



NTNU – Trondheim
Norwegian University of
Science and Technology

Security in Computer Networks

TTM4200 – Computer Networks

individual Readiness Assurance Test

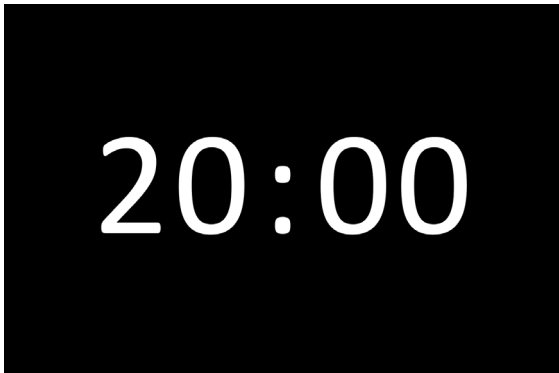
- Find the right answer **alone**.
- Choose the answer that fits **best**.
- **One** handwritten A4 page as helping material.



20:00

team Readiness Assurance Test

- Find the right answer **in teams**.
- Choose the answer that fits **best**.
- **One** handwritten A4 page as helping material.



20:00

Time for a break

we restart at 11:15

TIP

What is Triple DES and why is it disallowed?

Triple DES offered 112-bit security through its three-step encryption process, but NIST deprecated it in 2018 and disallowed its use after 2023.

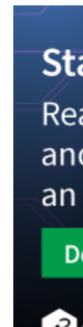


By **Michael Cobb**

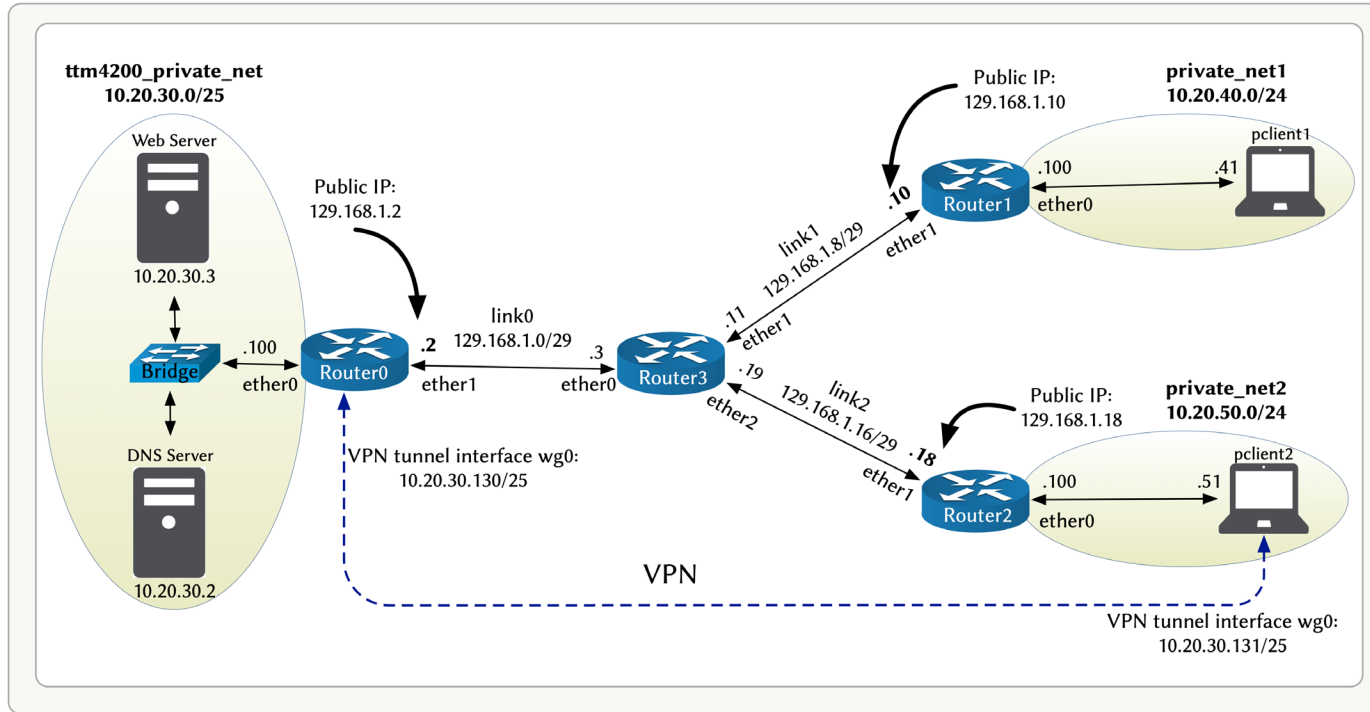
Published: 03 Sep 2025

Triple Data Encryption Algorithm was used widely across many industries and in many popular network protocols to encrypt data at rest and data in motion. NIST deprecated the algorithm in 2018 and disallowed its use after 2023.

<https://www.techtarget.com/searchsecurity/tip/Expert-advice-Encryption-101-Triple-DES-explained>



Final labs



Review Problem 6 (pg. 710)

Suppose N people want to communicate with each of $N-1$ other people **using symmetric key encryption**. All communication between any two people, i and j , is visible to all other people in this group of N , and **no other person** in this group should be able to decode their communication.

a) How many keys are required in the system as a whole?

Now suppose that **public key encryption** is used.

b) How many keys are required in this case?

