Security in Computer Networks

ECE 6607
GATECH Shenzhen, Fall 2022

Dr. Min Luo

1

Chapter Goals

- Understand principles of network security:
  - Cryptography and its *many* uses beyond "confidentiality"
  - Authentication
  - Message integrity
- Security in practice:
  - Firewalls and intrusion detection systems
  - Security in application, transport, network, link layers

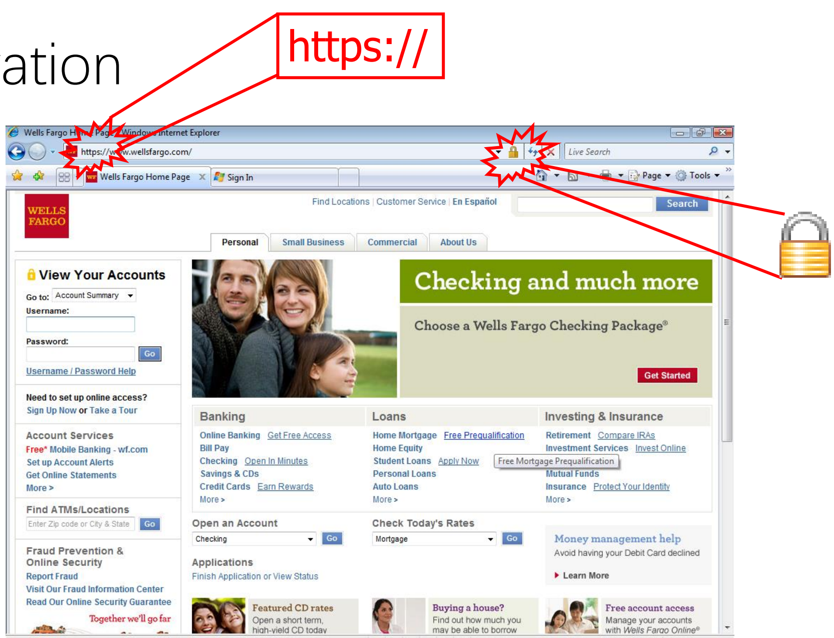
2

Outline

- **What is network security?**
- Principles of cryptography
- Message integrity, authentication
- Securing e-mail
- Securing TCP connections: SSL
- Network layer security: IPsec
- Securing wireless LANs
- Operational security: firewalls and IDS

3

Motivation



4

Security Attacks

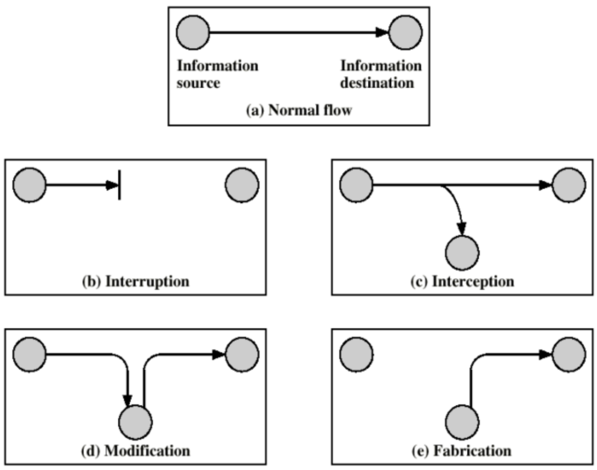


Figure 1.1 Security Threats

Interruption: an attack on availability

Interception: an attack on confidentiality

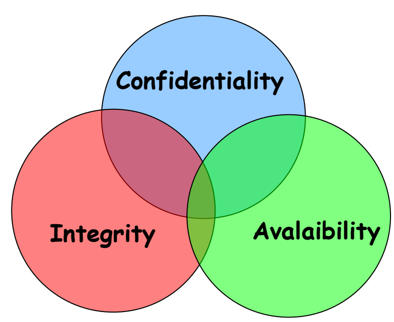
Modification: an attack on integrity

Fabrication: an attack on authenticity

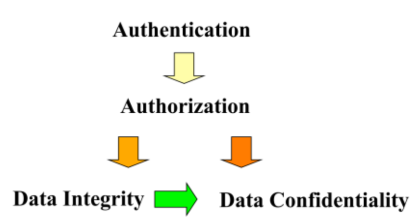
5

Desirable Security Properties

- *Confidentiality*: Only sender, intended receiver should "understand" message contents
  - Sender *encrypts* message
  - Receiver *decrypts* message
- *Authentication*: Sender, receiver want to confirm *identity* of each other
- *Message integrity*: Sender, receiver want to ensure message not altered (in transit, or afterwards) without detection
- *Access and availability*: Services must be *accessible and available* to users
- Accountability and non-repudiation
- Privacy of collected information
- …

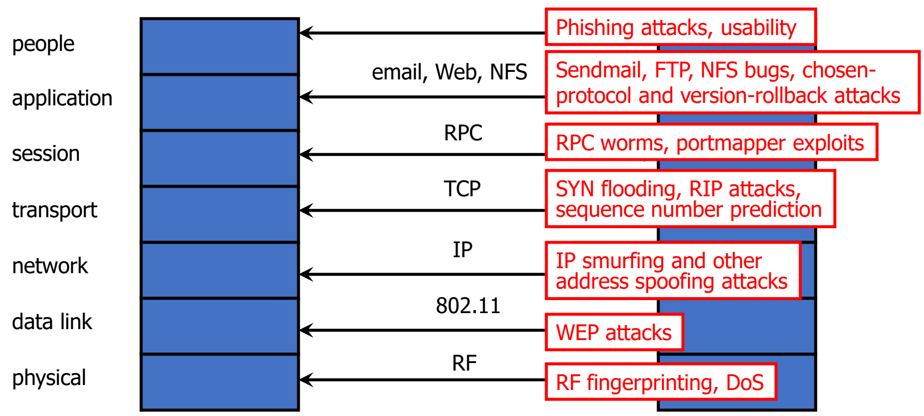


Security Service Dependencies



6

Network Stack



Only as secure as the ***single weakest layer***…… or ***interconnection*** between the layers

slide 7

7

Friends and enemies: Alice, Bob, Trudy

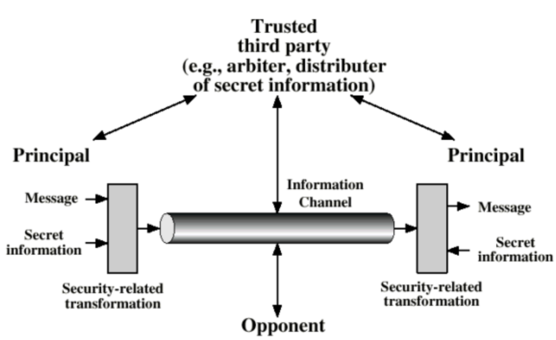
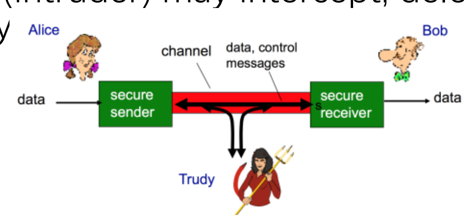


Figure 1.3 Model for Network Security

- Bob, Alice (lovers, negotiators…) want to communicate "securely"
- Trudy (intruder) may intercept, delete, modify



… well, *real-life* Bobs and Alices ➔

Web browser/server for electronic transactions

E.g., on-line purchases…

On-line banking client/server:

Digital currency/certificate

DNS servers

Routers exchanging routing table updates

Other examples?

8

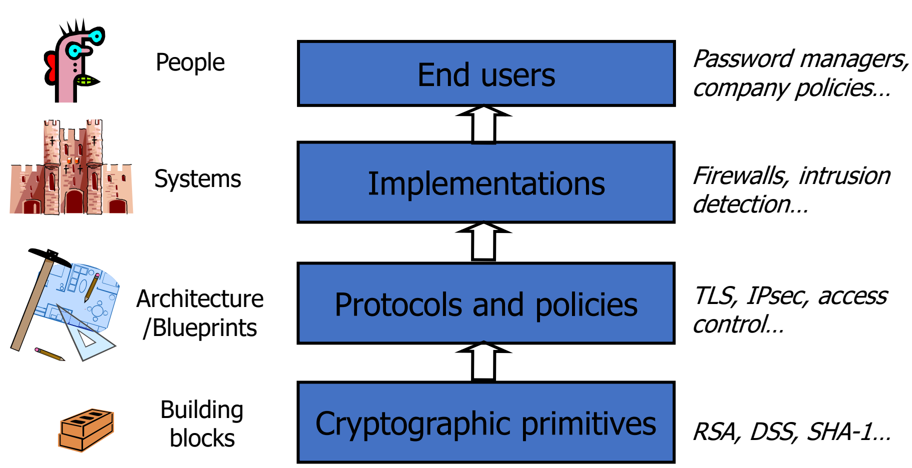
There are bad guys (and girls) out there ➔ TRUDY!

