Chapter Four

Review of Shared Key Cryptography and Hash Functions

What is Cryptography?

- The goal of the cryptography is to protect private communication in the public world. The assumption is that two entities wanting to communicate - Alice and Bob - are shouting their messages in a room full of people. Everyone can hear what they are saying.
- The goal of cryptography is to protect this communication so that only Alice and Bob can *understand* the content of the messages.
- Beyond confidentiality - ensuring the secrecy of communication - cryptography is used for many other purposes, such as:

Continued...

- **Authentication** - Alice can *sign* her message and Bob can verify that she sent it based on this signature
- **Integrity checking** - Alice can generate a *checksum* of the message. Bob can either extract it from the message or recalculate it and verify that the message has not been changed.
- **Non-repudiation** - if Alice signs the message she cannot deny later that she sent it, because no one else could generate that same signature.
- Exchanging a secret with someone you have never met before, in a room full of people
- Proving to someone you know a secret without giving it away
- Sending secret messages to any m out of n people so that only those m can retrieve them and the rest cannot
- Sending secret messages to a group of n people, that can be retrieved only if m people work together

Continued...

- How do we encrypt messages? Alice could use knowledge of some common secret that only she and Bob know, like "meet at that place where we first had lunch together". This does not work for arbitrary messages, nor for the case where Alice and Bob never communicated before.
- Alice could use a secret protocol to encrypt (transform plaintext into ciphertext) and reverse it to decrypt (transform ciphertext into plaintext). This is a bad idea for several reasons:
- It is hard to come up with a very secure cipher scheme (encryption and decryption protocol). Simple schemes (such as replacing letters with other letters and shuffling them around) are easily broken.
- Alice has to somehow communicate this new protocol to Bob and Bob has to implement it correctly. This does not scale and requires a human to implement each new protocol. Consequently protocols would be simple and thus not secure.

Continued...

- Alice can use a public protocol that has been well designed, well implemented and analyzed by many people for security. She can use a secret key that she communicates to Bob - a key is much easier to communicate than a whole new protocol. This is how today's crypto schemes work.

Types of Cryptographic Functions

- There are three main types of cryptographic functions that are the building blocks of security:

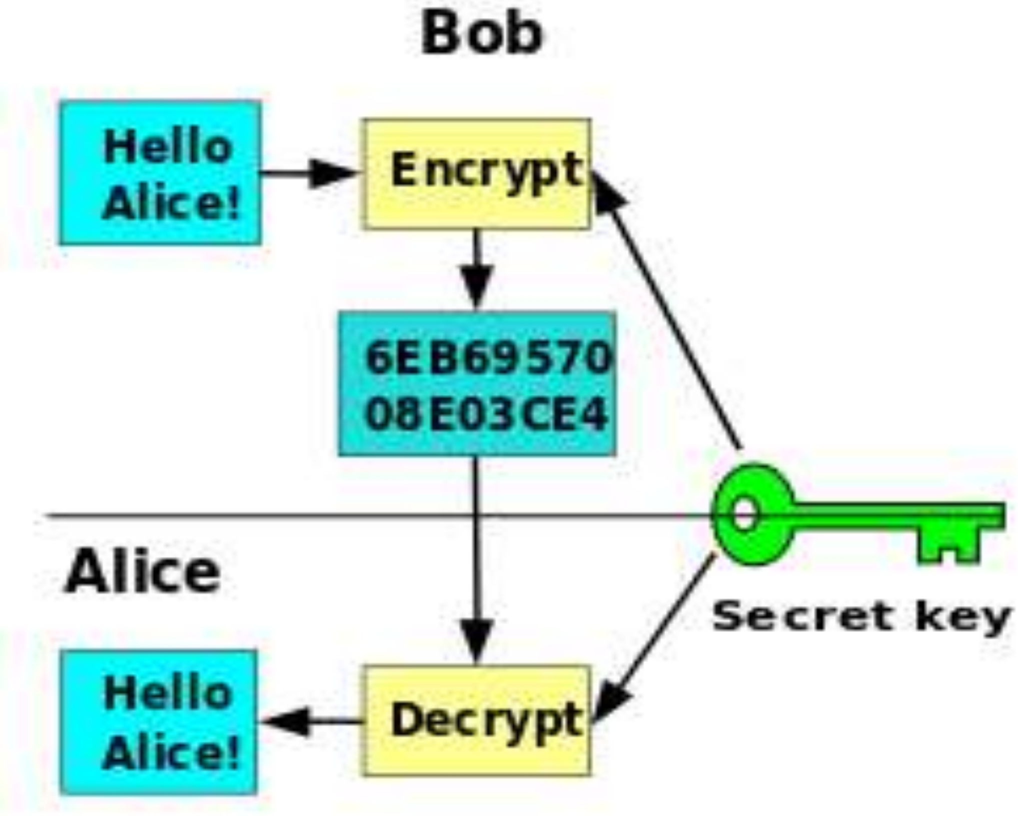
1. ***Symmetric cryptography*** - Alice and Bob know the same key and use it for encryption and decryption.

- Symmetric crypto can be used to ensure secrecy - Alice and Bob exchange the secret key and use it to communicate privately.

Continued…

- It can also be used for secure storage - Alice encrypts the files she stores in the cloud. If the cloud is compromised no one can read her files.
- Symmetric crypto can also be used for authentication, aka proving that you know a secret without revealing it. Alice and Bob want to prove to each other they know the same secret.
- Alice chooses a random number and encrypts it with the key. Bob decrypts it and sends it back. Then Bob chooses a secret number and encrypts it and Alice decrypts it and sends it back. Why do we have to do this twice? Who is assured of what in each round?

Continued…


Bob
Hello Alice!
Encrypt
6EB69570 08E03CE4
Alice
Secret key
Hello Alice!
Decrypt

Continued…

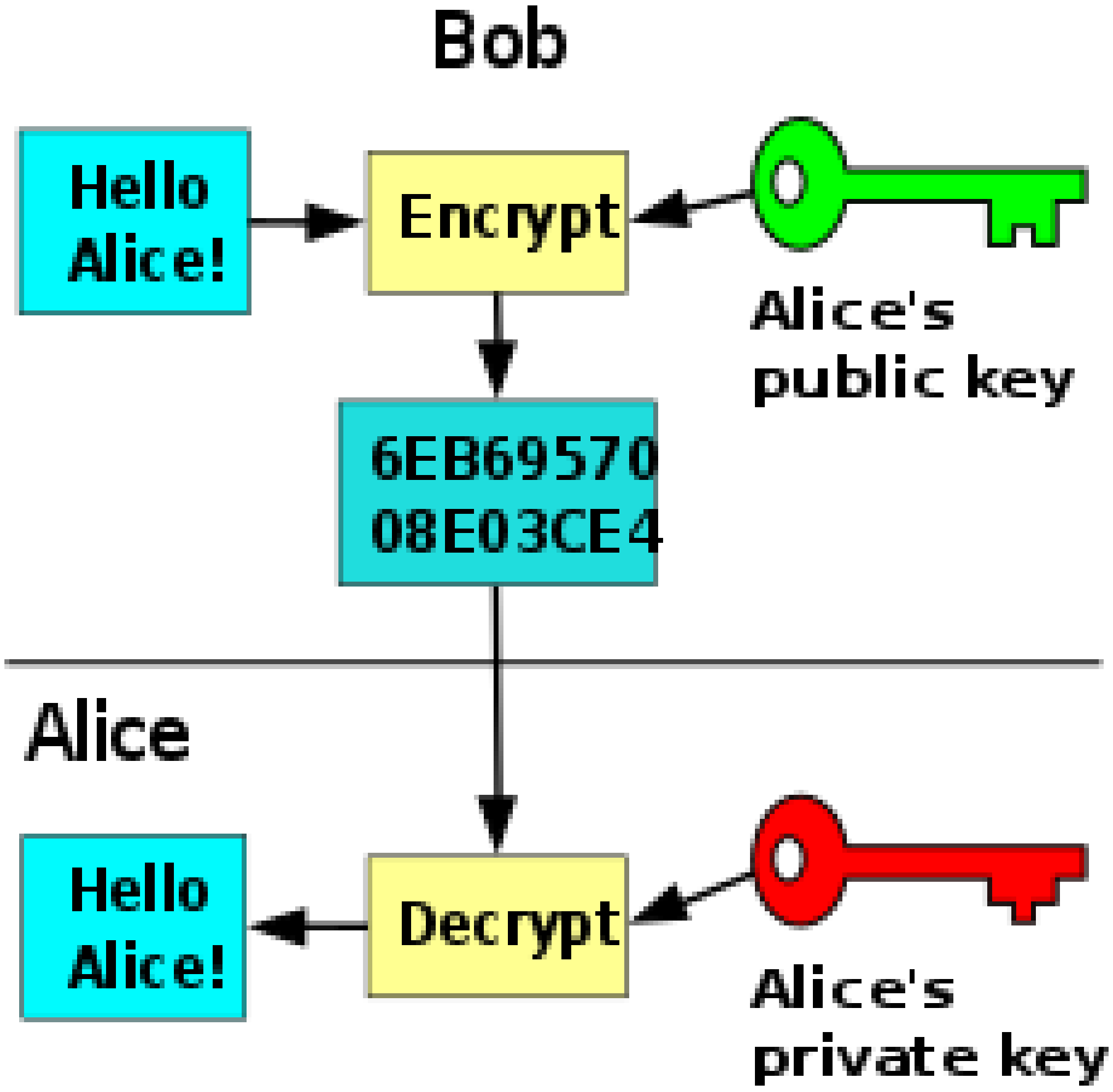
2. ***Asymmetric cryptography*** - Alice has a pair of keys - a private key known only to her and a public key that anyone can find out. She can encrypt with one key from a pair (any one) and decrypt with another. Asymmetric crypto can do whatever symmetric crypto can but much slower (about 1,500 times slower). However it can also provide some extra functionality.

