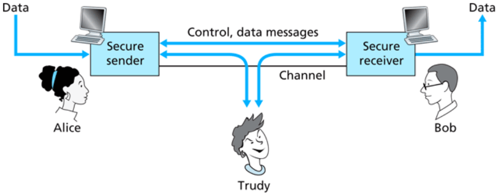
Network Security

CS 360 Internet Programming

Daniel Zappala

Brigham Young University
Computer Science Department

Attacks



- intruder can overhear, modify, insert, or delete messages
- packet sniffing
  - overhear packets sent on the link
  - particularly useful on wireless links
- IP spoofing
  - nothing prevents a host from sending a packet with any IP address
- man-in-the-middle
  - insert a malicious node into the conversation between two hosts
  - can sniff, inject, modify, or delete packets

Denial of Service

- denial of service attack: render a computer unusable by legitimate users
  - vulnerability attack: send crafted messages to stop a service or crash a host
  - bandwidth flooding: send so many packets that the network at a server gets clogged
  - connection flooding: establish a large number of TCP connections at a server
- DDoS: distributed DoS, much harder to detect and defend against

Critical Infrastructure is Vulnerable

- DNS: bandwidth flooding attack
  - flood the DNS root servers with pings
  - carried out Oct 21, 2002 using a botnet
  - many root servers screened out the traffic
  - caching eliminates much of the danger
- other possible DNS attacks
  - flood TLD servers with queries
  - send bogus DNS replies
  - DNS poisoning: send bogus replies to a DNS server
  - send a lot of queries to a server using a spoofed source IP address (reflection attack)

Your Servers are Vulernable

- port scanning: determine which ports are open on a host
- check open ports in case a server with a known security flaw is running
  - e.g. Microsoft SQL Server on port 1434 vulnerable to buffer overflow, exploited by the Slammer worm in 2003-2004
- many port scanners available, e.g. nmap

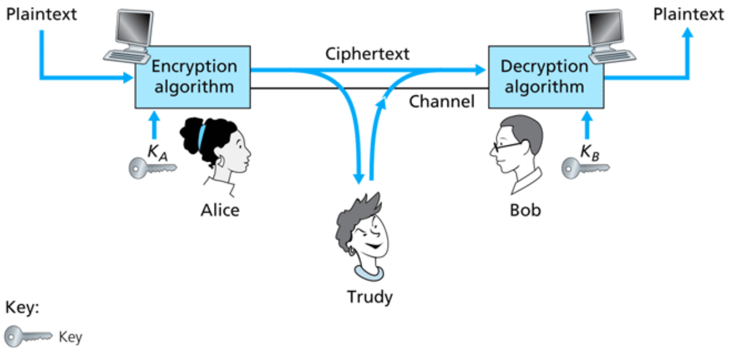
How Did the Internet Get This way?

- *The Design Philosophy of the DARPA Internet Protocols*, David Clark, Proceedings of ACM SIGCOMM 1988, pp. 106–114.
    - primary goal: interoperability among existing networks
    - secondary goals: fault tolerance, multiple transport protocols, minimum assumptions about network capabilities
    - additional goals: distributed management, cost effective, low effort for host attachment, accountability
- no mention of security: assumed that network participants were trustworthy

Security Properties

- confidentiality
  - only the sender and receiver should be able to understand the contents of the message
  - may also want more general confidentiality – obscure the fact that you are talking with someone and the pattern
- integrity
  - ensure that communication is not altered in transit
- authentication
  - confirm the identity of the other party
- operational security/availability
  - preventing access to resources by unauthorized users

Confidentiality



- use cryptography to achieve confidentiality
  - plaintext: the message Alice wants to share with Bob
  - ciphertext: the encrypted form of the plaintext
  - K_A used to encrypt, K_B used to decrypt
  - messages sent over a public channel

Cryptography

- symmetric key cryptography
  - Alice and Bob share a private key
  - encrypt and decrypt messages with the same key
- public key cryptography
  - Alice and Bob each assigned a public key and a private key
  - encrypt a message in the other's public key
  - private key decrypts the message

