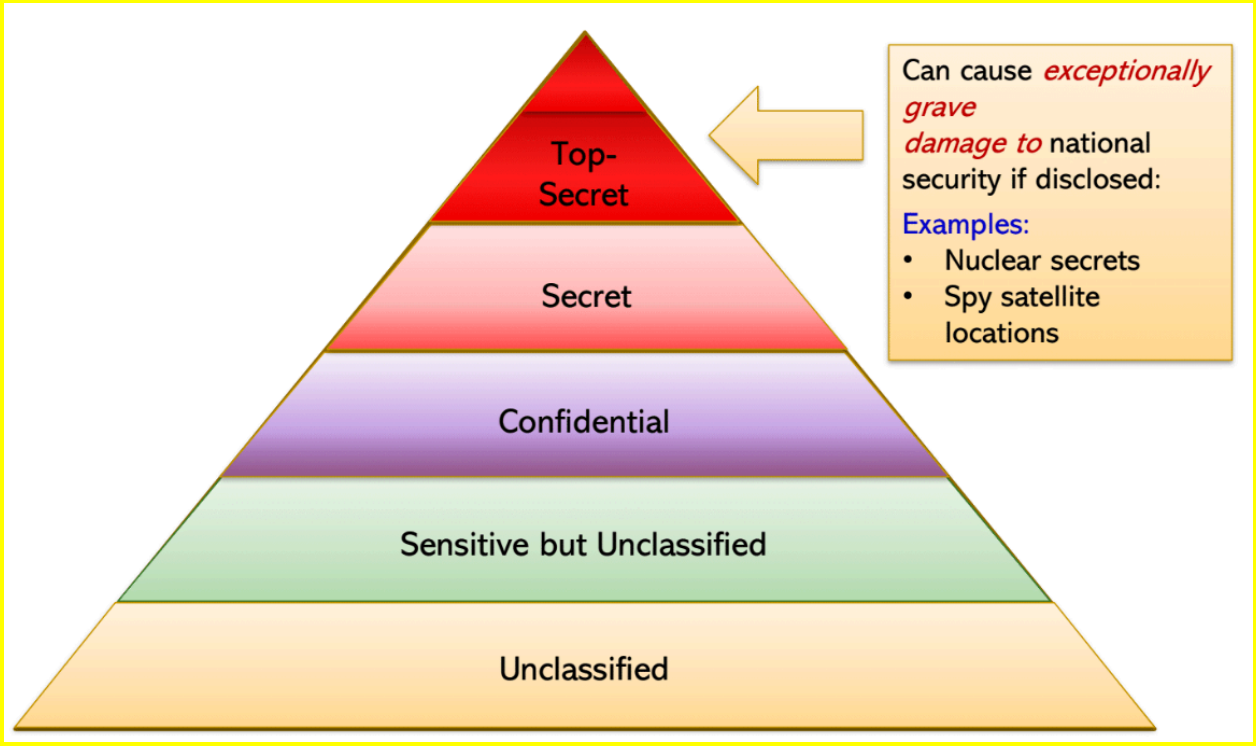
Confidential: Extremely sensitive and for company's internal use only
• Significant negative impact to the company will result if the data is disclosed:
• Example: data that if disclosed can seriously harm the company's competitive edge: e.g., unreleased product designs
Private: •Data of a private or personal nature and intended for internal use only
• A significant negative impact could occur for the company or individuals if private data is disclosed
Confidential vs Private:
• Both require same level of security
• Confidential generally refers to company data (e.g., product designs)
• Private generally refers to the data related to individuals (e.g., medical records)
Sensitive: More classified than public data
• Negative impact will result to the company if disclosed
• Examples: List of laid off employees
Public: Will result in no negative impacts for the organization if disclosed
• Examples: Data on the public website, News

FIPS-140 Series

• What is FIPS-140? What does it provide?
FIPS-140 is a security standard which was developed by the US government.

• For what type of government systems was FIPS-140 compliance designed for?
The government system that FIPS-140 compliance was designed for is “Sensitive but Unclassified” data.

• Understand U.S. Government data classification hierarchy.
– Example: Describe the Top-Secret data category. What type of data does it contain? Give examples.

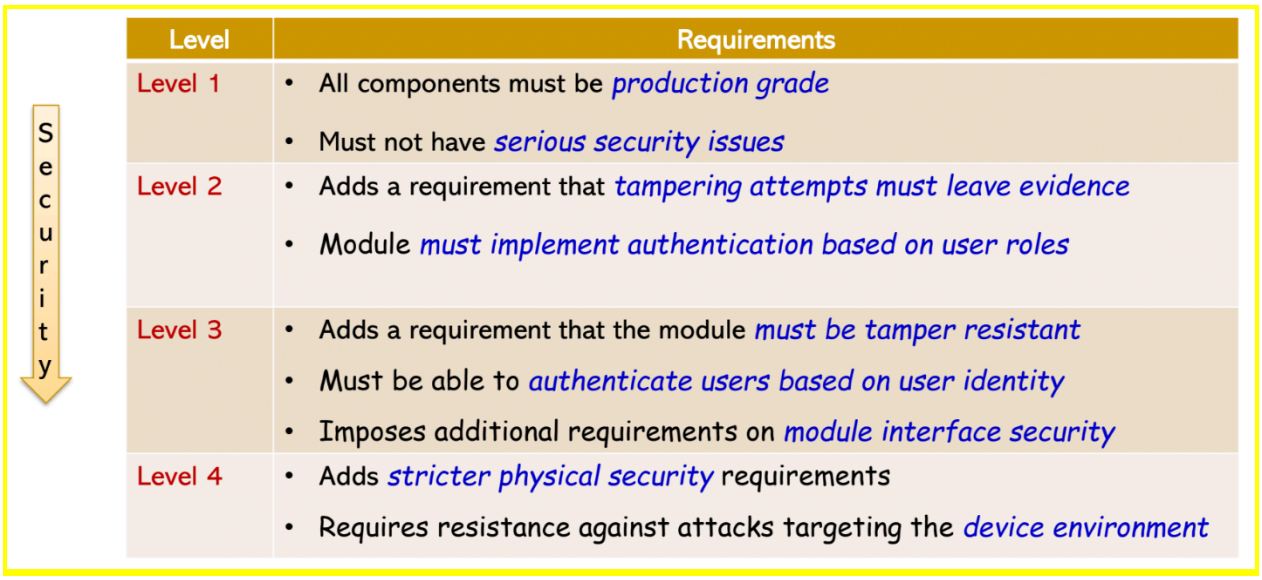


Secret: Can cause serious damage to national security if disclosed
Examples:
• Troop deployments
• Military operation plans
Confidential: Can cause damage to national security if disclosed
Examples:
• NSA secrets (what Edward Snowden leaked)
Sensitive but Unclassified: Will not cause serious damage to national security if disclosed
Examples:
• Payroll data
• Tax IDs of individuals
• Purchasing information

Unclassified: Will not impact national security if revealed; accessible to all
Examples:
• Public news briefings
• Information on the DoD public website
A mnemonic (bottom up) : US Can Stop Terrorism

• Do not need to know all 11 areas of FIPS-140 requirements. However, be able to give examples of and explain a few such areas.
Cryptographic module specification: what must be documented
Cryptographic module ports and interfaces: what information enters and exits the module and how it should be separated (How will people or systems be interacting with this module, how does data enter and exit the module)
Roles, services and authentication: what users are allowed to perform what actions with the module and how these users are authenticated (What are users able to do based on their role, how are users authenticated before using the module)
Finite state model: documentation describing what states of the module and how transitions between the states happen (State diagram and how you transition from each state)
Physical Security: Whether module should be resistant to physical tampering; and Whether if physically tempered with, tempering should be easily detectable; and Whether the module should be robust to extreme environmental conditions
Operational Environment: the type of operating system utilized by the module and what operating systems can use the module (What OS does it run?)
Cryptographic key management: Specifies how the keys are generated, stored, and destroyed
Electromagnetic Interference (EMI)/Electromagnetic Compatibility (EMC): All electronics generate electromagnetic fields that can expose sensitive data and interfere with other electronics; This area outlines concerns and security requirements regarding module's EMI/EMC
Self-tests: specifies how the module should test itself and when and how to act if a particular test fails (e.g., power-up tests, conditional tests, etc)
Design assurance: documentation requirements showing that the module was well designed and implemented
Mitigation of attacks: if the module is designed to mitigate a specific attack, then it must included in the documentation

• Know that FIPS-140 provides 4 levels of compliance with the first level being the highest. Do not need to know what each level encompasses.
Level 1 is least secure while Level 4 is the most secure.



