CS 268: Network Security

Ion Stoica
April 3, 2002

Security Requirements

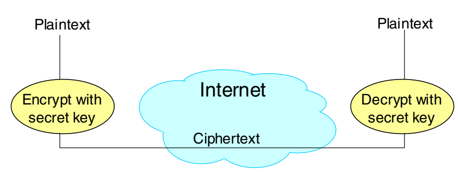
- Authentication
  - Ensures that the sender and the receiver are who they are claiming to be
- Data integrity
  - Ensure that data is not changed from source to destination
- Confidentiality
  - Ensures that data is red only by authorized users
- Non-repudiation
  - Ensures that the sender has strong evidence that the receiver has received the message, and the receiver has strong evidence of the sender identity, strong enough such that the sender cannot deny that it has sent the message and the receiver cannot deny that it has received the message (not discussed in this lecture)

Cryptographic Algorithms

- Security foundation: cryptographic algorithms
  - Secret key cryptography, Data Encryption Standard (DES)
  - Public key cryptography, RSA algorithm
  - Message digest, MD5

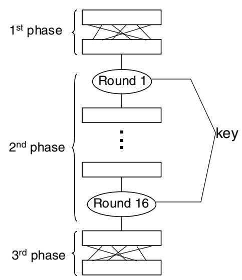
Symmetric Key

- Both the sender and the receiver use the same secret keys



Data Encryption Standard (DES)

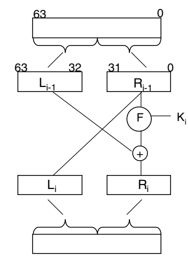
- DES encrypts a 64-bit block of plain text using a 64-bit key
- Three phases
  1. Permute the 64 bits in the block
  2. Apply a given operation 16 times on the 64 bits
  3. Permute the 64 bits using the inverse of the original permutation



Operation in Each Round of 2nd Phase

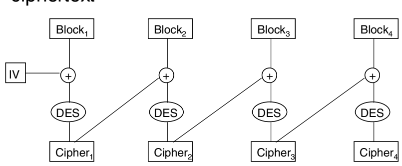
- Key is 56 bits
- Each round the key is modified and 48 bits are selected from it. Given result K_i, the following operations are performed

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$



Encrypting Larger Messages

- Initialization Vector (IV) is a random number generated by sender and sent together with the ciphertext

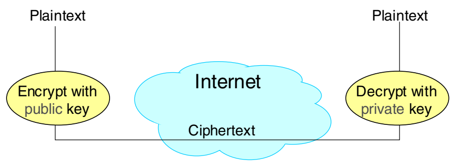


Discussion

- Can provide confidentiality
- No mathematical proof, but practical evidence suggests that decrypting a message without knowing the key requires exhaustive search
- To increase security use triple-DES, i.e., encrypt the message three times

RSA (Rivest, Shamir, and Adleman)

- Sender uses a public key
- Receiver uses a private key



Generating Public and Private Keys

- Choose two large prime numbers p and q (>= 256 bit long) and multiply them: $n = p*q$
- Chose encryption key e such that e and $(p-1)*(q-1)$ are relatively prime
- Compute decryption key d, where

  $d = e^{-1} \bmod ((p-1)*(q-1))$
- Construct public key from pair (n, e)
- Construct private key from pair (d, n)

RSA Encryption and Decryption

- Encryption:
  - $c = m^e \bmod n$
- Decryption:
  - $m = c^d \bmod n$

Example

- Choose p = 7 and q = 11 → n = p*q = 77
- Compute encryption key e: (p-1)*(q-1) = 6*10 = 60 → chose e = 13 (13 and 60 are relatively prime numbers)
- Compute decryption key d: d = 13^{-1} mod 60 → 13*d mod 60 = 1 mod 60 → d = 37 (37*13 = 481)

Example (cont'd)

- n = 77; e = 13; d = 37
- Send message m = 7
- Encryption: c = m^e mod n = 7^{13} mod 77 = 35
- Decryption: m = c^d mod n = 35^{37} mod 77 = 7

