CS 798: Digital Forensics and Incident Response

Lecture 13 - Covert Channels and Traffic Obfuscation

Diogo Barradas

Winter 2025

University of Waterloo

Why are we studying covert channels?

- Criminals may wish to transfer sensitive/unauthorized information through a channel that is not supposed to transmit that information
  - Makes it more difficult to detect data exchanges
  - Makes a digital investigator's life harder

Croissant-based covert channel

Why should we care?

- **Corporate Espionage**
  - Loss of competitive advantage
- **Government or military activities**
  - Increased threat to National Security
- **Criminal Activities**
  - Transfer of pornography or commercial software
- **Financial Impact**
  - Transfer of confidential financial data

Outline

1. Definition

2. Network information hiding

   Network covert channels

   Traffic obfuscation

Definition

Covert channel

- A **covert channel** is a path for the illegal flow of information between subjects within a system, utilizing system resources that were not designed to be used for inter-subject communication.
- What information can/cannot be transmitted through a channel may be determined by a policy/guidelines/physical limitations, etc.

Types of covert channel

- Several dimensions to be considered:
  - Local vs. remote
  - Storage vs. timing
  - Noisy vs. noiseless
- Important characteristics:
  - **Bandwidth**: how many Bps can be transmitted through the covert channel?
  - **Noise**: Is the information transmitted through the covert channel distorted in any way?

Local vs. remote covert channels

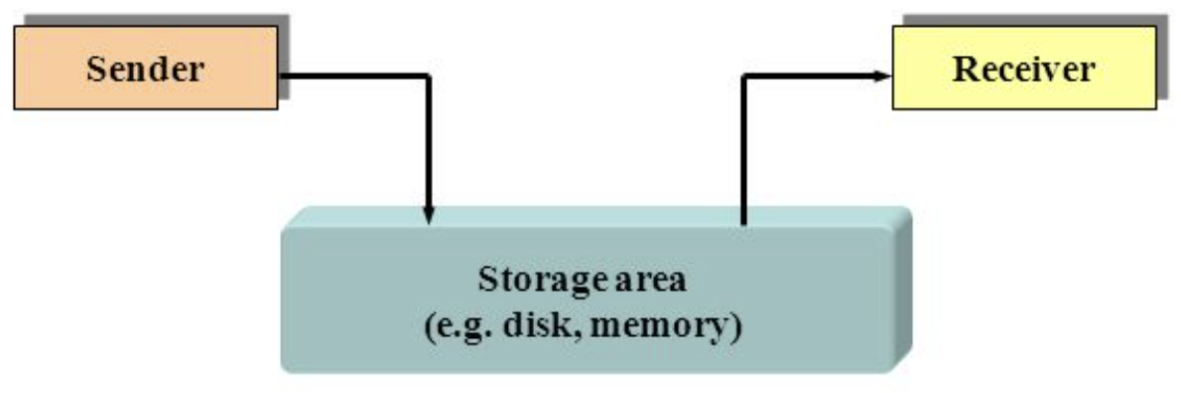
- Local covert channels leverage a machine's shared resources:
  - CPU, RAM, Disk...
- Remote covert channels leverage transmission mechanisms
  - Typically the network (but also others...)

ETHERLED: Air-gapped systems leak data via network card LEDs

By **Bill Toulas** August 23, 2022 07:28 AM

Covert storage channels

- To use a **covert storage channel**:
  - Both sender and receiver must have access to some attribute of a shared object.
  - The sender must be able to modify the attribute.
  - The receiver must be able to view that attribute
  - A mechanism must be in place for initiating the sender and receiver processes, and there must be a way to sequence their accesses to the shared resource (e.g., sync header)