Cryptographic Concepts

• What is the plaintext, ciphertext, encryption, decryption, and secret key? How are these elements used to protect data?
Plaintext: original message (anyone can read)
Ciphertext: coded message
Enciphering (encryption): converting plaintext to ciphertext
Deciphering (decryption): restoring the plaintext from ciphertext
You use encryption to the plaintext using the key to get the ciphertext so unwanted parties cannot read the ciphertext. The approved party then would be able to decrypt the ciphertext with the key to be able to read it.

• What are the differences between substitution, transposition, and product ciphers?
Substitution: each element (a bit or a letter) in the plaintext is mapped into another element
Transposition: elements in the plaintext are rearranged
Product: multiple stages of substitutions and transpositions

• What is the difference between symmetric and public-key cryptography?
Symmetric key algorithm uses a single key while public-key cryptography uses two keys.

• What is the difference between a block and stream cipher?
Block: process one block of elements at a time, producing an output block for each input block (AES converts 16 elements at a time (uses padding if it is not 16 elements))
Stream: process the input elements continuously, producing one element at a time (a bit/byte/character at a time)

• Describe the building blocks and dynamics of the symmetric cryptography.
Two parties share the same secret key. The same key is used to encrypt and decrypt (Does not mean you use the same algorithm for both encryption or decryption).

• What are the security requirements for secure symmetric key cryptography?
A strong encryption algorithm: the opponent should be unable to decrypt ciphertext or discover the key even if he/she has a number of ciphertexts and the plaintext that produced each ciphertext
Must assume that the opponent knows the algorithm. However, that knowledge must not suffice in deducing the secret key or the message contents.
A third party that can securely distribute the key to the two communicating parties

X: message, K: encryption key, Y: ciphertext E: encryption function D: decryption function
Y = E(K,X)
X = D(K,Y)
An opponent can observe Y, but do not have access to K or X.

• What is Kerckhoff's Principle? Explain.
Kerckoff's Principle is that a cryptographic system must remain secure even if everything about it except the secret key is known. Assume the attacker knows everything about your scheme.

• Be able to use openssl for encryption, decryption, digital signature, etc.
Generating a (symmetric) AES 256-bit cryptographic key:
openssl enc - nosalt - aes - 256 - ecb - k hello - aes - P
openssl: a popular *nix system cryptographic utility
-aes-256-ecb: the encryption algorithm
-k: the passphrase used for generating the key ("hello - aes " in this example)

Encrypting using the key generated in the previous slide
openssl enc - nosalt - aes - 256 - ecb - in plain.txt - out message.txt.enc - base64 - K E8B6C00C9ADC5E75BB656ECD429CB1643A25B111FCD22C6622D53E0722439993
enc: tells the program to perform encryption
openssl: a popular *nix system cryptographic utility
-nosalt : do not use salting (more later; don't worry about this now)
-aes-256-ecb: encrypt using the AES symmetric key encryption algorithm with a 256-bit key
-in plain.txt: specifies the name of the file to encrypt (plain.txt in this example)
-out message.txt: the name of the file where to save the encrypted contents (message.txt.enc in this example)
-base64 –K: the encryption key to use (we used the key generated in the previous slide)

openssl enc -nosalt -aes-256-ecb –d -in message.txt.enc -out message.txt.dec -base64 -K E8B6C00C9ADC5E75BB656ECD429CB1643A25B111FCD22C6622D53E0722439993
In order to decrypt the file in the previous slide (message.txt.enc), we simply:
Add a – d argument after – aes - 256 - ecb as shown above

Provide the name of the encrypted file as input (message.txt.enc ; the file we created in the previous slide)
Provide the name of the file to store the decrypted contents (message.txt.dec)

• What is the difference between a bruteforce and a cryptanalytic attack?
A bruteforce attack is when the attacker tries all possible keys on a piece of ciphertext until an intelligible translation is obtained. A cryptanalytic attack is when the attacker relies on the nature of the algorithm and uses some knowledge of the plaintext (uses mathematical analysis to try to analyze the difference between the plaintext and the ciphertext).

• Describe the differences between the ciphertext-only, known-plaintext, and chosen-plaintext attack.
Ciphertext-only: Attacker only has access to the ciphertext.
Attempts to use statistical techniques to discover the key and/or plaintext.
Known-plaintext attack: Attacker has access to the ciphertext and knows some properties of the plaintext
Example: the plaintext is known to contain a certain sentence.
Attacker only has access to a limited amount of plaintext and ciphertext pairs
Chosen-plaintext attack: Attacker can encrypt any plaintext using the target encryption scheme
Uses analysis in order to figure out the function of the cryptographic system (and hopefully the secret key!).
Attacker can generate as many plaintext and ciphertext pairs

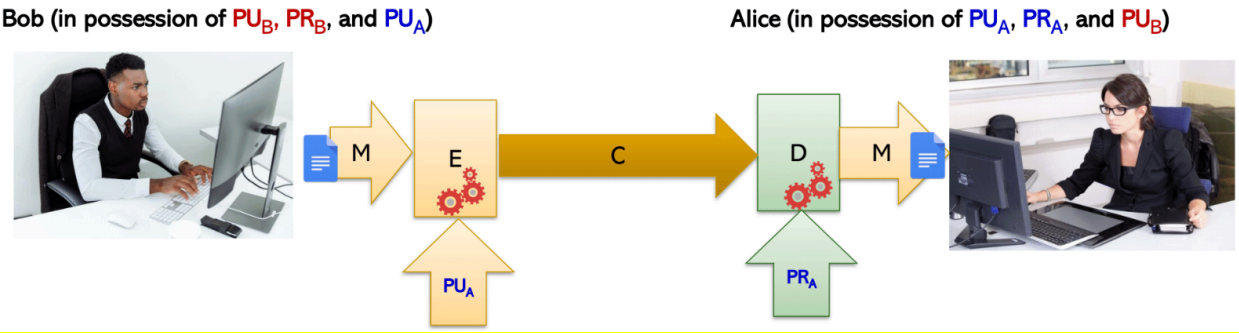
Public Key Cryptography

• Describe the requirements and the basic building blocks of public cryptography?
Each communicator generates two mathematically linked keys. The keys are called public and private, respectively. The public key is made available to the public.The private key is kept private. Sender sends the message encrypted using the receiver's public key. The receiver decodes the messages using their private key.

• Show how public key cryptography can be used to provide confidentiality.
Only the receiver can decrypt the message since only the receiver has their private key.

Confidentiality Example: Bob wants to send a message (M) confidentially to Alice

- Bob obtains Alice's public key (PU_A) and uses it to encrypt the message
- Alice decrypts the message Alice's private key (PR_A)

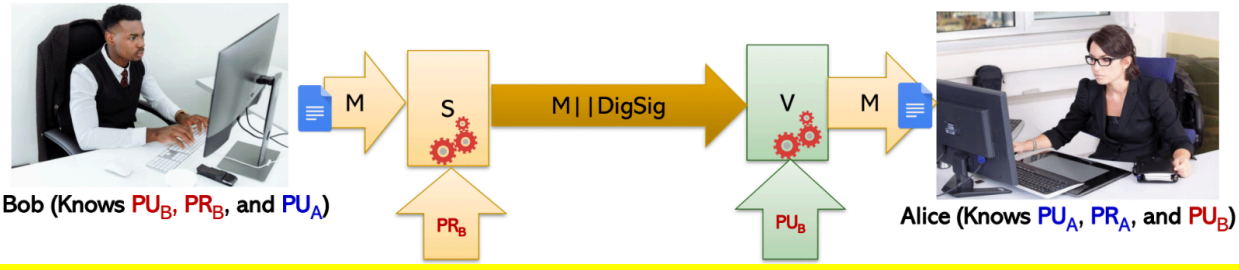


• Show how public key cryptography can be used to provide digital signature. Explain why the digital signature provides assurance that the message is from the source claimed?