Discussion

- Can provide confidentiality
- Receiver A computes n, e, d, and sends out (n, e)
  - Everyone who wants to send a message to A uses (n, e) to encrypt it
- How difficult is to recover d ? (Someone that can do this can decrypt any message sent to A!)
- Recall that

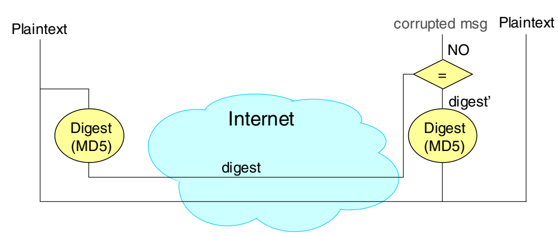
  $d = e^{-1} \bmod ((p-1)*(q-1))$
- So to find d, you need to find primes factors p and q
  - This is provable very difficult

Message Digest (MD) 5

- Can provide data integrity
  - Used to verify the authentication of a message
- Idea: compute a hash on the message and send it along with the message
- Receiver can apply the same hash function on the message and see whether the result coincides with the received hash

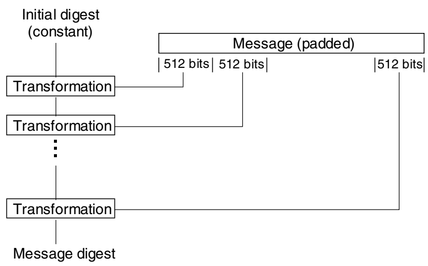
MD 5 (cont'd)

- Basic property: digest operation very hard to invert
  - in practice someone cannot alter the message without modifying the digest



Message Digest Operation

- Transformation contains complex operations (see Peterson&Davie, page 583)



Authentication

- Validate a mapping between two entities
  - alice@cs.berkeley.edu↔Alice
  - www.whitehouse.gov↔Whitehouse of USA
  - www.whitehouse.com↔entertainment provider (not Whitehouse of USA)
- Solutions
  - Passwords
  - Encryption
  - Biometrics

Key Distribution Problem

- Many of the previous algorithms rely on keys
- How do two parties securely get keys to do privacy, authentication, etc.?
- Set up a secure connection using different key
  - How to bootstrap?
- Out-of-band key distribution
  - Floppy disk, piece of paper, telephone, etc.
  - High latency, wastes human time
- Must be done whenever key is compromised, entity is added, keys expire

Needham and Schroeder

- Addresses key distribution problem
- Reduces number of keys distributed out-of-band
- Assumes malicious user can read, modify, drop, and fabricate messages

Interactive Connection, Symmetric Key

1) A$\rightarrow$AS: A,B,I_{A1}

to get CK from AS

no encryption

2) AS $\rightarrow$A: $\{I_{A1}$,B,CK,$\{CK,A\}^{KB}\}^{KA}$

to send CK to A

Encrypted with KA so only A can read it and so A knows it came from AS

I_{A1} so that A knows this isn't a replay (why?)

B so that A knows this isn't a man in middle attack (why?)

Interactive Connection, Symmetric Key

3) A$\rightarrow$B: $\{CK,A\}^{KB}$

to send CK to B

encrypted with KB so that B knows it came from the AS and A is authenticated

4) B$\rightarrow$A: $\{I_B\}^{CK}$

5) A$\rightarrow$B: $\{I_B\text{-}1\}^{CK}$

so B can determine if 3) is a replay

Interactive Connection, Symmetric Key

- What if CK is compromised?
  - Attacker
    - Listens to previous conversation between A and B
    - Breaks CK eventually
    - Spoofs A, sends copy of messages 3,4,5 to B
- Add timestamp to messages:

  2) AS$\rightarrow$A: $\{I_{A1}$,B,CK,$\{CK,A,TS\}^{KB}\}^{KA}$

  3) A$\rightarrow$B: $\{CK,A,TS\}^{KB}$

  B ignores if TS is too old
  - Need synchronized clock (why?)
  - How to secure clock synchronization protocol?