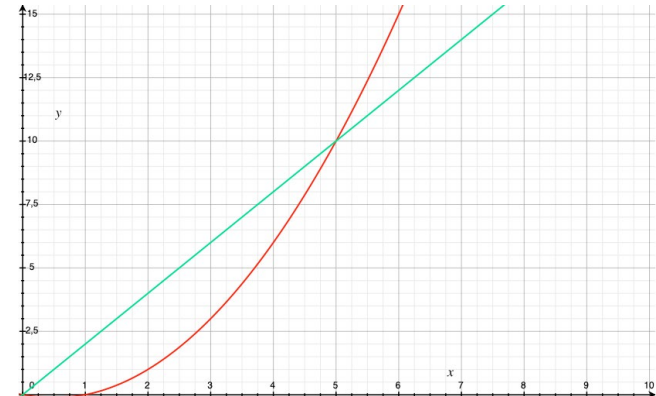
Solving Problem 6 (pg. 710)

a) Each **pair** of N users should have a key so:

$$\#keys = N(N - 1)/2$$

b) With a public-key system each user has 2 keys, one public and one private, so:

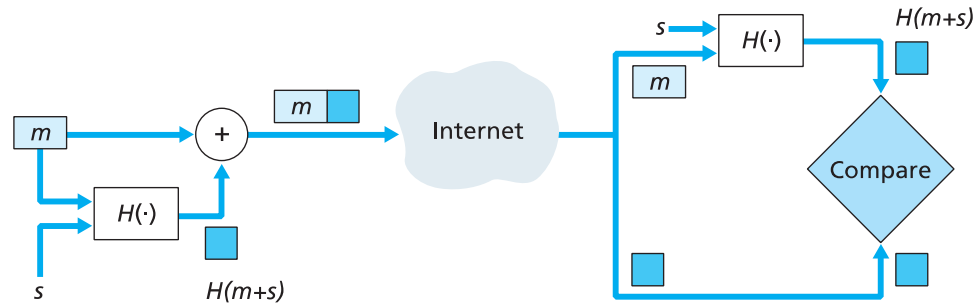
$$\#keys = 2N$$



Review Problem X

Consider a variation of the MAC algorithm (figure below) where the **sender instead sends** $(m, H(m) + s)$, where $H(m) + s$ is the concatenation of $H(m)$ and s .

Is this variation flawed? Why or why not?



Key:

m = Message

s = Shared secret

Solving Problem X

- This scheme is flawed.
- For example, an attacker, can sniff the communication and obtain the shared secret s by extracting the last portion of digits from $H(m) + s$.
- Then the attacker can masquerade as the sender by creating her/his own message t and send $(t, H(t) + s)$.

Exam exercise 1

- Assume that you want to send a confidential and digitally signed contract via a public network. Which combination of information security services and mechanisms are needed to do this securely?
 - w = Digital Signature
 - x = Message Authentication Code (MAC)
 - y = Certification Authority (CA)
 - z = Encryption

Solving Exam exercise 1

- w, y and z but not x
 - Digital Signature for signing the document.
 - CA for verifying the certificate's validity (i.e., obtain/verify public key).
 - Encryption for confidentiality

Exam exercise 2

For message integrity, which of the following statement(s) is/are correct?

- a) *Message integrity is the property that the identity of the sender can be confirmed to be who or what they claim to be.*
- b) *Message integrity is the property that the receiver can detect whether the message sent was altered in transit.*
- c) *Both checksumming and hashing techniques may be used.*
- d) *Generally, a hash provides a better message integrity check than a checksum.*
- e) *To ensure message integrity, the transport layer protocol used to communication the message has to be TCP.*

Solving Exam exercise 2

- b), c) and d) but not a) nor e)
 - Digital Signatures for identity and we can use, for example, HMAC for routing messages
 - Checksumming is a poor hashing function (see page 656).
 - Encryption for confidentiality

Enough for today

See you on Thursday

Welcome

we start at 10:15

Join at [menti.com](https://www.menti.com) | use code 5766 1732

Instructions

Go to

www.menti.com

Enter the code

5766 1732



Or use QR code



RAT

Question 1: Network Security is concerned with:

all of the options. (This is the best answer.)

42.9%

confidentiality, integrity and authentication.

57.1%

access and availability.

0.0%

operational aspects.

0.0%

RAT

Question 6: A Message Authentication Code (MAC):

can be used by routers to \textbf{verify} messages' integrity. (This is the best answer.)

28.6%

requires \textbf{both} a hash function and an encryption algorithm.

26.5%

can be used by routers to \textbf{authenticate} another router's messages.

30.6%

provides \textbf{both} message integrity and digital signatures.

14.3%

RAT

Question 9: Suppose a Certification Authority (CA) contains Alice's certificate, which binds Alice's public key to Alice. This certificate is signed with:

The CA's private key. (This is the best answer.)

32.7%

The CA's public key.

32.7%

Alice's private key.

20.4%

Alice's public key.

14.3%

Exam exercise 1

- Assume that you want to send a confidential and digitally signed contract via a public network. Which combination of information security services and mechanisms are needed to do this securely?
 - w = Digital Signature
 - x = Message Authentication Code (MAC)
 - y = Certification Authority (CA)
 - z = Encryption

Solving Exam exercise 1

- w, y and z but not x
 - Digital Signature for signing the document.
 - CA for verifying the certificate's validity (i.e., obtain/verify public key).
 - Encryption for confidentiality

Exam exercise 2

For message integrity, which of the following statement(s) is/are correct?

- a) *Message integrity is the property that the identity of the sender can be confirmed to be who or what they claim to be.*
- b) *Message integrity is the property that the receiver can detect whether the message sent was altered in transit.*
- c) *Both checksumming and hashing techniques may be used.*
- d) *Generally, a hash provides a better message integrity check than a checksum.*
- e) *To ensure message integrity, the transport layer protocol used to communication the message has to be TCP.*

Solving Exam exercise 2

- b), c) and d) but not a) nor e)
 - Digital Signatures for identity and we can use, for example, HMAC for routing messages
 - Checksumming is a poor hashing function (see page 656).
 - Encryption for confidentiality

Problem XX

Consider the 3-bit block cipher in the Table on the right. Suppose the **plaintext is 100100100**.