SKE: Substitution Ciphers

- substitute one letter for another
- Caesar cipher
  - substitute each letter for the one that is k letters later
  - plaintext: “bob, i love you. alice” “bobiloveyoualice”
  - ciphertext: “ree, l oryh brx. dolfh” “reeloryhbrxdolfh”
  - 25 possible key values
- monoalphabetic cipher
  - any letter can substitute for any other letter
  - plaintext: “bobiloveyoualice”
  - ciphertext: “nknsgktcwkymgsbc
  - 26! (10^{26}) possible keys
  - used in Cryptoquote puzzles

Plaintext letter:	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext letter:	m	n	b	v	c	x	z	a	s	d	f	g	h	j	k	l	p	o	i	u	y	t	r	e	w	q

SKE: Substitution Ciphers

- Polyalphabetic Cipher
  - use multiple monoalphabetic ciphers, depending on the position of a letter in the plaintext
  - for example, use C_1, C_2 for alternating letters
  - plaintext: “bobiloveyou”
  - ciphertext: “ghunetoxdhz”
  - note that two ’b’s are coded differently

Plaintext letter:	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
$C_1(k = 5)$:	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
$C_2(k = 19)$:	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s

SKE: Breaking a Cipher

- statistical analysis: e and t are most frequently occurring letters in the English language, common two and three letter words
- code breaking scenarios
  - ciphertext-only: only have access to the ciphertext
  - known-plaintext: know some plaintext, ciphertext combinations
  - chosen-plaintext: intruder can choose plaintext and obtain its ciphertext form

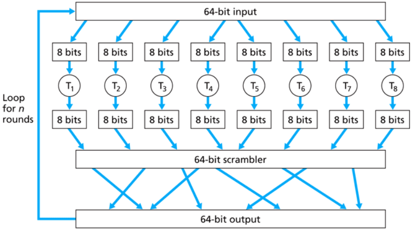
SKE: Block Ciphers

- used in PGP, SSL, IPsec
- divide message into blocks of k bits
- map each block of plaintext to ciphertext
  - plaintext: 010110001111
  - ciphertext: 101000111001
- 2^3 possible inputs, $8! = 40,320$ permutations
- typically use blocks of 64 bits or larger

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

SKE: Implementation of a Block Cipher

- keeping a full table of 2^{64} mappings is infeasible
- instead use a function to simulate randomly permuted tables
- example
  - break 64-bit blocks into 8-bit blocks
  - process by an 8-bit table and reassemble
  - scramble the order of the bits
  - loop for n rounds to make each input bit affect most of the output bits



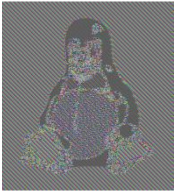
SKE: Block Cipher Details

- algorithms
  - Data Encryption Standard (DES): 64-bit blocks, 56-bit key
  - 3DES
  - Advanced Encryption Standard (AES): 128-bit blocks, 128-, 192-, or 256-bit key
- key length determines table mappings and permutations
- brute-force attacks
  - cycle through all keys: 2^n possible keys for a key length n
  - DES cracked in 6.4 days using $10,000 of hardware, March 2007
  - a system that can crack DES in one second would take 149 trillion years to crack AES

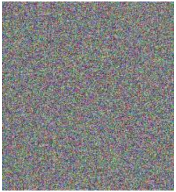
SKE: Cipher Block Chaining

- weakness of block ciphers
  - the same plaintext is encrypted to the same ciphertext
  - any patterns in cleartext are evident in ciphertext
- calculate ciphertext block i: $c(i) = K_s(m(i) \oplus c(i-1))$
- $c(0) =$ initialization vector
- need to distribute initialization vector in the clear, without modification