Interactive Connection, Public Key

1) A$\rightarrow$AS: A,B

to get PKB from AS

2) AS$\rightarrow$A: $\{PKB,B\}^{SKAS}$

to send PKB to A

assume that A knows PKAS securely

encryption for integrity not privacy

B so that A knows 1) was good

3) A$\rightarrow$B: $\{I_A,A\}^{PKB}$

tells B that A wants to talk

Interactive Connection, Asymmetric key

4) $B \rightarrow AS$: B,A

5) $AS \rightarrow B$: $\{PKA,A\}^{SKAS}$

Same as 1) and 2)

6) $B \rightarrow A$: $\{I_A,I_B\}^{PKA}$

Prevent replay from B to A

7) $A \rightarrow B$: $\{I_B\}^{PKB}$

Prevent replay from A to B

Discussion

- Messages sent
  - Symmetric key: 5, 3 with caching
  - Asymmetric key: 7, 3 with caching
  - Caching introduces vulnerabilities
    - key could have been compromised
- Resists some attacks
  - Eavesdropping
  - Replay
- AS
  - Symmetric key: needs secret dbase, secure transactions
  - Asymmetric key: doesn't need secret dbase; needs only to store items of form: A: $\{PKA, A\}^{SKAS}$

Problems

- Authentication Server
  - Single point of failure
    - Could be compromised, crashed, overloaded
  - Must be securely administered
    - Must have administrator trusted by all principals
    - Adding principals requires contacting administrators $\rightarrow$ very slow
- Inter-domain communication
  - Each domain has separate authentication server
  - Hierarchy of domains
    - parent domains must be trusted by child domains
  - Must go through administrator

One-Way Communication

- Symmetric key

  $A \rightarrow B$: $(CK, A)^{KB}$

  add at the head of message encrypted with CK; self-authenticated
- Public Key

  $A \rightarrow B$: $(A, I, B^{SKA})^{PKB}$

  I is a nonce in the message

Conclusion

- Systems derived from Needham-Schroeder
  - Kerberos
    - Popular in large centralized organizations
    - Centralized structure does not suit Internet
  - SSL
    - Used for secure TCP connections
- Key distribution is still a hard problem
  - Many systems more vulnerable to key distribution attacks than crypto failure

```
The authenticity of host 'host.domain.com (10.0.0.1)'
can't be established.RSA key fingerprint is
be:3c:a3:8f:6d:70:32:78:e1:df:68:0f:ec:d2:f4:19.

Are you sure you want to continue connecting (yes/no)?
```


Denial of Service

- Huge problem in current Internet [MVS01]
  - Yahoo!, Amazon, eBay, CNN, Microsoft attacked
  - 12,000 attacks on 2,000 organizations in 3 weeks
  - some more that 600,000 packets/second
    - more than 192Mb/s
  - most documented perpetrators are determined teenagers using freely available tools
    - consider if the attacker is a large, well-funded group of professionals using secret tools
    - may have already happened
  - preventing deployment of critical applications
    - medical, energy, transportation

Problem: 0wning

- Attacker compromises a large number of hosts
  - 1M compromised hosts is plausible
- exploits security flaws in OS and applications
  - bugs, e.g., buffer overruns ("`strcpy(dest, src);`")
  - poor security policy, e.g., automatically executed email attachments
  - crypto, authentication systems do not prevent
  - firewalls do not prevent email viruses
- hosts usually have high bandwidth connections (e.g., DSL)

Problem: Attack

- Compromised hosts send TCP SYN packets to target
  - sent at max rate with spoofed source address
  - more sophisticated attacks possible
    - attack DNS, BGP
    - reflection
      - cause one non-compromised host to attack another
      - examples?
- Affect on target host
  - may crash or slow down drastically
  - connection to the Internet is saturated