- Initially assume that CBC is **not** used. What is the resulting ciphertext?
- Suppose Trudy sniffs the ciphertext. Assuming she knows that a 3-bit block cipher **without** CBC is being employed (but doesn't know the specific cipher), what can she surmise?
- Now suppose that CBC is used with $IV = 111$. What is the resulting ciphertext?

Input	Output
000	110
001	111
010	101
011	100
100	011
101	010
110	000
111	001

XOR truth table	Input		Output
	A	B	
	0	0	0
	0	1	1
	1	0	1
	1	1	0

Solving Problem XX

XOR truth table	Input		Output
	A	B	
	0	0	0
	0	1	1
	1	0	1
	1	1	0

a) 100 100 100 $\rightarrow$ 011 011 011

b) Trudy will know the three block plaintexts are the same.

c) $c(i) = K_S(m(i) \oplus c(i-1))$ and IV = 111

- $c(1) = K_S(100 \oplus 111) = K_S(011) = 100$

- $c(2) = K_S(100 \oplus 100) = K_S(000) = 110$

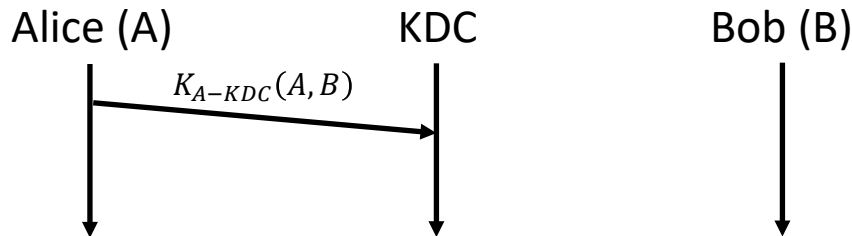
- $c(3) = K_S(100 \oplus 110) = K_S(010) = 101$

Problem 10 (pg. 714)

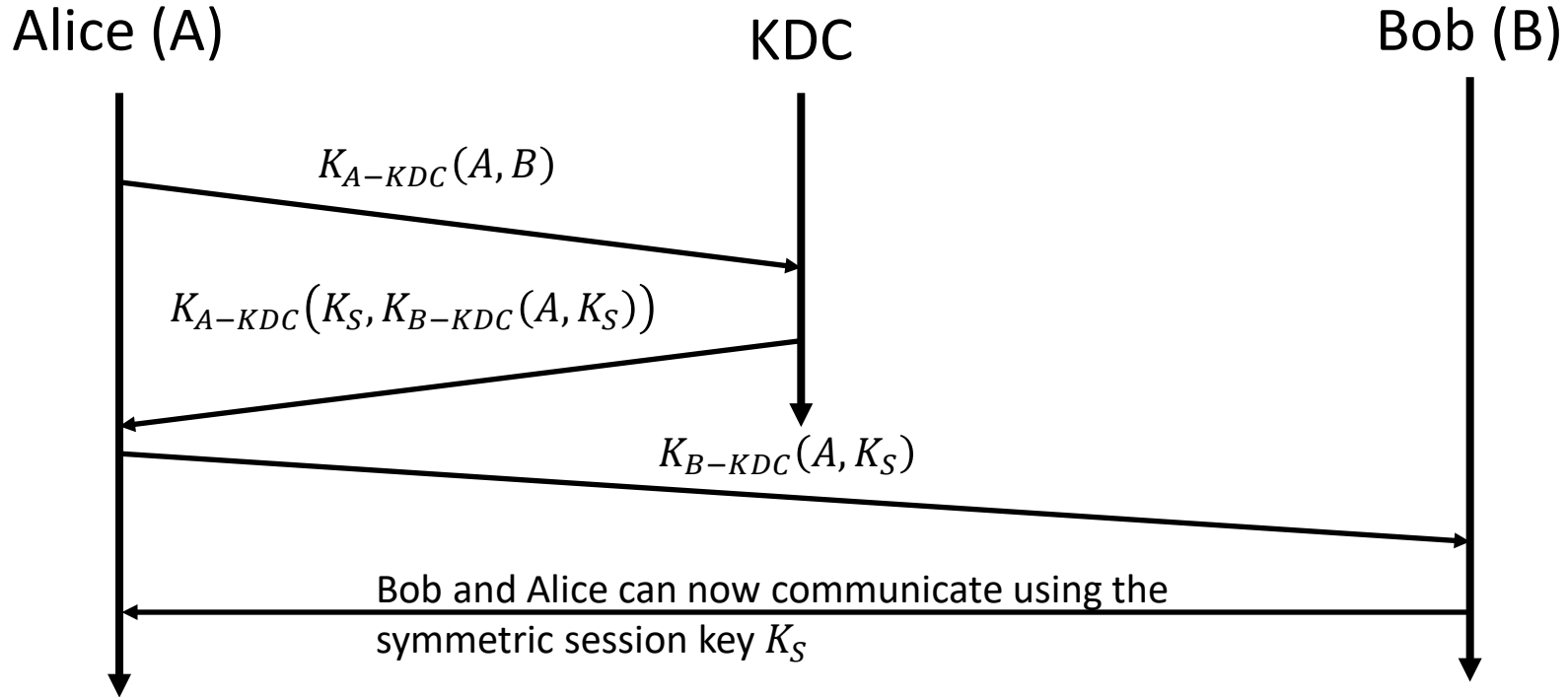
Suppose Alice wants to communicate with Bob using symmetric-key cryptography using a session key K_S . We learned how public-key cryptography can be used to distribute the session key from Alice to Bob. In this problem, we explore how the session key can be distributed — **without public key cryptography** — using a key distribution center (KDC). The KDC is a server that shares a unique secret symmetric key with each registered user. For Alice and Bob, denote these keys by K_{A-KDC} and K_{B-KDC} .