Q: What can a "bad guy" do?
A: A lot!

- *Eavesdrop:* Intercept messages
- Actively *insert* messages into connection
- *Impersonation:* Can fake (spoof) source address in packet (or any field in packet)
- *Hijacking:* "Take over" ongoing connection by removing sender or receiver, inserting himself in place
- *Denial of service*: Prevent service from being used by others (e.g., by overloading resources)

9

Network Defenses



<ins>All</ins> defense mechanisms must work correctly and securely

slide 10

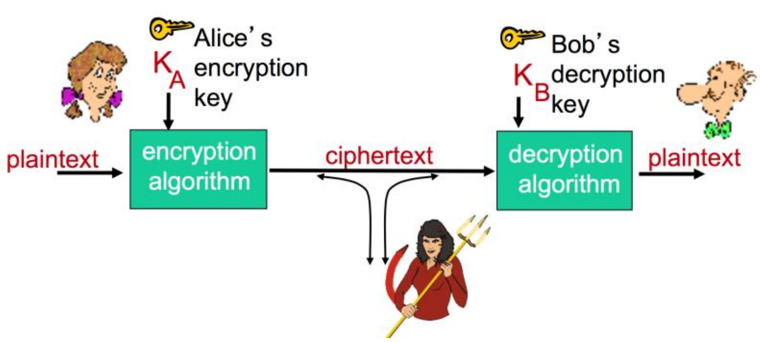
10

Outline

- What is network security?
- **Principles of cryptography**
- Message integrity, authentication
- Securing e-mail
- Securing TCP connections: SSL
- Network layer security: IPsec
- Securing wireless LANs
- Operational security: firewalls and IDS

11

The language of cryptography



Sent m = plaintext message

Transported $K_A(m)$ = ciphertext, encrypted with key K_A

Received ***$m = K_B(K_A(m))$***

12

Simple encryption scheme

- *Substitution cipher: S*ubstitute one thing for another
- Mono alphabetic cipher: Substitute one letter for another

```
plaintext:   abcdefghijklmnopqrstuvwxyz

ciphertext:  mnbvcxzasdfghjklpoiuytrewq
```

e.g.:
```
  Plaintext: bob. i love you. alice
 ciphertext: nkn. s gktc wky. mgsbc
```

Encryption key: mapping from set of 26 letters to set of 26 letters

A more sophisticated encryption approach

- n substitution ciphers, M_1, M_2, …, M_n
- Cycling pattern:
  - e.g., n=4: M_1,M_3,M_4,M_3,M_2; M_1,M_3,M_4,M_3,M_2; ..
- For each new plaintext symbol, use subsequent substitution pattern in cyclic pattern
  - Dog: d from M_1,o from M_3, g from M_4

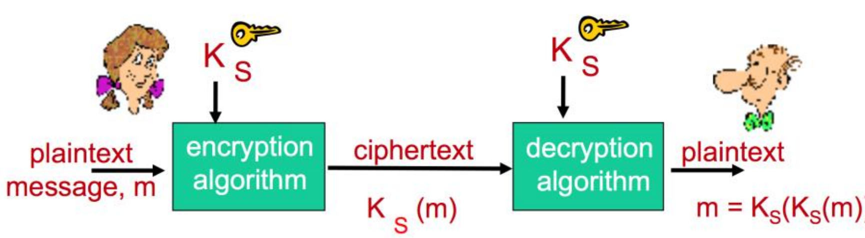
Encryption key: n substitution ciphers, and cyclic pattern
- Key need not be just n-bit pattern

Breaking an encryption scheme

- Cipher-text only attack: Trudy has ciphertext she can analyze
- Two approaches:
  - Brute force: Search through all keys
- Statistical analysis: Correlation, inference, …
- Known-plaintext attack: Trudy has plaintext corresponding to ciphertext
  - e.g., in mono alphabetic cipher, Trudy determines pairings for a,l,i,c,e,b,o,
- Chosen-plaintext attack: Trudy can get ciphertext for chosen plaintext

15

Symmetric key cryptography



Symmetric key crypto: Bob and Alice share *same (symmetric) key*: K_S

- e.g., key is some known substitution pattern in mono alphabetic substitution cipher

Q: How do Bob and Alice agree on key value?

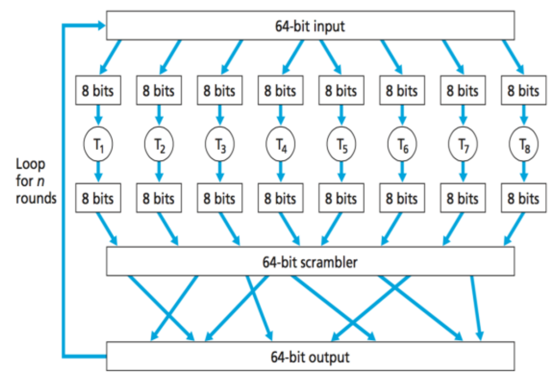
16

Symmetric key crypto: DE

input	output	input	output
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

DES: **Data Encryption Standard**

- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64-bit plaintext input
- Block cipher with cipher block chaining:
  - Msg is broken into k (64)-bits block, each block is *encrypted independently*
  - K(64)-bit is further divided into 8-bit chunks, each processed by applying permutation rules specified in T_i
- Making DES more secure:
  - 3DES: *Encrypt 3 times* with 3 different keys



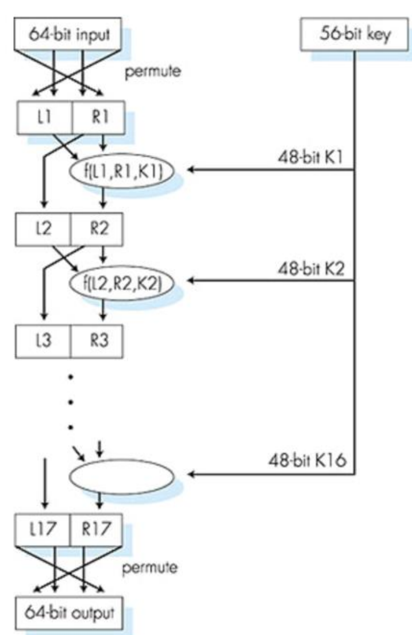
17

Symmetric key crypto: DES

DES operation

- Initial permutation
- 16 identical "rounds" of function application, each using different 48 bits of key
- Final permutation

A Complete Example: http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm



18

AES: Advanced Encryption Standard

- Symmetric-key NIST standard, replaced DES (Nov 2001)
- Processes data in 128 bit blocks
- 128, 192, or 256 bit keys
- If brute force decryption (try each key) taking *1 sec on DES*, takes *149 trillion years for AES*

19

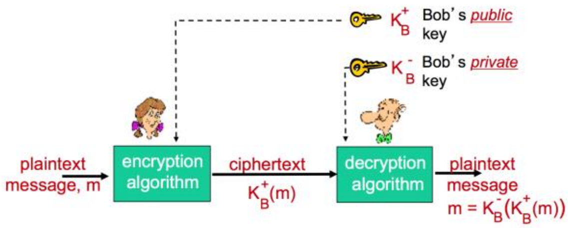
Public Key Cryptography

Symmetric Key Crypto
- Requires sender, receiver know shared secret key
- Q: How to agree on key in first place (particularly if never “met”)?

Public Key Crypto
- Radically different approach [Diffie- Hellman76, RSA78]
- Sender, receiver do *not* share secret key
- *Public* encryption key known to *all*
- *Private* decryption key known only to receiver

20

Public key cryptography



requirements:

1. need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that
$$K_B^-(K_B^+(m)) = m$$
2. given public key K_B^+, it should be *infeasible* to compute private key K_B^-

The RSA (**Rivest, Shamir, Adelson**) Algorithm

- A public-key cryptosystem
- Based on the idea that *factorization of integers into their prime factors* is hard.