Dealing with Attack

- distinguish attack from flash crowd (why?)
- prevent damage [M+01]
  - distinguish attack traffic from legitimate traffic
  - rate limit attack traffic
- stop attack
  - identify attacking machines
  - shutdown attacking machines
  - usually done manually, requires cooperation of ISPs, other users
- identify attacker
  - very difficult, except
  - usually brags/gloats about attack on IRC
  - also done manually, requires cooperation of ISPs, other users

Incomplete Solutions

- Fair queueing (why?)
- Integrated Services and Differentiated Services (why?)
- RSVP (why?)
- Quality of service mechanisms usually assume that users are selfish, but not malicious

Identifying Attacking Machines

- Defeat spoofed source addresses
- Does not stop or slow attack
- Egress filtering
  - a domain's border router drop outgoing packets which do not have a valid source address for that domain
  - if universal, could abolish spoofing (why isn't it universal?)
- IP Traceback [many proposals]
  - similar to DPS
  - routers probabilistically tag packets with an identifier
  - destination can infer path to true source after receiving enough packets

Aggregate Congestion Control [M+01]

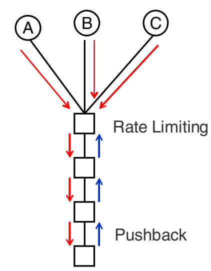
- goal: prevent damage from both attacks and flash crowds
- distinguish attack traffic from legitimate traffic
  - identify an aggregate of flows causing many drops
- limit aggregate
  - decide on bandwidth that limits drops
- convey decision to up stream routers
  - so up stream routers do not waste bandwidth delivering traffic that will be dropped

Distinguishing Aggregates

- Cluster together flows
- Too specific: does not affect drop rate (why?)
- Not specific enough: slow down legitimate traffic
- Cluster attributes: source/dest addr, source/dest port
- Examples
  - dest: cnn.com (+/-?)
  - dest: cnn.com/port 80 (+/-?)
  - dest: cnn.com/port 80, src: dosrus.com
- Clustering algorithm may have to be kept secret
- Current solutions use heuristics
  - open research problem

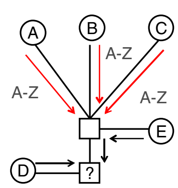
Pushback

- Convey information about high rate aggregate up stream
  - Why not necessary for flash crowd?
  - Why is it necessary for upstream routers to drop traffic?
  - Why do upstream routers need notification from downstream routers?



Pushback Issues

- Necessary if downstream router cannot identify aggregate
- Attack may still be too broad to distinguish
- Why would upstream routers trust downstream routers in different domains?



Conclusions

- Most significant problem in Internet today
- Traditional solutions ineffective
  - QoS, crytography, authentication
- Pushback provides general framework for solution
- Many problems remain

Network Intrusion Detection System (NIDS)

- Goal: automatically detect unauthorized access to hosts over the network
  - assume attacker has already compromised system
  - exploited inevitable flaws in system
    - bugs
    - compromised keys, passwords because of user mistakes
- maintain database of rules
  - e.g., “host X should never allow remote access”, “host Y should only be sent valid DNS queries”
- capture packets at border router and compare with database
- notify administrator in real time or automatically block intruder

Network Intrusion Detection Issues

- Why use NIDS in addition to firewall
  - NIDS doesn’t block traffic, so it can protect hosts outside of firewall
  - Firewall doesn’t prevent all forms of intrusion (e.g. email virus)
- Accuracy
  - rules are too general → too many false positives
  - rules are too specific → intruders undetected
- Fundamental rules
  - rules specific to application implementation → rule must change when application changes
  - application generic rules are difficult to formulate
  - e.g., interactive traffic can be characterized by distribution of human inter-character typing interval

- Little advantage for interactive communication
  - most people connect to only a fraction of the hosts in a domain → n is small
  - many hosts share same keys → n is small
  - user changes set of hosts with distinct keys infrequently
    - with PK, user can collect all PKs (n) and copy them to all hosts (n) → 2n key distribution instead of n^2