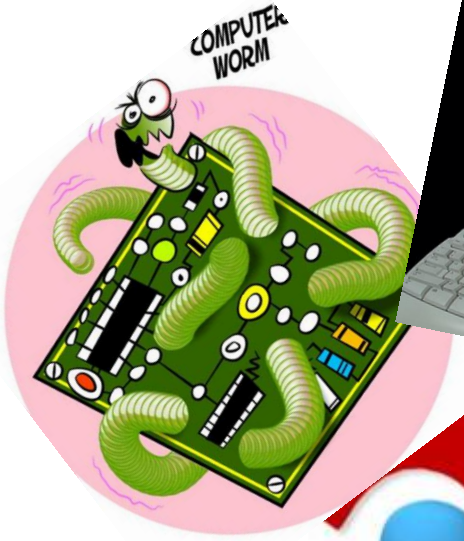
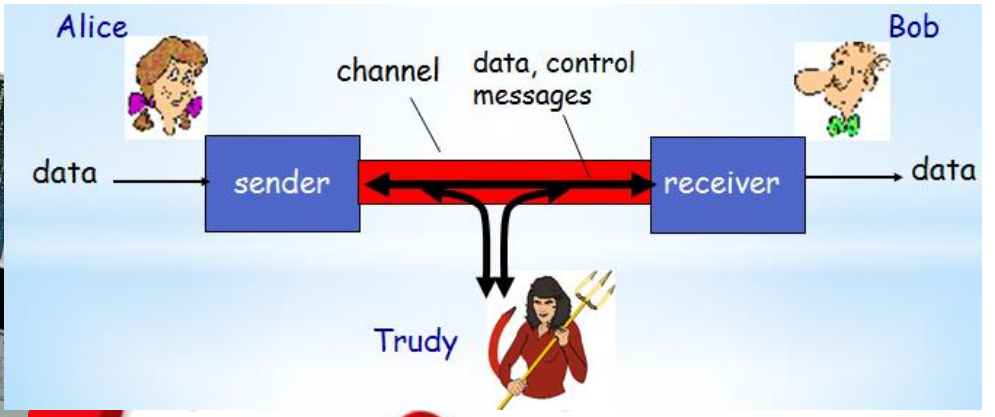
Chapter 7
Network Security

A Brief History of the World


COMPUTER
WORM


Alice
Bob
channel
data, control
messages
data
sender
receiver
data
Trudy


Virus Automatic Alert
VIRUS
DETECTED
Repair
Delete

Who is vulnerable?

- Financial institutions and banks
- Internet service providers
- Pharmaceutical companies
- Government and defense agencies
- Contractors to various government agencies
- Multinational corporations
- **ANYONE ON THE NETWORK**

Common security attacks and their countermeasures

- Finding a way into the network
  - Firewalls
- Exploiting software bugs, buffer overflows
  - Intrusion Detection Systems
- Denial of Service
  - Ingress filtering, IDS
- TCP hijacking
  - IPSec
- Packet sniffing
  - Encryption (SSH, SSL, HTTPS)
- Social problems
  - Education

Denial of Service

TCP Attacks

- Say hello to Alice, Bob and Mr. Big Ears

TCP Attacks

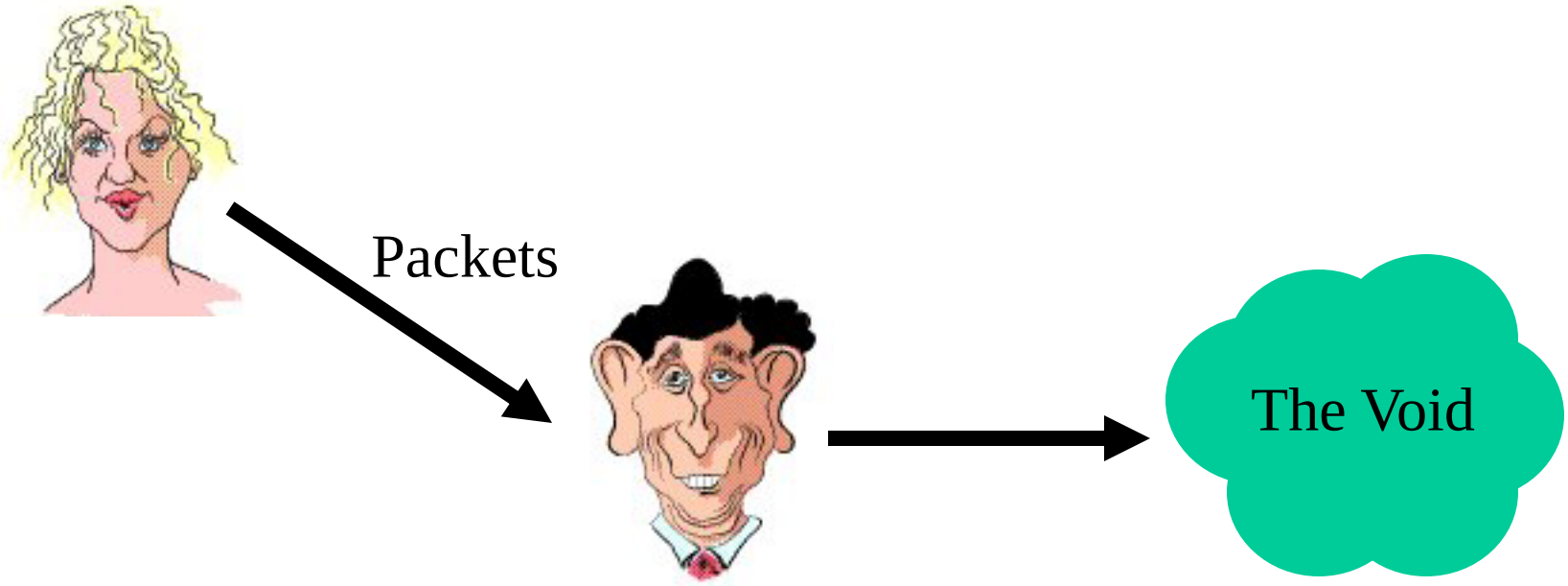
- Alice and Bob have an established TCP connection

TCP Attacks

- Mr. Big Ears lies on the path between Alice and Bob on the network
  - He can intercept all of their packets