- In symmetric crypto secret communication between Alice and Bob is only possible if they know the same secret key. But how do they find out this key?
- They can use asymmetric crypto to exchange it. Bob could advertise his public key on his Web page. Alice could use it to encrypt a secret key and send it to Bob - only he will be able to retrieve it.

Continued…

- If Alice wants to authenticate Bob she encrypts a message with his public key and he decrypts it and sends it back. Alice can do all this without storing any secret information.
- If Alice orders something from Bob and signs every message by encrypting it with her private key, anyone can verify her signature and no one can forge it.
- This ensures ***non-repudiation*** - Alice cannot deny she sent the message.

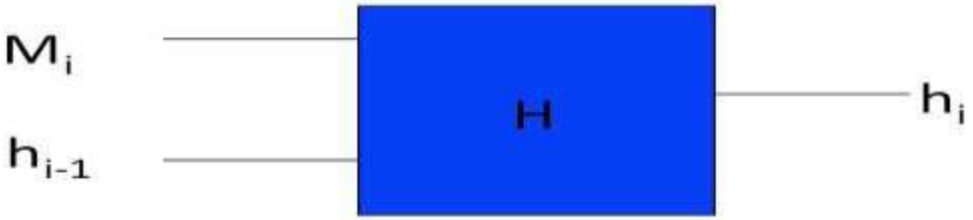
Continued...


Bob
Hello Alice!
Encrypt
Alice's public key
6EB69570 08E03CE4
Alice
Hello Alice!
Decrypt
Alice's private key

Continued...

3. ***Hash functions*** - these are publicly known functions that reduce a large message to a fixed-size hash, applying a non-linear transformation (aka one-way hashes or message digests).

- Hash functions are useful to prove message integrity. One can hash the message and display its hash somewhere publicly.
- Hashing a large message requires the hash algorithm to break it into chunks, and combine each chunk with the hash of the previous chunks to generate new hash. This process is shown in the picture below.



Continued…

Attacks on Cryptography

- We now briefly mention some attacks one could launch on a cryptographic protocol with the goal to learn the decryption key or in some cases the plaintext that was input to the protocol. We assume that Alice and Bob talk using the cryptographic protocol. Eve is the enemy that can passively observe encrypted messages but cannot change them, while Mallory is the enemy that sits on the path of the messages and can modify them at will.

Ciphertext-only attack

- In this attack intruders observes cipher texts and uses them to guess plaintext and the decryption key. Such attack e.g., would be possible on *mono alphabetic* cipher that replaces each letter in the alphabet with another letter, according to a map. Because the frequency of the letters does not change with this cipher Eve can use *frequency analysis* to break it.

Continued...

Known-plaintext attack

- In this attack intruders knows some portions of the plaintext. How does she know this? Perhaps messages are exchanged using a known protocol that has predictable fields such as timestamp, a keyword chosen from a small set, etc. This way intruders can collect plaintext-cipher text pairs and use them to obtain the key if the cipher is simple.

Chosen-plaintext attack

- In this attack intruders can inject some messages to be encrypted and observe the output.

Man-in-the-middle attack

- In this attack intruders sits on the path of the messages and can do whatever intruders likes with them. intruders can modify them, inject new messages, drop the existing ones or replay messages. MITM attacks are notoriously hard to detect and defend from.

Continued...

Brute-force attack

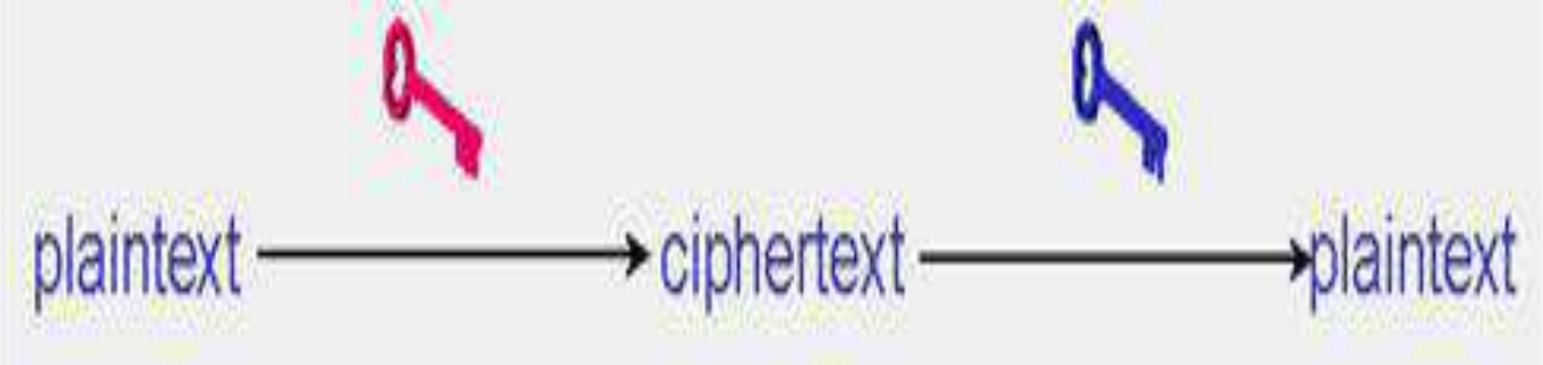
- In this attack intruders will simply try all possible keys until intruders finds the one that works. Again, there is an assumption that intruders can tell which one has worked because decryption of a cipher text will result in something intelligible in a chosen language. This attack can be made arbitrarily hard if the key is long. What if the decryption key depend on a password in some known way, how could Eve use this to break the encryption?

Overview - Public Key Cryptography

- Modern PKC was first described publicly by Stanford University professor Martin Hellman and graduate student Whitfield Diffie in 1976.

- Generic PKC employs two keys that are mathematically related although knowledge of one key does not allow someone to easily determine the other key. One key is used to encrypt the plaintext and the other key is used to decrypt the ciphertext.
- The important point here is that it does not matter which key is applied first, but that both keys are required for the process to work. Because a pair of keys are required, this approach is also called asymmetric cryptography.
- In PKC, one of keys designated the public key and may be advertised as widely as owner wants. The other key is designated the private key and is never revealed to another party.
- Suppose Alice wants to send Bob a message. Alice encrypts some information using Bob's public key; Bob decrypts the ciphertext using his private key. This method could be also used to prove who sent a message;

Continued…



Public key (asymmetric) cryptography. PKC uses two keys, one for encryption and the other for decryption.

Continued…

- Alice, for example, could encrypt some plaintext with her private key; when Bob decrypts using Alice's public key, he knows that Alice sent the message (authentication) and Alice cannot deny having sent the message (non-repudiation).