The DigSig or digital signature helps provide assurance that the message is from the source claimed since the receiver can use a verification function that takes in the public key of the source, the message, and the digital signature.

Digital Signature Example: Bob wants to send a digitally signed message (M) to Alice

- Bob encrypts the message using his private key (PR_B)
- Alice obtains Bob's public key (PU_B) and uses it to decrypt the message
- S(PR, M): a function that creates a digital signature of message M using the private key PR
- V(PU, M, DigSig): a function that verifies the signature DigSig of message M using the public key PU. Outputs true if the signatures match, and false otherwise.
- || is the append operation
- If the decryption succeeds, the message must be from Bob



• Show how public key cryptography can be used to provide both confidentiality and digital signature.

To show both confidentiality and digital signature, Bob would sign the message using his private key and then encrypt the message using Alice's public key.

• Is public key cryptography inherently more secure than symmetric key cryptography
Yes, if one party leaks the symmetric key then the channel is no longer secure. On the other hand if a party leaks their private key, only the messages of the party are no longer secure.

• Can public key encryption viably replace symmetric key encryption schemes?
No, public key encryption is much slower and less expensive.

openssl genrsa –out private -key.pem 2048
Generates the private key
genrsa: generate a private key for the RSA algorithm – a popular public key encryption algorithm
Specifies to save the generated private key in the file called private - key.pem
2048: the size of the key to generate (in bits)

openssl rsa -pubout -in private-key.pem -out public-key.pem
Generates a public and private keys
rsa: specifies to use the RSA algorithm – a popular public key encryption algorithm
Specifies to save the generated public key in the file called public-key.pem
Specifies to save the generated private key in the file called private-key.pem

openssl rsautl -encrypt -in plain.txt -out plain.txt.enc -inkey public-key.pem – pubin
Encrypt file plain.txt using RSA and public key stored in file public-key.pem (the one we generated in the previous example)
Store the encrypted file contents in plain.txt.enc

openssl rsautl - decrypt -in plain.txt.enc -out plain.txt.dec -inkey private-key.pem
-decrypt: decrypt the file
plain.txt.enc: the file to decrypt using RSA and private key stored in file private-key.pem (the one we generated in the previous example)
Store the decrypted file contents in plain.txt.enc

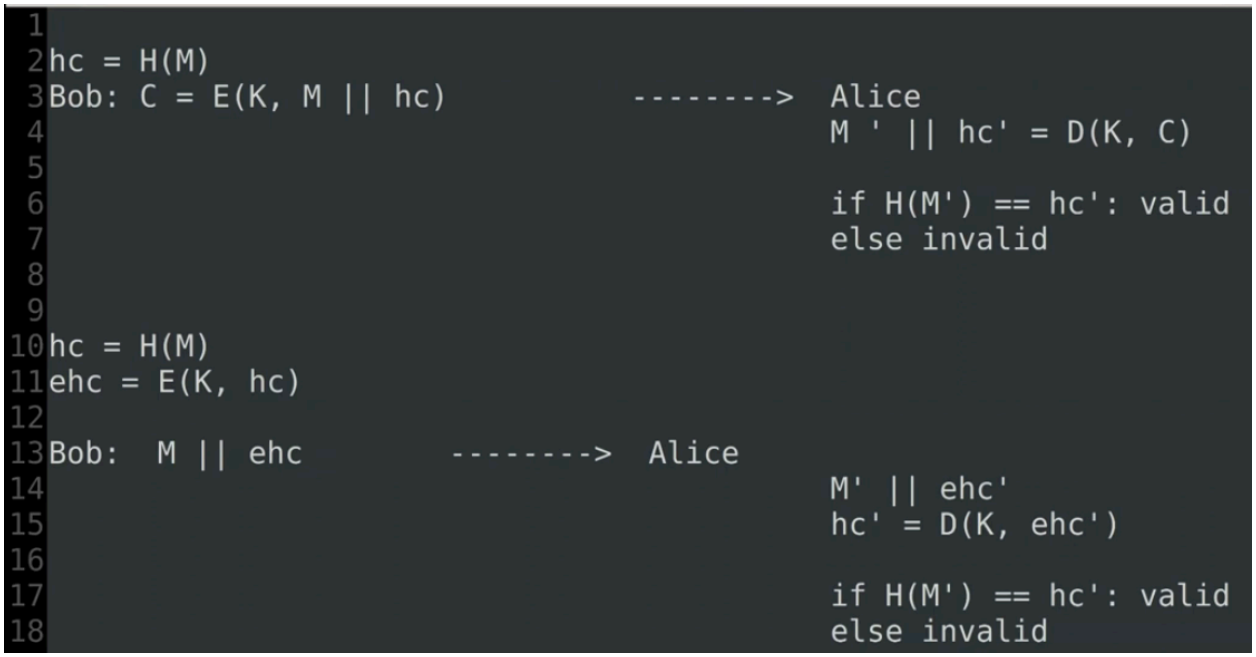
Hashing and Message Authentication Codes (MACs)

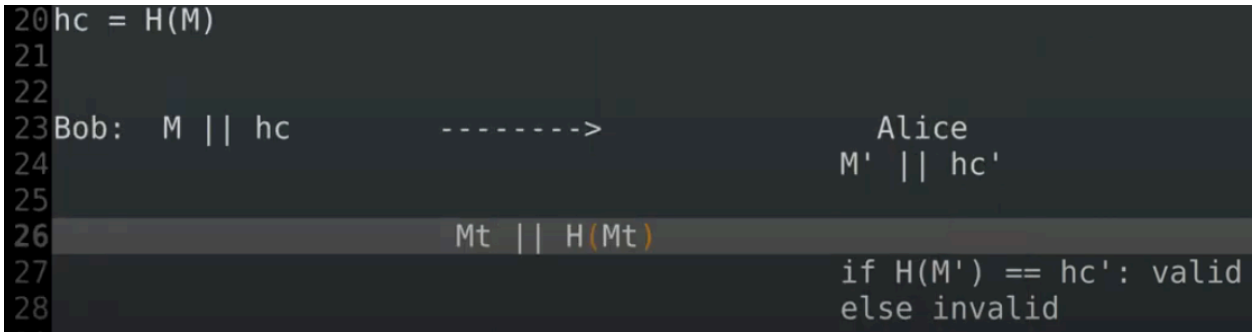
• What is a hash function? Why is it important? What are its applications?
– Example: Illustrate how hashing can be used to provide proof of work?
A hash function is a one-way function that accepts a variable-size message M as input and produces a fixed-size output, referred to as a hash code H(M). The purpose of a hash is to produce a fingerprint of a file. This can be used to detect changes to a message. Often used to create a digital signature or verify file integrity.

• What are the security requirements of the hash function?
More bits for the hash means better since it protects against birthday attacks.

• Be able to illustrate the use of hash functions for e.g., integrity and digital signature.

You have to encrypt a hash since an attacker can just modify the message and the hash.





• What is the difference between a hash function and a MAC?
A MAC function requires a secret key to generate the correct output. A hash function does not require a secret key.
A MAC does not provide a digital signature since if you have the MAC key you can generate another message (provides authentication but not confidentiality).

• Compare and contrast symmetric cryptography and a MAC function.
Symmetric cryptography is two way since the message can both be encrypted and decrypted. A MAC function is a function so you cannot go backwards.

Basic Cryptographic Techniques for Authentication and Confidentiality

• Be able to explain schemes for confidentiality, integrity, and digital signature discussed in class and be able to design your own based on requirements.
Symmetric Encryption:

Provides confidentiality (Force the plaintext to have some structure, e.g. add a hash function to the message)
Does not provide digital signature
Public-key Encryption:
Provides confidentiality and authentication
Digital signature is created using hash and public key cryptography

```
Digital Signature: hash and public key cryptography

Alice wants to send a digitally signed message to Bob:

Let M be the message
Let PUa/PRa be Alice's public/privats keys, respectively

hc = H(M)
DigSig = Sign(PRa, hc)

M || DigSig ------------------> M'||DigSig'
                                hc' = hash(M')
                                Verify(hc', PUa, DigSig')

Digital Signature Only
```

```
Digital Signature + Confidentiality

Let Bob and Alice share a symmetric key K

hc = H(M)
DigSig = Sign(PRa, hc)

C = E(K, M || DigSig) ------------------> M'||DigSig' = D(K, C)
                                hc' = hash(M')
                                Verify(hc', PUa, DigSig')
```

```
Two people who need to sign the same message:

Alice:       PUa/PRa      (public/private)
Mad Hatter:  PUmt/PRmt    (PUMt/PRmt)

Alice: hc = H(M)
DigSigA = Sign(PRa, hc)


Bob: hc = H(M)
DigSigB = Sign(PRb, hc)

M || SigSigA || DigSigB ---------------------->

----------
Two messages one signature:

Two messages: M1 and M2
```

```
Two messages one signature:

Two messages: M1 and M2

hc = H(M1 || M2)
DigSig = Sing(PRa, hc)

M1 || M2 || digSigA
```

```
Message M

mc = MAC(K, M)
wdDigSig = Sign(PRa, mc)   M || wdDigSig ---->   mc' = MAC(K, M)
Verify(mc', PUa, wddigSig)
```

• What is the difference between unconditional and computational security?

