Network Security - Summary

Manuel Waibel

May 4, 2023

Contents

1 Threats and Security Objectives

1.1 Types of Network Attacks

- Eavesdropping

  Passive wiretapping done secretly

- Man in the Middle

  Active wiretapping attack, in which the attacker intercepts and selectively **modifies** communicated data to masquerade as one or more entities involved the communication.

- Denial-of-Service (DoS)

  The **prevention of authorized access** to a system resource or the **delaying** of system operations and functions

1.2 Security Objectives

1.2.1 Confidentiality

. . . data is not disclosed to system entities unless they have been authorized to know the data.

$\Longrightarrow$ encryption, access control

1.2.2 Integrity

. . . data has not been changed, destroyed, or lost in an unauthorized or accidental manner.

$\Longrightarrow$ message authentication, device authentication

1.2.3 Availability

. . . a system or a system resource being accessible, or usable or operational upon demand, by an authorized system entity, according to performance specifications for the system . . .

$\Longrightarrow$ protection of infrastructure, authorization

1.3 Types of scans

1.3.1 Horizontal Scan (Host Scan)

- Search if a **specific port** is open on **multiple hosts**

- Packets to
  - a **single** destination **port**
  - on **multiple** destination IP **addresses**

1.3.2 Vertical Scan (Port Scan)

- Search if **ports** are open on a **specific host**
- Packets to
  - a **multiple** destination **port**
  - on **single** destination IP **addresses**

1.4 Types of Attacks

- (D)DoS
- Viruses
- Worms
- Botnets
- ...

1.4.1 Denial of Service

Types:

- Physical disruption

  Cable, radio signal
- Overloading network and network devices

  e.g., ICMP flooding
- Altering configuration

  e.g., ARP, Routing, DNS
- Overloading computational resources

  CPU, memory

  Data structures, table entries

  State keeping $\implies$ Claim & Hold

Example Dos Attacks:

- Reflection DoS Attack

  Attacker send Requests to multiple other Hosts with spoofed address $\Longrightarrow$ all hosts answer to victim with said spoofed address.

  Example: Smurf Attack - Attacker sends `ICMP ECHO REQUEST` to multiple hosts, which then all reply to victim

- SYN-Flooding (Claim & Hold)

  Attacker sends multiple `TCP-SYN` packets to server, which then reserves space for the attackers connection. The attacker never completes the Handshake and tries to claim & hold all available connections to the server.

1.5 Backscatter

Reply to spoofed addresses

1.6 Worms vs. Virus

Virus: A self-replicating (and usually hidden) section of computer software, that propagates by infecting other programs. A virus **cannot run by itself**, it requires that its host program to be run, in order to make the virus active.

Worm: A computer program that **can run independently**, can propagate a complete version of itself onto other hosts on a network and may consume system resources destructively.

1.6.1 Kinds of Worms

Monomorphic Worm:

- Payload does not change
- Signature-based detection possible (because file signature does not change)

Polymorphic Worm:

- Payload changes (e.g. different encryptions used)
- Behaviour (and algorithm) remain the same

Metamorphic Worm:

- Payload changes
- Behaviour changes

1.7 Botnets

Multiple infected targets (bots) listen to one or more Command and Control (C&C) centers for instructions.

1.7.1 C&C Structures

Centralized: Single C&C controls bots

+ Low latency
- Easier to detect
- Single point of failure

P2P: Bots communicate with each other. Special bots, which are directly connected to C&C propagate commands to the other bots, which themselves propagate the commands, ...

+ Robust
- Latency, Loss
- Scalability
- complexity (manageability)

Hybrid: Mix between centralized and P2P nets - multiple servent bots (with public IPs) control client bots (with private IPs)

1.7.2 Registering to a C&C

- **Hardcoded IP address**
  + No need for DNS lookup
  - Bot code reveals C&C
  - Denylisting/Filtering C&C IPs
- **Peer list**
  + Separate from bot binary
  + Can be updated
- **Hardcoded Domain name**
  + Can be mapped to different IP addresses
- **Domain Generation Algorithm (DGA)**
  + Algorithm known to C&C and bots

2 Cryptography - Basics

1. Number of possible keys should be large.
2. Keys should be long.
3. Keys should be chosen randomly

2.1 One Time Pad (OTP)

Proven to be unbreakable, **iff** only used once.