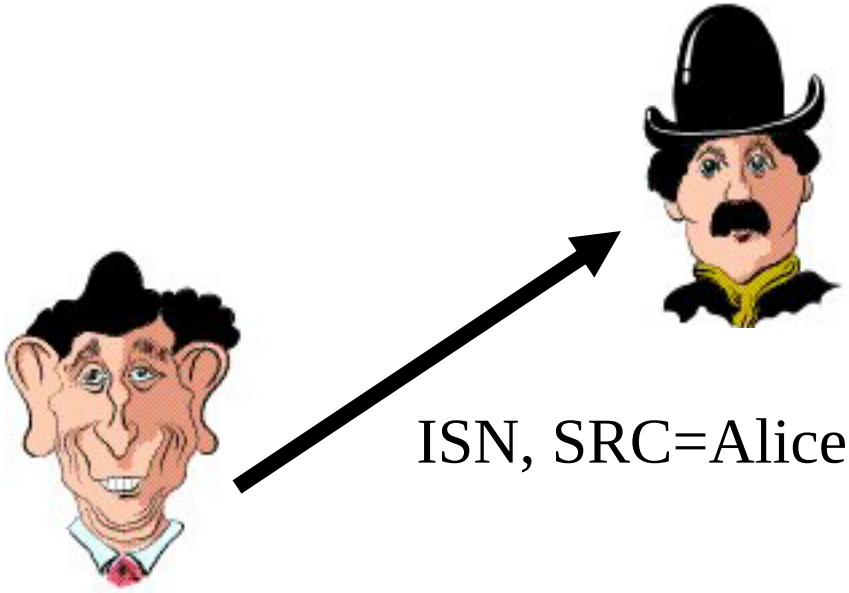
TCP Attacks

- First, Mr. Big Ears must drop all of Alice's packets since they must not be delivered to Bob (why?)



TCP Attacks

- Then, Mr. Big Ears sends his malicious packet with the next ISN (sniffed from the network)



TCP Attacks

- Why are these types of TCP attacks so dangerous?

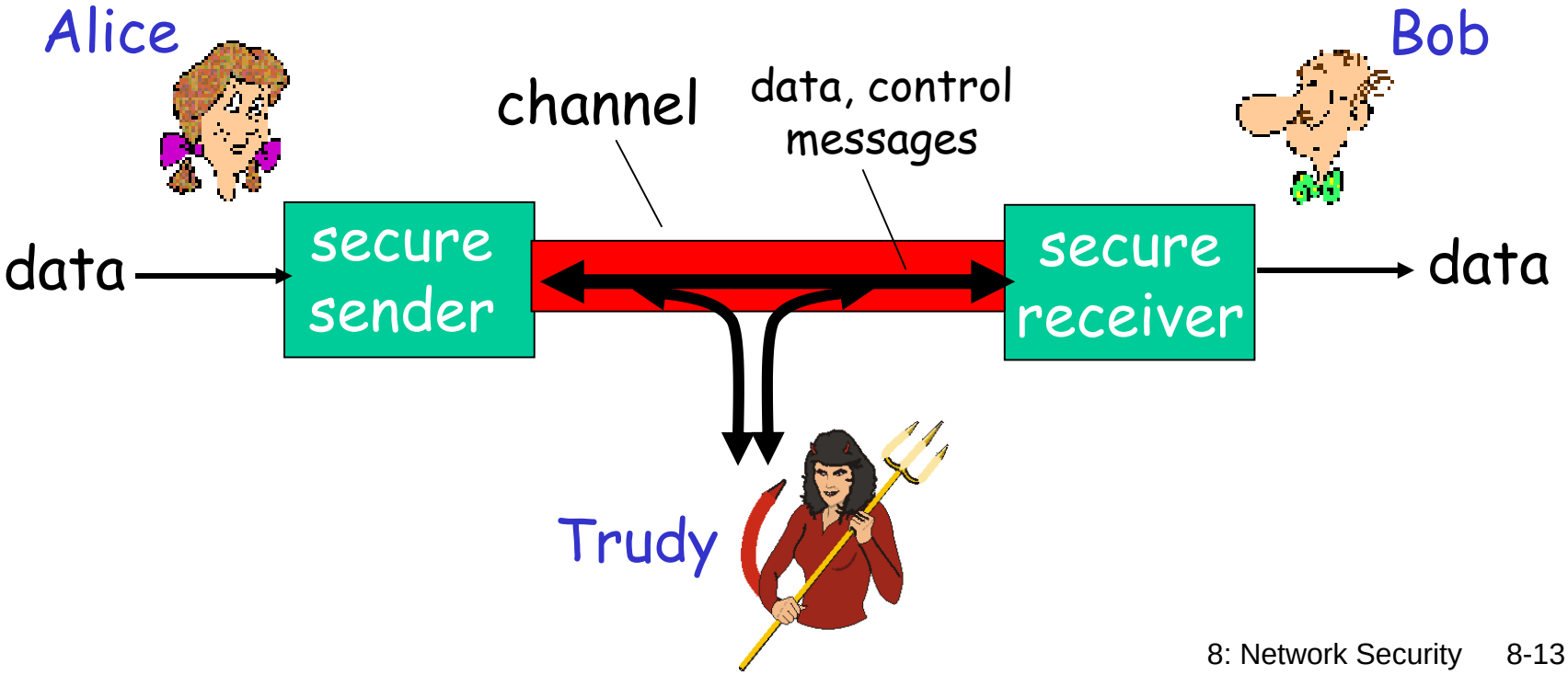
Web server

Malicious user

Trusting web client

Friends and enemies: Alice, Bob, Trudy

- well-known in network security world
- Bob, Alice (lovers!) want to communicate "securely"
- Trudy (intruder) may intercept, delete, add messages



Packet Sniffing

- Recall how Ethernet works …
- When someone wants to send a packet to some else …
- They put the bits on the wire with the destination MAC address …
- And remember that other hosts are listening on the wire to detect for collisions …
- It couldn't get any easier to figure out what data is being transmitted over the network!

Social Problems (Engineering)

- People can be just as dangerous as unprotected computer systems
  - People can be lied to, manipulated, bribed, threatened, harmed, tortured, etc. to give up valuable information
  - Most humans will breakdown once they are at the "harmed" stage, unless they have been specially trained
    - Think government here…

Why network security?