★ n=p・q, where p and q are distinct primes

- Proposed by **R**ivest, **S**hamir, and **A**dleman in 1977 and a paper was published in The Communications of ACM in 1978

- Bob *chooses* two primes *p, q* and compute *n = pq*
- Bob *chooses e* with gcd(e,(p-1)(q-1)) = gcd(e, ψ(n))=1
- Bob *solves* $de \equiv 1 \pmod{\psi(n)}$
- Bob *makes (e,n) public* and *(p,q,d) secret*
- Alice encrypts *m* as $c \equiv m^e \pmod{n}$
- Bob decrypts by computing $m \equiv c^d \pmod{n}$

RSA Algorithm

- p=885320963, q=238855417,
- n=p・q=211463707796206571
- Let e=9007 ➔ d=116402471153538991
- M="cat"=30120, C=113535859035722866

(e,n) public key
(p,q,d) secret/private key

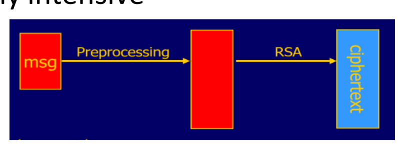
23

RSA in practice: session keys

Exponentiation in RSA ➔ Computationally intensive

- DES is *at least 100 times faster* than RSA

➔Preprocessing & Enhancements!



- The "work horse" of Internet security:
  - Most Public Key Infrastructure (PKI) products.
  - SSL/TLS: Ssecure connection; Certificates and key-
- *Session key, K_S*
  - Bob and Alice use RSA to exchange a symmetric key K_S
  - Once both have K_S, they use symmetric key cryptography

24

Outline

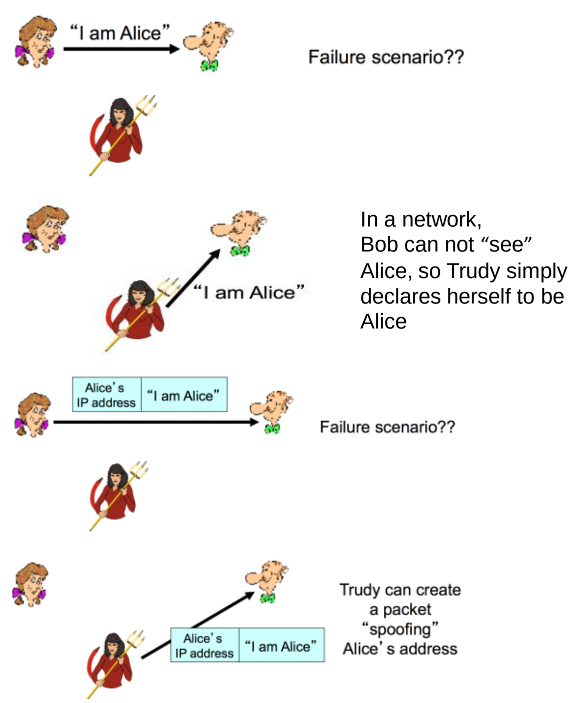
- What is network security?
- Principles of cryptography
- Message integrity, **authentication**
- Securing e-mail
- Securing TCP connections: SSL
- Network layer security: IPsec
- Securing wireless LANs
- Operational security: firewalls and IDS

Authentication

- *Goal:* Bob wants Alice to "prove" her identity to him

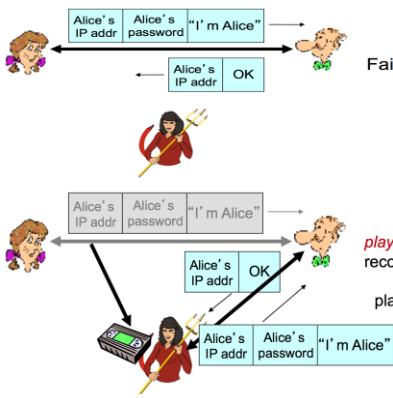
- *Protocol ap1.0:* Alice says "I am Alice"

- *Protocol ap2.0:* Alice says "I am Alice" in an IP packet containing her source IP address



Authentication

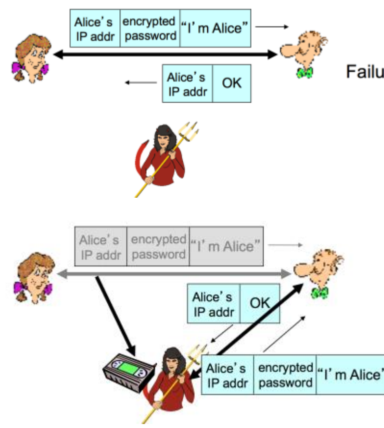
- *Protocol ap3.0:* Alice says “I am Alice” and sends her secret password to “prove” it.



Failure scenario??

playback attack: Trudy records Alice’s packet and later plays it back to Bob

- *Protocol ap3.1:* Alice says “I am Alice” and sends her *encrypted* secret password to “prove” it.



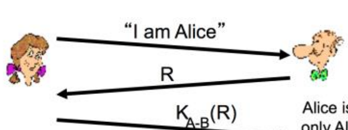
Failure scenario??

record and playback *still* works!

27

Authentication

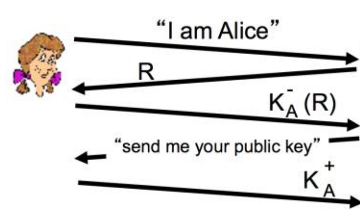
- *Goal:* Avoid playback attack
- ***Nonce:*** number (R) used only *once-in-a-lifetime*
- *ap4.0:* to prove Alice “live”, Bob sends Alice *nonce*, R. Alice must return R, encrypted with shared secret key
  - Requires shared symmetric key
  - Can we authenticate using public key techniques?



Alice is live, and only Alice knows key to encrypt nonce, so it must be Alice!

Failures, drawbacks?

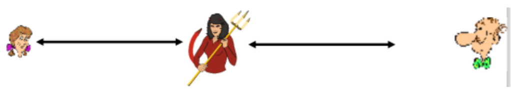
- *Ap5.0:* Use nonce, public key cryptography

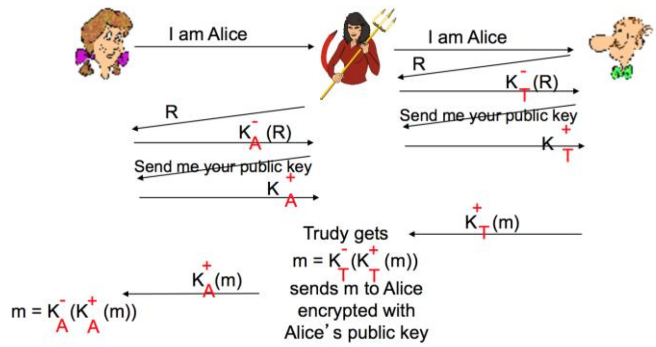


Bob computes $K_A^+(K_A^-(R)) = R$ and knows only Alice could have the private key, that encrypted R such that $K_A^+(K_A^-(R)) = R$

28

ap5.0: security hole





- *Man (or woman) in the middle attack:* Trudy poses as Alice (to Bob) and as Bob (to Alce)

- Difficult to detect:
  Bob receives everything that Alice sends, and vice versa (e.g., so Bob, Alice can meet one week later and recall conversation!)
  ➔ Problem is that Trudy receives all messages as well!

29

Outline

- What is network security?
- Principles of cryptography
- **Message integrity**, authentication
- Securing e-mail
- Securing TCP connections: SSL
- Network layer security: IPsec
- Securing wireless LANs
- Operational security: firewalls and IDS

30

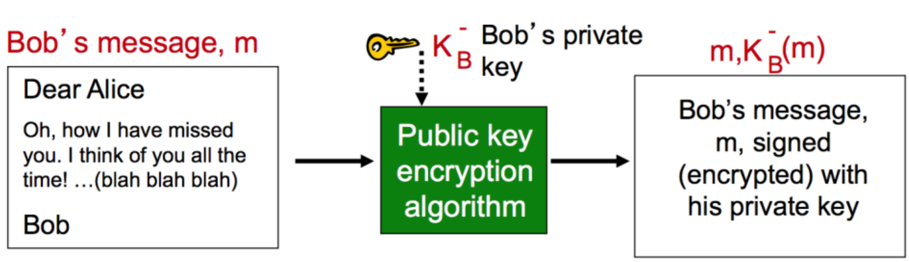
Digital signatures

- Cryptographic technique analogous to *hand-written signatures*
- Sender(Bob) *digitally signs* document, establishing he is document *owner/creator*.
- *Verifiable, non-forgeable:* Recipient (Alice) can prove to someone that Bob, and no one else (including Alice), must have signed document

31

Digital signatures

- Simple digital signature for message *m*:
- Bob signs *m* by encrypting with his private key K_B, creating "signed" message, $K_B(m)$



32

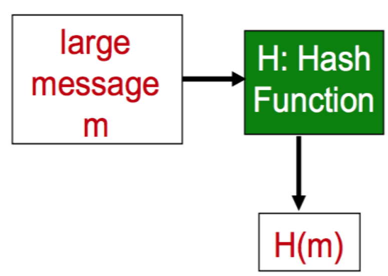
Digital signatures

- Suppose Alice receives msg *m*, with signature: $m, K^-_B(m)$
- Alice verifies *m* signed by Bob by applying Bob's public key K^+_B to $K^-_B(m)$ then checks $K^+_B(K^-_B(m)) = m$.
- If $K^+_B(K^-_B(m)) = m$, whoever signed *m* must have used Bob's private key.