Design a scheme that uses the KDC to distribute K_S to Alice and Bob. Your scheme should use three messages to distribute the session key: a message from Alice to the KDC; a message from the KDC to Alice; and finally, a message from Alice to Bob. **The first message is $K_{A-KDC}(A, B)$.** Using the notation, K_{A-KDC} , K_{B-KDC} , K_S , A , and B answer the following questions:

- a) What is the second message?
- b) What is the third message?



Solving Problem 10 (pg. 714)



Time for a break

we restart at 11:15

Hashing

- `echo -n "This is my message" | sha256sum`
- `echo -n "This is my much much mmmmmmmmmmuch
longggggggggger message" | sha256sum`
- <https://www.freebsd.org/releases/14.0R/signatures/>
- `sha256sum -c CHECKSUM.SHA256-FreeBSD-15.0-RELEASE-
amd64 2>/dev/null | grep FreeBSD-15.0-RELEASE-
amd64-bootonly.iso`

Public signature

- `gpg --verify CHECKSUM.SHA512-FreeBSD-14.0-RELEASE-amd64.asc`
- `gpg --verify FAKECHECK`

Readings for next week

Book pages

- 669–683 (Sec. 8.5, 8.6, 8.7 and 8.7.1)
- 697–705 (8.9, 8.9.1)
- Optional:
 - 705–708 (Sec. 8.9.2)

Key concepts

Securing Email (PGP)

Securing TCP Connections (TLS)

Virtual Private Networks

Firewalls

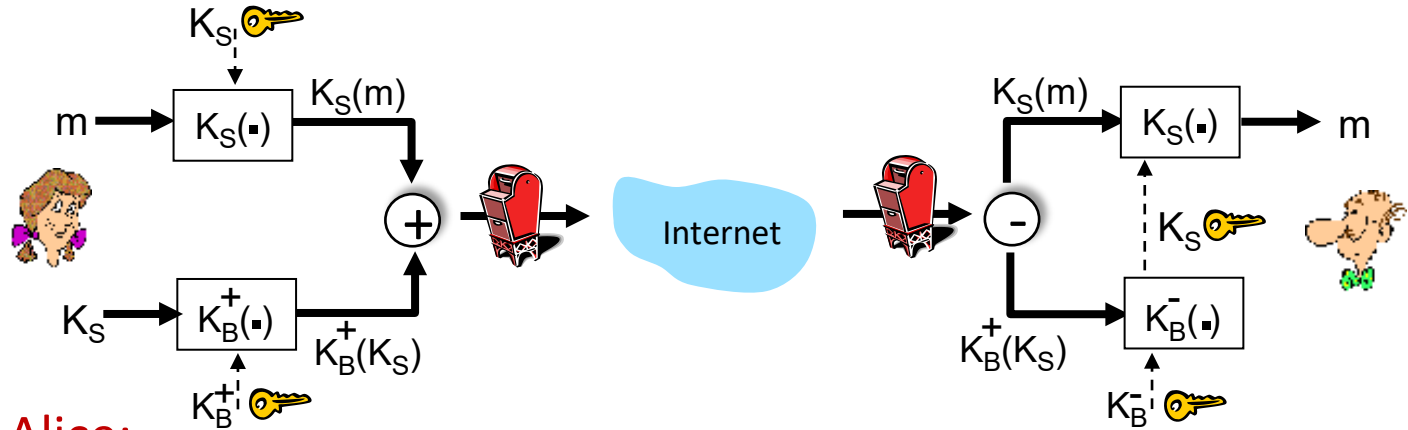
Chapter 8 outline

- What is network security?
- Principles of cryptography
- Authentication, message integrity
- **Securing e-mail**
- Securing TCP connections: TLS
- ~~■ Network layer security: IPsec~~
- ~~■ Security in wireless and mobile networks~~
- Operational security: firewalls and IDS



Secure e-mail: confidentiality

Alice wants to send *confidential* e-mail, m , to Bob.

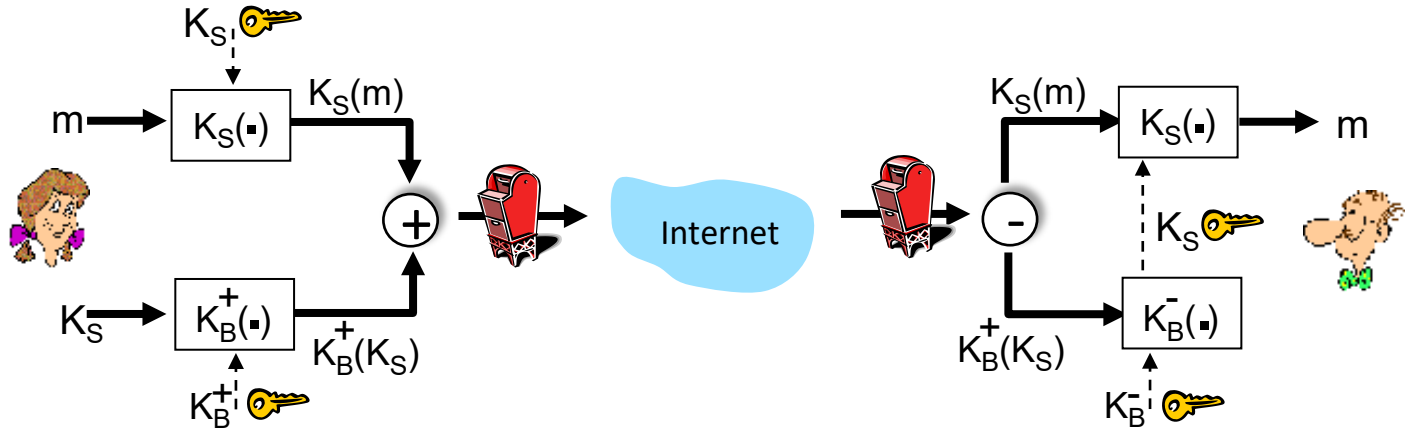


Alice:

- generates random *symmetric* private key, K_S
- encrypts message with K_S (for efficiency)
- also encrypts K_S with Bob's public key
- sends both $K_S(m)$ and $K_B^+(K_S)$ to Bob

Secure e-mail: confidentiality (more)

Alice wants to send *confidential* e-mail, m , to Bob.