Confidentiality: only sender, intended receiver should "understand" message contents

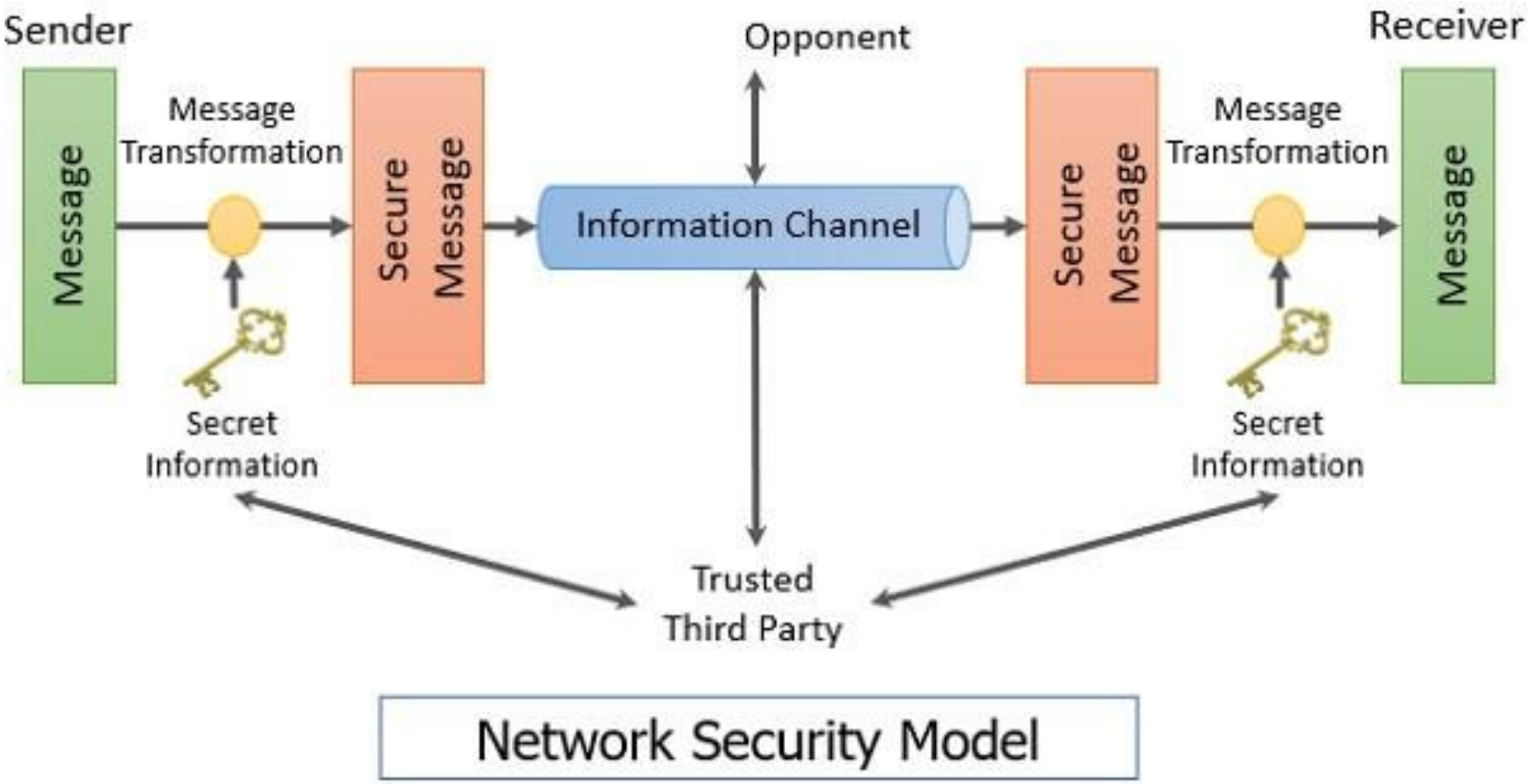
- sender encrypts message
- receiver decrypts message

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

A Network Security Model exhibits how the security service has been designed over the network to prevent the opponent from causing a threat to the confidentiality or authenticity of the information that is being transmitted through the network. For a message to be sent or receive there must be a sender and a receiver.



Network Security Model

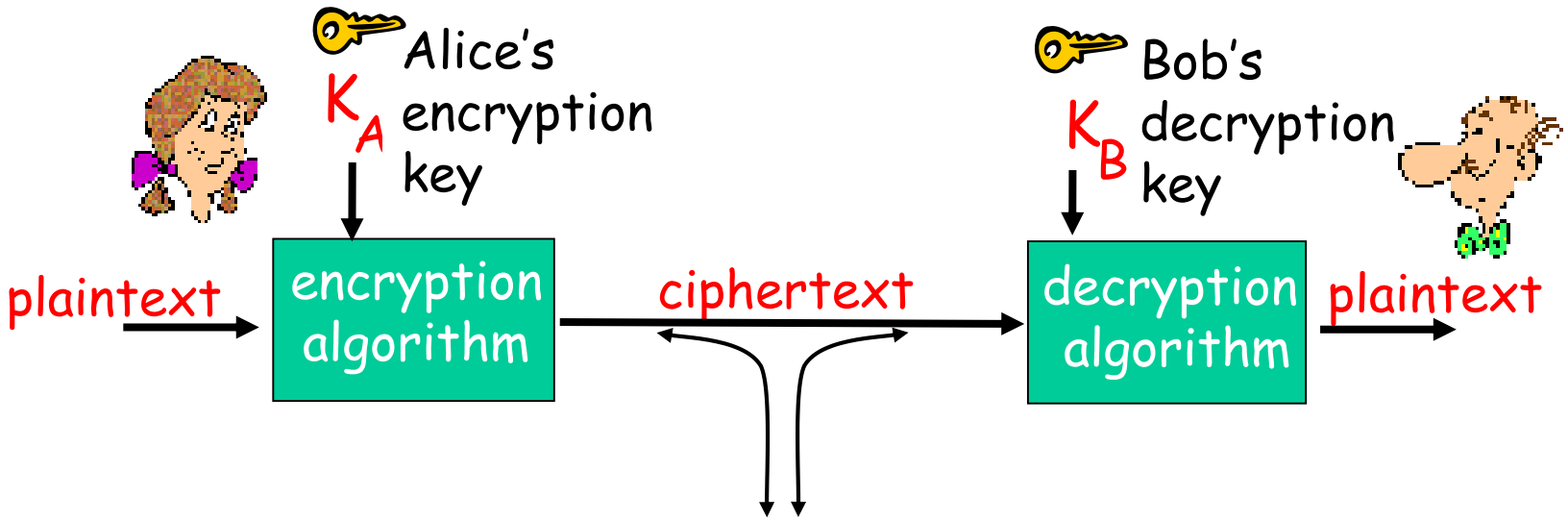
Model for Network Access Security

Using this model requires us to implement:

1. Authentication
   - select appropriate gatekeeper functions to identify users
2. Authorization
   - implement security controls to ensure only authorized users access designated information or resources