original



block cipher

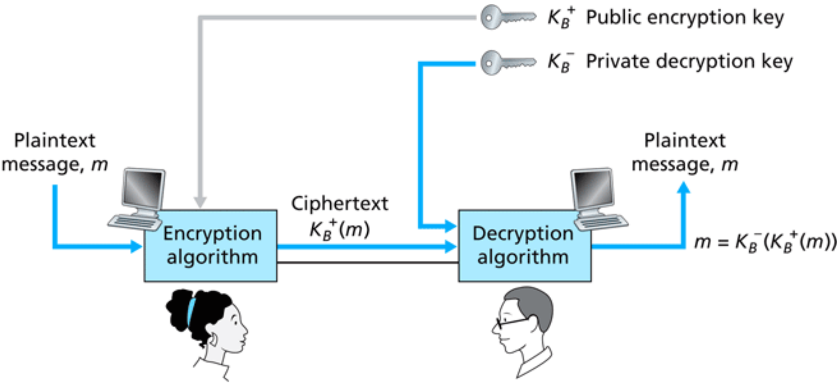


cipher block chaining

Public Key Encryption

- private key encryption requires two parties to share a secret
  - must somehow share the secret
  - meet in person, talk on phone
- public-key encryption
  - communicate securely without sharing a private key
  - can also be used for authentication and digital signatures

PKE: Example



1. Alice fetches Bob's public key, K_B^+
2. Alice encrypts and sends her message, $K_B^+(m)$, using a well-known encryption technique
3. Bob decrypts with private key, $K_B^-(K_B^+(m))$

PKE: Issues

- chosen-plaintext attack
    - choose some text, encrypt with Bob's public key, try to learn the private key
    - must choose keys so that this is hard
- anyone can send an encrypted message to Bob
    - scales much better than private key encryption
    - must use other techniques to verify identity of other party

PKE: RSA

- choose public and private keys
  1. choose two large prime numbers, p and q, such that their product has at least 1,024 bits, preferably 2,048
  2. compute $n = pq$ and $z = (p-1)(q-1)$
  3. choose $e < n$ that has no common factors with z
  4. find a number d such that $ed - 1$ is evenly divisible by z
  5. public key: (n, e), private key (n, d)
- encryption: $c = m^e \mod n$
- decryption: $m = c^d \mod n$

PKE: RSA Toy Example

- values
  - $p = 5,\ q = 7$
  - $e = 5,\ d = 29$

Alice's Encryption

Plaintext Letter	m: numeric representation	m^e	Ciphertext $c = m^e \bmod n$
l	12	248832	17
o	15	759375	15
v	22	5153632	22
e	5	3125	10

Bob's Decryption

Ciphertext c	c^d	$m = c^d \bmod n$	Plaintext Letter
17	4819685721067509150915091411825223071697	12	l
15	127834039403948858939111232757568359375	15	o
22	851643319086537701956194499721106030592	22	v
10	100000000000000000000000000000	5	e

PKE: Issues

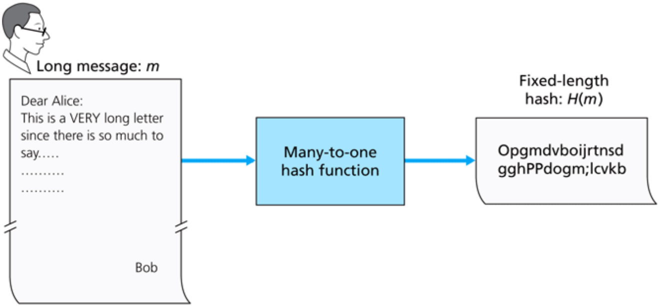
- session keys
  - exponentiation is an expensive process
  - use public key encryption to exchange a private session key
- how do you compute these values?
  - choose large prime number
  - choose e and d
  - perform exponentiation with large numbers
  - see references listed in the Kurose book
- why does it work?
  - see Kurose book for brief discussion, references for more details

Integrity

- ensure the message is not altered (data integrity)
- ensure the message is from the other party (source integrity or message authentication)