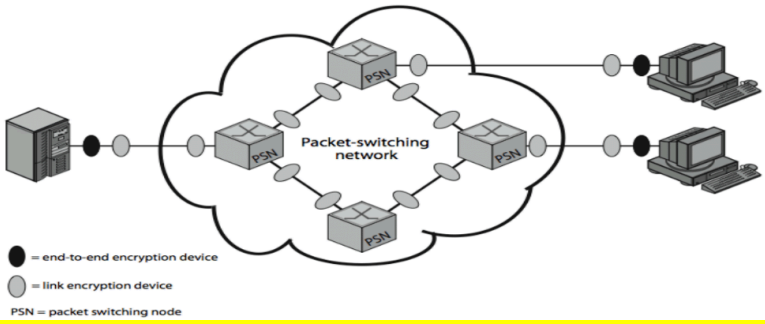
Unconditional security (Unbreakable):
No matter how much computer power or time is available, the cipher cannot be broken since the ciphertext provides insufficient information to determine the corresponding plaintext
No encryption algorithm is unconditionally secure except the one - time pad

Computational security (Breakable but will take a long time):
Given limited computing resources (e.g., time needed for calculations is greater than age of universe), the cipher cannot be broken
The cost of breaking the cipher exceeds the value of the encrypted information
The time required to break the cipher exceeds the useful lifetime of the information

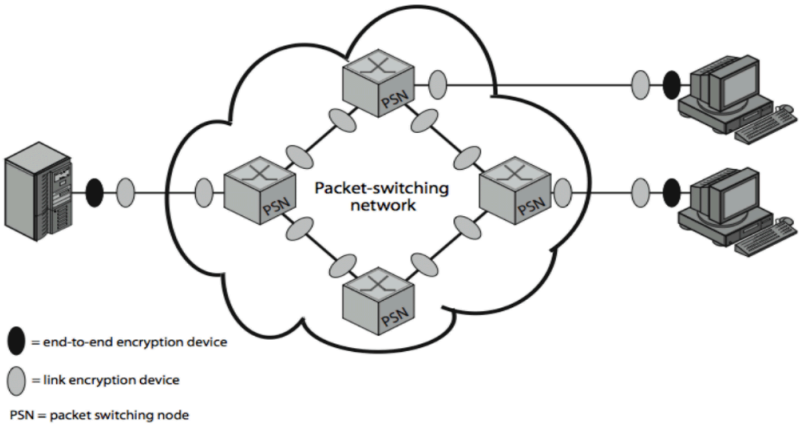
Confidentiality using Symmetric Encryption

• Be able to explain the implementation, advantages, and disadvantages of link and end-to end encryption.

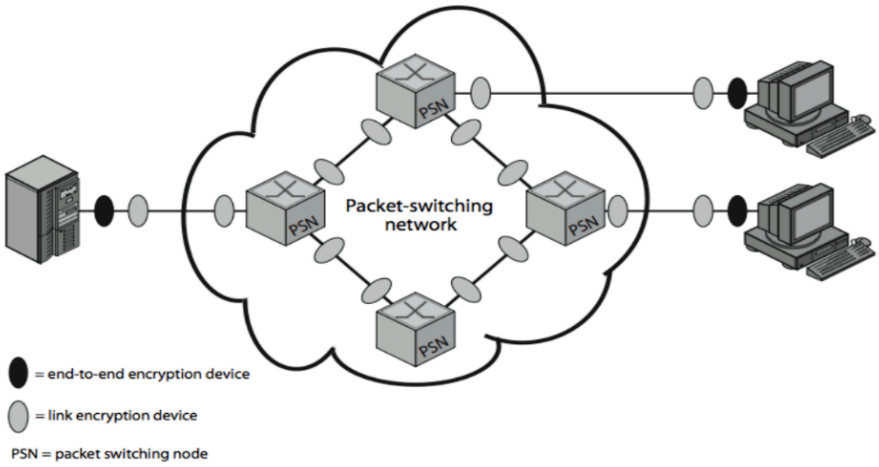
- Link encryption
  - Encryption occurs independently on every link
  - Each vulnerable communications link is equipped on both ends with an encryption device.
  - Requires many encryption devices in a large network
  - Each pair of nodes that share a link should share a unique key – many keys must be provided



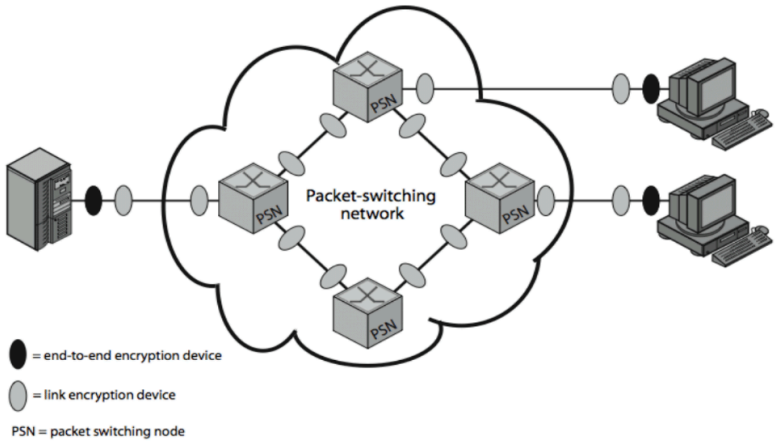
- Disadvantage: the message must be decrypted each time it enters a switch because the switch must read the address in the packet header in order to route the message - the message is vulnerable at each switch.



- End-to-end encryption
  - Encryption occurs between original source and final destination.
  - The data in encrypted form are transmitted across the network to the destination terminal or host.



- End-to-end encryption
  - The destination shares a key with the source and hence is able to decrypt the data.

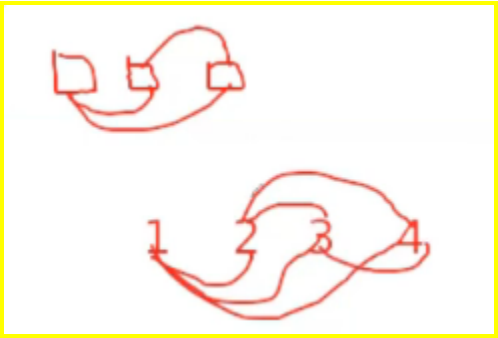


– Example: Suppose we would like to implement link encryption on a network with 5 links. How many symmetric keys will we need?
You would need 5 symmetric keys since you have 5 links.

– Example: In link encryption, can we encrypt the entire packet including headers? Explain.
Yes, the whole packet is encrypted. This is because encrypt the packet before putting it on the link and the packet travels one way. However, in end-to-end encryption the packet header is unable to be encrypted. This is because the switches would be unable to know where to forward the packet.

N = systems
# of e2e keys = N *(N-1)/2

For # of keys in end-to-end encryption.



• Why do we (ideally) need both link and end-to-end encryption?

End-to-end protects data contents over the entire path and provides authentication. Link protects traffic flows from monitoring. Both cover each other's weaknesses well since link is able to encrypt headers and end-to-end protects against compromised switches.

• What is traffic analysis? What is the countermeasure?
Traffic analysis is the process of monitoring communications flows in order to deduce information from patterns in communication. The countermeasure is to use traffic padding. By creating a constant stream of data the attacker would be unable to monitor the flows of data.