Trusted computer systems may be useful to help implement this model

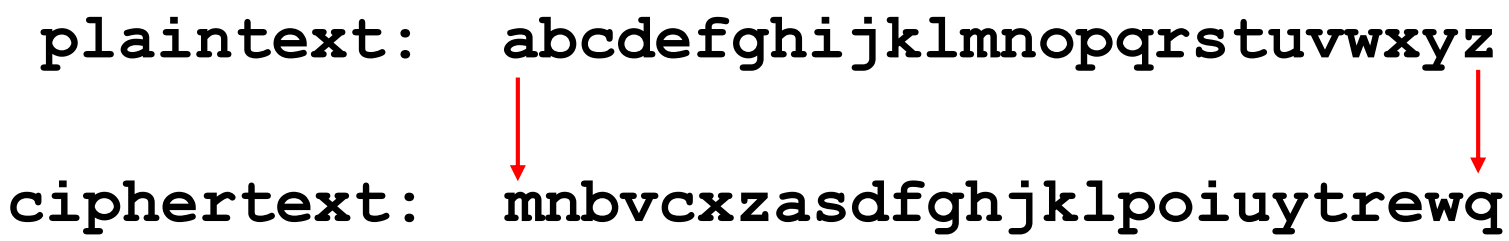
The principle of cryptography



Symmetric key cryptography

substitution cipher: substituting one thing for another

- monoalphabetic cipher: substitute one letter for another



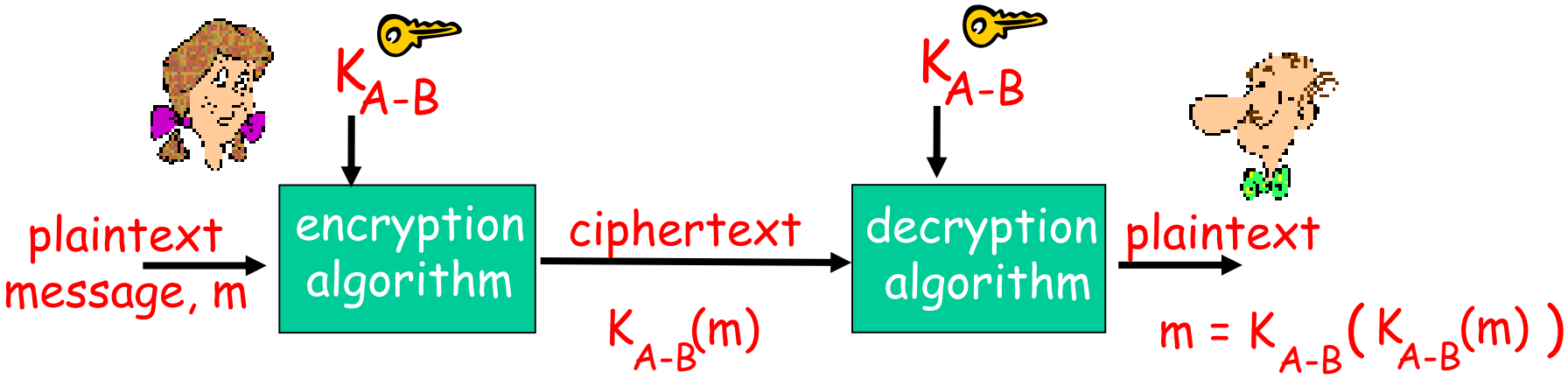
E.g.: Plaintext: bob. i love you. alice

ciphertext: nkn. s gktc wky. mgsbc

Q: How hard to break this simple cipher?:

- brute force (how hard?)

Symmetric key cryptography



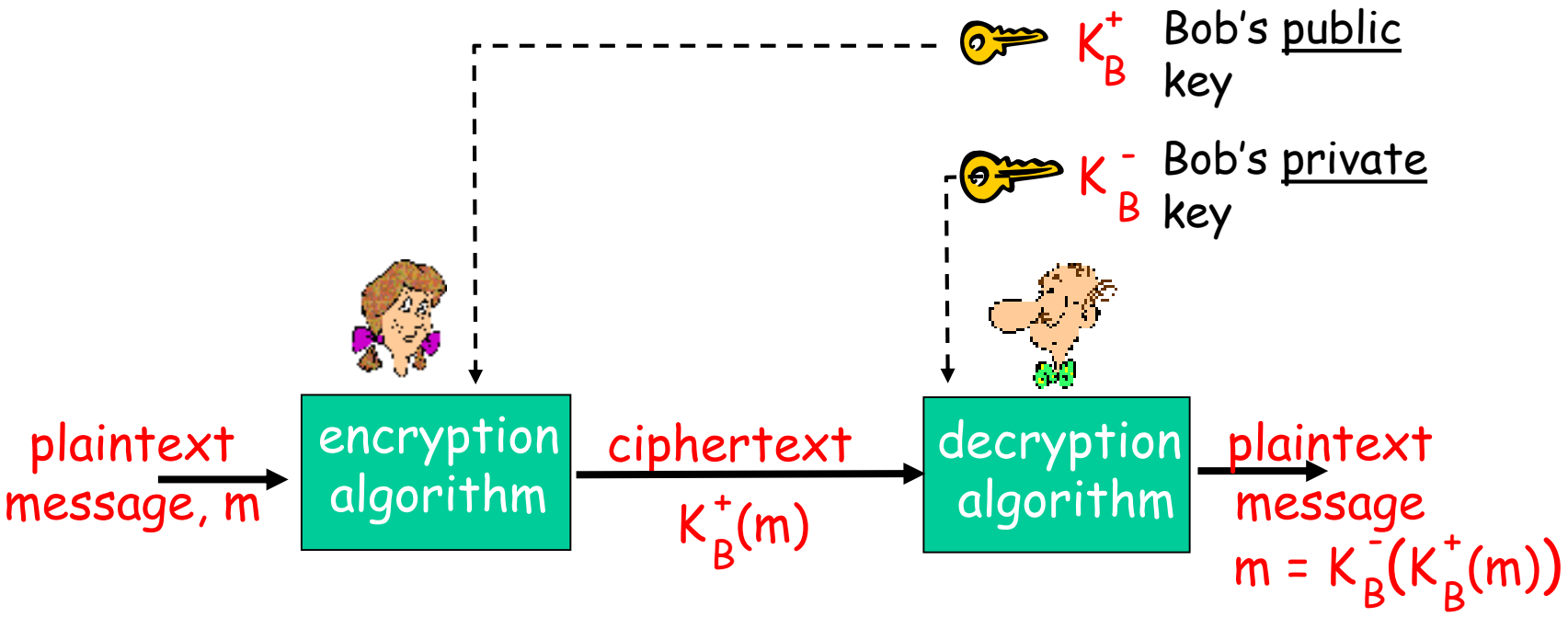
DES: Data Encryption Standard

- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64-bit plaintext input
- How secure is DES?
  - DES Challenge: 56-bit-key-encrypted phrase ("Strong cryptography makes the world a safer place") decrypted (brute force) in 4 months
  - no known "backdoor" decryption approach
- making DES more secure:
  - use three keys sequentially (3-DES) on each datum
  - use cipher-block chaining