Cryptographic Hash Function

- takes an input, m, and computes a fixed-size string (hash)
- hash function chosen so that it is computationally infeasible to:
    - reverse the hash and recreate the original message
    - find two messages that hash to the same value



Choosing a Hash Function

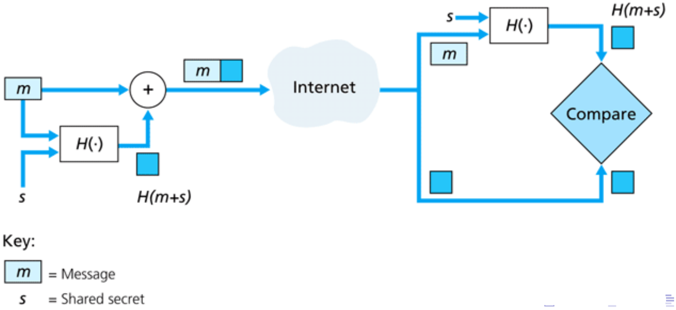
- easy to foil a simple checksum
- MD5: 128 bit hash, security is currently questionable
- SHA-1: 160-bit hash, more secure but recently discovered weakness
  - 2005 Chinese researchers reported to find collisions 2,000 times faster than brute force
  - still very very very hard to break, but attacks always get better
- with any sign of weakness, time to work to develop new hash functions

Data Integrity

- easy to provide data integrity without authentication
  1. Alice creates hash $H(m)$
  2. Alice sends $(m, H(m))$ to Bob
  3. Bob receives (m, h) and checks if $H(m) = m$
- Bob can't be sure the message came from Alice
- useful anyway
  - checking that you downloaded an unmodified version of a file
  - assumes that the MD5 hasn't been modified

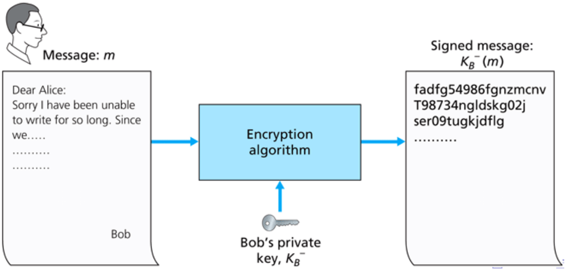
Data Integrity and Authentication

- Alice and Bob share a secret, the authentication key
  1. Alice calculates $H(m+s)$, the *message authentication code*
  2. Alice sends Bob $(m, H(m+s))$
  3. Bob receives (m, h) and checks if $H(m+s) = h$
- popular standard: HMAC
- anyone who shares the key can generate an authenticated message



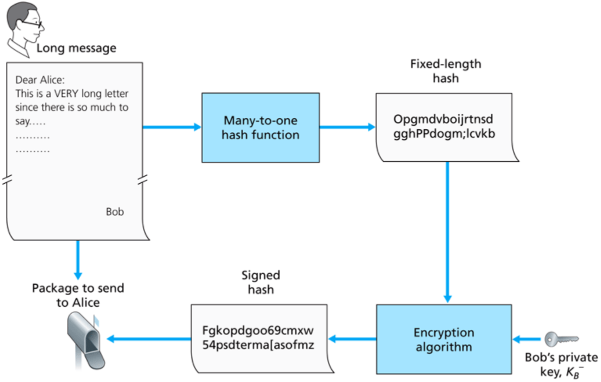
Digital Signatures

- want to be able to verify the owner or creator of a document, or signify agreement with the document's content
- properties
  - verifiability: can prove it was signed by a person
  - non-repudiation: can prove that only that person could have signed it
  - integrity: signature fails if document modified
- $K_B^+(K_B^-(m)) = m$