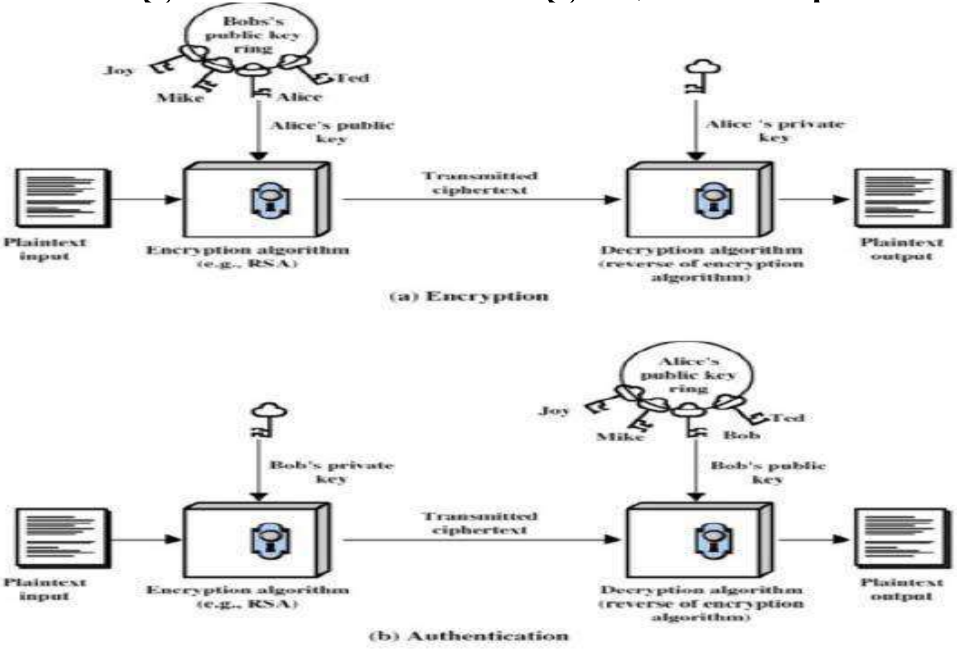


Figure 8.1: Public Key Cryptography.

Continued...

Figure illustrates the P-K process. The steps are:

1. Each system generates a pair of keys.

2. Each system publishes its encryption key (public key) keeping its companion key private.

3. If A wishes to send a message to B it encrypts the message using B's public key.

- When B receives the message, it decrypts the message using its private key. No one else can decrypt the message because only B knows its private key.

Continued…

Four types of Public Key Cryptography:

1. Diffie Hellam Algorithm (DH)

2. Rivest Shamir Adleman Algorithm (RSA)

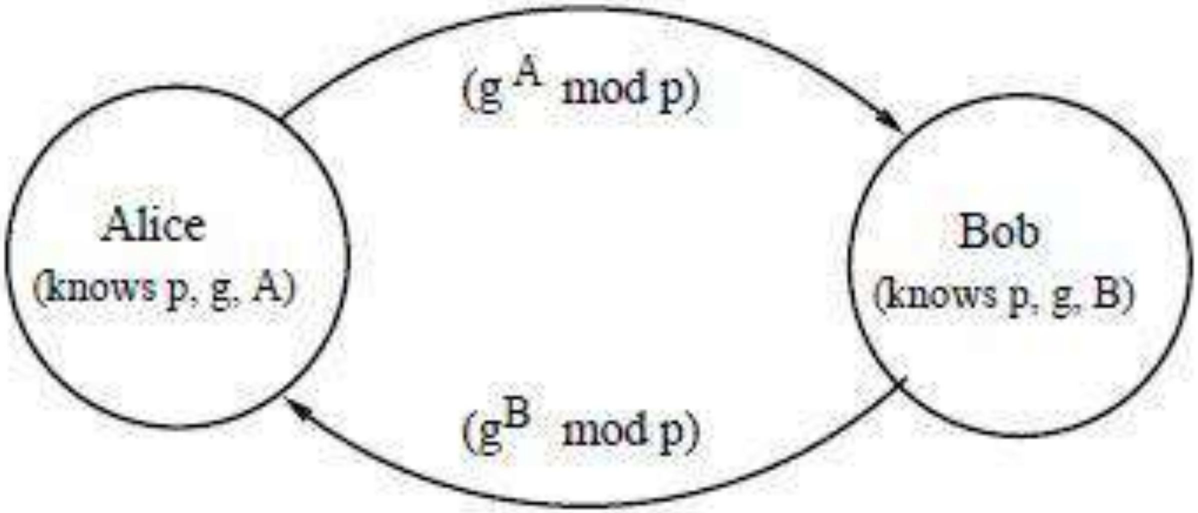
3. Certificate Authority (CA)

4. Public Key Infrastructure (PKI)

1. Diffie-Hellman Algorithm

- The Diffie-Hellman key agreement protocol (1976) was the first practical method for establishing a shared secret over an unsecured communication channel.
- The point is to agree on a key that two parties can use for a symmetric encryption, in such a way that an eavesdropper cannot obtain the key.

Continued…



Continued…

Steps in the algorithm:

1. Alice and Bob agree on a prime number p and a base g.

2. Alice chooses a secret number a, and sends Bob (g^a mod p).

3. Bob chooses a secret number b, and sends Alice (g^b mod p).

4. Alice computes ((g^b mod p)^a mod p).

5. Bob computes ((g^a mod p)^b mod p).

Both Alice and Bob can use this number as their key. Notice that p and g need not be protected.

Continued...

Diffie Hellman Example:

1. Alice and Bob agree on p = 23 and g = 5.

2. Alice chooses a = 6 and sends 5^6 mod 23 = 8.

To Prove:

5^6 mod 23 = 15625 % 23

= 679.347826087 – 679 =

= 0.347826087 x 23

= 8

Continued...

3. Bob chooses b = 15 and sends 5^{15} mod 23 = 19.

To Prove:

5^{15} mod 23 = 5 x 5^{14} % 23

= 5 x $(5^{2})^{7}$ % 23

= 5 x $(25)^{7}$ % 23

= 5 x (25 % 23) 7 % 23

= 5 x $(2)^{7}$ % 23

= 5 x (128 % 23) % 23

= 5 x 13 % 23

= 65 % 23

= 19

Continued...

4. Alice computes 19^6 mod 23 = 2.

To Prove:

19^6 mod 23 = 47045881 % 23

= 47045881 / 23

= 2045473.0869565 - 2045473

= 0. 0869565217 x 23

= 1.9999 = 2

Continued…

5. Bob computes 8^{15} mod 23 = 2.

To Prove:

8^{15} mod 23 = 8 x 8^{14} % 23

= 8 x $(8^2)^{7}$ % 23

= 8 x $(64)^{7}$ % 23

= 8 x $(64 \% 23)^{7}$ % 23

= 8 x $(18)^{7}$ % 23

= 8 x (612220032 % 23) % 23

= 8 x 6 % 23 = 48 % 23 = 2

- Then 2 is the shared secret.
- Clearly, much larger values of a, b, and p are required. An eavesdropper cannot discover this value even if she knows p and g and can obtain each of the messages.

Continued…

RSA Algorithm

- The RSA algorithm was developed by Ron Rivest, Adi Shamir and Len Adleman at MIT in 1978. Since this time it has reigned supreme as the most widely accepted and implemented general-purpose approach to public-key encryption.
- The RSA scheme is a block cipher in which the plaintext and cipher text are integers between 0 and n – 1 for some n. The scheme makes use of an expression with exponentials.
- Plaintext is encrypted in blocks having a binary value less than some number n.
- For some plaintext block M and ciphertext block C we have:

Continued...

RSA Algorithm

Overview of Key Generation

1. Generate two large prime numbers, p and q.

2. Let n = pq.

3. Let Φ (n) = (p-1) (q-1).

4. Choose a small number e, ie relatively prime to the quotient and is 1 < e < Φ (n) and gcd(e, Φ (n))=1, where e will be part of private key.

5. Find d, we will calculate by 2 methods

a) Eucliden Algorithm d = e-1 mod Φ (n)

b) Conherent equation method d = 1+ k(Φ (n))/e

Continued...