AES: Advanced Encryption Standard

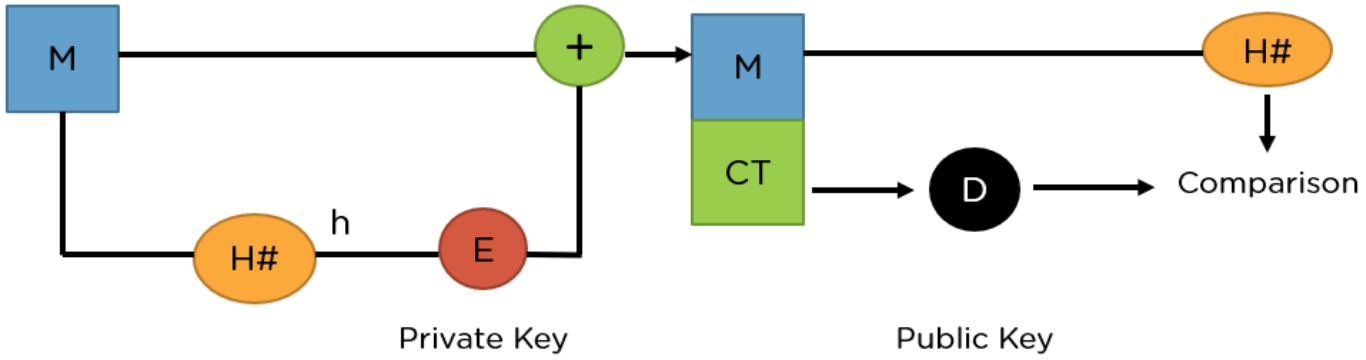
- new (Nov. 2001) symmetric-key NIST standard, replacing DES
- processes data in 128 bit blocks
- 128, 192, or 256 bit keys
- brute force decryption (try each key) taking 1 sec on DES, takes 149 trillion years for AES

Public key cryptography



What Is the RSA Algorithm?

The RSA algorithm is a public-key signature algorithm developed by Ron Rivest, Adi Shamir, and Leonard Adleman. Their paper was first published in 1977, and the algorithm uses logarithmic functions to keep the working complex enough to withstand brute force and streamlined enough to be fast post-deployment. The image below shows it verifies the digital signatures using RSA methodology.



RSA can also encrypt and decrypt general information to securely exchange data along with handling digital signature verification. The image above shows the entire procedure of the RSA algorithm. You will understand more about it in the next section.

For RSA algorithms

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 impossible to compute private key K_B^-

RSA: Rivest, Shamir, Adleman algorithm

RSA:

1. Choose two large prime numbers p, q. (e.g., 1024 bits each)
2. Compute $n = pq$, $z = (p-1)(q-1)$
3. Choose e (with $e<n$) that has no common factors with z. (e, z are "relatively prime").
4. Choose d such that $ed-1$ is exactly divisible by z. (in other words: $ed \bmod z = 1$).
5. *Public* key is $\underbrace{(n,e)}_{K_B^+}$. *Private* key is $\underbrace{(n,d)}_{K_B^-}$.

RSA: Encryption, decryption

0. Given (n,e) and (n,d) as computed above

1. To encrypt bit pattern, m, compute

   $c = m^e \bmod n$ (i.e., remainder when m^e is divided by n)

2. To decrypt received bit pattern, c, compute

   $m = c^d \bmod n$ (i.e., remainder when c^d is divided by n)

Magic happens!

$$m = \underbrace{(m^e \bmod n)}_{c}{}^d \bmod n$$

RSA example:

Bob chooses $p=5$, $q=7$. Then $n=35$, $z=24$.

$e=5$ (so e, z relatively prime).

$d=29$ (so $ed-1$ exactly divisible by z.

encrypt:

letter	m	m^e	$c = m^e \bmod n$
l	12	1524832	17

decrypt:

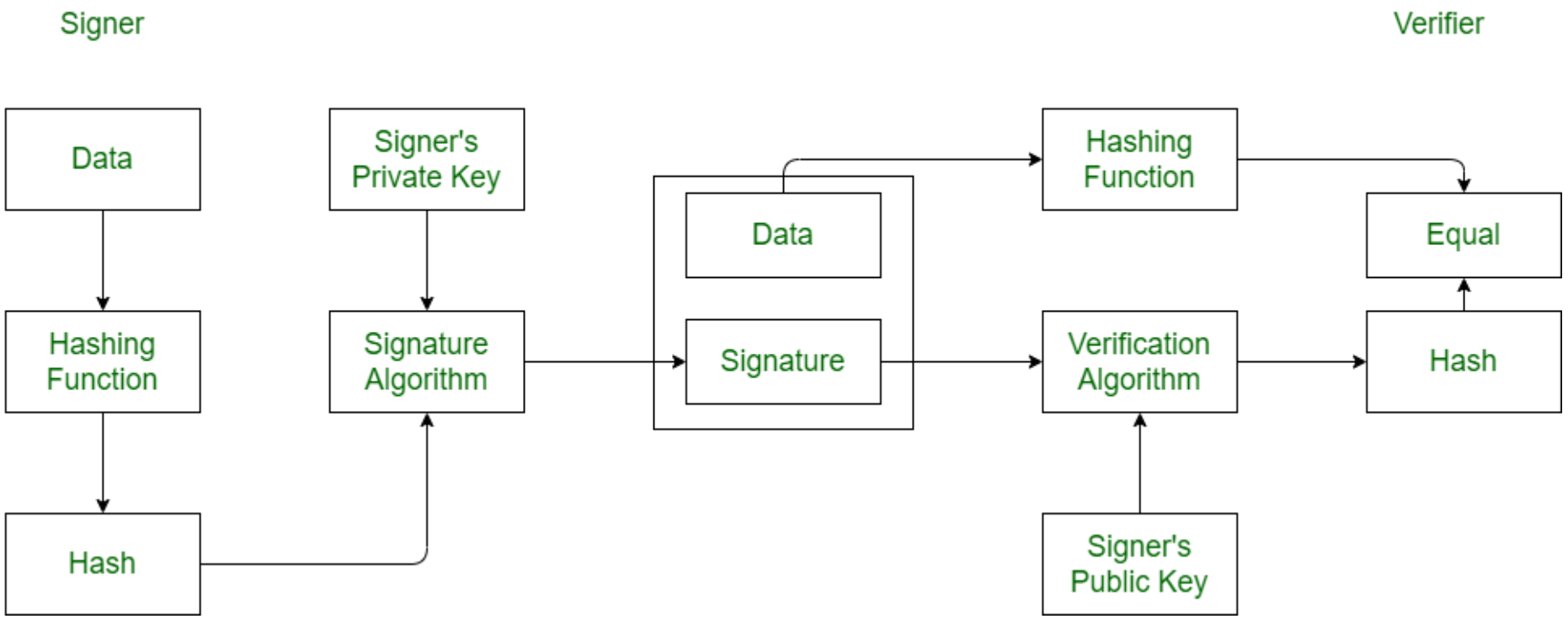
c	c^d	$m = c^d \bmod n$	letter
17	48196857210675091509141182522307169 7	12	l

Digital Signature Algorithm (DSA) :

DSA stand for **Digital Signature Algorithm**. It is used for digital signature and its verification. It is based on mathematical concept of modular exponentiation and discrete logarithm. It was developed by **National Institute of Standards and Technology (NIST)** in 1991.