- Alice thus *verifies* that:
  - Bob signed m
  - No one else signed m
  - Bob signed m and not m'
- Non-repudiation: -
  - Alice can take *m*, and signature $K^-_B(m)$ to court and prove that Bob signed *m*

33

Message digests

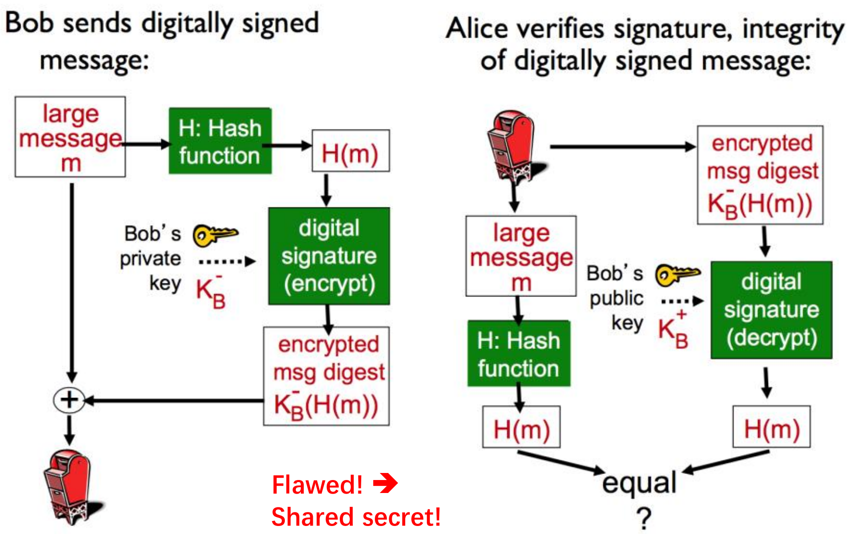


- Computationally expensive to public-key-encrypt long messages
- *Goal: fixed-length, easy-to-compute digital **"fingerprint"***
  - Apply *hash function* H to *m*, get fixed size message digest, *H(m)*.

- Hash function properties:
  - Many-to-1
  - Produces fixed-size msg digest (fingerprint)
  - Given message digest *x*, *computationally infeasible* to find *m* such that $x = H(m)$

34

Digital signature = signed message digest

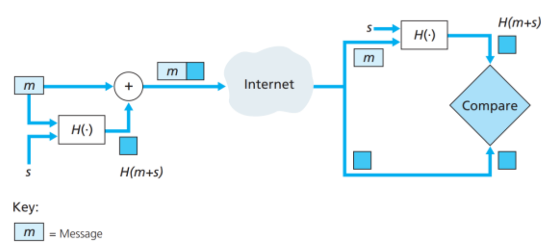


Flawed! ➔ Shared secret!

35

Message authentication code (MAC)

- Authentication key : A shared secret s, just a string of bits
- Message integrity can be performed:
  1. Alice creates message *m*, concatenates *s* with ***m*** to create *m + s*, and calculates the hash *H(m + s)*
     - *H(m + s)* is the *message authentication code* (MAC).
  2. Alice then appends the MAC to the message *m*, creating an *extended message (m, H(m + s))*, and sends the extended message to Bob.
  3. Bob receives an extended message *(m, h)* and knowing *s*, calculates the MAC *H(m + s)*. If *H(m + s) = h*, Bob concludes that everything is fine.
- No need for complex/expensive enc/dec algorithm!



36

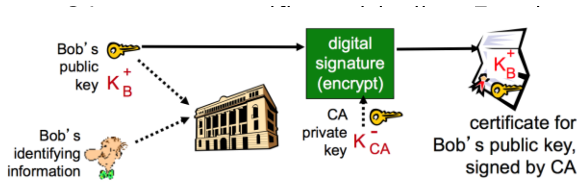
Hash function algorithms

- *MD5* hash function widely used (RFC 1321)
  - Computes 128-bit message digest
  - Arbitrary 128-bit string x, *difficult* to construct msg m whose MD5 hash is equal to x
- *SHA-1* is also used
  - US standard [NIST, FIPS PUB 180-1]
  - 160-bit message digest

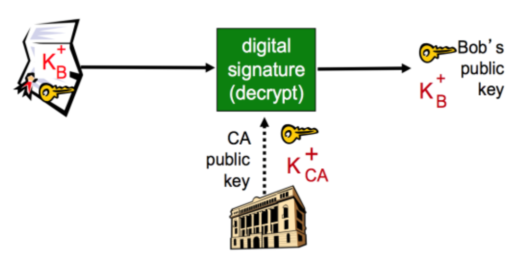
Parameters of Comparison	SHA/SHA-1...	MD5
Definition	SHA is a cryptographic hash function algorithm created by NIST to facilitate the creation of message digests.	MD5 was created by Ron Rivest and is used to convert messages of indiscriminate length into 128-bit message digests.
Full Form	Secure Hash Algorithm	Message Digest
Maximum Message Length	SHA can convert a message of 2^{64} – to – 2^{128} bits to form a *160-* 512 bit message digest.	MD5 can convert messages of any length into a 128-bit message digest.
Security	Balanced and tolerable: More secured than MD5.	Less secured than SHA and its improved SHA-1 version.
Speed	The original version of the algorithm is slower than MD5. However, its subsequent installments like SHA-1 offer much more enhanced speeds.	MD5 is faster than the *original* SHA version.
Vulnerability	Less vulnerable to cyber threats and hacker attacks.	More vulnerable to cyber threats and hacker attacks.
Number of Attacks	Fewer attacks have been able to breach the algorithm.	Several severe attacks have been reported.
Uses Today	Used in applications like SSH, SSL, etc.	MD5's usage is mostly limited to *verifying* the integrity of files due to its *poor security protocols.*

Certification authorities

- *Certification Authority (CA):* Binds public key to particular entity, E.
- E (person, router) registers its public key with CA.
  - E provides "proof of identity" to CA.



- When Alice wants Bob's public key:
  - Gets Bob's certificate (Bob or elsewhere).
  - Apply CA's public key to Bob's certificate, get Bob's public key



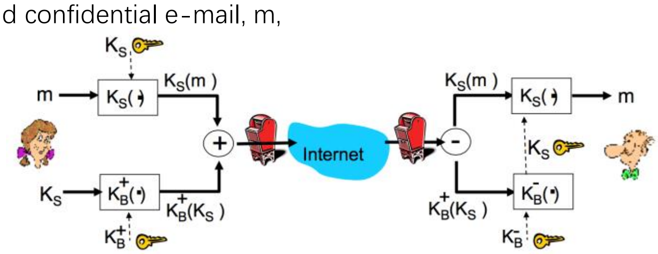
Outline

- What is network security?
- Principles of cryptography
- Message integrity, authentication
- **Securing e-mail**
- Securing TCP connections: SSL
- Network layer security: IPsec
- Securing wireless LANs
- Operational security: firewalls and IDS

39

Secure e-mail

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



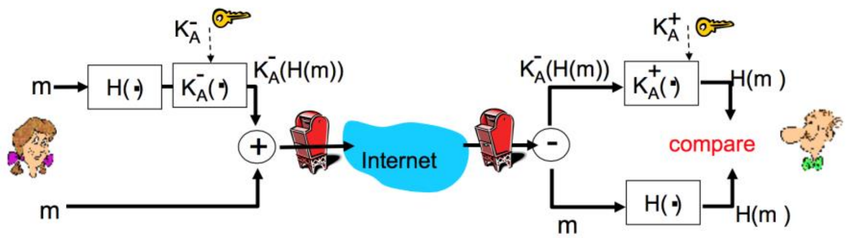
1. Generates *random symmetric* private key, K_S
2. Encrypts message with K_S (for efficiency)
3. Also encrypts K_S with Bob's public key
4. Sends both $K_S(m)$ and $K_B(K_S)$ to Bob

- Bob:
  - Uses his private key to decrypt and recover K_S
  - Uses K_S to decrypt $K_S(m)$ to recover m

40

Secure e-mail

- Alice wants to provide sender authentication message integrity .