- **Encryption**

6. $C = M^e \bmod n$

- **Decryption**
- 7. $M = C^d \bmod n$

1. Generate two large prime numbers, p and q

- To make the example easy to follow, I am going to use small numbers, but this is not secure. Lets have:
- $p = 7$
- $q = 19$

2. Let n = p q

- $n = p \times q$
- $= 7 \times 19$

$= 133$

Continued...

3. Let Φ (n) = (p-1) (q-1)

= 6 x 18

= 108

4. Choose a small number, e coprime to m.

- e coprime to m, means that the largest number that can exactly divide both e and m (their greatest common divisor, or GCD) is 1.
- e = 2 GCD (2,108) = 2 (No !)
- e = 3 GCD (3,108) = 3 (Yes !)
- e = 4 GCD (4,108) = 4 (Yes !)
- e = 5 GCD (5,108) = 1 (Yes !)
- Let have e= 5

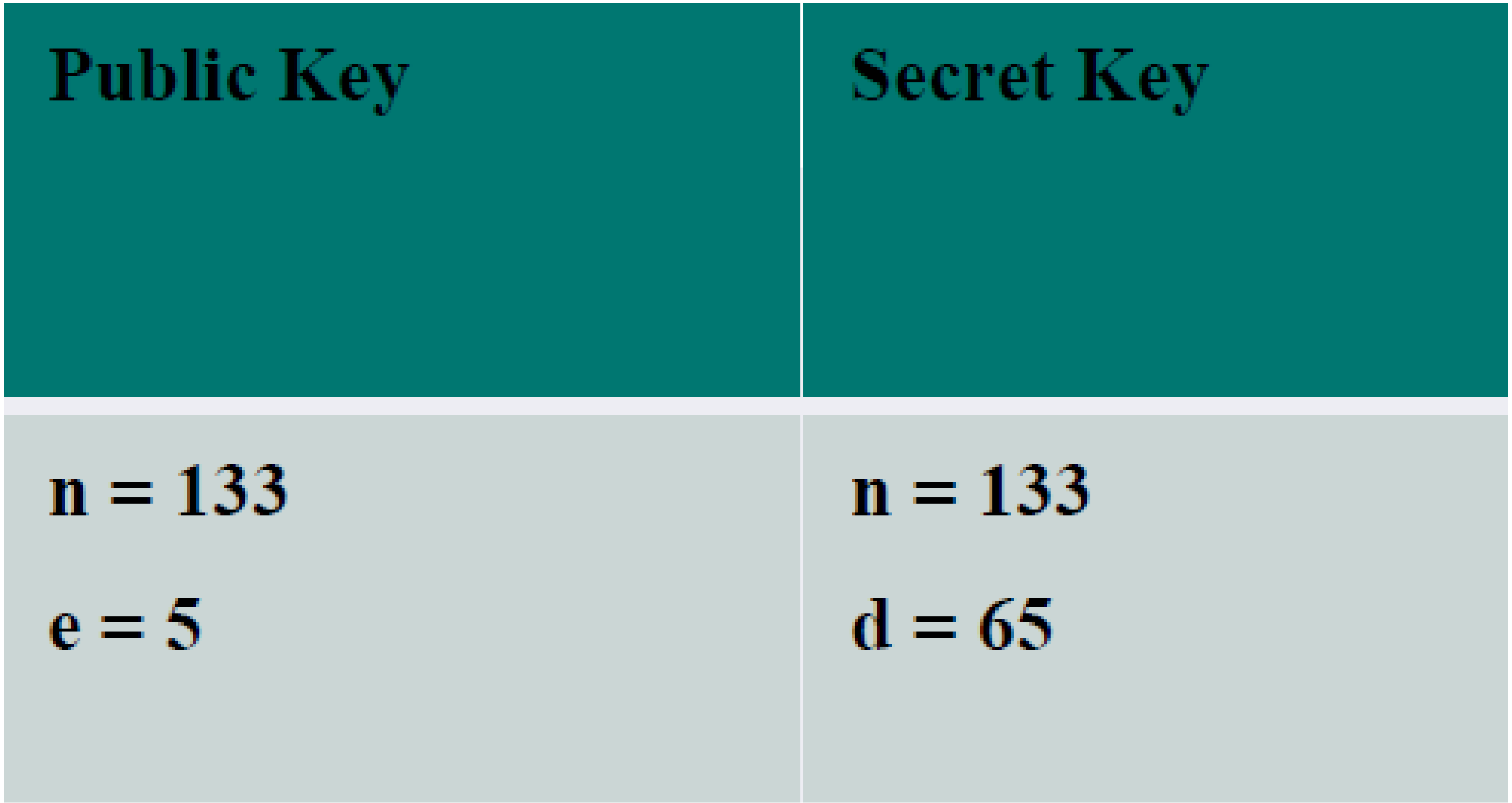
Continued...

5. **Find d,** Conherent equation method d = 1+ k(Φ (n))/e

- This is equivalent to finding d which satisfies de = 1 +k (Φ (n)), Where k is any integer.
- We can rewrite this as,
- d = (1 + nm)/ e
- Now we work through values of n until an integer solution for e is found:
- k = 0
- d = (1 + 0 * 108) / 5 = 1/5 (no)
- k = 1
- d = (1 + 1 * 108) / 5 = 109/5 (no) k= 2
- d = (1 + 2 * 108) / 5 = 217/5 (no)
- k = 3

d = (1 + 3 * 108) / 5 = 325/5 = 65 (yes !)

Continued...



Public Key	Secret Key

n = 133
e = 5
	n = 133
	d = 65

Continued...

Communication Encryption

- This message must be a number less than the smaller of p and q.
- However, at this point we don’t know p or q, so in practice a lower bound on p and q must be published.
- This can be published below their true value and so isn't a major security concern.
- For e.g., lets use the message “ 6”.
- $C = P^{e} \% n$
- $= 6^{5} \% 133$
- $= 7776 \% 133$
- $= 62$

Continued…

Decryption

- This works very much like encryption, but involves a larger exponentiation which is broken down into several steps.
- $M = c^{d} \% n$

$= 62^{65} \% 133$

$= 62 \text{ x } 62^{64} \% 133$

$= 62 \text{ x } (62^{2})^{32} \% 133$

$= 62 \text{ x } (3844)^{32} \% 133$

$= 62 \text{ x } (\mathbf{3844 \% 133})^{32} \% 133$

$= 62 \text{ x } (\mathbf{120})^{32} \% 133$

Continued...

- **We now repeat the sequence of operation that reduced 62^{65} to 120^{32} to reduce the exponent down to 1.**

$= 62 \text{ x } (120^2)^{16} \% 133$

$= 62 \text{ x } (14400)^{16} \% 133$

$= 62 \text{ x } (\mathbf{14400 \% 133})^{16} \% 133$

$= 62 \text{ x } (\mathbf{36})^{16} \% 133$

$= 62 \text{ x } (36^2)^{8} \% 133$

$= 62 \text{ x } (1296)^{8} \% 133$

$= 62 \text{ x } (\mathbf{1296 \% 133})^{8} \% 133$

$= 62 \text{ x } (\mathbf{99})^{8} \% 133$

$= 62 \text{ x } (992)^{4} \% 133$

$= 62 \text{ x } (\mathbf{9801 \% 133})^{4} \% 133$

Continued...

$= 62 \text{ x } (92)^{4} \% \ 133$

$= 62 \text{ x } (92^{2})^{2} \% \ 133$

$= 62 \text{ x } (92^{2})^{2} \% \ 133$

$= 62 \text{ x } (\mathbf{8464 \ \% \ 133})^{2} \% \ 133$

$= 62 \text{ x } (\mathbf{85})^{2} \% \ 133$

$= 62 \text{ x } (7225) \% \ 133$

$= 62 \text{ x } (\mathbf{7225 \ \% \ 133}) \% \ 133$

$= 62 \text{ x } (\mathbf{43})^{1} \% \ 133$

$= 2666 \% \ 133$

$= \mathbf{6}$

□ And that matches the plaintext we put in at the beginning, so that algorithm worked

Continued...