It involves four operations:

1.Key Generation
2.Key Distribution
3.Signing
4.Signature Verification


Signer
Verifier
Data
Signer's Private Key
Hashing Function
Signature Algorithm
Hash
Data
Signature
Hashing Function
Verification Algorithm
Signer's Public Key
Equal
Hash

Steps in DSA Algorithm

Keeping the image above in mind, go ahead and see how the entire process works, starting from creating the key pair to verifying the signature at the end.

1. Key Generation

There are two steps in the key generation process: parameter generation and per-user keys.

Parameter Generation

•Initially a user needs to choose a cryptographic hash function (H) along with output length in bits |H|. Modulus length N is used in when output length |H| is greater.

•Then choose a key length L where it should be multiple of 64 and lie in between 512 and 1024 as per Original DSS length. However, lengths 2048 or 3072 are recommended by NIST for lifetime key security.

•The values of L and N need to be chosen in between (1024, 60), (2048, 224), (2048, 256), or (3072, 256) according to FIPS 186-4. Also, a user should chose modulus length N in such a way that modulus length N should be less than key length (N<L) and less than and equal to output length (N<=|H|).

•Later a user can choose a prime number q of N bit and another prime number as p of L bit in such a way that p-1 is multiple of q. And then choose h as an integer from the list (2……..p-2).

•Once you get p and q values, find out

g = h^(p-1)/q*mod(p). If you get g = 1, please try another value for h and compute again for g except 1.

p, q and g are the algorithm parameters that are shared amongst different users of the systems.

Per-user Keys

To compute the key parameters for a single user, first choose an integer x (private key) from the list (1…….q-1), then compute the public key, y=g^(x)*mod(p).

2. Signature Generation

•It passes the original message (M) through the hash function (H#) to get our hash digest(h).

•It passes the digest as input to a signing function, whose purpose is to give two variables as output, s, and r.

•Apart from the digest, you also use a random integer k such that 0 < k < q.

•To calculate the value of r, you use the formula r = (gk mod p) mod q.

•To calculate the value of s, you use the formula s = [K-1(h+x . R)mod q].

•It then packages the signature as {r,s}.

•The entire bundle of the message and signature {M,r,s} are sent to the receiver.

3. Key Distribution

While distributing keys, a signer should keep the private key (x) secret and publish the public key (y) and send the public key (y) to the receiver without any secret mechanism.

Signing

Signing of message m should be done as follows:

•first choose an integer k from (1……q-1)

•compute

r = g^(k)*mod(p)*mod(q). If you get r = 0, please try another random value of k and compute again for r except 0.

•Calculate

s=(k^(-1)*(H(m)+xr))*mod(q). If you get s = 0, please try another random value of k and compute again for s except 0.

•The signature is defined by two key elements (r,s). Also, key elements k and r are used to create a new message. Nevertheless, computing r with modular exponential process is a very expensive process and computed before the message is known. Computation is done with the help of the Euclidean algorithm and Fermat's little theorem.

4. Signature Verification

•You use the same hash function (H#) to generate the digest h.

•You then pass this digest off to the verification function, which needs other variables as parameters too.

•Compute the value of w such that: s*w mod q = 1

•Calculate the value of u1 from the formula, u1 = h*w mod q

•Calculate the value of u2 from the formula, u2 = r*w mod q

•The final verification component v is calculated as v = [((gu1 . yu2) mod p) mod q].

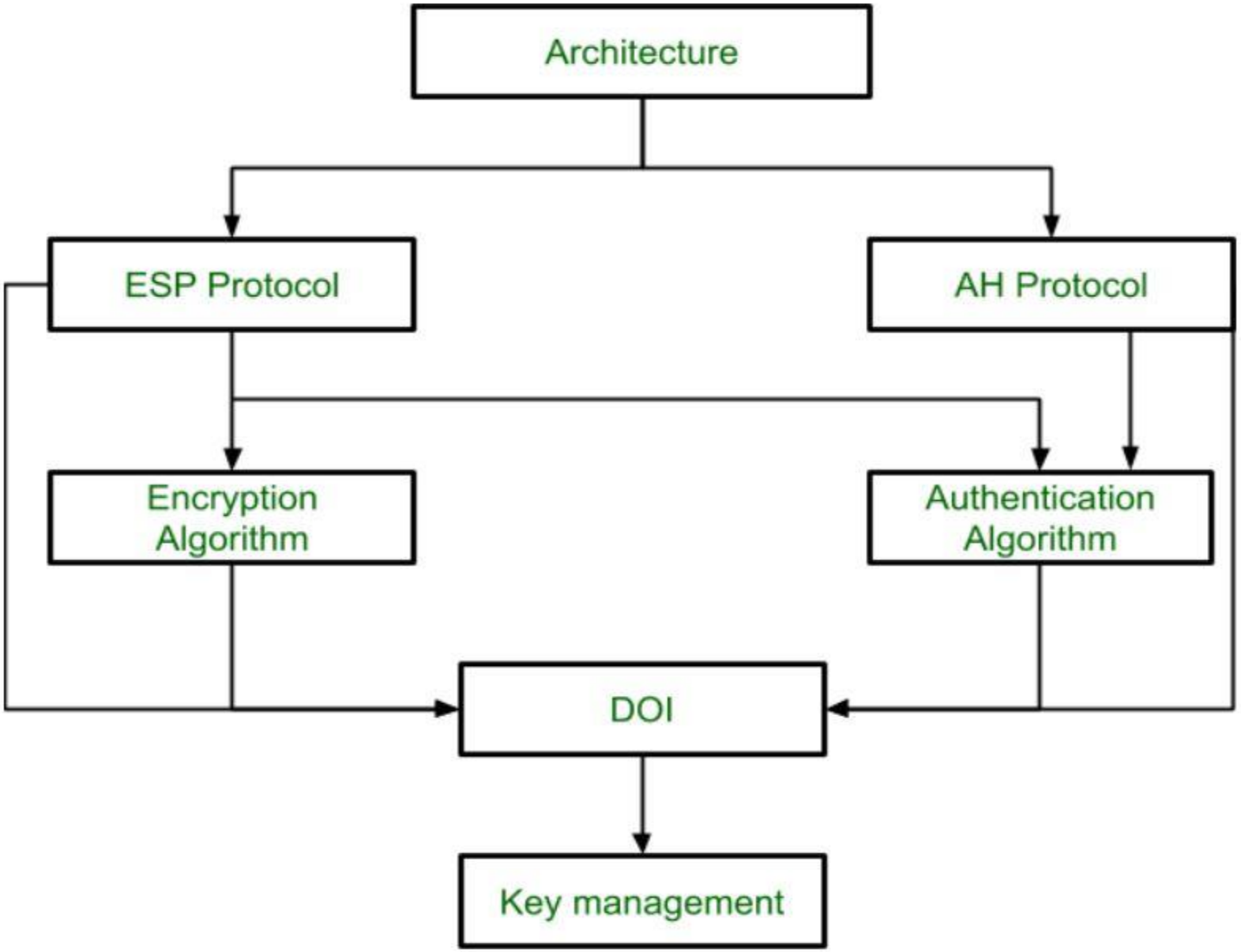
•It compares the value of v to the value of r received in the bundle.