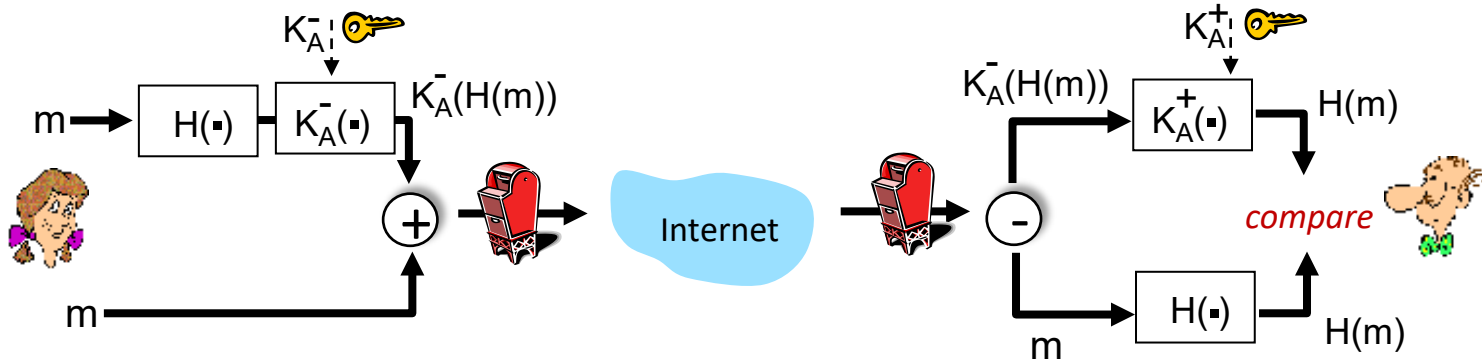


Bob:

- uses his private key to decrypt and recover K_S
- uses K_S to decrypt $K_S(m)$ to recover m

Secure e-mail: integrity, authentication

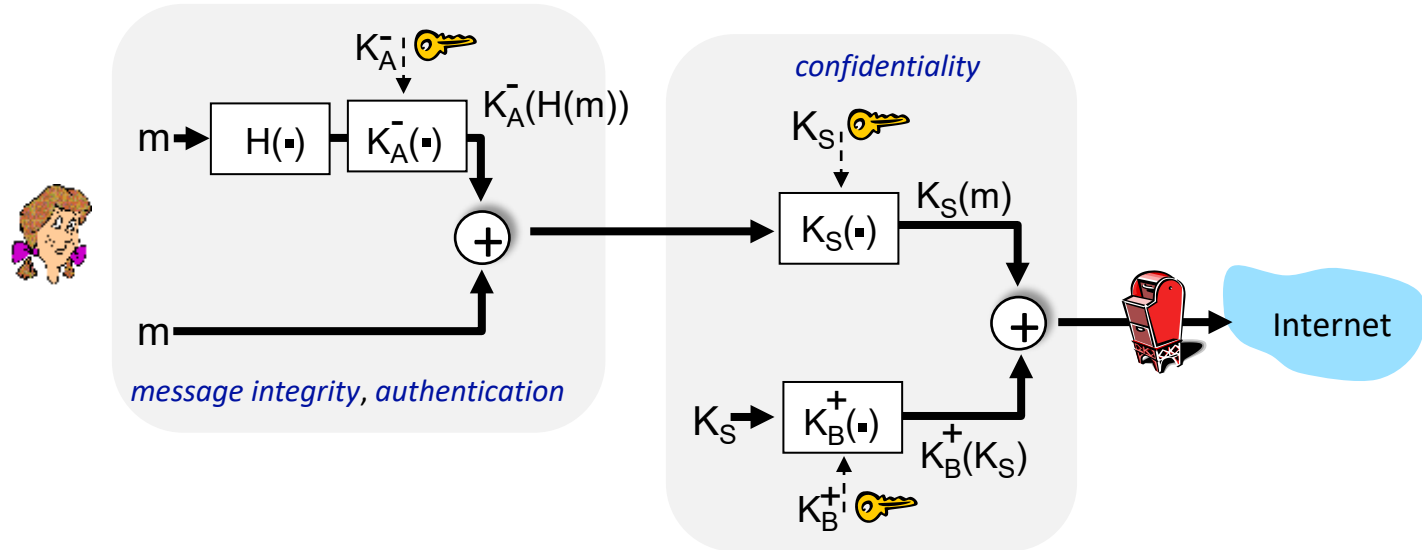
Alice wants to send m to Bob, with *message integrity, authentication*



- Alice digitally signs hash of her message with her private key, providing integrity and authentication
- sends both message (in the clear) and digital signature

Secure e-mail: confidentiality, integrity, authentication

Alice sends m to Bob, with *confidentiality, message integrity, authentication*



Alice uses three keys: her private key, Bob's public key, new symmetric key

What are Bob's complementary actions?

Chapter 8 outline

- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing e-mail
- **Securing TCP connections: TLS**
- ~~Network layer security: IPsec~~
- ~~Security in wireless and mobile networks~~
- Operational security: firewalls and IDS



Transport-layer security (TLS)

- widely deployed security protocol above the transport layer
 - supported by almost all browsers, web servers: https (port 443)
- provides:
 - **confidentiality**: via *symmetric encryption*
 - **integrity**: via *cryptographic hashing*
 - **authentication**: via *public key cryptography*

} *all techniques we have studied!*
- history:
 - early research, implementation: secure network programming, secure sockets
 - secure socket layer (SSL) deprecated [2015]
 - TLS 1.3: RFC 8846 [2018]

Transport-layer security: what's needed?