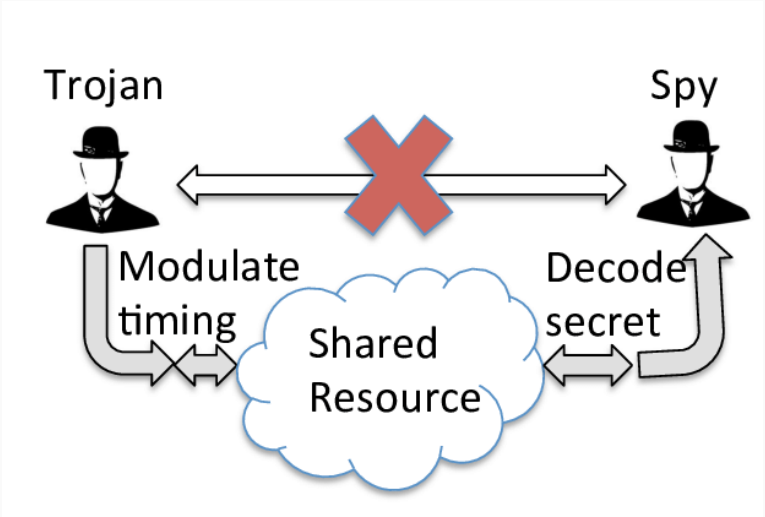


Examples of covert storage channels

- **File permissions**:
  - A user with top secret clearance creates a file and manipulates access rights. A user with no clearance level tries to read the file. They will either get "file does not exist" (0) or "access denied" (1).
- **Other examples**: File naming, print spacing, memory occupied by a folder

Covert timing channels

- To use a **covert timing channel**:
  - Both sender and receiver must have access to some attribute of a shared object.
  - Both sender and receiver have access to a time reference (real-time clock, timer, ordering of events).
  - The sender must be able to control the timing of the detection of a change in the attribute of the receiver.



Examples of covert timing channels

- **Cache-based covert channels**:
  - Leak information cross-VMs via L2 cache
- Meltdown & co.

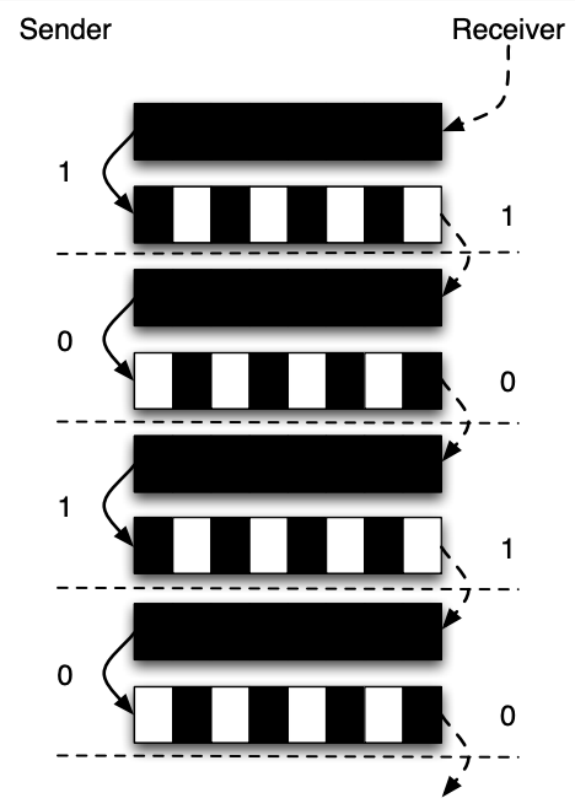


Figure 1: An illustration of a covert channel using L2 cache to encode information. For each bit, the sender evicts half of the cache lines from the L2 cache saturated previously by the receiver (solid lines). The receiver then decodes the information by measuring the difference of timing in accessing different subsets of the cache (dashed lines).

Can't we just get rid of covert channels?

- It is typically infeasible to eliminate every potential covert channel in a (networked) computer system, but we can:
  - Eliminate them by modifying the system implementation.
  - Reduce their bandwidth by introducing noise into the channel.
  - Monitor for usage patterns that indicate someone is trying to exploit a covert channel.