•If it matches, the signature verification is complete.

IPSec (IP Security) architecture uses two protocols to secure the traffic or data flow. These protocols are ESP (Encapsulation Security Payload) and AH (Authentication Header).

IPSec Architecture includes protocols, algorithms, DOI, and Key Management. All these components are very important in order to provide the three main services:

- Confidentiality
- Authentication
- Integrity


Architecture
ESP Protocol
AH Protocol
Encryption Algorithm
Authentication Algorithm
DOI
Key management

IPsec: Security

- network-layer secrecy:
  - sending host encrypts the data in IP datagram
  - TCP and UDP segments; ICMP and SNMP messages.
- network-layer authentication
  - destination host can authenticate source IP address
- two principal protocols:
  - authentication header (AH) protocol
  - encapsulation security payload (ESP) protocol
- for both AH and ESP, source, destination handshake:
  - create network-layer logical channel called a security association (SA)
- each SA unidirectional.
- uniquely determined by:
  - security protocol (AH or ESP)
  - source IP address
  - 32-bit connection ID
  - **DOI (Domain of Interpretation):** DOI is the identifier that supports both AH and ESP protocols. It contains values needed for documentation related to each other.

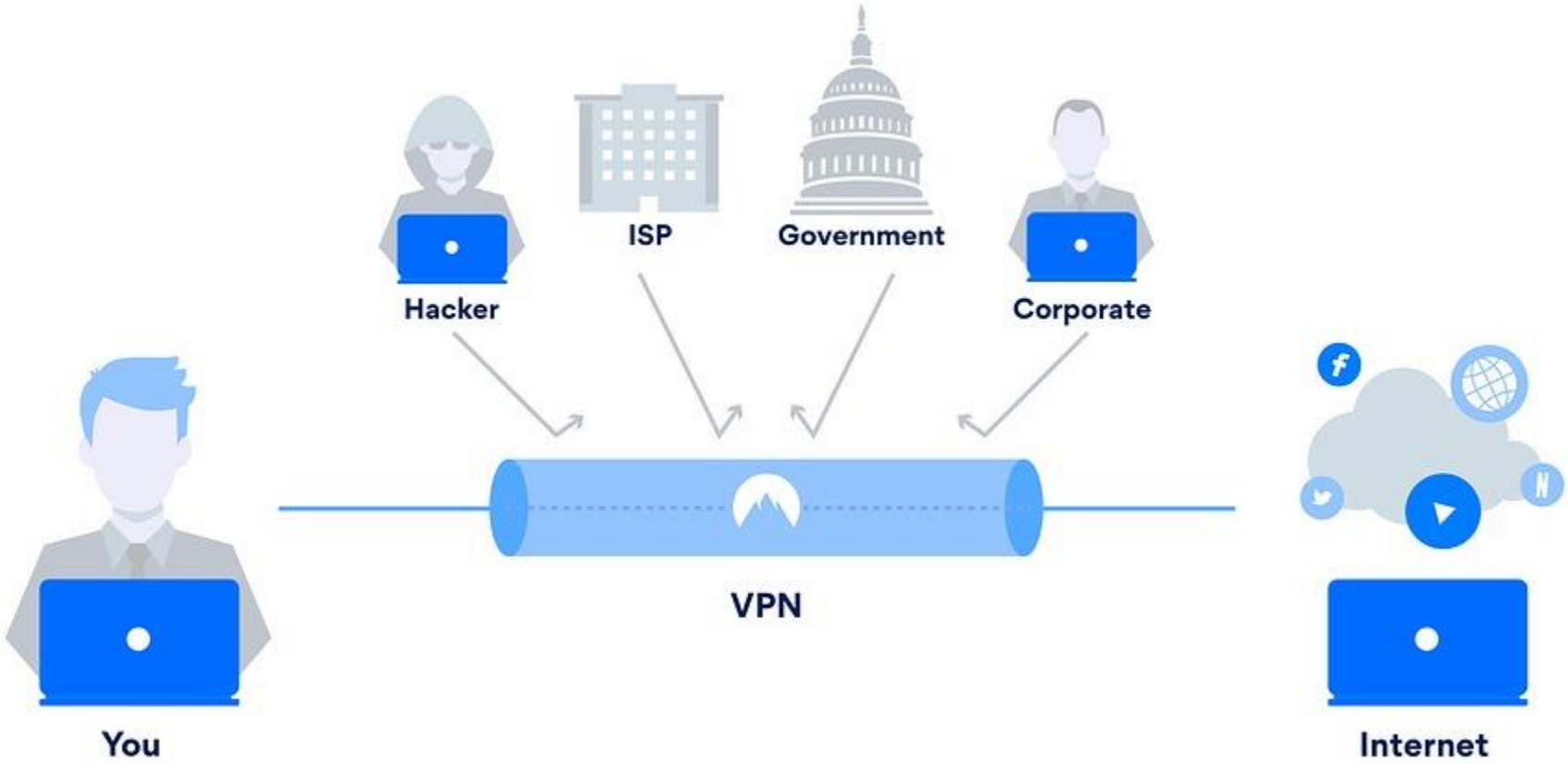
A virtual private network (VPN) is an Internet security service that allows users to access the Internet as though they were connected to a private network.

VPNs use encryption to create a secure connection over unsecured Internet infrastructure.

VPNs are one way to protect corporate data and manage user access to that data.

VPNs protect data as users interact with apps and web properties over the Internet, and they can keep certain resources hidden.