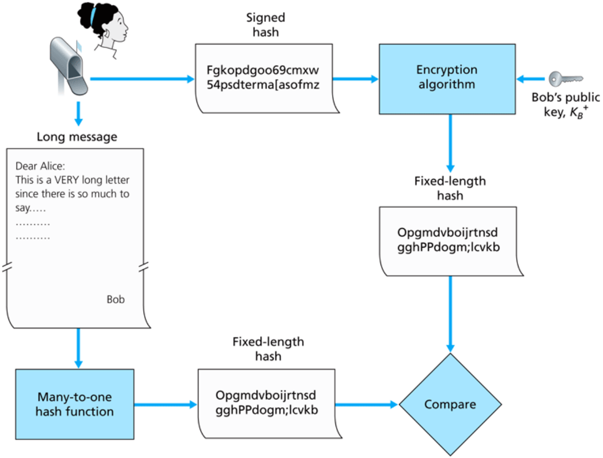


Simplifying Computation

- encryption is computationally expensive
- sign a hash of the message instead

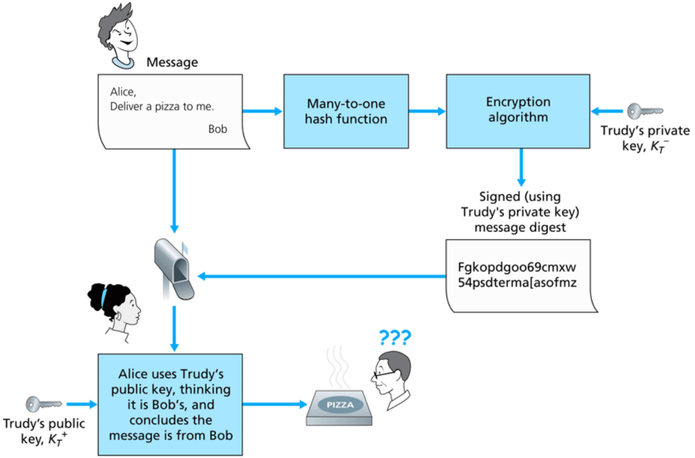


Verifying a Signature



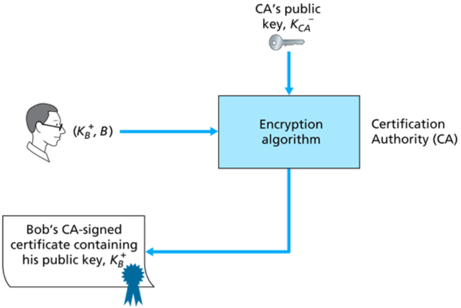
Public Keys are No Guarantee

- must certify that a public key belongs to a certain entity



Public Key Certification

- typically performed by a Certificate Authority (CA)
  - verifies that an entity (person, computer) is who it says it is - verification procedure is left to the CA
  - creates a certificate that contains the public key and a unique identifier for the entity (e.g. email address, IP address)
  - signs the certificate

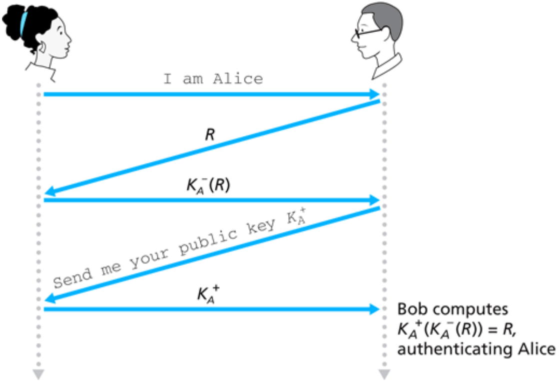


Authentication

- prove your identity to someone over the network
  - message authentication verifies only that the message came from a particular person
  - subject to a replay attack
- authenticate first, then exchange messages