Distribution of Symmetric Session Keys Using Symmetric Key Cryptography

• What is symmetric key distribution? Why is it important?
Symmetric key distribution refers to the same session key that is being distributed between two users. They encrypt the session using that key. This key only exists until the session is over.

Key Distribution

- For symmetric encryption to work, the two parties must share a common key and that key must be protected from access by others.
- Frequent key changes are usually desirable to limit the amount of data compromised if an attacker learns the key.
- Key distribution: refers to the means of delivering a key to two parties who wish to exchange data, without allowing others to see the key
- Often secure systems fail due to a break in the key distribution scheme

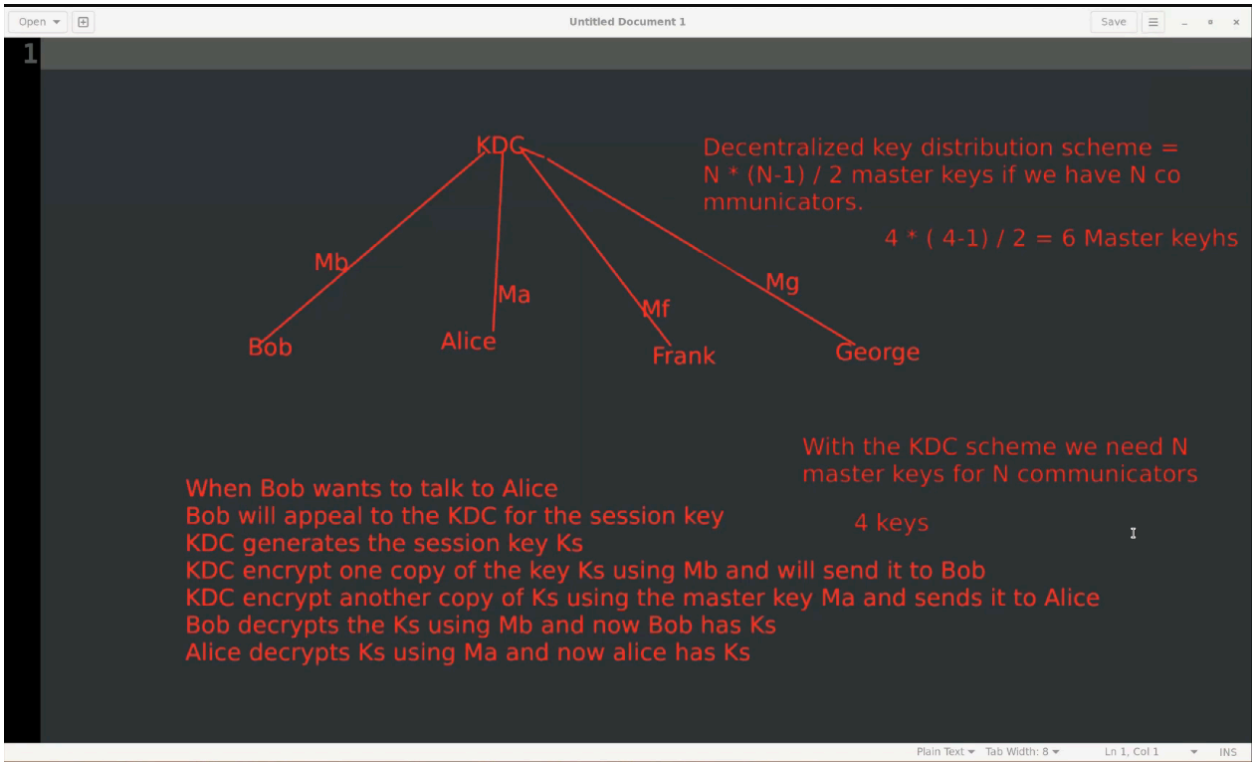
• Be able to explain the key hierarchy.
Use master keys to securely distribute session keys. You use a single key (master key) to distribute multiple session keys.

Key Hierarchy

- The use of a key distribution center is based on the use of a hierarchy of keys.
- Master key
  - Used to encrypt session keys
  - Shared by user & key distribution center
- Session key
  - Temporary key
  - Used for encryption of data between users

• Why use symmetric master keys to distribute session keys?

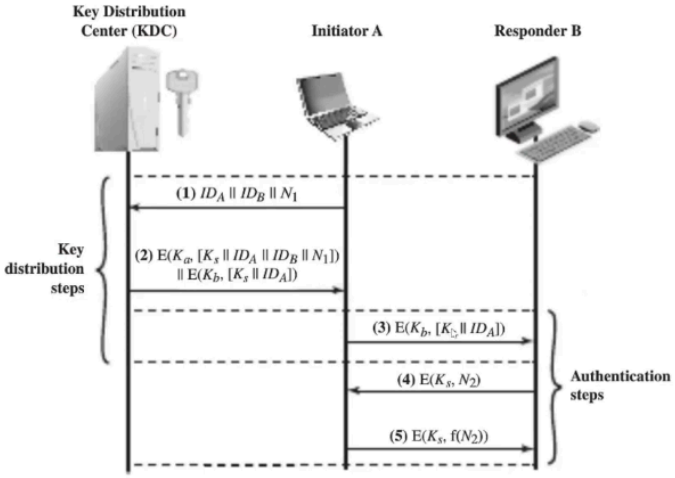
Master keys do not change very often, while new session keys are needed for each new session.

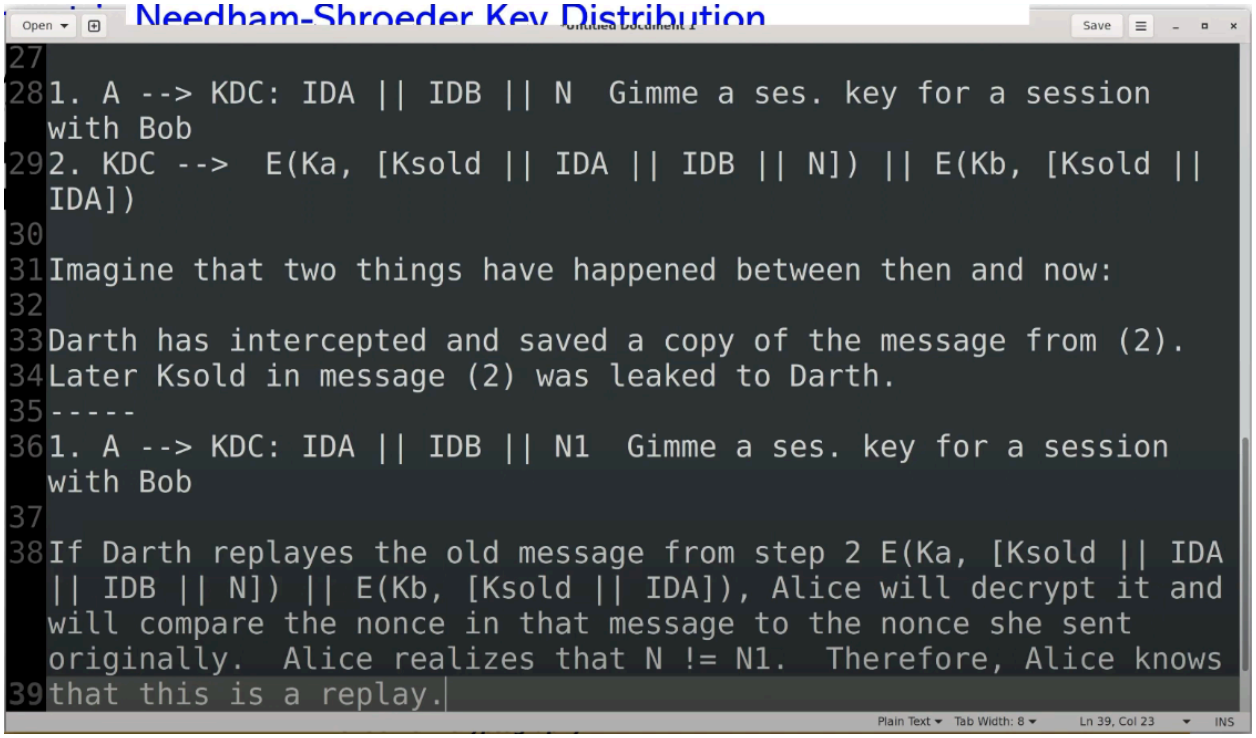


• Understand the Symmetric Needham-Schroeder Key Distribution protocol. What are its key components? What is the basic flow of messages? How does the protocol use nonces and time stamps to prevent replay attacks?