Some attempts at detection

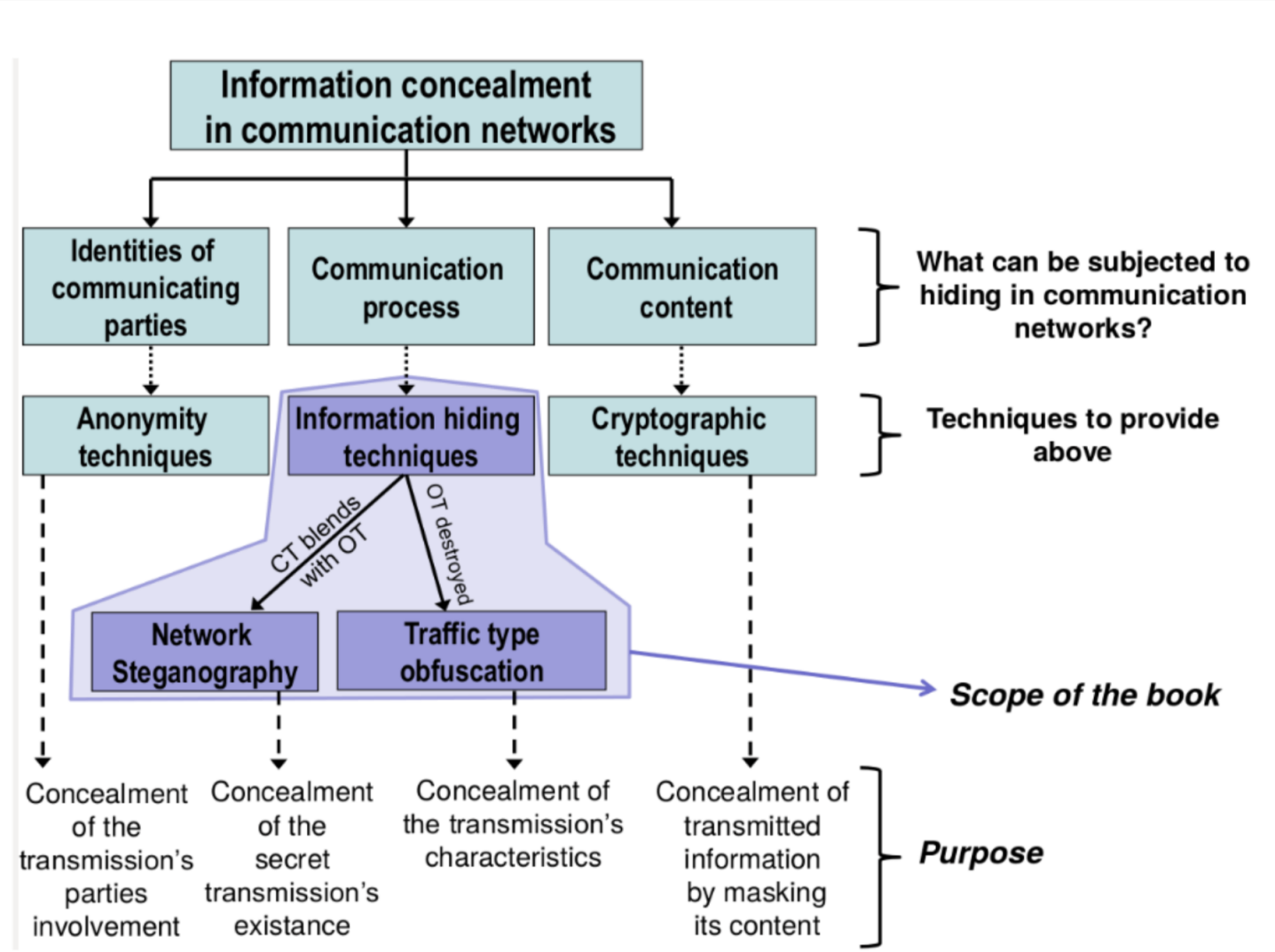
- Kemmerer's **Shared Resource Matrix**
  - Systematic way to investigate potential covert channels
  - Requires substantial knowledge about the semantics and implementation of system operations.