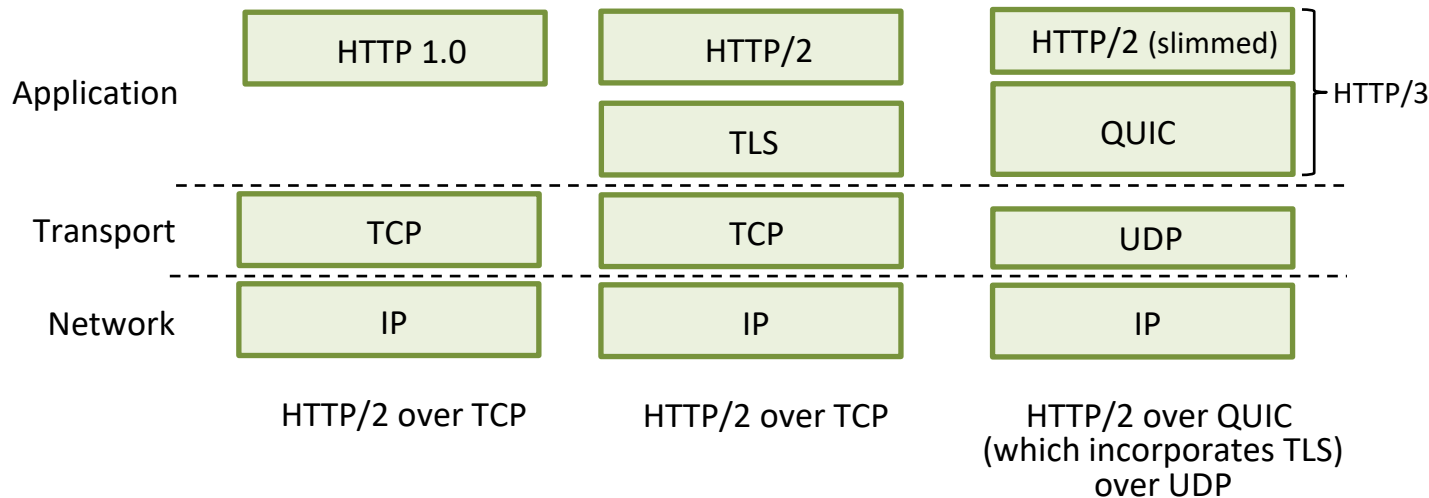
- let's *build* a toy TLS protocol, *t-tls*, to see what's needed!
- we've seen the "pieces" already:
 - **handshake**: Alice, Bob use their certificates, private keys to authenticate each other, exchange or create shared secret
 - **key derivation**: Alice, Bob use shared secret to derive set of keys
 - **data transfer**: stream data transfer: data as a series of records
 - not just one-time transactions
 - **connection closure**: special messages to securely close connection

t-tls: cryptographic keys

- considered bad to use same key for more than one cryptographic function
 - different keys for message authentication code (MAC) and encryption
- four keys:
 - 🔑 K_c : encryption key for data sent from client to server
 - 🔑 M_c : MAC key for data sent from client to server
 - 🔑 K_s : encryption key for data sent from server to client
 - 🔑 M_s : MAC key for data sent from server to client
- keys derived from key derivation function (KDF)
 - takes master secret and (possibly) some additional random data to create new keys

Transport-layer security (TLS)

- TLS provides an API that *any* application can use
- an HTTP view of TLS:



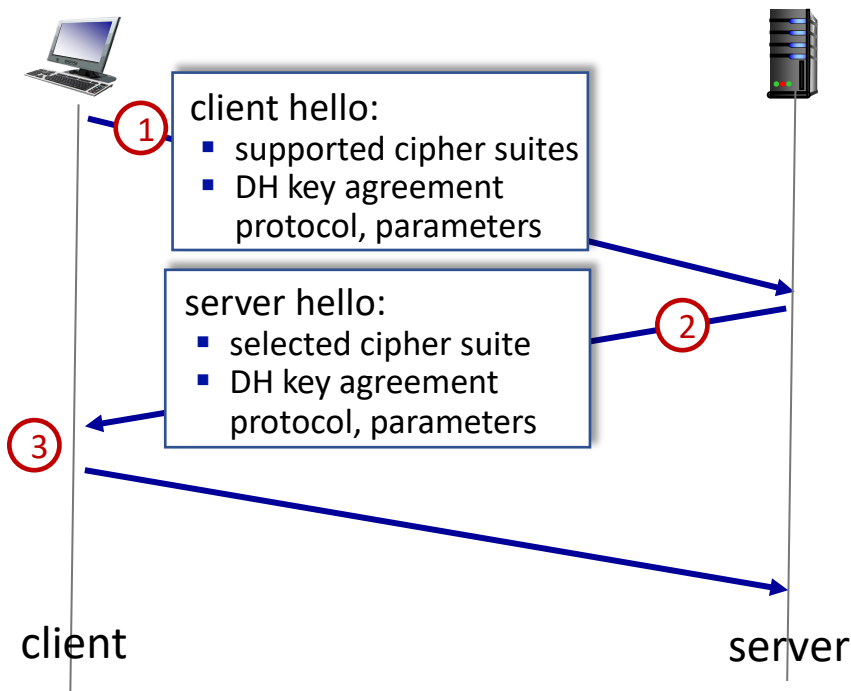
TLS: 1.3 cipher suite

- “cipher suite”: algorithms that can be used for key generation, encryption, MAC, digital signature
- TLS: 1.3 (2018): more limited cipher suite choice than TLS 1.2 (2008)
 - only 5 choices, rather than 37 choices
 - *requires* Diffie-Hellman (DH) for key exchange, rather than static DH or RSA
 - combined encryption and authentication algorithm (“authenticated encryption”) for data rather than serial encryption, authentication
 - 4 based on AES
 - HMAC uses SHA (256 or 384) cryptographic hash function