Symmetric Needham-Shroeder Key Distribution

- A wishes to establish a logical connection with B and requires a one-time session key to protect the data transmitted over the connection
- A shares the master key K_a with the KDC
- B shares the master key K_b with the KDC





N is nonce to prevent replay attacks.
Step 1: Send the ID of Alice and Bob to the KDC.
Step 2: Send a session key and a copy of Step 1 back to Alice that is encrypted with her master key. Also send a session key and the ID of Alice that is encrypted with Bob's master key.
Step 3: Alice sends Bob the second part of Step 2.
Step 4: Bob encrypted a nonce with the session key (to see if confidentiality has been breached).
Step 5: Alice returns the nonce with a transformation function (etc. adding 1 to the nonce). Cannot return the nonce without a transformation due to replay attack.

– Example: Why does KDC echo Alice's message in step (2)?
KDC echoes Alice's message in step 2 so Alice knows which key corresponds to which session.

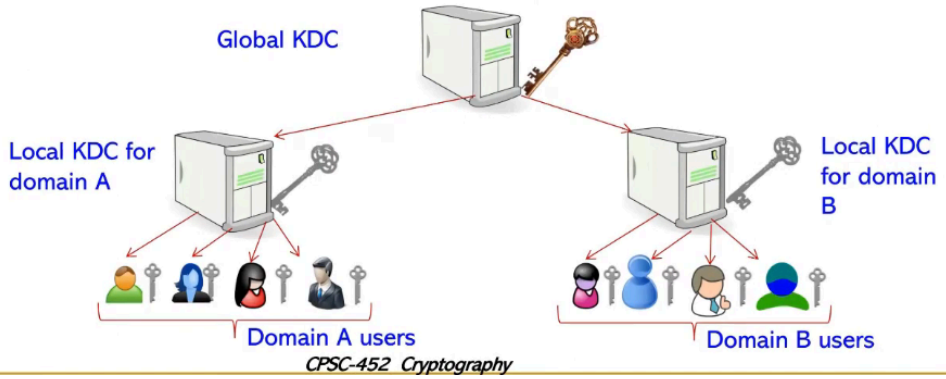
– Example: Is Alice able to decrypt the entire message from the KDC? Explain.
No, because Alice does not have Bob's master key. This simplifies the job of the KDC so Alice can directly forward the message to Bob.

– Example: Five users want to communicate with each other securely using symmetric key cryptography only. How many keys must each user have in the Key Distribution Center (KDC)-based scheme?
We need as many master keys as we have communicators. Therefore, 5 keys are needed.

• What are the advantages of using a hierarchy of KDCs?

Hierarchical Key Control

- It is not necessary to limit the key distribution function to a single KDC – for large networks, a hierarchy of KDCs can be established
  - E.g. local KDCs, each responsible for a small domain
  - If two entities are in different domains, then local KDCs can communicate through a global KDC.



CPSC-452 Cryptography 21/121

Local KDC can manage session keys for a domain while the Global KDC handles session keys between two different domains.

• Be able to find flaws in symmetric key distribution protocols.
– Example: Assume a key distribution scheme where Bob and Alice both share the symmetric master key MBA. To distribute the session key Alice generates the session key KS encrypts it using MBA and sends the message IDA || E(MBA, KS).

This is vulnerable to a replay attack since there is no nonce to confirm that the message is being replayed.

Distribution of Secret Keys Using Public-Key Cryptography

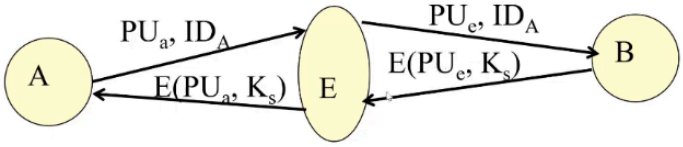
• Why use public key cryptography to distribute session keys? That is, why not just use public key encryption to send bulk data?
Public key cryptography is much slower than symmetric key cryptography since there are much more complicated calculations involved. Instead, we use public key cryptography to distribute session keys.

• Be able to identify vulnerabilities in key distribution protocols.
– Example: Consider the Merkle Simple Key Distribution protocol. Is it secure? If so, explain how so and if not so, explain the flaw.

No, this protocol is susceptible to a man in the middle attack.

Simple Secret Key Distribution: Attack

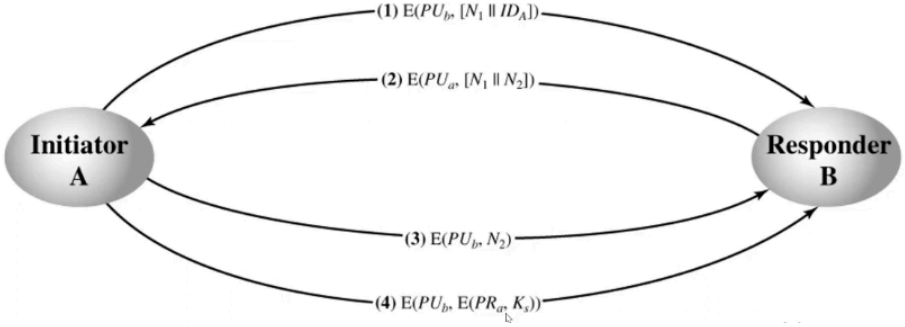
- Problem: man-in-the-middle attack – an adversary can intercept messages and then replay the intercepted message or send another message.



- A transmits a message intended for B consisting of PU_a and A's identifier ID_A
- E intercepts the message, creates its own public/private key pair $\{PU_e, PR_e\}$ and transmits $PU_e||ID_A$ to B
- B generates a secret key K_s and transmits $E(PU_e,K_s)$
- E intercepts the message, and learns Ks by computing $D(PR_e, E(PU_e, K_s))$
- E transmits $E(PU_a,K_s)$ to A
- Both A and B know K_s and are unaware that K_s has been revealed to E.

Simple Secret Key Distribution: Solution

1. A uses B's public key to encrypt a message to B containing ID_A and a nonce N_1
2. B sends a message to A encrypted with PU_a and containing A's nonce N_1 and a new nonce generated by B (N_2). Because only B could have decrypted message 1, N_1 in message 2 assures A that the correspondent is B



• Explain how distribution of symmetric keys using public keys can be combined with the Symmetric Key Needham Shroeder protocol for greater efficiency.
Public key cryptography is slow and expensive while session keys need to be created often. It is easier to use public key cryptography to generate session keys.

• Diffie-Hellman Key distribution: – What is the application of the Diffie-Hellman approach? What are its advantages?
– Be able to show that number a is a primitive root of number q.
$a^i \bmod q$
∗ Example: Show that 3, 5, 6, 7, 10, 11, 12, and 14 are primitive roots of 17.
a = 3
q = 17

Is q = 17 prime? Yes

- An integer a is a primitive root of a prime number q if $a \bmod q$, $a^2 \bmod q$, …, $a^{q-1} \bmod q$ are distinct and consist of the integers from 1 through q-1 in some permutation.

$a^1 \bmod 17 = 3$
$a^2 \bmod 17 = 9$
$a^3 \bmod 17 = 10$
$a^4 \bmod 17 = 13$
$a^5 \bmod 17 = 5$
$a^6 \bmod 17 = 15$
$a^7 \bmod 17 = 11$
$a^8 \bmod 17 = 16$
$a^9 \bmod 17 = 14$
$a^{10} \bmod 17 = 8$
$a^{11} \bmod 17 = 7$
$a^{12} \bmod 17 = 4$
$a^{13} \bmod 17 = 12$
$a^{14} \bmod 17 = 2$
$a^{15} \bmod 17 = 6$
$a^{16} \bmod 17 = 1$

etc. all values must be from 1 to q-1 (values cannot repeat)

– Know how to compute public, private, and symmetric keys.