3. Certificate Authority (CA)

- A certificate authority (CA) is a trusted entity that issues electronic documents that verify a digital entity's identity on the Internet. The electronic documents, which are called digital certificates, are an essential part of secure communication and play an important part in the public key infrastructure (PKI).
- Certificates typically include **the owner's public key, the expiration date of the certificate, the owner's name and other information about the public key owner**.
- Operating systems (OSes) and browsers maintain lists of trusted CA root certificates to verify certificates that a CA has issued and signed. The format of these certificates is specified by the X.509 standard.

Continued…

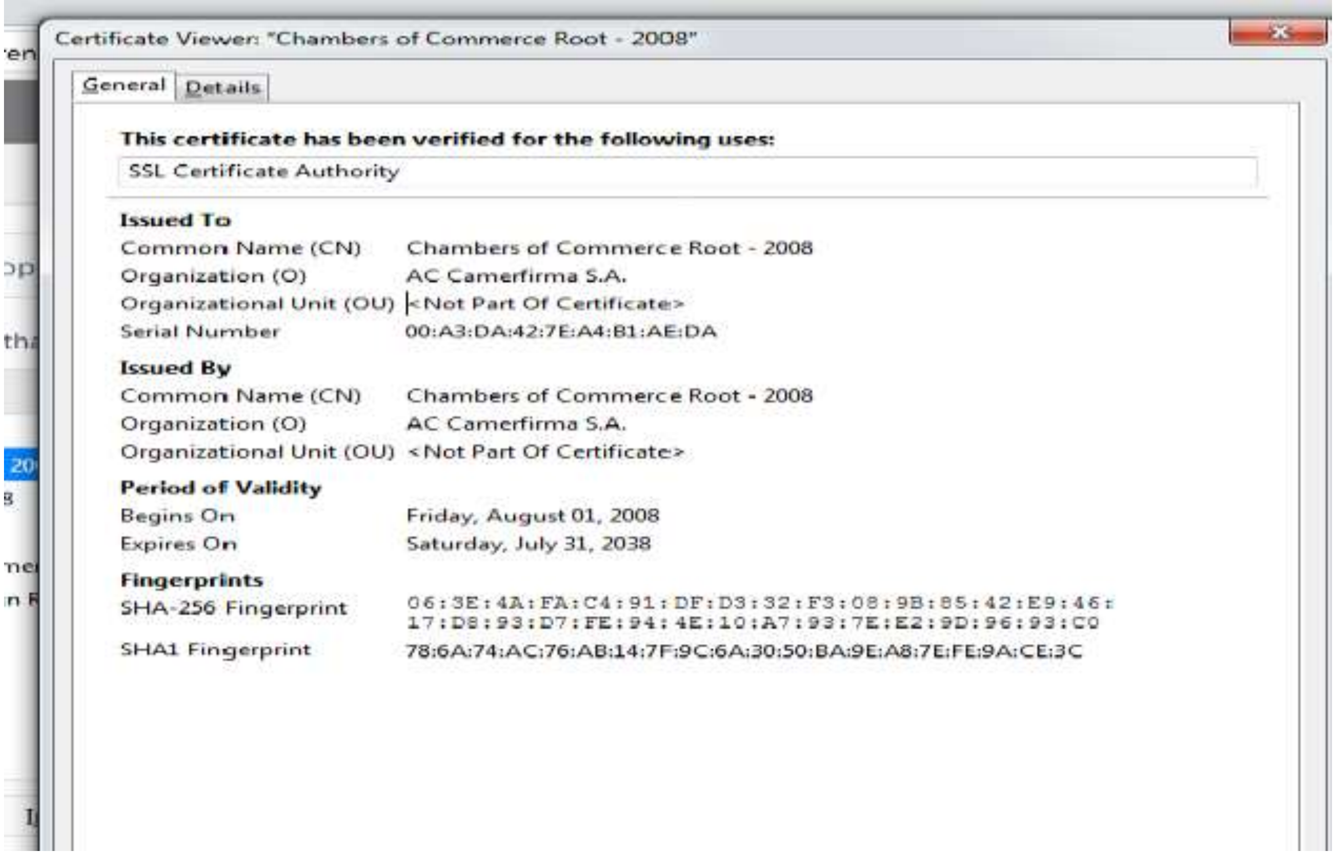
Certificate Authority

- **Certificates** can be stored anywhere.
- A principal can provide its certificate as part of authentication.
- If the CA is compromised it can destroy the integrity of the entire network.

CA have

- Users Name
- Users Public Name o Expiration Time
- Serial Number
- Issuing CA's Signature

Continued...



Certificate Viewer: "Chambers of Commerce Root - 2008"

This certificate has been verified for the following uses:

SSL Certificate Authority

Issued To

Common Name (CN)	Chambers of Commerce Root - 2008
Organization (O)	AC Camerfirma S.A.
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	00:A3:DA:42:7E:A4:B1:AE:DA

Issued By

Common Name (CN)	Chambers of Commerce Root - 2008
Organization (O)	AC Camerfirma S.A.
Organizational Unit (OU)	<Not Part Of Certificate>

Period of Validity

Begins On	Friday, August 01, 2008
Expires On	Saturday, July 31, 2038

Fingerprints

SHA-256 Fingerprint	06:3E:4A:FA:C4:91:DF:D3:32:F3:08:9B:85:42:E9:46: 17:D8:93:D7:FE:94:4E:10:A7:93:7E:E2:9D:96:93:C0
SHA1 Fingerprint	78:6A:74:AC:76:AB:14:7F:9C:6A:30:50:BA:9E:A8:7E:FE:9A:CE:3C

Continued...

- Certificates can be used for:
- **Authentication**, which verifies the identity of someone or something.
- **Privacy**, which ensures that information is only available to the intended audience.
- **Encryption**, which disguises information so that unauthorized readers are unable to decipher it.
- **Digital signatures**, which provide nonrepudiation and message integrity.
- These services can be important to the security of your communications. In addition, many applications use certificates, such as e-mail applications and Web browsers.

Certificate File Formats

- Certificate import and export operations support four file formats. Choose the format that meets your specific requirements.

Continued...

- **Personal Information Exchange (PKCS #12)**
- The Personal Information Exchange format (PFX, also called PKCS #12) supports secure storage of certificates, private keys, and all certificates in a certification path. The PKCS #12 format is the only file format that can be used to export a certificate and its private key.
- **Cryptographic Message Syntax Standard (PKCS #7)** The PKCS #7 format supports storage of certificates and all certificates in the certification path.
- **DER-encoded binary X.509** The Distinguished Encoding Rules (DER) format supports storage of a single certificate. This format does not support storage of the private key or certification path.
- **Base64-encoded X.509** The Base64 format supports storage of a single certificate. This format does not support storage of the private key or certification path.

Continued...

- There are so many SSL certificate providers that competes the other to prove that it is the best. The following are the Top 10 *Secure Sockets Layer* (SSL) Providers in the industry based on the survey conducted by W3Techs and any websites using one of these certificates assure its customers that they are safe and secure while performing sensitive transactions over the websites.

1. **Comodo**

- Comodo ranks the top on the list as the best SSL Provider to secure websites. It offers great security solutions to business that range from small to big. Comodo SSL offers 256-bit encryption which is trusted by all browsers. It also assures technical support from experts with a **warranty that is worth $250,000**. Comodo SSL provides unlimited server licenses. Extended Validation certificate is offered at **an annual cost of $249.**

Continued...

2.IdenTrust

- IdenTrust is next on the line after Comodo and it is doing good in securing major set of bank websites. It offers **256 bit encryption** using the **SHA-2 algorithm** and have more than 5 million users across the globe.

3. Symantec

- Symantec comes next after Identrust. It assures its customers trust when it comes to website security. It offers **256 bit encryption**. It is bit expensive with the cost of its products are sometimes over $1,000 with $1.5 million warranty.

Continued…

4. GoDaddy

- GoDaddy is also famous, though they are well known for domain registration they are much famous for their SSL certificates that are available at a price ranging from **$69 to $299**. It also lets the customers to customize their needs. It also provides **24/7**technical Support.

5. GlobalSign

- GlobalSign offers **2048 bit** future proof SSL Certificates, it also ensures to detect phishing. It offers technical support for its customers.

6. Digicert

- Digicert offers SSL certificates that is good enough to secure a website protecting its customer while performing transactions online. They provide **2048 bit SSL certificates**. The certificates costs from $175 to $595 a year.

Continued...