Key: Both communication partners need the same key. The key has to have the length of the message to encrypt.
Encryption: Message is encrypted by applying a *XOR* to the message with the key
Decryption: The Message is decrypted by again applying a *XOR* to the encrypted message, revealing the original message.

If the key is used twice the intersection of the two messages can be recovered.

$$c_A = E(k, m_A) = m_A \oplus k$$
$$c_B = E(k, m_B) = m_B \oplus k$$

$$c_A \oplus c_B = m_A \oplus k \oplus m_B \oplus k = m_A \oplus m_B$$

2.2 Cipher types

1. **Substitution ciphers**: Substitute original letters of message with others

   Frequency analysis possible (but requires much effort)

   Monoalphabetic: fixed alphabet to substitute letters (e.g. Caesar cipher)

   Polymorphic: multiple alphabets used, substitution rules change (e.g. Vignère, Enigma)

2. **Transposition ciphers:** Rearrange the original letters of the message

   Letters of plaintext remain $\implies$ frequencies remain

   can give hint to original message

Substitution ciphers can be made secure (OTP), transposition ciphers not.

In practice usually a combination of substitution and transposition is used.

3 Modern Ciphers

3.1 Stream Ciphers

Stream ciphers are symmetric ciphers, where the key has the same length as the message. The message is encrypted by applying a XOR operation to the message and the key ($c = m \oplus k$).

- Fast
- Simple
- Easy to implement in hardware
- Useful for
  - Resource constraint devices
  - High data rates
  - High error rates

Examples: OTP, RC4 and A5/1

3.1.1 Advantage

Compares probabilities ($P(X)$) for pseudo-random generator (PRG) sequence $G(s)$ and real random sequence R

$$Adv(A, G(s)) = |P(A(G(s)) = 1) - P(A(R) = 1)|$$

The **smaller** $Adv(A, G(s))$ is, the **better** the PRG is.
If the difference is high, $G(s)$ is detectably different from a real random sequence $\implies$ weak PRG.

Example:
$P(A(R) = 1) = 0.9$
$P(A(G(s)) = 1) = 0.2$

$$\begin{aligned} Adv(A, G(s)) &= |P(A(G(s)) = 1) - P(A(R) = 1)| \\ &= |0.2 - 0.9| \\ &= 0.7 \implies \text{weak PRG} \end{aligned}$$

3.1.2 Ron's Code 4 - RC4

1. Two byte arrays

   S: Permutation of all possible bytes (=256 entries)

   K: initial key of length n ($1 \leq n \leq 256$, typically 40 - 128 bits)

2. 2 indices i and j which are some distance away from each other (calculation omitted)

3. S is shuffled according to K, i and j

   Swap `S[i]` and `S[j]`

4. Output to keystream is `S[t]`, where `t = (S[i] + S[j]) mod 256`

RC4 is used in network encryption cipher WEP. WEP prepends an additional initialization vector (IV) as padding to prevent key repetitions (seed = IV + key).
Subsceptible to a known-plaintext attack: Discover secret key (seed) with known IV.
Prerequisites:

- Attacker knows many ciphertext messages
- Attacker knows (or can guess) first byte of plaintext (e.g. from standard message formats)

3.2 Block Ciphers

Algorithm operates on message blocks of fixed length.
Message usually goes through multiple iterations of S-Box and P-Box operations. The key is usually adapted for every round (sub-keys).

- Good for encryption of bulk data
- Better security than stream ciphers (but slower)
- On transmission errors, the whole block has to be retransmitted

3.2.1 Feistel Cipher

Plaintext is split into left (L) and right (R) part. The sides then alternate in getting combined with the key and the corresponding other side is XOR'ed with the result (= combination with side + key).

$$\begin{aligned} \text{Input: } & (L_0, R_0) \\ & L_{i+1} = R_i \\ & R_{i+1} = L_i \oplus F(R_i, K_i) \\ \text{Output: } & (R_{n+1}, L_{n+1}) \end{aligned}$$