Diffie-Hellman Key Exchange

- An integer a is a primitive root of a prime number q if $a \bmod q$, $a^2 \bmod q$, …, $a^{q-1} \bmod q$ are distinct and consist of the integers from 1 through $q-1$ in some permutation.
- There are two publicly known numbers:
  - A prime number q
  - An integer a that is the primitive root of q
- Each user generates his/her key
  - Chooses a private key (number): $x < q$
  - Computes their public key: $y = a^x \bmod q$
- Each user keeps the x value private and makes the y value available publicly

∗ Example: Assume Bob and Alice choose the public values a = 3 and q = 31. If this is a valid setup, show how Bob and Alice can compute the public/private keys and how they use them to arrive at the same symmetric key.

Yes, a is a primitive root of q.

Bob picks private key Xb = 8. Since Xb = 8 < q = 31, this is a valid choice.
Alice picks private key Xa = 9. Since Xa = 9 < q = 31, this is a valid choice.

Public Key Computation:

Bob: $Yb = a^{Xb} \bmod q = 3^8 \bmod 31 = 20$
Alice: $Ya = a^{Xa} \bmod q = 3^9 \bmod 31 = 29$

Computing Shared Session Key:

Bob receives the public key Ya from Alice
$K = Ya^{Xb} \bmod q = 29^8 \bmod 31 = 8$

Alice gets Bob's public key Yb
$K = Yb^{Xa} \bmod q = 20^9 \bmod 31 = 8$

Let a = 2 and q = 11.

1. Is q = 11 a prime number? Yes
2. Is a = 2 a primitive root of q = 11?

$2^1 \bmod 11 = 2$
$2^2 \bmod 11 = 4$
…
$2^{11-1} \bmod 11 = 1$
[2, 4, 8, 5, 10, 9, 7, 3, 6, 1]

Bob picks private key Xb = 5 which is less than q = 11.
Alice picks private key Xa = 7 which is less than q = 11.

Bob computes public key: $Yb = 2^5 \bmod 11 = 10$
Alice computes public key: $Ya = 2^7 \bmod 11 = 7$

Bob receives Ya = 7 from Alice
$K = Ya^{Xb} \bmod q = 7^5 \bmod 11 = 10$

Alice receives Yb = 10 from Bob
$K = Yb^{Xa} \bmod q = 10^5 \bmod 11 = 10$

$a^1 \bmod q =$
$a^2 \bmod q =$
$a^3 \bmod q =$
…
$a^{q-1} \bmod q =$

– What makes it difficult for the attacker to derive the private keys or the shared symmetric keys even if the attacker has access to the public information used for the Diffie-Hellman key exchange?
All numbers from 1 to q-1 are possible private keys. Therefore if q is a larger number, the attacker needs to go through a long list of possible private keys.

– Understand how man-in-the-middle attack against Diffie-Hellman works. What is the fix? Ensuring the Authenticity of Public Keys in Public Distribution Schemes.
a = 5
q = 23
a is a primitive root of q.

Private Keys
Alice: XA = 10
Bob: XB = 5
Darth: XD = 12

Public key Yb for Bob
$YB = a^{XB} \bmod q = 5^5 \bmod 23 = 20$

Public key Ya for Alice
YA= a^{XA}modq = 5^{10}mod23 = 9

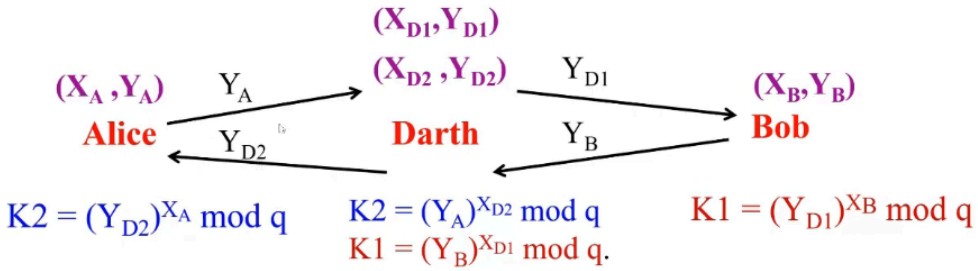
Alice PU + Darth PR: KAD = YA^{XD}mod23 = 9^{12}mod23 = 9

Darth PU + Bob PR: KBD = YD^{XB}mod23 = 18^{5}mod23 = 3

Bob PU + Darth PR: KBD = YB^{XD}mod23 = 20^{12}mod23 = 3

Darth PU + Alice PR: KAD = YD^{XA}mod23 = 18^{10}mod23 = 9

- Man-in-the-Middle-Attack

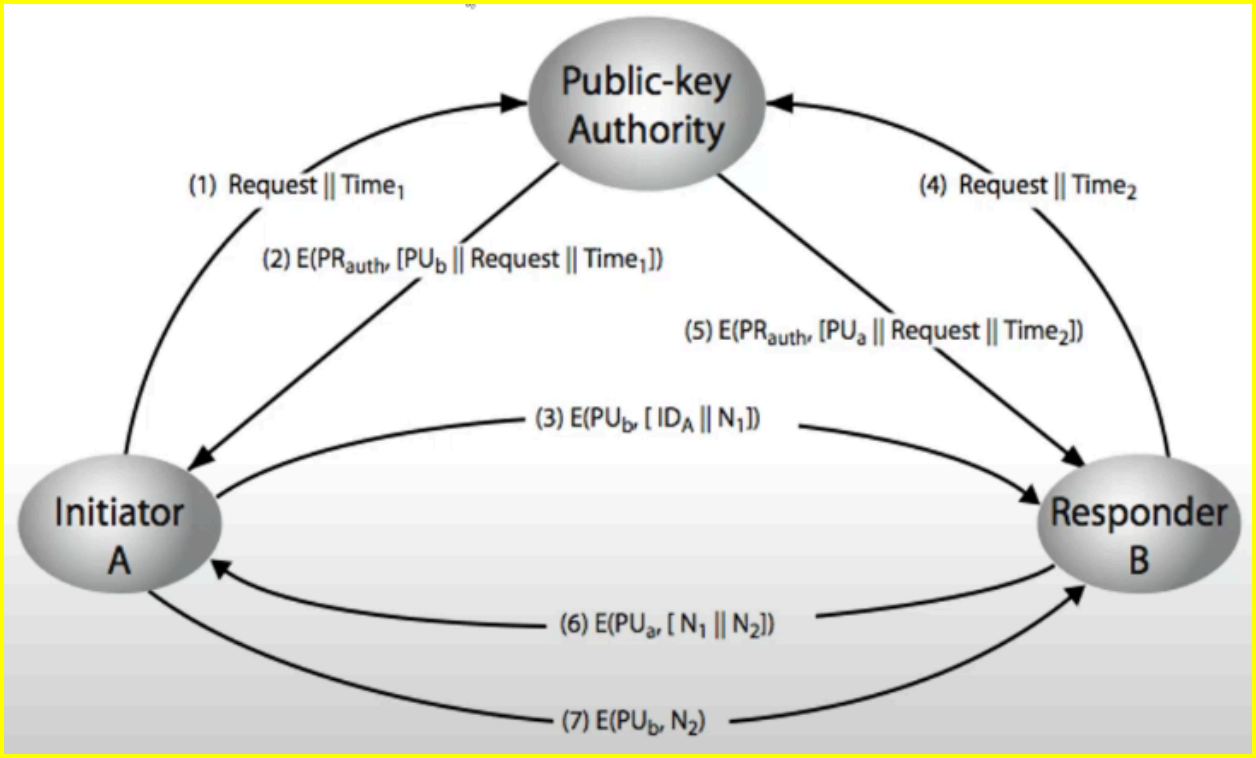


- Now, Alice and Bob think that they share a secret key, but instead Bob and Darth share secret key K1. Darth and Alice share secret key K2.

The fix is to authenticate public keys.