7. Certum

- Certum offers a good deal of protection while making online transactions. They provide **128 bit encryption** with a month of trial. The certificates costs from $23 to $ 299 per year.

8. Entrust

- Entrust is also one of the companies that falls in the top 10 list in providing the best SSL
- Certificates. It helps the website owners to secure customer transactions and **increase in the rate of sale conversion**. The SSL certificates of Entrust are affordable.

Continued…

9. Startcom

- Startcom assures complete encryption to meet the real demand in protecting the computer networks. The SSL certificate that it provides **encrypts the transactions** done between the web browser and the server.

10. Secom Trust

- Secom Trust is a Japanese based company that offers **SSL certificates to protect websites** and hence the online business.

Introduction to the TCP/IP Stack

TCP/IP:The Language of the Internet

- TCP/IP (Transport Control Protocol/Internet Protocol) is the ``language'' of the Internet. Anything that can learn to ``speak TCP/IP'' can play on the Internet. This is functionality that occurs at the Network (IP) and Transport (TCP) layers in the ISO/OSI Reference Model. Consequently, a host that has TCP/IP functionality (such as UNIX, OS/2, MacOS, or Windows NT) can easily support applications (such as Netscape's Navigator) that uses the network.

Open Design

- One of the most important features of TCP/IP isn't a technological one: The protocol is an ``open'' protocol, and anyone who wishes to implement it may do so freely. Engineers and scientists from all over the world participate in the IETF (Internet Engineering Task Force) working groups that design the protocols that make the Internet work. Their time is typically donated by their companies, and the result is work that benefits everyone.

Continued...

IP

- As noted, IP is a ``network layer'' protocol. This is the layer that allows the hosts to actually ``talk'' to each other. Such things as carrying datagrams, mapping the Internet address (such as 10.2.3.4) to a physical network address (such as 08:00:69:0a:ca:8f), and routing, which takes care of making sure that all of the devices that have Internet connectivity can find the way to each other.

Understanding IP

- IP has a number of very important features which make it an extremely robust and flexible protocol. For our purposes, though, we're going to focus on the security of IP, or more specifically, the lack thereof.

Internet Protocol Security

- In short, Internet Protocol security (IPSec) is a framework of open standards for helping to ensure private, secure communications over Internet Protocol (IP) networks through the use of cryptographic security services.

Continued…

Attacks against IP

- A number of attacks against IP are possible. Typically, these exploit the fact that IP does not perform a robust mechanism for authentication, which is proving that a packet came from where it claims it did.
- *A packet simply claims to originate from a given address, and there isn't a way to be sure that the host that sent the packet is telling the truth*.
- This isn't necessarily a weakness, but it is an important point, because it means that the facility of host authentication has to be provided at a higher layer on the ISO/OSI Reference Model.
- Today, applications that require strong host authentication (such as cryptographic applications) do this at the application layer.

Continued…

What attacks does IPSec protect against?

- It is integrated at the Internet layer (layer 3), it provides security for almost all protocols in the TCP/IP suite (through different policies) and it can help against:
- Network-based attacks from untrusted computers, attacks that can result in the denial-of-service of applications, services, or the network
- Data corruption
- Data theft
- User-credential theft
- Administrative control of servers, other computers, and the network
- Block untrusted communications
- Most of these threats occur throught some sort of Man in The Middle attack

Continued...

IP Spoofing.

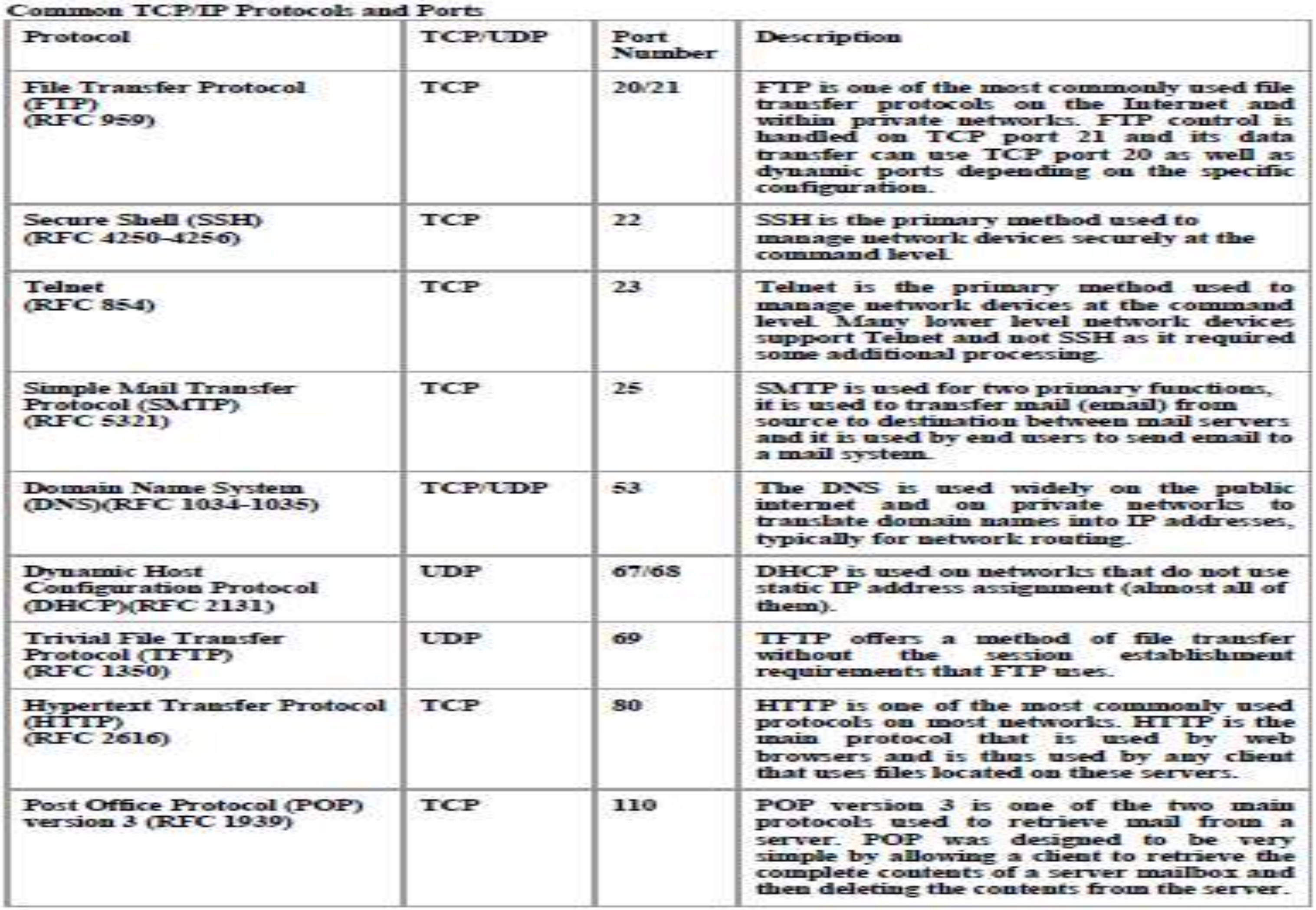
- This is where one host claims to have the IP address of another. Since many systems (such as router access control lists) define which packets may and which packets may not pass based on the sender's IP address, this is a useful technique to an attacker: he can send packets to a host, perhaps causing it to take some sort of action.
- Additionally, some applications allow login based on the IP address of the person making the request (such as the Berkeley r-commands).
- These are both good examples how trusting untrustable layers can provide security that is - - at best -- weak.

Continued...

IP Session Hijacking.

- This is a relatively sophisticated attack, first described by Steve Bellovin.
- This is very dangerous, however, because there are now toolkits available in the underground community that allow otherwise unskilled bad-guy-wannabes to perpetrate this attack.
- IP Session Hijacking is an attack whereby a user's session is taken over, being in the control of the attacker.
- If the user was in the middle of email, the attacker is looking at the email, and then can execute any commands he wishes as the attacked user. The attacked user simply sees his session dropped, and may simply login again, perhaps not even noticing that the attacker is still logged in and doing things.