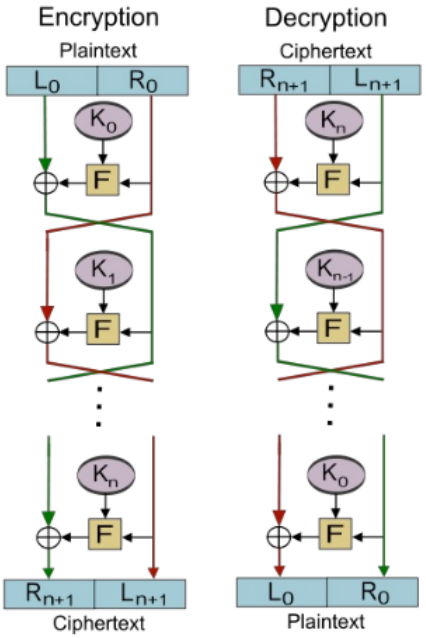


Figure 1: Feistel Cipher operation

The **DES** Structure uses the Feistel Cipher for 16 Rounds with a 56 bit key seed (today considered not secure).

- DES function: $c = E(m, k)$
- Double DES function: $c = E(E(m, k_A), k_B)$
- Triple DES (3DES) function: $c = E(\underline{\mathbf{D}}(E(m, k_A), k_B), k_C)$
  - 3DES uses encrypt-decrypt-encrypt for backwards compatibility. If all keys are the same, it behaves like single DES ($k_A = k_B = k_C \implies$ single DES).

Successor is the Advanced Encryption Standard **AES**.

- Block length: 128 bits
- Rounds: 10-14 (depending on key length)
- Key lengths: 128, 192 or 256 bits

AES uses substitution and permutation.

3.3 Block Cipher Modes

3.3.1 Electronic Codebook (ECB)

1. Message is split into equal sized blocks (if needed padding is added)

2. Each block is encrypted independently

$\Longrightarrow$ **Problem:** Equal plaintext generates equal ciphertext
$\Longrightarrow$ Add randomness to ciphertext be applying (XOR) a random initialization vector to each block (random vectors have to be transmitted with ciphertext)

3.3.2 Cipher Block Chaining (CBC)

1. Message is split into equal sized blocks (if needed padding is added)
2. First block is XOR'rd with an initialization vector IV and then encrypted (adds randomness to plaintext before encrypting)
3. Digest is then used as vector for next block (XOR with digest and then encrypt)
4. repeat step 3. until all blocks are encrypted

$\Longrightarrow$ Solves problem of needing to transmit random numbers of all blocks with cipher text. Instead only the first initialization vector IV has to be transmitted.
$\Longrightarrow$ corruption of one block or IV will corrupt the whole message (or at least all following blocks)

3.3.3 Output Feedback (OFB)

1. Message is split into equal sized blocks - no padding needed in this mode
2. Initialization vector IV is encrypted and then combined with plaintext block
3. The digest of the encrypted IV is then encrypted again and again XOR'rd with the plaintext
4. Double encrypted IV is then again encrypted (and so on...) and combined with plaintext until all plaintext blocks are used

Note only IV is encrypted again and again. To create the ciphertext the encrypted vector at the current stage is combined with the corresponding plaintext (see Figure 2).

3.3.4 Cipher Feedback (CFB)

Just like OFB, but instead not only the encrypted IV is propagated, but instead the combination with the encrypted IV and the plaintext (see Figure 3).