<https://www.rfc-editor.org/rfc/rfc8446#section-1.2>

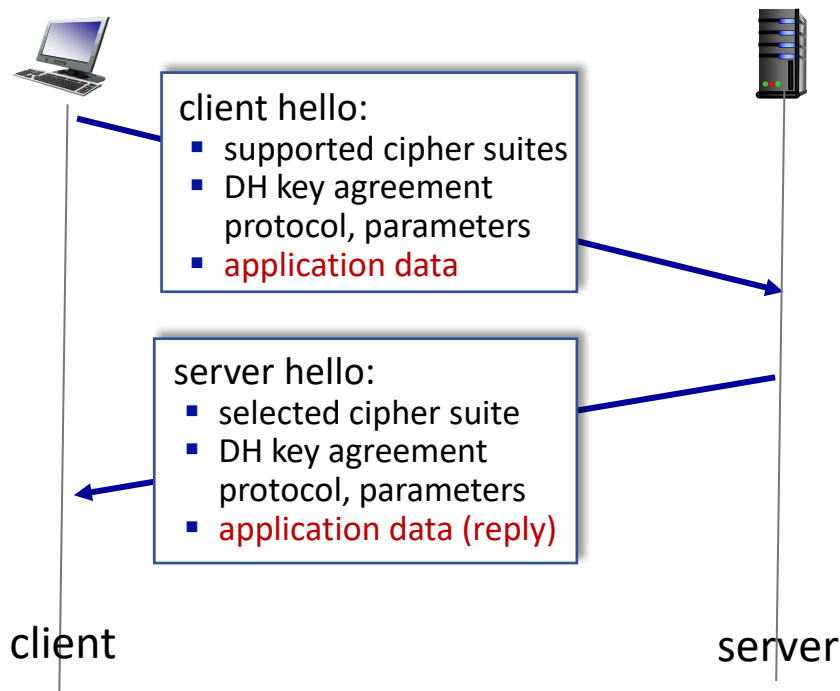
<https://www.ietf.org/archive/id/draft-ietf-tls-deprecate-obsolete-kex-02.html>

TLS 1.3 handshake: 1 RTT



- ① client TLS hello msg:
 - *guesses* key agreement protocol, parameters
 - indicates cipher suites it supports
- ② server TLS hello msg chooses
 - key agreement protocol, parameters
 - cipher suite
 - server-signed certificate
- ③ client:
 - checks server certificate
 - generates key
 - can now make application request (e.g., HTTPS GET)

TLS 1.3 handshake: 0 RTT



- initial hello message contains encrypted application data!
 - “resuming” earlier connection between client and server
 - application data encrypted using “resumption master secret” from earlier connection
- vulnerable to replay attacks!
 - maybe OK for an HTTP GET or client requests not modifying server state

Chapter 8 outline

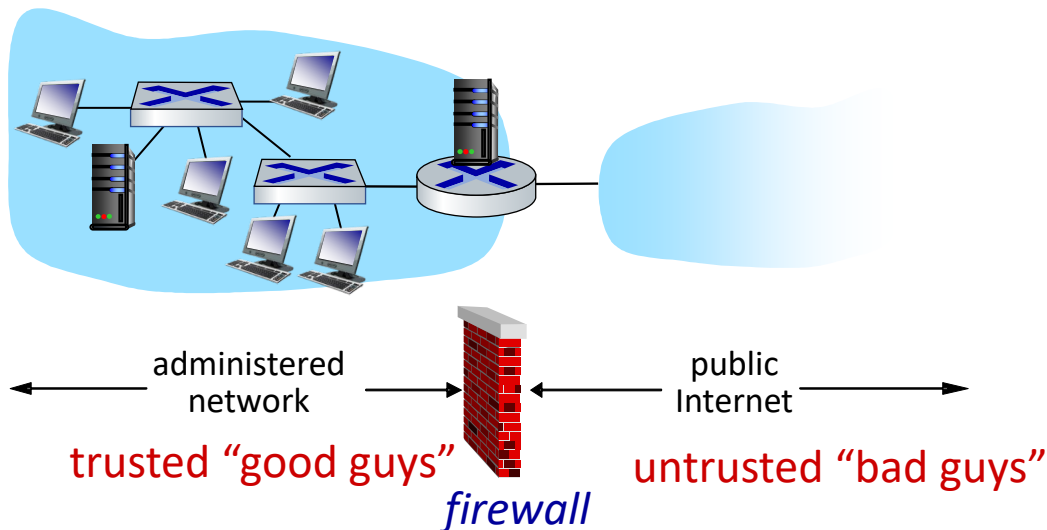
- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing e-mail
- Securing TCP connections: TLS
- Network layer security: IPsec
- Security in wireless and mobile networks
- **Operational security: firewalls and IDS**



Firewalls

firewall

isolates organization's internal network from larger Internet, allowing some packets to pass, blocking others



Firewalls: why

prevent denial of service attacks:

- SYN flooding: attacker establishes many bogus TCP connections, no resources left for “real” connections