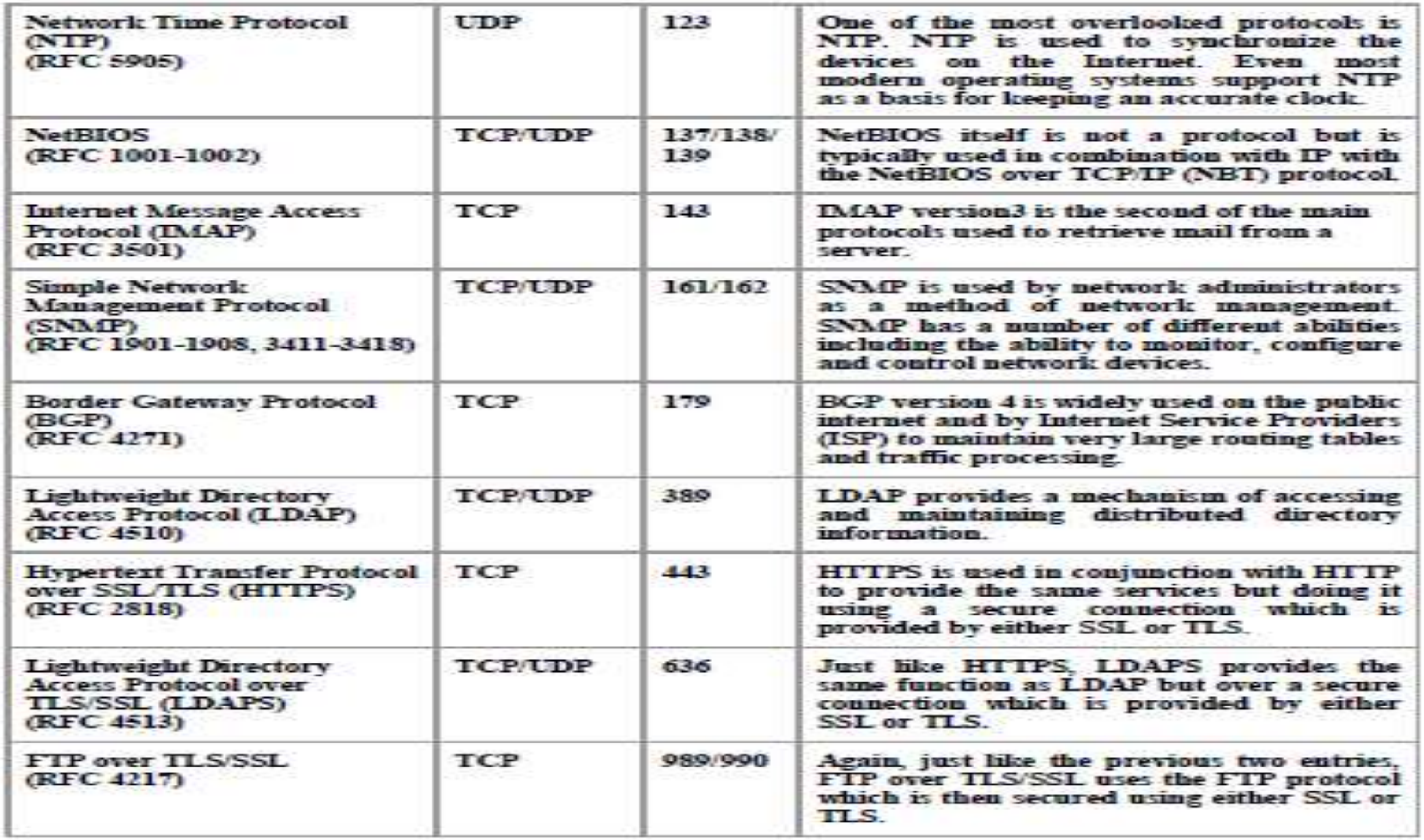
Network Security (Ports and Protocols)
Common TCP/IP Protocols and Ports



Common TCP/IP Protocols and Ports

Protocol	TCP/UDP	Port Number	Description
File Transfer Protocol (FTP) (RFC 959)	TCP	20/21	FTP is one of the most commonly used file transfer protocols on the Internet and within private networks. FTP control is handled on TCP port 21 and its data transfer can use TCP port 20 as well as dynamic ports depending on the specific configuration.
Secure Shell (SSH) (RFC 4250-4256)	TCP	22	SSH is the primary method used to manage network devices securely at the command level.
Telnet (RFC 854)	TCP	23	Telnet is the primary method used to manage network devices at the command level. Many lower level network devices support Telnet and not SSH as it required some additional processing.
Simple Mail Transfer Protocol (SMTP) (RFC 5321)	TCP	25	SMTP is used for two primary functions, it is used to transfer mail (email) from source to destination between mail servers and it is used by end users to send email to a mail system.
Domain Name System (DNS)(RFC 1034-1035)	TCP/UDP	53	The DNS is used widely on the public internet and on private networks to translate domain names into IP addresses, typically for network routing.
Dynamic Host Configuration Protocol (DHCP)(RFC 2131)	UDP	67/68	DHCP is used on networks that do not use static IP address assignment (almost all of them).
Trivial File Transfer Protocol (TFTP) (RFC 1350)	UDP	69	TFTP offers a method of file transfer without the session establishment requirements that FTP uses.
Hypertext Transfer Protocol (HTTP) (RFC 2616)	TCP	80	HTTP is one of the most commonly used protocols on most networks. HTTP is the main protocol that is used by web browsers and is thus used by any client that uses files located on these servers.
Post Office Protocol (POP) version 3 (RFC 1939)	TCP	110	POP version 3 is one of the two main protocols used to retrieve mail from a server. POP was designed to be very simple by allowing a client to retrieve the complete contents of a server mailbox and then deleting the contents from the server.

Continued…



Network Time Protocol (NTP) (RFC 5905)	UDP	123	One of the most overlooked protocols is NTP. NTP is used to synchronize the devices on the Internet. Even most modern operating systems support NTP as a basis for keeping an accurate clock.
NetBIOS (RFC 1001-1002)	TCP/UDP	137/138/139	NetBIOS itself is not a protocol but is typically used in combination with IP with the NetBIOS over TCP/IP (NBT) protocol.
Internet Message Access Protocol (IMAP) (RFC 3501)	TCP	143	IMAP version3 is the second of the main protocols used to retrieve mail from a server.
Simple Network Management Protocol (SNMP) (RFC 1901-1908, 3411-3418)	TCP/UDP	161/162	SNMP is used by network administrators as a method of network management. SNMP has a number of different abilities including the ability to monitor, configure and control network devices.
Border Gateway Protocol (BGP) (RFC 4271)	TCP	179	BGP version 4 is widely used on the public internet and by Internet Service Providers (ISP) to maintain very large routing tables and traffic processing.
Lightweight Directory Access Protocol (LDAP) (RFC 4510)	TCP/UDP	389	LDAP provides a mechanism of accessing and maintaining distributed directory information.
Hypertext Transfer Protocol over SSL/TLS (HTTPS) (RFC 2818)	TCP	443	HTTPS is used in conjunction with HTTP to provide the same services but doing it using a secure connection which is provided by either SSL or TLS.
Lightweight Directory Access Protocol over TLS/SSL (LDAPS) (RFC 4513)	TCP/UDP	636	Just like HTTPS, LDAPS provides the same function as LDAP but over a secure connection which is provided by either SSL or TLS.
FTP over TLS/SSL (RFC 4217)	TCP	989/990	Again, just like the previous two entries, FTP over TLS/SSL uses the FTP protocol which is then secured using either SSL or TLS.

Continued...