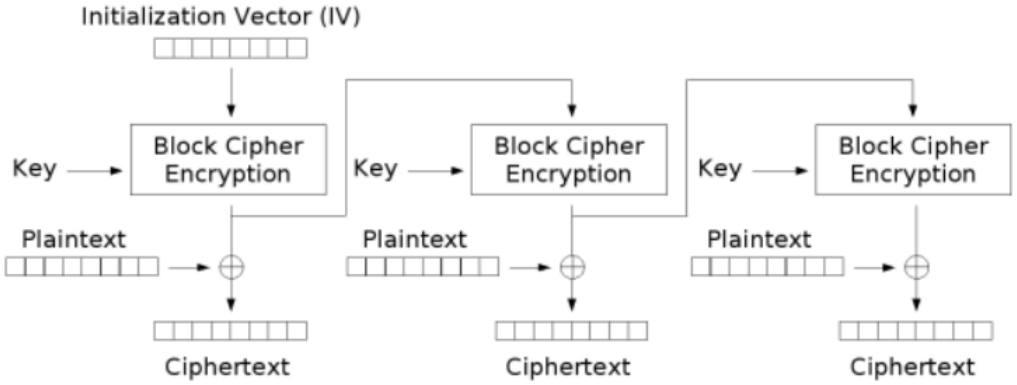


Figure 2: Output Feedback Mode

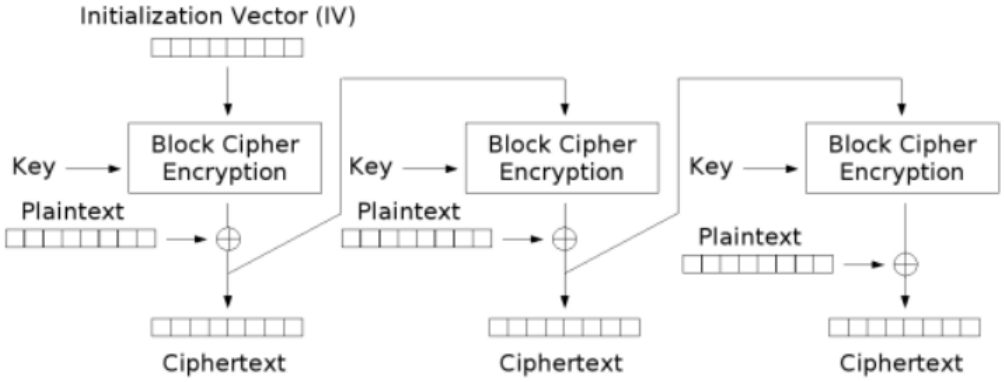


Figure 3: Cipher Feedback Mode

3.3.5 Counter Mode (CTR)

Essentially realizes a stream cipher

1. Message is split into equal sized blocks
2. IV consists of random nonce and counter
3. Counter is incremented for every block ($\implies$ every block has different IV)
4. IV is encrypted and *XOR*'rd with plaintext

3.4 Message Authentication

Protect a message against modification:
Integrity - data has not been changed, destroyed, or lost in an unauthorized or accidental manner.

3.4.1 Message Authentication Code (MAC)

Usually computed with cryptographic hash function and secret key.

1. Sender generates a tag t via a key
2. Append tag to message as MAC
3. Receiver calculates tag t' via key
4. Compares tags t and $t' \implies$ accepts or rejects

Why MAC and not CRC?
Cyclic redundancy check (CRC) is a way to check if messages have been altered. Though this could be used in principal, it does not use a key and is therefore "not secure enough" to be used alone for message authentication. It is however used as part of encryption algorithms such as WEP.

MAC vs. Digital Signatures:

MAC	Digital Signature
• Verify that message was not altered • Generation and Verification of MAC with the same key (in sym. cryptography) ◦ everyone who knows secret key can generate MAC	• Prove that message was sent by a specific sender • Only originator of message should be able to provide correct signature (possible with asym. cryptography)

Cryptographic Hash Functions:

1. Compression

   Hash has a fixed size

   Length of hash is usually smaller than of message $len(h(x)) < len(x)$
2. Efficiency

   It should be easy to compute the hash of a message
3. One-way

   No feasible way to invert hash
4. Weak collision resistance

   Given x and $h(x)$, not feasible to find a y such that $h(x) = h(y)$

5. Strong collision resistance

   Not feasible to find x and y such that $h(x) = h(y)$

4 Asymmetric Cryptography

Private/secret key k_s and public key k_p are used. Only the public key k_p is ever published!
One can only encrypt using the public key k_p (but not decrypt).
One can only decrypt using the private key k_s (but not encrypt).

Signatures:
Sender can proof the he/she is the one sending the message by signing hash or message with with his/her private key. Recipient can then verify signature with the persons public key.