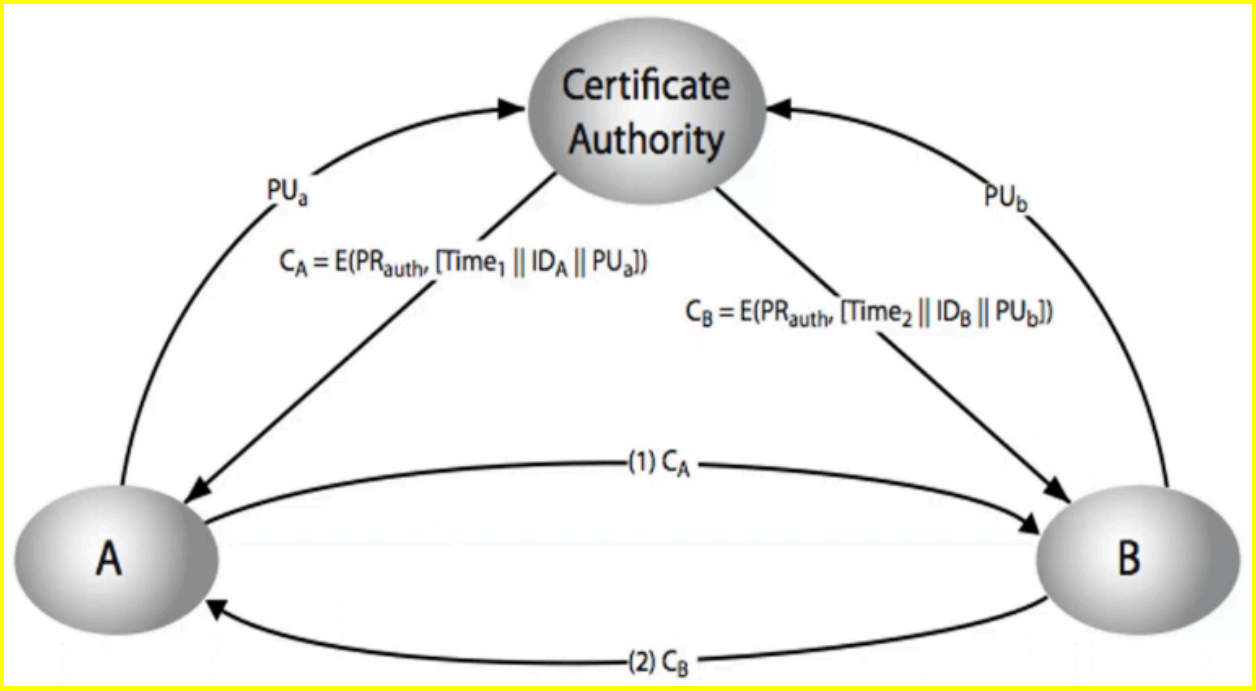
• Understand how all public key management schemes work (and be able to understand the diagrams):
– Explain the public announcement approach to distributing public keys? What are its advantages and disadvantages?
Take your public key and announce it (etc. put it in a forum post)
Advantages: easy to publish your public key
Disadvantage: forgery

– Explain the public key authority-based distribution schemes. What are the requirements, advantages, and disadvantages? (slides 16-20).
Introduce a trusted third party to verify the authenticity of the public keys



1. A sends request to the public-key authority with a timestamp
2. The public-key authority will respond with B's public key, the original request, and the original time stamp (will be digitally signed with the public-key authority's private key)
3. A will store B's public key and use it to encrypt a message to B containing an identifier of A and a nonce (This is used to verify B's identity)
4 & 5. B retrieves A's public key from the authority the same way A retrieved B's public key
6. B sends a message to A that is encrypted using A's public key that contains A's nonce and a new nonce generated by B
7. A returns N2 encrypted using B's public key, to ensure B is corresponding with A
Advantages: has a trusted third party hosting the public keys
Disadvantage: the public-key authority can be a performance bottleneck (public-key authority can have downtime and is the only way to get public keys)

– What are public key certificates? What are their requirements, advantages, and disadvantages? (slides 22-26)?
Create certificates to verify the public key of each user



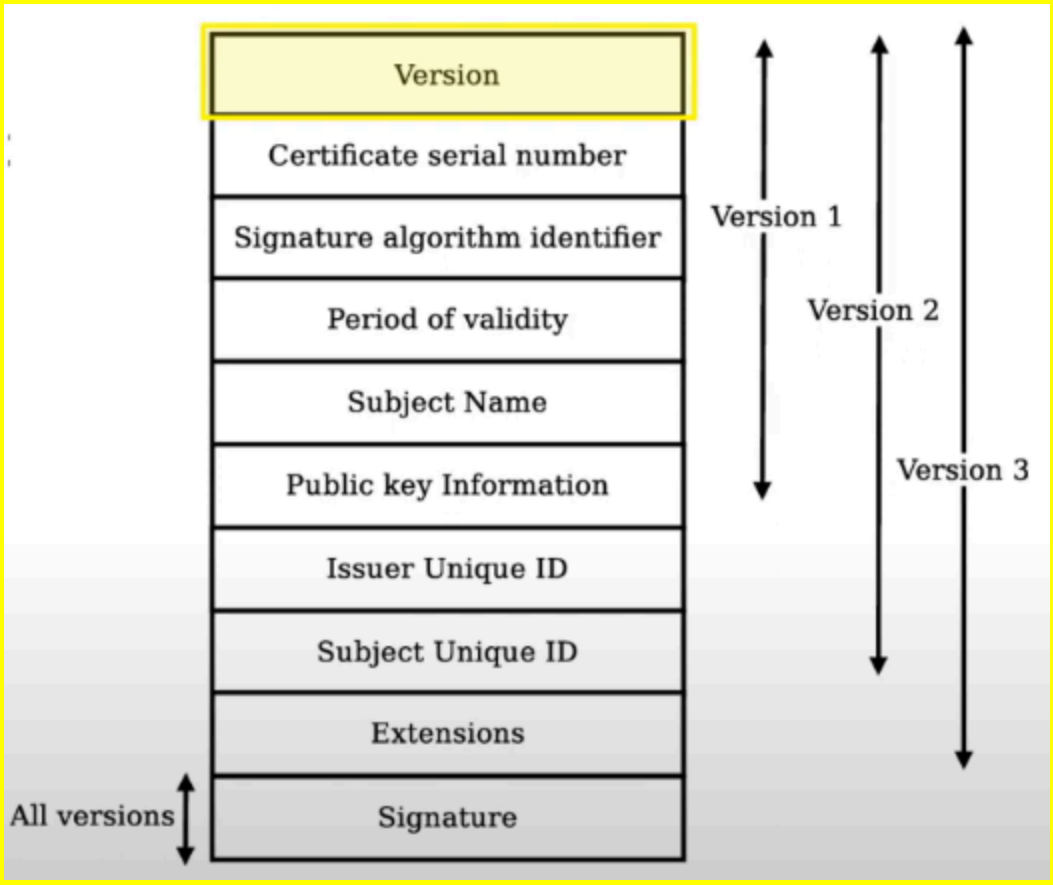
1. Alice applies to the certificate authority supplying a public key and requesting a certificate
2. The returned certificate that is signed by the certificate authority contains the timestamp (expiration date), Alice's ID, and Alice's public key
3. Alice may send the certificate to any other participant since they can decrypt the certificate using the authority's public key

Advantages: can directly ask the user for a certificate for the public key

Disadvantage:

– What are the X.509 certificates? Why are they important? What important information do they contain? How are they created and used and digitally signed?

X.509 is the standard format for public key certificates. Some important information they contain are period of validity, public key information, and signature algorithm identifier.



Certificate version: A value of 1, 2, or 3
Certificate serial number: A unique identifier of the certificate
Signature Algorithm: The digital signature algorithm the certificate authority used to sign the certificate
Validity Period: The time period for which the certificate is considered valid
Subject Name: Name of the entity represented by the certificate
Public Key Info: Public key owned by the subject
Issuer Unique ID: A unique identifier of the issuing CA as defined by the issuing CA
Subject Unique ID: A unique identifier for the certificate subject as defined by the issuing CA
Extensions: The subject of the CA and a serial number of the CA that issued the certificate; or A hash of the public key of the CA that issued the certificate

– What is a self-signed certificate? Does it provide authenticity?
A self-signed certificate is a certificate signed by the user. This does not provide authenticity.

– What are TLS certificate chains? Given a certificate chain, be able to trace and explain how the authenticity of the chain is verified? public key certificate.
TLS certificate chains are when a certificate authority verifies another certificate authority who issued it. shibboleth.fullerton.edu is verified from Incommon RSA server which is verified by USERTrust RSA Certification. The top level is self certified.

– What is a certificate revocation list?
A certificate revocation list is a list of revoked certificates. If the serial number is in the certificate revocation list, then the certificate is revoked.