RESOURCE ATTRIBUTE \ PRIMITIVE		WRITE FILE	READ FILE	LOCK FILE	UNLOCK FILE	OPEN FILE	CLOSE FILE	FILE LOCKED	FILE OPENED	PROCESS SLEEP
PROCESS	ID									
	ACCESS RIGHTS	R	R	R	R	R	R	R	R	
	BUFFER	R	R,M							
FILES	ID									
	SECURITY CLASSES	R	R	R	R	R	R	R	R	
	LOCKED BY	R	R	R,M	R	R	R	R	R	
	LOCKED	R	R	R,M	R,M	R	R	R	R	
	IN-USE SET	R	R	R	R	R,M	R,M	R	R	
	VALUE	R,M	R							
CURRENT PROCESS		R	R	R	R	R	R	R	R	R,M
SYSTEM CLOCK		R	R	R	R	R	R	R	R	R

Kemmerer 1983, "Shared Resource Matrix Methodology: An Approach to Identifying Storage and Timing Channels"

Network information hiding

Information hiding in the network



Information Hiding in Communication Networks, W. Mazurczyk et al., Wiley 2016

Network information hiding

Network covert channels

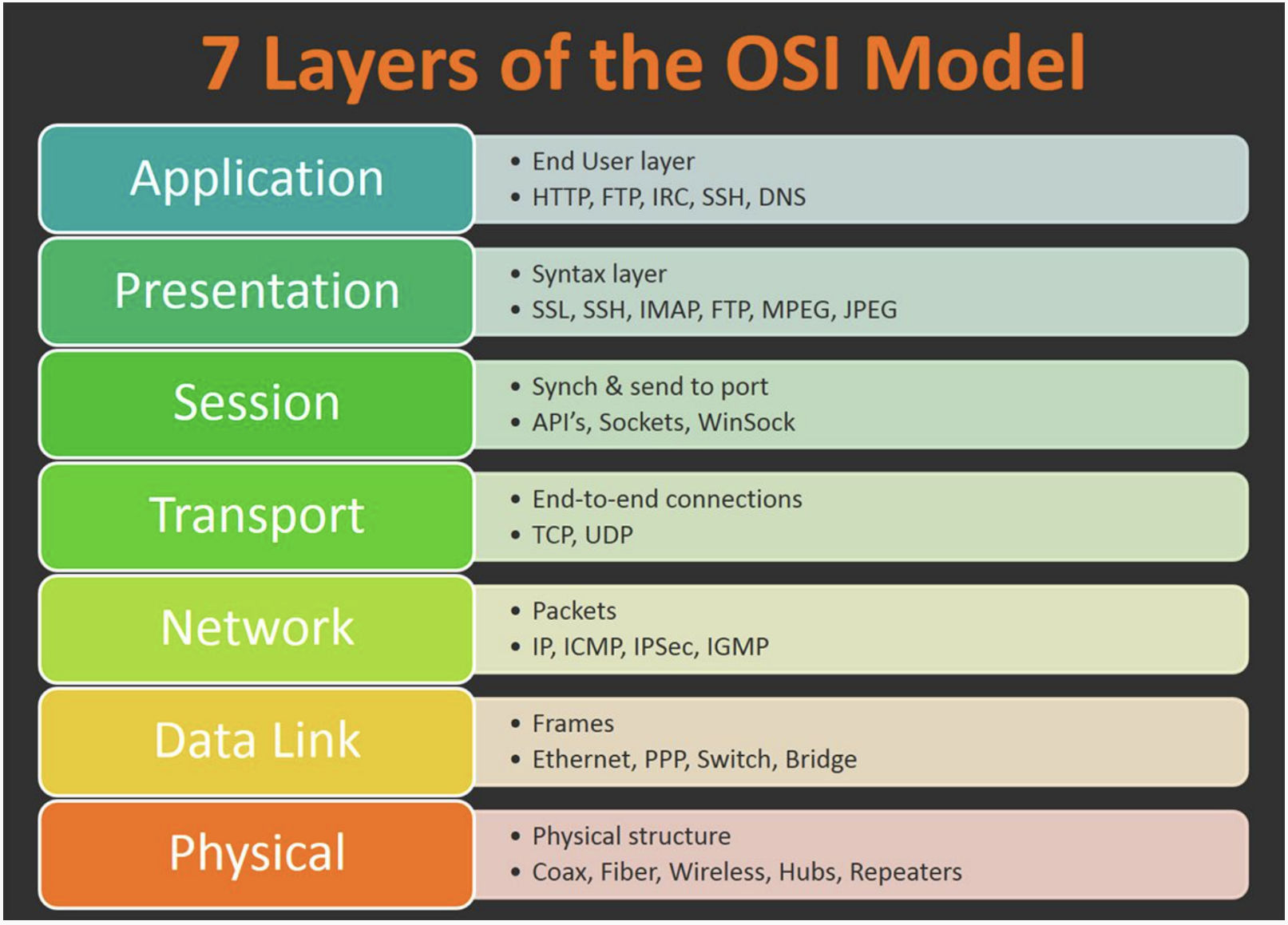
How do we create a network covert channel?