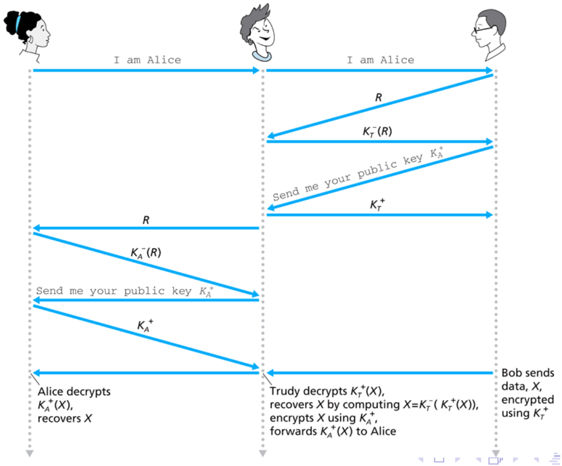
Nonce + Public Key

- use a public key to decrypt a nonce encrypted with a private key
- requires public key infrastructure



Dangers of Insecure Public Key Infrastructure

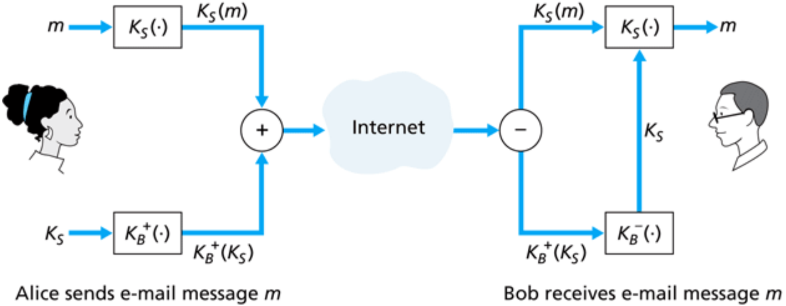
- Trudy can impersonate Alice
- man-in-the-middle attack



Securing Email

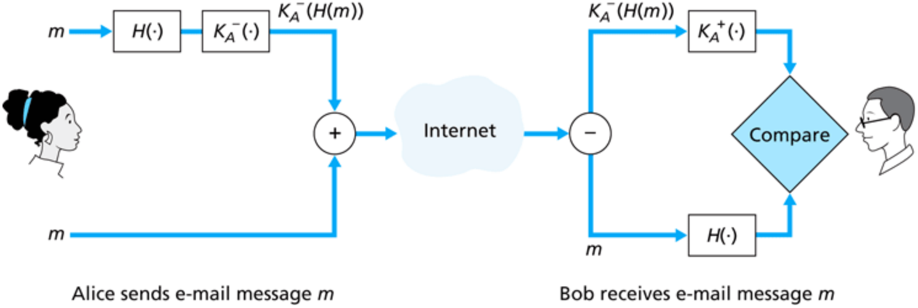
- goals
  - confidentiality
  - message integrity
  - sender authentication
  - receiver authentication

Email Confidentiality



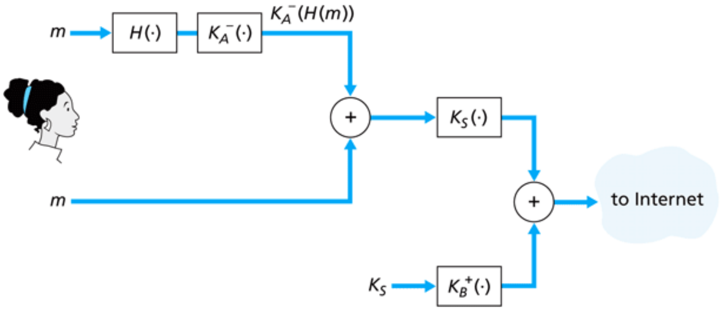
- Alice
  - encrypts message with a symmetric key
  - encrypts session key with Bob's public key
  - sends Bob the encrypted message and session key
- Bob
  - decrypts session key using private key
  - uses session key to decrypt message

Email Integrity and Sender Authentication



- Alice
  - creates a message digest with a hash function
  - signs the digest with her private key
  - sends unencrypted message and digest to Bob
- Bob
  - checks digest using Alice's public key, hash of message
  - reads the message

Confidentiality, Integrity, Sender Authentication



- Alice sends signed digest and message, encrypted with shared private key, plus shared key encrypted in Bob's public key
- Bob reverses the process