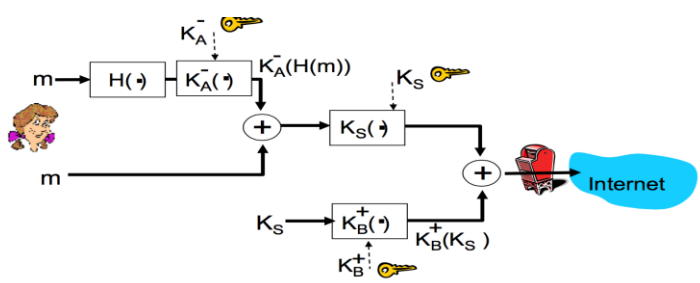


- Alice digitally signs message
- Sends both message (in the clear) and digital signature

41

Secure e-mail

- Alice wants to provide secrecy, sender authentication, message integrity.



- Alice uses *three keys:* her private key, Bob's public key, newly created symmetric key

42

Outline

- What is network security?
- Principles of cryptography
- Message integrity, authentication
- Securing e-mail
- **Securing TCP connections: SSL**
- Network layer security: IPsec
- Securing wireless LANs
- Operational security: firewalls and IDS

43

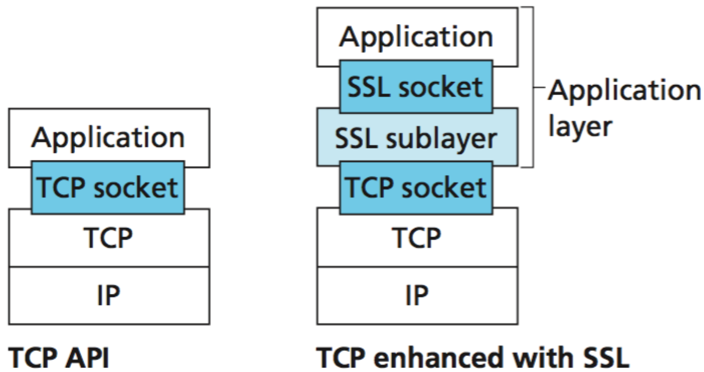
SSL: Secure Sockets Layer

- Widely deployed security protocol
  - Supported by almost all browsers, web servers
  - https
  - $Billions/year over SSL
- Mechanisms: [Woo 1994], implementation - Netscape
- Variation - Transport Layer Security (TLS), RFC 2246
- Provides ➔ Enhancing TCP services
  - *Confidentiality*
  - *Integrity*
  - *Authentication*

- Original goals:
  - Web *e-commerce* transactions
  - Encryption (especially credit-card numbers)
  - Web-server authentication
  - Optional client authentication
  - Minimum hassle in doing business with new merchant
- Available to all TCP applications
  - Secure socket interface

44

SSL and TCP/

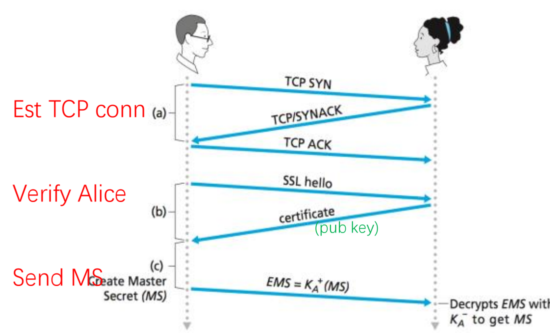


- SSL provides application programming interface (API) to applications ➔ App layer
- C/C++, Java and Python SSL libraries/classes readily available

45

Toy SSL: a simple secure channel

- ***Handshake***: Alice and Bob use their certificates, private keys to *authenticate* each other and exchange *shared secret*
- ***Key derivation***: Alice and Bob use shared secret to derive ***set of 4 keys***, by *slicing* the master key MS
  - Session encrypt key (data): E_A, E_B
  - Session MAC key (integrity verification): M_A, M_B
- ***Data transfer***: Data to be transferred is broken up into series of records, each appended with MAC
- ***Connection closure***: Special messages to securely close connection



- ***Message Authentication Code*** (**MAC**): A cryptographic checksum on data that uses a session key to detect both *accidental and intentional modifications* of the data.
- A **MAC** requires two inputs:
  - A message
  - A secret key known only to the originator of the message and its intended recipient(s).
- Hashed or Hash-based MAC (HMAC)

46

SSL cipher suite

- Cipher suite
  - Public-key algorithm
  - Symmetric encryption algorithm
  - MAC algorithm
- SSL supports several cipher suites
- Negotiation: Client, server agree on cipher suite
  - Client offers choice
  - Server picks one

Common SSL symmetric ciphers
- DES – Data Encryption Standard: block
- 3DES – Triple strength: block
- RC2 – Rivest Cipher 2: block
- RC4 – Rivest Cipher 4: stream

SSL Public key encryption
- RSA

47

Real SSL: handshake

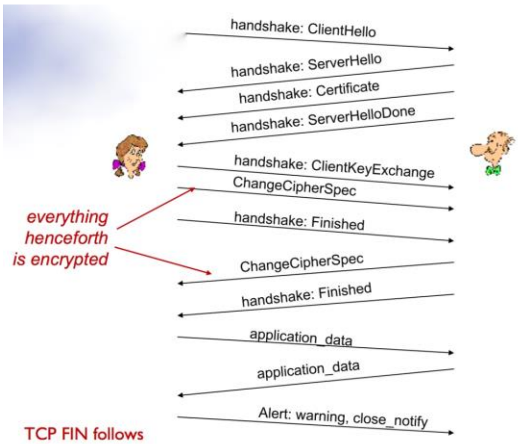
Purpose
1. Server authentication
2. Negotiation: agree on crypto algorithms
3. Establish keys
4. Client authentication (optional)

Procedure:

1. Client sends *list of algorithms* it supports, along with *client nonce*
2. Server chooses algorithms from list; sends back: *choice + certificate + server nonce*
3. Client verifies certificate, extracts server's public key, generates pre_master_secret, encrypts with server's public key, sends to server
4. Client and server independently compute encryption and MAC keys from pre_master_secret and nonces
5. Client sends a MAC of *all* the handshake messages
6. Server sends a MAC of *all* the handshake messages

48

Real SSL connection



49

Outline

- What is network security?
- Principles of cryptography
- Message integrity, authentication
- Securing e-mail
- Securing TCP connections: SSL
- **Network layer security: IPsec**
- Securing wireless LANs
- Operational security: firewalls and IDS

50

network-layer confidentiality

Between two network entities:

- Sending entity encrypts datagram payload:
  - TCP or UDP segment, ICMP message, OSPF message….
- All data sent from one entity to other would be hidden:
  - web pages, e-mail, P2P file transfers, TCP SYN packets …

➔ "*Blanket Coverage*"

51

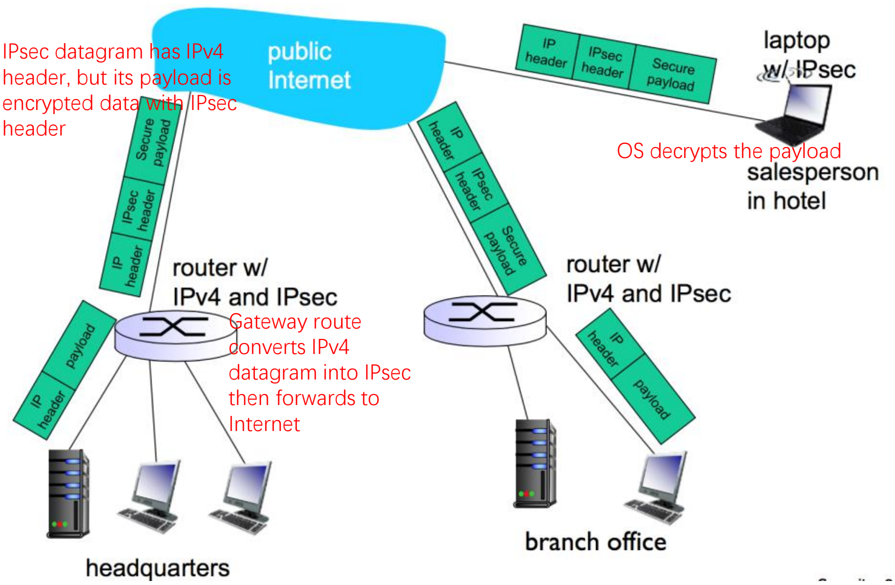
Virtual Private Networks

Motivation:

- Institutions often want *private networks* for security.
  - Separate routers, links, DNS infrastructure ➔ COSTS!
- VPN: Institution's *inter-office traffic* is sent *over public Internet* instead
  - Encrypted before entering public Internet
  - Logically separate from other traffic

52

Virtual Private Networks (VPNs)



53

IPSec standards

- 10+ RFCs
- RFC 4301: Overall IP security architecture
- RFC 6071: Overview of the IPsec protocol suite
  - Two IPsec protocols providing different service models:
    - Authentication Header (AH) protocol: Source authentication & data integrity but *not* confidentiality
    - Encapsulation Security Protocol (ESP): Source authentication, data integrity, ***and*** *confidentiality*
      - More widely used than AH

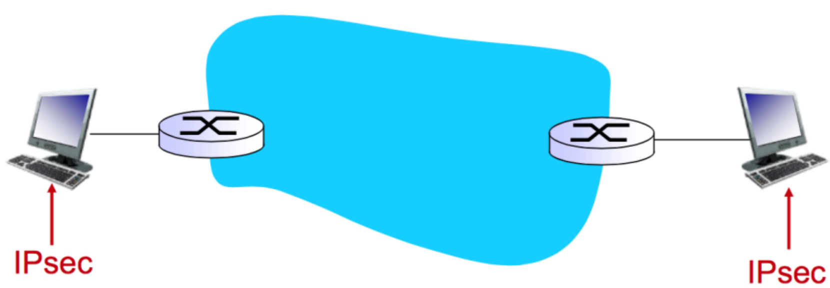
54

IPsec services

- Data ***integrity***
- Origin/Source ***authentication***
- ***Confidentiality***
- Replay attack prevention:
  - Trudy sniffs all messages between Bob and Alice ➔"Played back" to Alice later (on a different TCP session), as Bob!
  - With a nonce in the protocol, Alice will send *different nonces for each TCP session*, causing the encryption keys to be different on the two sessions!

55

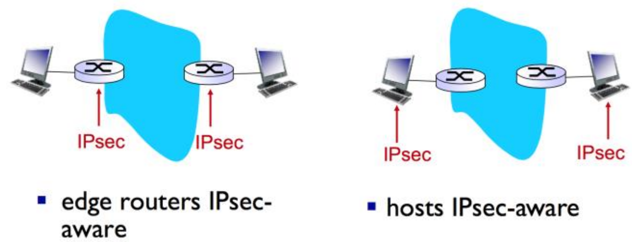
IPsec transport mode

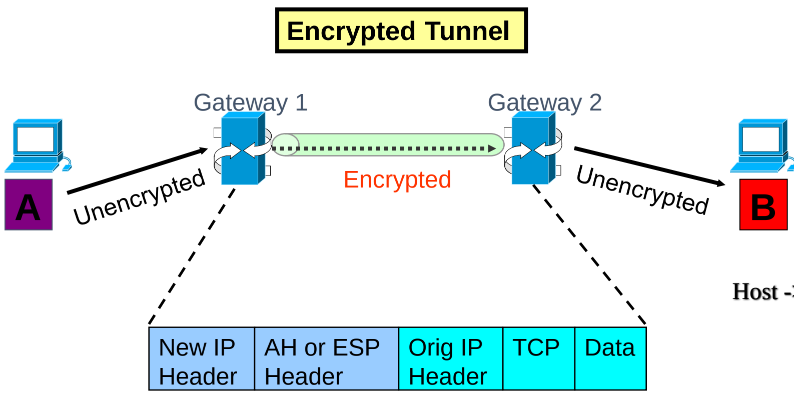


- IPsec datagram emitted and received by end-system : ***host -> host***
- Protects upper level protocols

56

IPsec tunneling mode





Host -> Gateway or Gateway -> Gateway

57

Four combinations are possible!

Host mode with AH	Host mode with ESP
Tunnel mode with AH	Tunnel mode with ESP

most common and most important

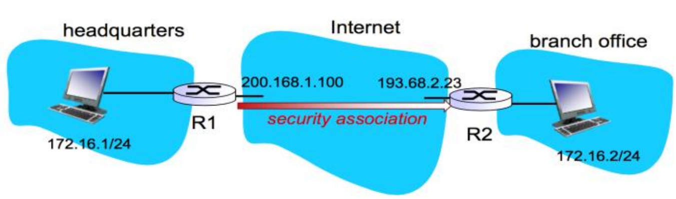
58

Security associations (SAs)

- Before sending data, "*security association* (SA)" is established from sending to receiving entity
  - SA: *Network-layer logical connection*
  - SAs are *simplex*: Only one direction
- Sending, receiving entitles maintain *state information* about SA
  - Recall: TCP endpoints also maintain state info
  - *IP is connectionless; IPsec is connection-oriented*!
- How many SAs in VPN w/ headquarters, branch office, and n traveling salespeople?
  - HQ – Branch & HQ – Sales ➔ 2 + 2n!

59

Example SA from R1 to R2



R1 stores for SA:

32-bit SA identifier: *Security Parameter Index (SPI)*
Origin SA interface (200.168.1.100)
Destination SA interface (193.68.2.23)
Type of encryption used (e.g., 3DES with CBC)
Encryption key
Type of integrity check used (e.g., HMAC with MD5)
Authentication key

60

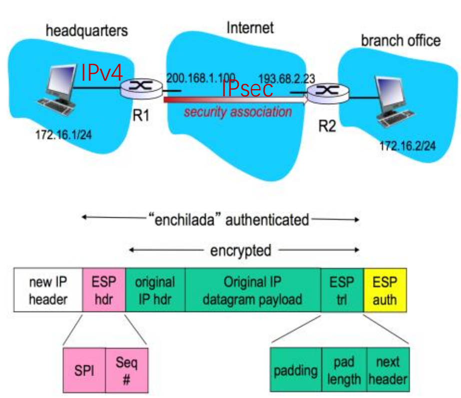
Security Association Database (SAD)

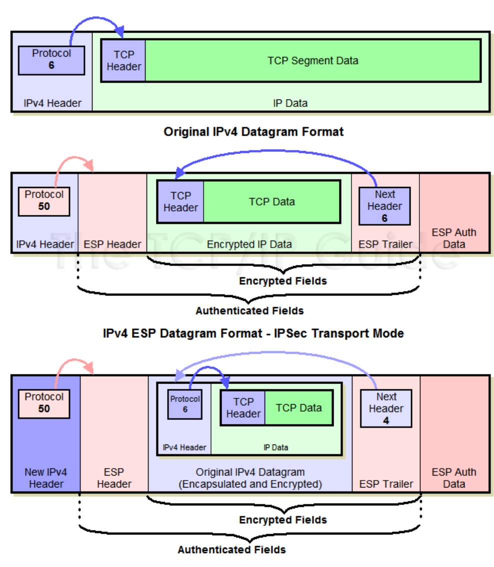
- Endpoint holds SA state in *security association database (SAD)*, where it can locate them during processing.
- SAD is a data structure in the endpoint's *OS kernel*
- With n salespersons, 2 + 2n SAs in R1's SAD
- When sending IPsec datagram, R1 accesses SAD to determine how to process datagram.
- When IPsec datagram arrives to R2, R2 examines SPI in IPsec datagram, *indexes* SAD with SPI, and processes datagram accordingly.

61

WHAT HAPPENS?

IPv4 ➔ Ipsec Datagram

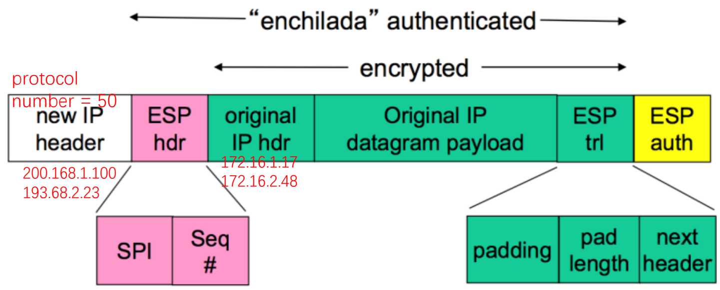




62

Inside the "enchilada"

- ESP trailer: Padding for block ciphers
- ESP header:
  - SPI, so receiving entity knows what to do
  - Sequence number, to thwart replay attacks
- MAC in ESP auth field is created with shared secret key



63

IPsec sequence numbers

- For new SA, sender initializes seq. # to 0
- Each time datagram is sent on SA:
  - Sender increments seq # counter
  - Places value in seq # field
- Goal:
  - Prevent attacker from sniffing and replaying a packet
  - Receipt of duplicate, authenticated IP packets may disrupt service
- Method:
  - Destination checks for duplicates
  - Doesn't keep track of *all* received packets; instead uses a *window*

64

Security Policy Database (SPD)

- Policy: For a given datagram,
  - Sending entity needs to know if it should use IPsec
    ➔ Source and destination IP address; protocol type
  - Also needs to know which SA to use
- An IPSec entity maintains SPD that indicates
  - “What” to do with arriving datagram :
  - “How” to do it ➔ By SAD!