- **Storage**
  - e.g., packet header manipulations
- **Timing**
  - e.g., timing between packets

- What about steganography? We may say that steganographic methods are used to create a network covert channel
- In a network covert channel:
  - Covert data is hidden in overt network transmissions
  - The "cover" medium is now called a "carrier"

OSI Layers

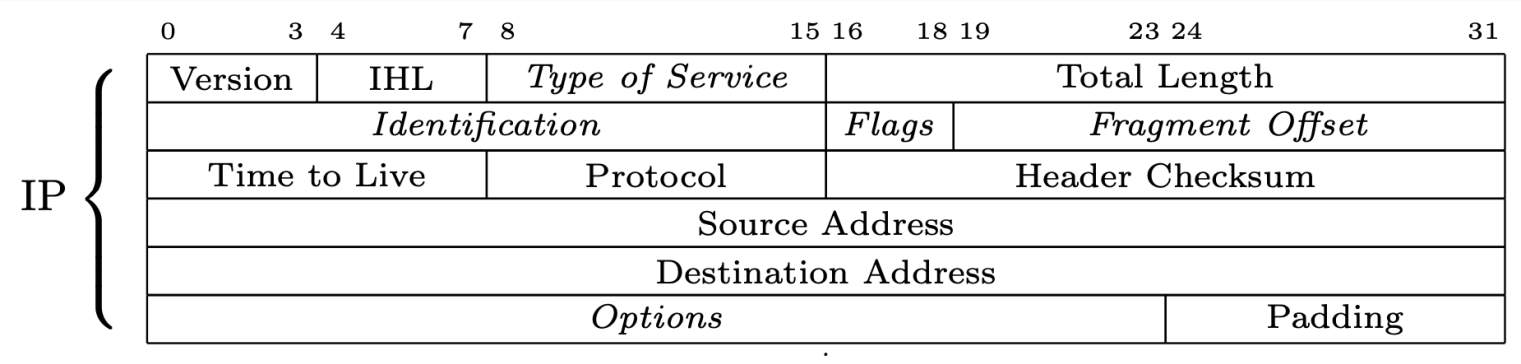
- We can implement covert channels across the OSI stack



Covert storage channels on TCP/IP

- TCP/IP packets have headers that provide extra information
  - Headers have different fields that are optional or disregarded in usual transmissions
- These fields can be used for hiding information!
  - IP identification
  - Offset
  - Options
  - TCP Checksum
  - TCP Sequence Numbers