PGP

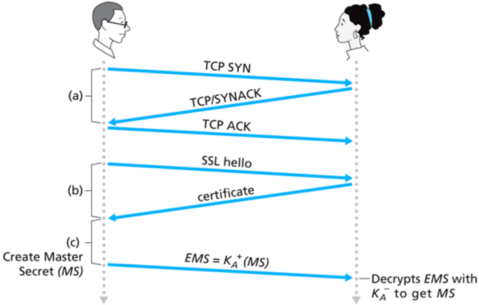
- design is basically the same as previous figure
- can use different hash functions, encryption algorithms
- simplifies creation of public and private key pairs, signed by a web of trust

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash:  SHA1
Bob:
Can I see you tonight?
Passionately yours, Alice
-----BEGIN PGP SIGNATURE-----
Version: PGP for Personal Privacy 5.0
Charset:  noconv
yhHJRHhGJGhgg/12EpJ+lo8gE4vB3mqJhFEvZP9t6n7G6m5Gw2
-----END PGP SIGNATURE-----
-----BEGIN PGP MESSAGE-----
Version: PGP for Personal Privacy 5.0
u2R4d+/jKmn8Bc5+hgDsqAewsDfrGdszX68liKm5F6Gc4sDfcXyt
RfdS10juHgbcfDssWe7/K=lKhnMikLo0+1/BvcX4t==Ujk9PbcD4
Thdf2awQfgHbnmKlok8iy6gThlp
-----END PGP MESSAGE
```

SSL

- provides confidentiality, data integrity, authentication for TCP connections
  - modified version of SSL3, standardized by IETF as TLS
  - used to secure nearly all e-commerce sites, signified by https
- goals
  - confidentiality - protect credit card information, order privacy
  - data integrity - ensure order is not modified
  - server authentication - ensure user is shopping at the right site
- provides an interface between the application and TCP

SSL Basics



- establish connection, get signed public key from Alice, then send Alice a master secret
- use master secret to generate
  - E_B: encryption key for data from Bob to Alice
  - M_B: MAC key for data from Bob to Alice
  - E_A: encryption keyf or data from Alice to Bob
  - M_A: MAC key for data from Alice to Bob

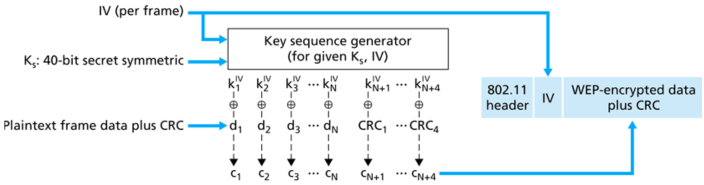
SSL Details

- data integrity
  - break data stream into records, append a MAC to each record
  - must use an SSL sequence number for each record to prevent reordering by an intruder (TCP sequence numbers are not secured)
- handshake
  - client sends a list of cryptographic algorithms it supports
  - server choose a symmetric algorithm, a public key algorithm, and a MAC algorithm
  - use a MAC to ensure handshake messages not altered
- closing
  - must send an SSL close message
  - prevents a truncation attack, where an intruder sends a FIN message for your connection

Wired Equivalent Privacy

- encryption between a host and an access point using a symmetric shared key
  1. wireless host requests authentication by access point
  2. access point responds with 128-bit nonce
  3. wireless host encrypts nonce with symmetric key
  4. access point decrypts nonce

$$c_i = d_i \oplus k_i^{IV}$$

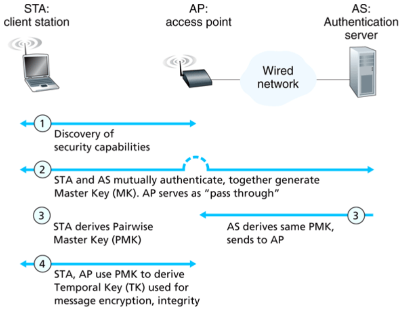


WEP Weaknesses

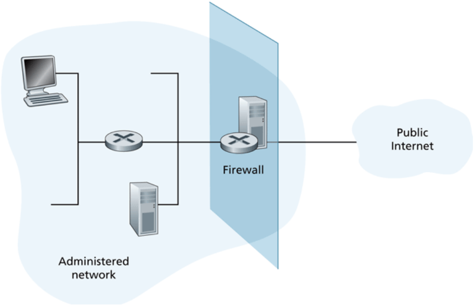
- WEP key, K_S, rarely changes, but IV must change for every frame
    - only 2^{24} possible IVs
    - if chosen randomly, the probability of having the same IV used twice is $> 99\%$ after only 12,000 frames
    - for 1K frames and 11 Mbps, this takes a few seconds
- attack
    - intruder sees IV in plaintext, can compute k_i^{IV} from the stream of data
    - eavesdropper sees the same IV, thus knows the k_i^{IV} for the message, and can decrypt it
- weakness in CRC algorithm that opens WEP to integrity attacks

IEEE 802.11i

- amendment to IEEE 802.11 standard, applicable to 802.11a,b,g