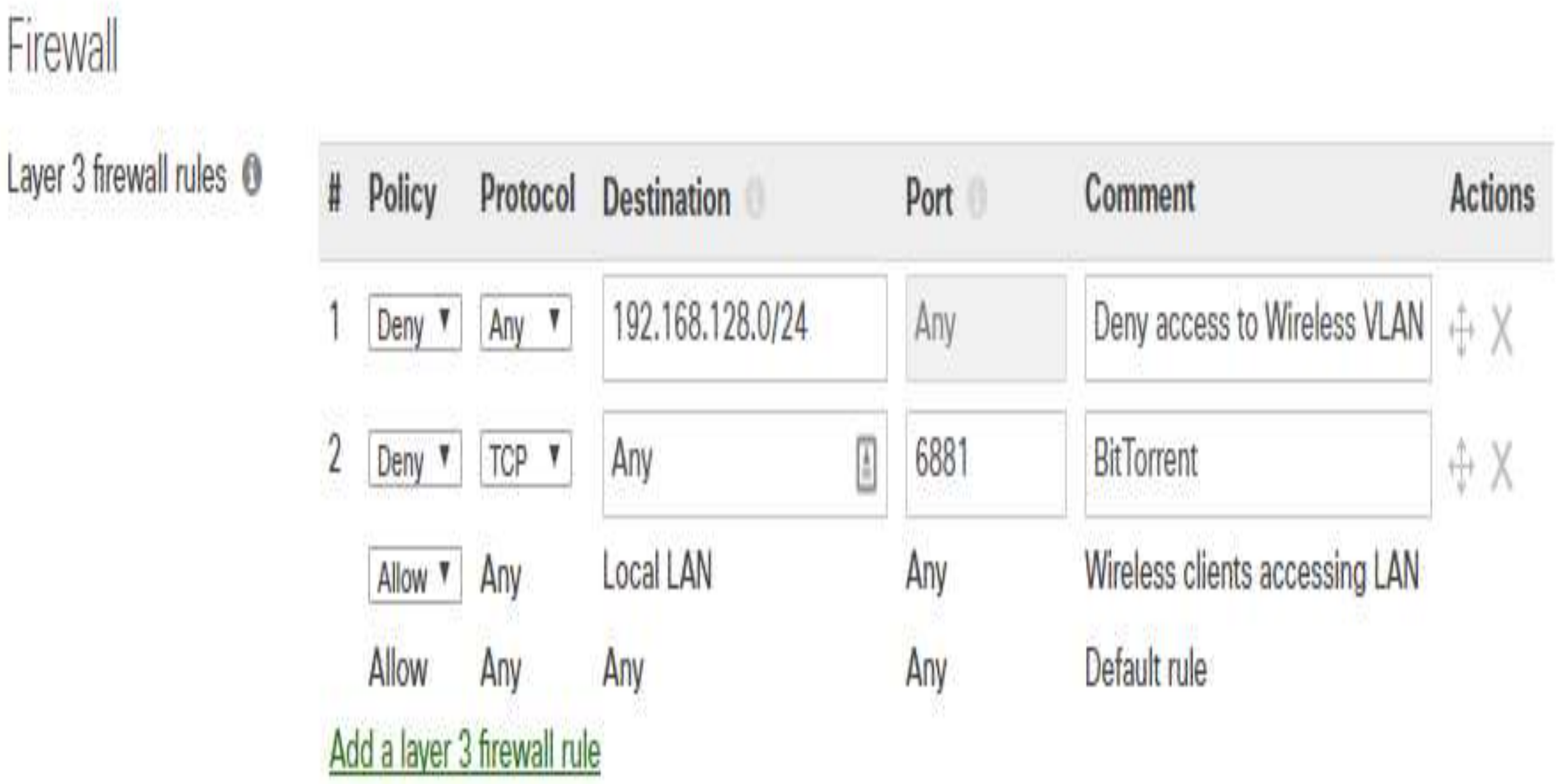
Firewall Rules

- Custom firewall rules provide an administrator with more granular access control beyond LAN isolation.
- An administrator can define a set of firewall rules that is evaluated for every request sent by a wireless user associated to that SSID.
- Firewall rules are evaluated from top to bottom. The first rule that matches is applied, and subsequent rules are not evaluated.
- If no rules match, the default rule (allow all traffic) is applied.

Layer 3 Firewall Rules

- As an example, the figure below depicts a sample set of custom firewall rules that will be enforced at layer 3.

Continued…



Firewall

Layer 3 firewall rules

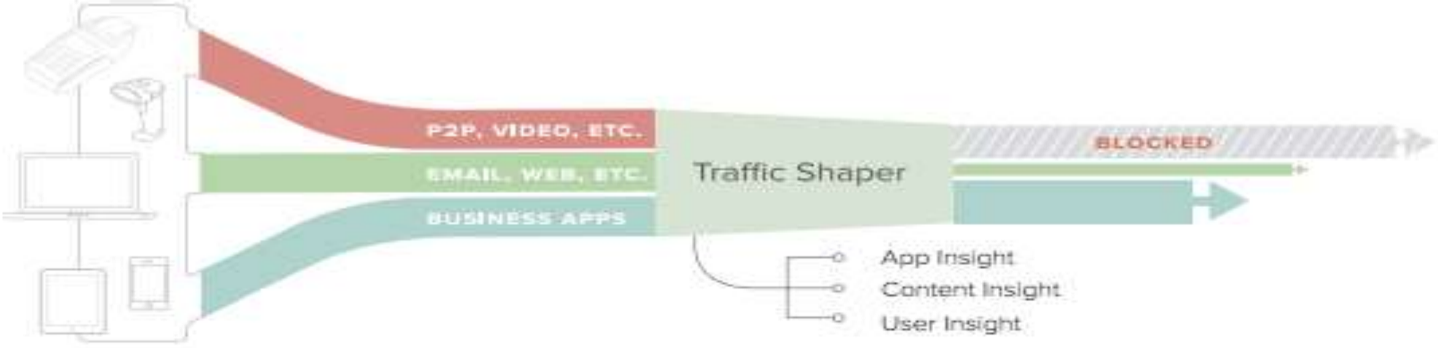
#	Policy	Protocol	Destination	Port	Comment	Actions
1	Deny	Any	192.168.128.0/24	Any	Deny access to Wireless VLAN	✥ X
2	Deny	TCP	Any	6881	BitTorrent	✥ X
	Allow	Any	Local LAN	Any	Wireless clients accessing LAN	
	Allow	Any	Any	Any	Default rule	

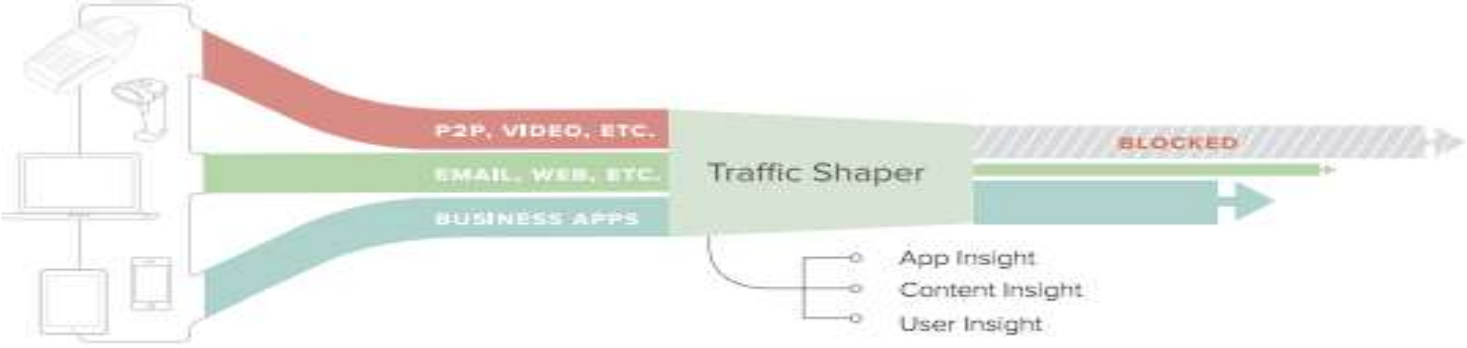
Continued...

Layer 7 Firewall Rules

- Layer 7 traffic analysis technology, it is possible to create layer 7 firewall rules to completely block certain applications without having to specify specific IP addresses or port ranges.
- This can be useful when applications use multiple or changing IP addresses or port ranges.
- It is possible to block applications by category (e.g. 'All video & music sites') or for a specific type of application within a category (e.g. only iTunes within the 'Video & music' category). The figure below illustrates a set of layer 7 firewall rules including applications blocked by entire categories and specific

Continued...


P2P, VIDEO, ETC.
EMAIL, WEB, ETC.
BUSINESS APPS
Traffic Shaper
BLOCKED
App Insight
Content Insight
User Insight


P2P, VIDEO, ETC.
EMAIL, WEB, ETC.
BUSINESS APPS
Traffic Shaper
BLOCKED
App Insight
Content Insight
User Insight

Thank You!