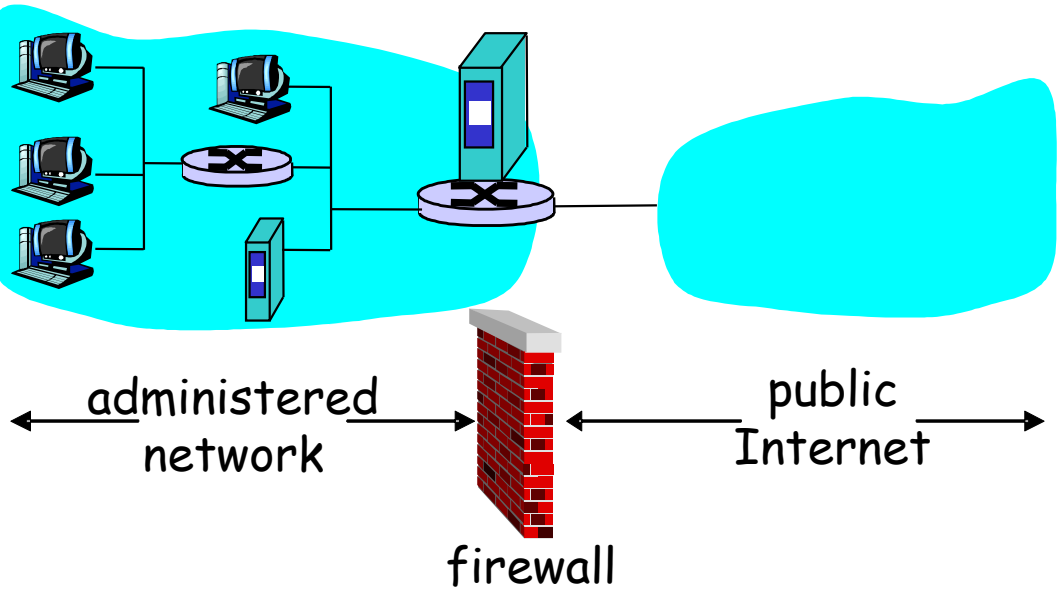
They are commonly used for access control — however, other identity and access management (IAM) solutions can also help with managing user access.


Hacker
ISP
Government
Corporate
You
VPN
Internet

Firewalls

firewall

isolates organization's internal net 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:

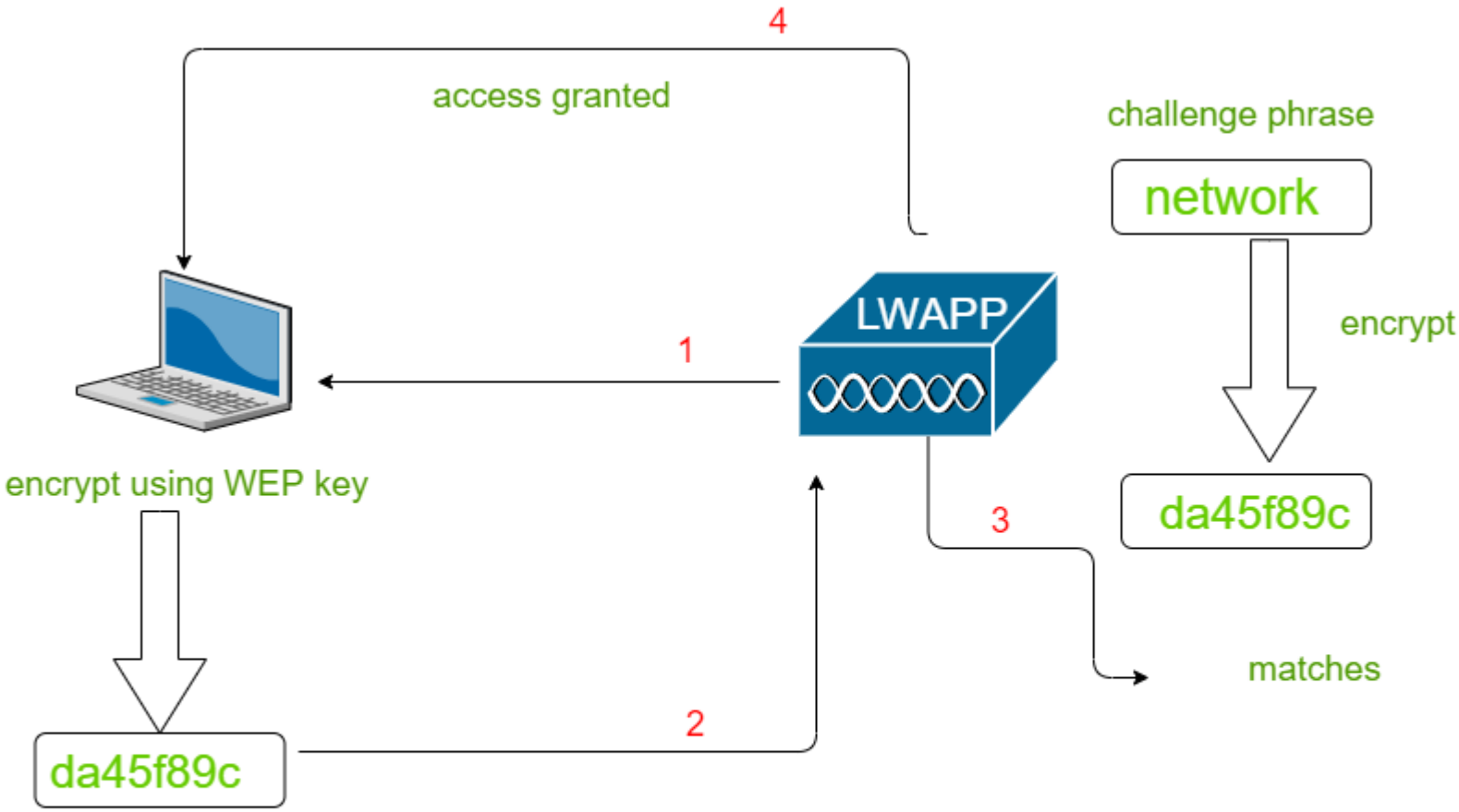
- stateless packet filters
- stateful packet filters
- application gateways

Wireless security: WEP, WPA, WPA2 and WPA3

WEP, WPA, WPA2 and WPA3: Which is best?

When choosing from among WEP, WPA, WPA2 and WPA3 wireless security protocols, experts agree WPA3 is best for Wi-Fi security. As the most up-to-date wireless encryption protocol, WPA3 is the most secure choice. Some wireless APs do not support WPA3, however. In that case, the next best option is WPA2, which is widely deployed in the enterprise space today.

At this point, no one should use the original wireless security protocol, WEP, or even its immediate successor, WPA, as both are outdated and make wireless networks extremely vulnerable to outside threats. Network administrators should replace any wireless AP or router that supports WEP or WPA with a newer device that's compatible with WPA2 or WPA3



Working of WEP Authentication