65

IKE: Internet Key Exchange

- *Example SA :* Manual establishment of IPsec SAs in IPsec endpoints
  - SPI: 12345
  - Source IP: 200.168.1.100
  - Dest IP: 193.68.2.23
  - Protocol: ESP
  - Encryption algorithm: 3DES-cbc
    HMAC algorithm: MD5
    Encryption key: 0x7aeaca…
    HMAC key:0xc0291f…
- Manual keying is impractical for VPN with 100/1000s of endpoints
- Instead use *IPsec IKE (Internet Key Exchange)* , RFC5996

66

IKE: PSK and PKI

- Authentication (prove who you are) with either
  - Pre-shared secret (PSK) or
  - With PKI (pubic/private keys and certificates).
- PSK: Both sides start with secret
  - Run IKE to authenticate each other and to generate IPsec SAs (one in each direction), including encryption, authentication keys
- PKI: Both sides start with public/private key pair, certificate
  - Run IKE to authenticate each other, obtain IPsec SAs (one in each direction).
  - Similar with handshake in SSL.

67

IKE phases

- IKE has two phases
  - *Phase 1:* Establish bi-directional IKE SA
    - Note: IKE SA different from IPsec SA
    - Aka ISAKMP security association
  - *Phase 2:* ISAKMP is used to securely negotiate IPsec pair of SAs
- Phase 1 has two modes:
  - Aggressive mode: Fewer messages
  - Main mode: Provides identity protection; More flexible

68

IPsec summary

- IKE message exchange for algorithms, secret keys, SPI numbers
- Either AH or ESP protocol (or both)
  - AH provides integrity, source authentication
  - ESP protocol (with AH) additionally provides confidentiality
- IPsec peers can be two end systems, two routers/firewalls, or a router/firewall and an end system

69

Outline

- What is network security?
- Principles of cryptography
- Message integrity, authentication
- Securing e-mail
- Securing TCP connections: SSL
- Network layer security: IPsec
- **Securing wireless LANs**
- Operational security: firewalls and IDS

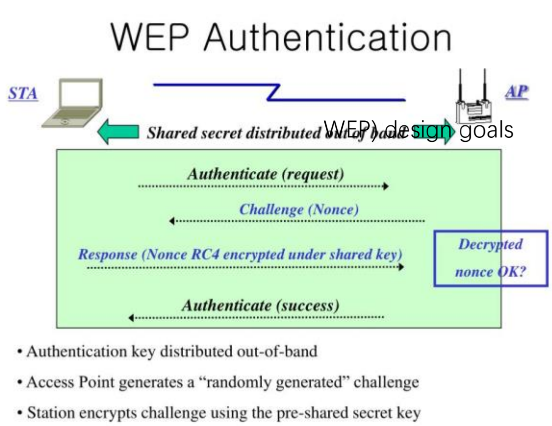
70

Insufficient Security services from 802.11

- Wireless LAN is much easier to be attacked!
- But the original 802.11:
  - Serious security flaws
  - Open to security attacks as if *no security* at all!

Wired Equivalent Privacy (WEP)

- Authentication and data encryption between a host and a wireless AP using a symmetric shared key approach, based on agreed-on keys



WEP) design goals

Notes:

- Not all APs do it, even if WEP is being used
- AP indicates if authentication is necessary in beacon frame
- Done before association

WEP design goals

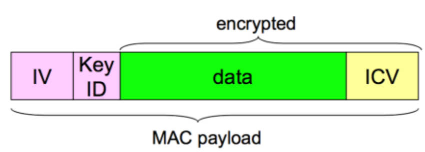
- Symmetric key crypto
  - Confidentiality
  - End host authorization
  - Data integrity
- Self-synchronizing: Each packet separately encrypted
  - Given encrypted packet and key, can decrypt
  - Can continue to decrypt packets when preceding packet was lost (unlike Cipher Block Chaining (CBC) in block ciphers)
- Efficient
  - Implementable in hardware or software

- Became wifi security standard in 1999, adopted in 802.11i in 2003
  ➔Flawed! Retired in 2004

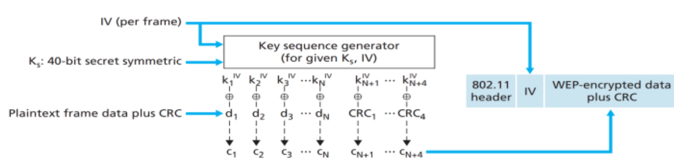
Wi-Fi Protected Access (WPA) : 2003 …

73

WEP encryption



- Sender calculates Integrity Check Value (ICV, four-byte hash/CRC) over data
- Each side has *104-bit shared key*
- Sender creates *24-bit initialization vector* (IV), appends to key: gives 128-bit key
- Sender also appends keyID (in 8-bit field)
- 128-bit key input into pseudo random number generator to get *keystream*
- Data in frame + ICV is encrypted with RC4:
  - Bytes of *keystream* are XORed with bytes of data & ICV
  - IV & keyID are appended to encrypted data to create payload
  - Payload inserted into 802.11 frame



74

Breaking 802.11 WEP encryption

security hole:

- 24-bit IV, one IV per frame, -> IV' s eventually reused
- IV transmitted in plaintext -> IV reuse detected

attack:

- Trudy causes Alice to encrypt known plaintext d_1 d_2 d_3 d_4 …
- Trudy sees: $c_i = d_i$ XOR k_i^{IV}
- Trudy knows c_i d_i, so can compute k_i^{IV}
- Trudy knows encrypting key sequence k_1^{IV} k_2^{IV} k_3^{IV} …
- Next time IV is used, Trudy can decrypt!

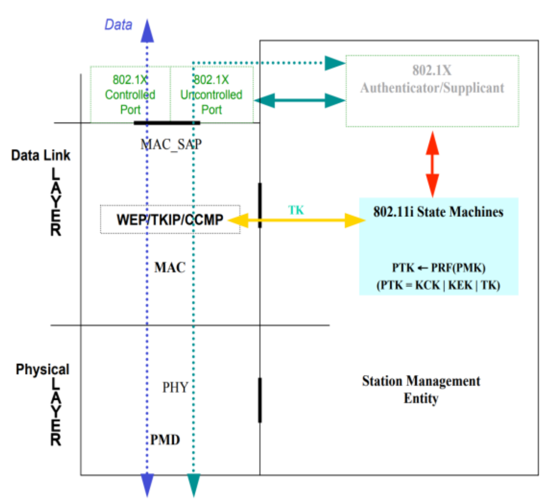
75

802.11i: Improved Security

- Numerous (stronger) forms of encryption possible
- Provides key distribution
- Uses *authentication server* separate from access point ➔ Centralized auth & access

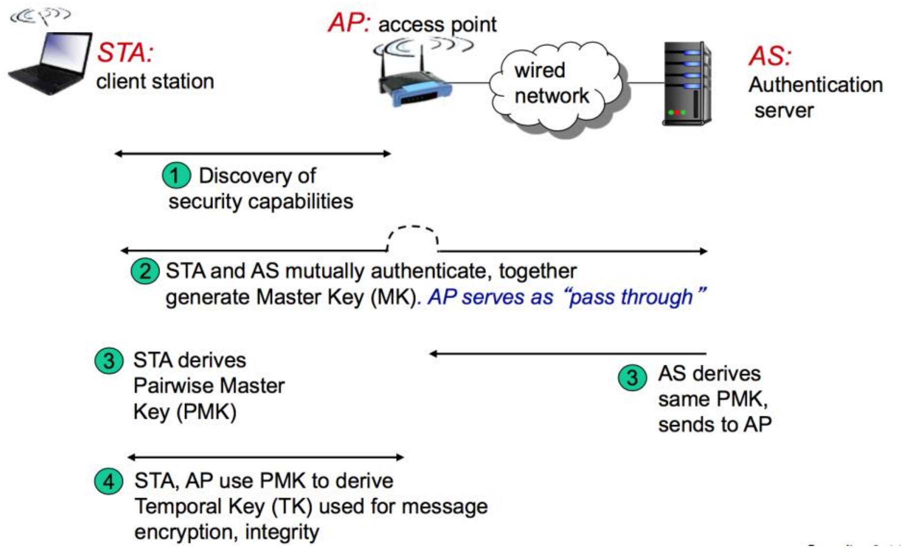
AES-CCMP – all new security protocol based on AES-128 in CCM mode
TKIP – designed as *a software patch to upgrade WEP* in already deployed equipment
WEP – the original 802.11i security protocol
RSNA State Machines – exercises *control* over 802.11i
PRF – Pseudo-Random Function, for *session key* construction
PMK – Pairwise Master Key = session authorization token
KCK – Key Confirmation Key = session "authentication" key
KEK – Key Encryption Key = session key for encrypting keys
TK – Temporal Key = session "encryption" key
4-Way Handshake – 802.11i key management protocol
RSN IE -- Data structure for *advertising and negotiating* security capabilities