4.1 Prime factorization

$$N = p \cdot q$$
$$g^r = mN + 1$$

For a random guess $g < N$ where $N \neq 0 \mod g$.
Then find the remainder r such that

$$g^r = 1 \mod N$$

Then split equation into

$$\underbrace{(g^{r/2}+1)}_{p}\underbrace{(g^{r/2}-1)}_{q} = m \cdot \underbrace{N}_{p \cdot q} = \varphi(N) \text{ (Totient function)}$$
$$(g^{r/2}+1) = k_1$$
$$(g^{r/2}-1) = k_2$$

Choose either k_1 or k_2 and then apply Euclid's algorithm:

$$k_1 = x \mod N \text{ or } k_2 = x \mod N$$
$$N = y \mod x$$
$$x = z \mod y$$
$$\vdots$$
$$\mathbf{p} = 0 \mod a$$

Then you can calculate q by dividing N with p:

$$\frac{N}{p} = \mathbf{q}$$

4.1.1 Example

$$N = 21$$
$$p \cdot q = 21$$

Calculate r with guess g (here $g = 8$):

$$8^1 = 8 \mod 21$$
$$8^2 = 1 \mod 21$$
$$\mathbf{r} = 2$$

Now we can fill in the equation:

$$(g^{r/2} + 1) =$$
$$(g^{r/2} - 1) =$$

$$(8^{2/2} + 1) =$$
$$(8^{2/2} - 1) =$$

$$(8 + 1) = 9$$
$$(8 - 1) = 7$$

Now we use Euclid's algorithm to find the greatest common divisor:

$$7 = 0 \mod 21$$
$$\mathbf{p} = 7$$

We have found **p**! Now we divide N through p to find **q**:

$$q = \frac{21}{7}$$

$$\mathbf{q} = 3$$

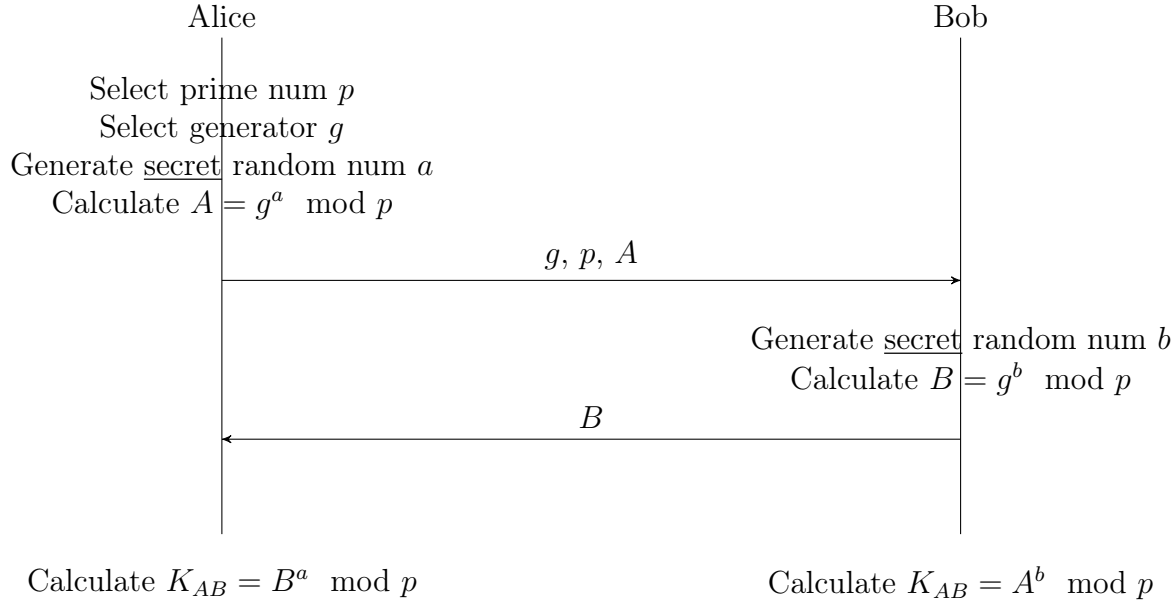
The prime factors of **21** are **3** and **7**.