prevent illegal modification/access of internal data

- e.g., attacker replaces CIA’s homepage with something else

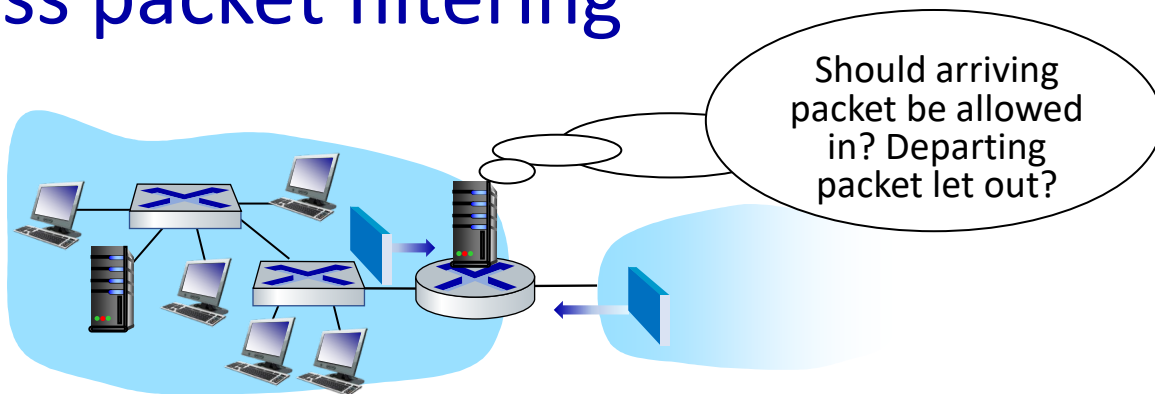
allow only authorized access to inside network

- set of authenticated users/hosts

three types of firewalls:

- stateless packet filters
- stateful packet filters
- application gateways

Stateless packet filtering



- internal network connected to Internet via router **firewall**
- filters **packet-by-packet**, decision to forward/drop packet based on:
 - source IP address, destination IP address
 - TCP/UDP source, destination port numbers
 - ICMP message type
 - TCP SYN, ACK bits

Access Control Lists

ACL: table of rules, applied top to bottom to incoming packets:
(action, condition) pairs: looks like OpenFlow forwarding (Ch. 4)!

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

Stateful packet filtering

- *stateless packet filter*: heavy handed tool
 - admits packets that “make no sense,” e.g., dest port = 80, ACK bit set, even though no TCP connection established:

action	source address	dest address	protocol	source port	dest port	flag bit
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK

- *stateful packet filter*: track status of every TCP connection
 - track connection setup (SYN), teardown (FIN): determine whether incoming, outgoing packets “makes sense”
 - timeout inactive connections at firewall: no longer admit packets

Stateful packet filtering

ACL augmented to indicate need to check connection state table before admitting packet

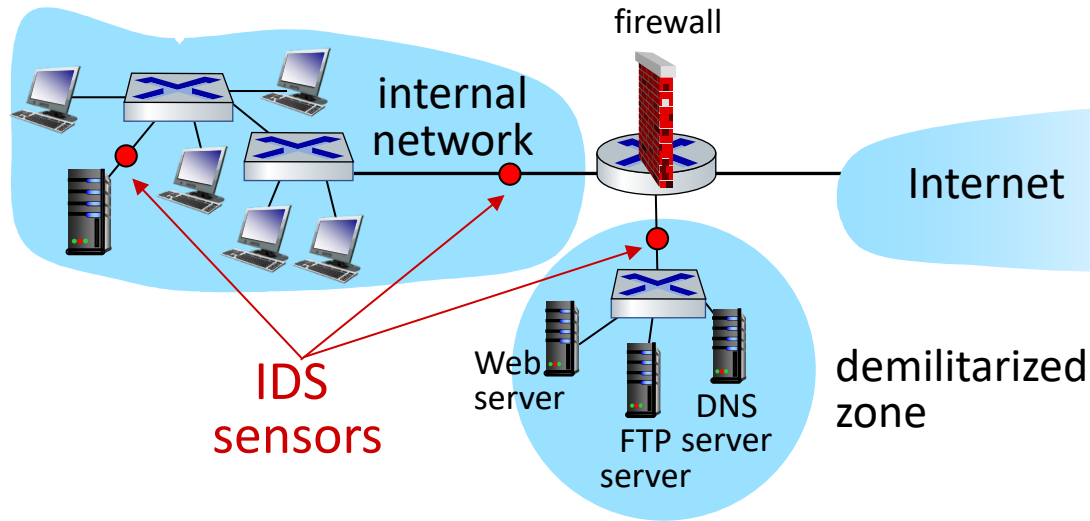
action	source address	dest address	proto	source port	dest port	flag bit	check connection
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any	
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

Intrusion detection systems

- packet filtering:
 - operates on TCP/IP headers only
 - no correlation check among sessions
- IDS: intrusion detection system
 - deep packet inspection: look at packet contents (e.g., check character strings in packet against database of known virus, attack strings)
 - examine correlation among multiple packets
 - port scanning
 - network mapping
 - DoS attack

Intrusion detection systems

multiple IDSs: different types of checking at different locations



Network Security (summary)

basic techniques.....

- cryptography (symmetric and public key)
- message integrity
- end-point authentication

.... used in many different security scenarios

- secure email
- secure transport (TLS)
- IP sec
- 802.11, 4G/5G

operational security: firewalls and IDS



Readings for next week

Book pages

- 669–683 (Sec. 8.5, 8.6, 8.7 and 8.7.1)
- 697–705 (8.9, 8.9.1)
- Optional:
 - 705–708 (Sec. 8.9.2)

Key concepts

Securing Email (PGP)

Securing TCP Connections (TLS)

Virtual Private Networks

Firewalls

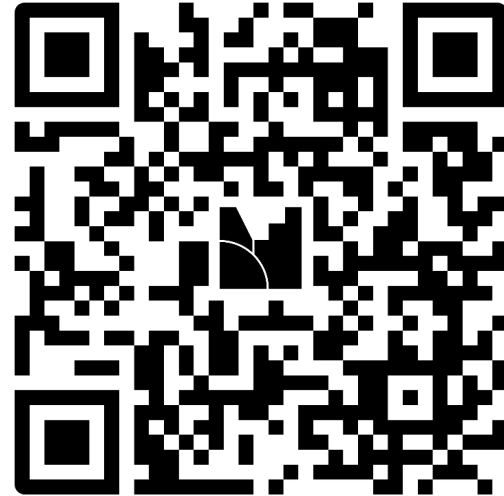
Instructions

Go to

www.menti.com

Enter the code

5766 1732



Or use QR code