802.11i Architecture



76

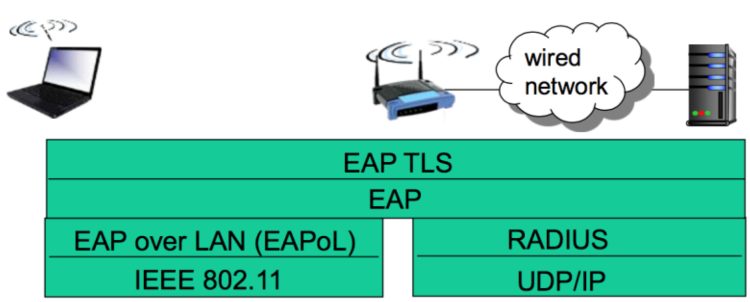
802.11i: Four Phases of Operation



77

Extensible Authentication Protocol (EAP)

- EAP: End-end *client (mobile) to authentication server* protocol
- EAP sent over *separate "links"*
  - Mobile-to-AP (EAP over LAN)
  - AP to authentication server (RADIUS over UDP)



78

Ranking the Wi-Fi Security Methods

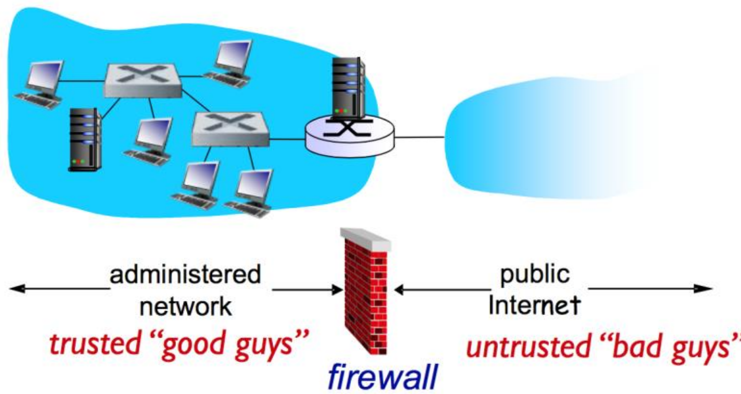
- WPA2 + AES
- WPA + AES
- WPA + TKIP/AES (TKIP is there as a fallback method)
- WPA + TKIP
- WEP
- Open Network (no security at all)

Outline

- What is network security?
- Principles of cryptography
- Message integrity, authentication
- Securing e-mail
- Securing TCP connections: SSL
- Network layer security: IPsec
- Securing wireless LANs
- **Operational security: firewalls and IDS**

Firewalls

- Goal: To isolate organization's internal netw from larger Internet, allowing some packets to pass, while blocking others



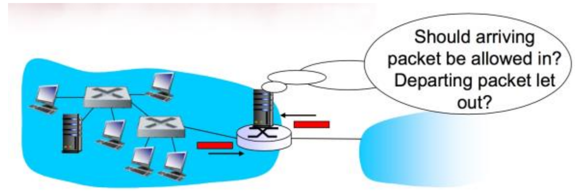
81

Why Firewalls?

- 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

82

Stateless packet filtering



- Internal network connected to Internet via *router firewall*
- Router *filters packet-by-packet,* decision to forward/drop packet based on:
  - Source *IP address*, destination IP address
  - TCP/UDP source and destination *port numbers*
  - ICMP *message type*
  - TCP *SYN and ACK* bits
- Heavy handed tool
  - Admits packets that "make no sense,"e.g.,destport= 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

83

Stateless packet filtering: examples

Policy	*Firewall Setting*
No outside Web access.	Drop all outgoing packets to any IP address, port 80
No incoming TCP connections, except those for institution's public Web server only.	Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
Prevent Web-radios from eating up the available bandwidth.	Drop all incoming UDP packets - except DNS and router broadcasts.
Prevent your network from being used for a smurf DoS attack.	Drop all ICMP packets going to a "broadcast" address (e.g. 130.207.255.255).
Prevent your network from being tracerouted	Drop all outgoing ICMP TTL expired traffic

- *Example 1:* Block incoming and outgoing datagrams with IP protocol field = 17 and with either source or dest port = 23
  - *Result:* All incoming, outgoing *UDP flows and telnet connections* are blocked
- *Example 2:* Block inbound TCP segments with ACK=0.
  - *Result:* Prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside.

84

Access Control Lists (ACL)

- ACL: Table of rules, applied *top to bottom* to incoming packets: (action, condition) pairs: similar to OpenFlow forwarding

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

85

Stateful packet filtering

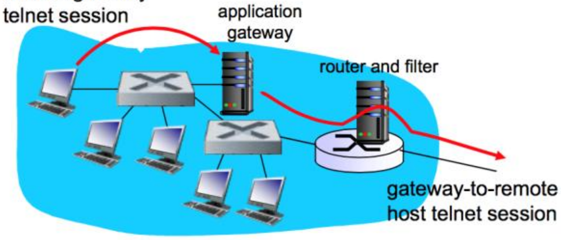
- *Stateful packet filter:* Track status of *every* TCP connection
  - Track connection setup (SYN), teardown (FIN): Determine whether incoming, outgoing packets "*makes sense*"
  - Timeout inactive connections at firewall: No longer admit packets
- ACL augmented to indicate need to check connection state table before admitting packet

action	source address	dest address	proto	source port	dest port	flag bit	check conxion
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	

86

Application gateways

- Filter packets on application data as well as on IP/TCP/UDP fields.
- *Example:* Allow selected internal users to telnet outside



1. Require all telnet users to *telnet through gateway*.

2. For authorized users, gateway sets up telnet connection to dest host. Gateway *relays* data between 2 connections

3. Router filter *blocks* all telnet connections not originating from gateway.

87

Limitations of firewalls & gateways

- *IP spoofing:* Router can't know if data "really" comes from claimed source
- If multiple apps need special treatment, each has own app gateway
- Client software must know how to contact gateway.
  - e.g., must set IP address of proxy in Web browser
- Filters often use *all or nothing policy for UDP*

- *Tradeoff:* Degree of communication with outside world, level of security
- Many highly protected sites still suffer from attacks

88

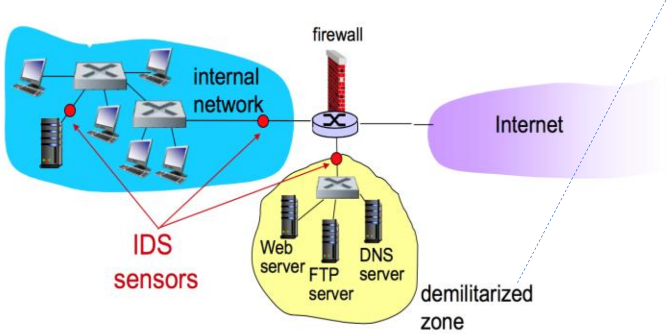
Intrusion detection systems

- 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

89

Intrusion detection systems

- Multiple IDSs: Different types of checking at different locations



- ***DMZ***: **A host or network that acts as a secure and intermediate network or path between an organization's internal network and the external network.**
  - As a front-line network that interacts directly with the external networks while logically separating it from the internal network
  - Limited access to the internal network, and all of its communication is scanned on a firewall before being transferred internally.
  - More secure, safer than a firewall, and can also work as a proxy server.
  - If an attacker intends to breach or attack an organization's network, a successful attempt will only result in the compromise of the DMZ network - not the core!

90

Summary

Basic techniques……
- Cryptography (symmetric and public)
- Message integrity
- End-point authentication

…. used in many different security scenarios
- Secure email
- Secure transport (SSL)
- IPsec
- 802.11

Operational security: Firewalls and IDS

91

Assignments

- Reading: 8.1, 8.2, 8.3, 8.6, 8.7, 8.9
- Problems: P2, P3, P4, P6, P8, P12, P16, P18, P21, P22, P25

92