4.2 Diffie-Hellman Key Exchange

Establish a joint secret key to use symmetric encryption for communication (because asym. encryption is rather slow). Challenge: establish secret key over insecure medium

1. Select a prime number p
2. Select a generator g

3. Generate a secret random number a
4. Calculate $A = g^a \mod p$
5. Send g, p and A to recipient
6. Recipient generates secret number b
7. Recipient calculates $B = g^b \mod p$
8. Recipient can calculate $A^b \mod p$
9. Recipient sends B to sender
10. Sender can calculate $B^a \mod p$



4.3 Elliptic Curve Cryptography (ECC)

Uses a different finite cyclic group than Diffie-Hellman: Points on an elliptic curve.
Elliptic curve (Weierstrass equation):

$$y^2 = x^3 + ax + b$$

Additional condition:

$$4a^3 + 27b^2 \neq 0$$

5 Security Protocols, Anomaly Detection

5.1 Certificate Authorities

Certificate Authorities (CA) confirm ones identity/public key. A CA manages the identities to certain public keys. A CA can build trust by being trusted by other CAs. If a CA is compromised, this CA is untrusted by others.

5.2 IPsec

Normal IP Packet structure:

IP Header	**Payload (Transport Header, Appl. data)**

5.2.1 Modes of Use

Endpoint-to-Endpoint Transport Mode

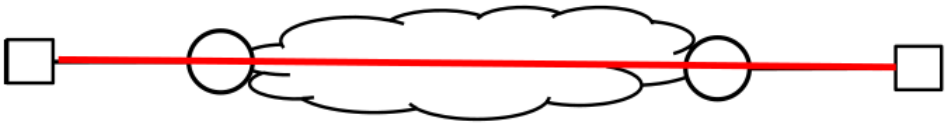


Figure 4: Endpoint-to-Endpoint Transport Mode

Gateway-to-Gateway Tunnel Mode

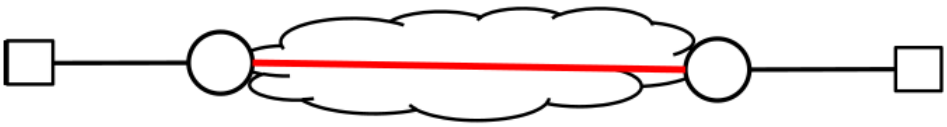


Figure 5: Gateway-to-Gateway Tunnel Mode

Endpoint-to-Gateway Tunnel Mode
Principle of a VPN.

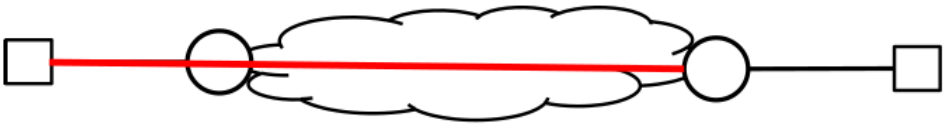


Figure 6: Endpoint-to-Gateway Tunnel Mode

5.2.2 Transport Mode

- Payload is encrypted
- IP header remains readable (attacker can know who communicates)
- Protection for higher layer protocols
- More efficient than Tunnel Mode

5.2.3 Tunnel Mode

- Original IP header is encrypted

  $\Longrightarrow$ New IP header is added for gateway address

5.3 Internet Key Exchange (IKE)

Goal:

- Establish a shared state between source and destination

  Which services provided

  Which cryptographic algorithms used

  Which keys used as input

  Establish IKE Security Association (SA)

- Mutual authentication

  Know identity of the communication partner

- Establish keying material to be used for IPsec

  Encryption

  Integrity Check

5.4 IP Darkspace

IP addresses announced by routing but no hosts attached.

5.4.1 Feature observation

Entropy:
The entropy is high, if the distribution of observed features is very diverse (e.g. each packets has a different destination IP) - plot (frequency vs. feature) is very flat and distributed.
The entropy is low of the distribution of the observed feature is very narrow (e.g. same port for every observed packet) - plot is very concentrated and has (a) peak(s).

5.5 Anomalies

Point Anomaly:
Points normally build clusters, but particular point stands out and cannot be grouped to others.

Contextual Anomaly:
A certain value is expected at a certain point, but the data shows something different (e.g. very low temperature in June)

Collective Anomaly:
A collective anomaly can be detected, if we have multiple comparable data sources. We expect a certain value, because it can be inferred from the other sources.

5.6 Support Vector Machines (SVMs)

SVMs try to optimize a classifier (e.g. a linear function to separate data points) with maximizing the margin between points. For this support vectors from the data points to the classifier function are used to maximize their "length".

5.6.1 What if a linear classifier cannot separate data because it is not linear?

Transform points into a higher dimension.