- allows the use of public keys to authenticate client and AS to each other, generate master key for client and AP
- WPA uses a pre-shared key between client and AP instead



Firewalls



- provide a gateway where traffic is checked before entering or exiting an organization
- only authorized traffic is allowed to pass

Packet Filter

- based on source and destination addresses, protocol type, source and destination ports, TCP flags, ICMP message type

Policy	Firewall Setting
No outside Web access.	Drop all outgoing packets to any IP address, port 80
No incoming TCP connections, except those for organization'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 packets.
Prevent your network from being used for a smurf DoS attack.	Drop all ICMP ping packets going to a "broadcast" address (eg 130.207.255.255).
Prevent your network from being tracerouted	Drop all outgoing ICMP TTL expired traffic

Policies for network 130.27/16 with web server at 130.207.244.203

Packet Filter Example

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

- first two rules allow internal users to surf the web
- second two rules allow DNS traffic to enter and leave the network

Stateful Packet Filter

- traditional packet filter: examine each packet individually
- stateful packet filter: track TCP connections
    - ensures that packets allowed by the filter must be part of an active connection
    - prevents an attacker from injecting malformed packets that happen to meet a filter rule

Stateful Packet Filter Example

source address	dest address	source port	dest port
222.22.1.7	37.96.87.123	12699	80
222.22.93.2	199.1.205.23	37654	80
222.22.65.143	203.77.240.43	48712	80

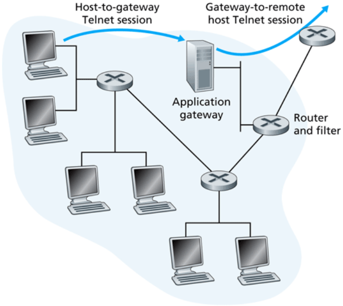
Connection table

action	source address	dest address	protocol	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	

Access control list

Application Gateway

- requires a user to authenticate before connecting outside the network
- technically, email server and web caches are application gateways



Intrusion Detection Systems

- even if an application gateway allows only authorized users to exit the company network, those connections that are authorized can still be attacked
  - a general Intrusion Detection System examines packet contents for attack signatures and generates appropriate alerts
  - an Intrusion Prevention System will also filter out the suspicious traffic
- can detect network mapping, port scans, TCP stack scans, DoS bandwidth flooding attacks, worms, viruses, OS vulnerability attacks, application vulnerability attacks

IDS Types

- signature-based system
  - maintains a database of attack signatures, including standard filter fields and strings found in the packet payload
  - crafted by people who investigate attacks, after they have been observed on the Internet
  - must compare every incoming packet to the list of signatures – requires very high-speed processing
  - snort: open source, comes with a large signature database that is constantly maintained
- anomaly-based system
  - observes traffic and examines patterns
  - anomalies, such as a burst of ICMP traffic or a large number of incoming or outgoing connections trigger a response
  - very challenging to distinguish between normal traffic and unusual traffic