IP header

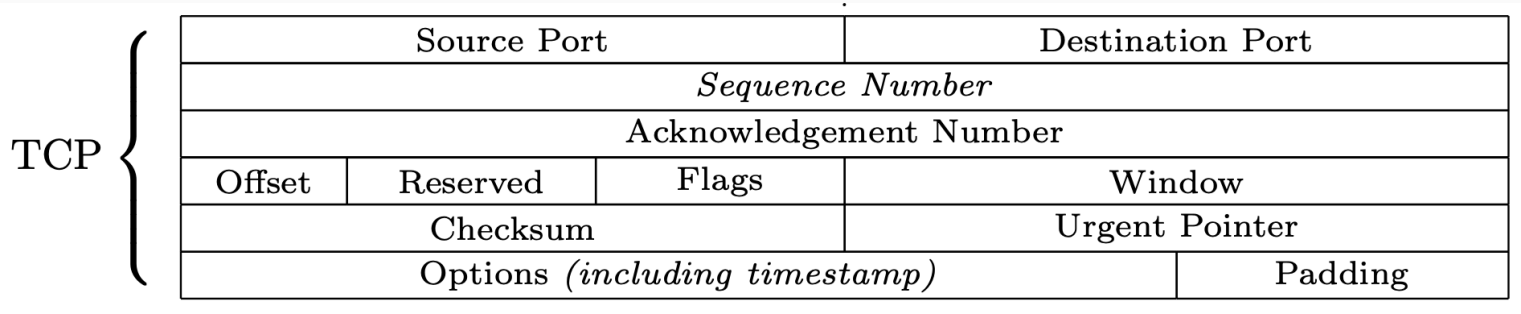

0 3 4 7 8 15 16 18 19 23 24 31
IP
Version
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding

Covert storage channels on IP

- IP ID: a value assigned by the sender to aid in assembling a packet's fragments
- Detection approaches:
  - OpenBSD toggles the most significant bit of the IP ID every 3 minutes or 30;000 IP IDs, so the MSB can be examined to check if it matches this pattern.
  - Within a rekey interval, the OpenBSD IP ID is nonrepeating

Embedding Covert Channels into TCP/IP, Murdoch and Lewis, International Workshop on Information Hiding, 2005

TCP header


TCP
Source Port
Destination Port
Sequence Number
Acknowledgement Number
Offset
Reserved
Flags
Window
Checksum
Urgent Pointer
Options (including timestamp)
Padding

Covert storage channels on TCP/IP

- TCP ISN: initial sequence number on TCP connections

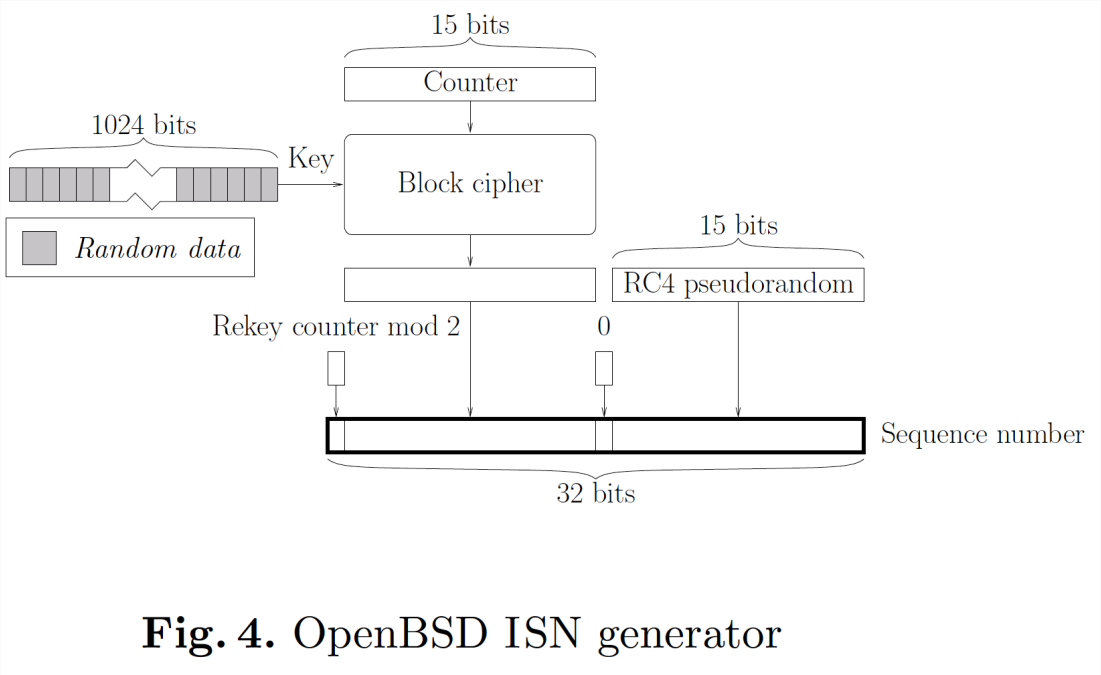


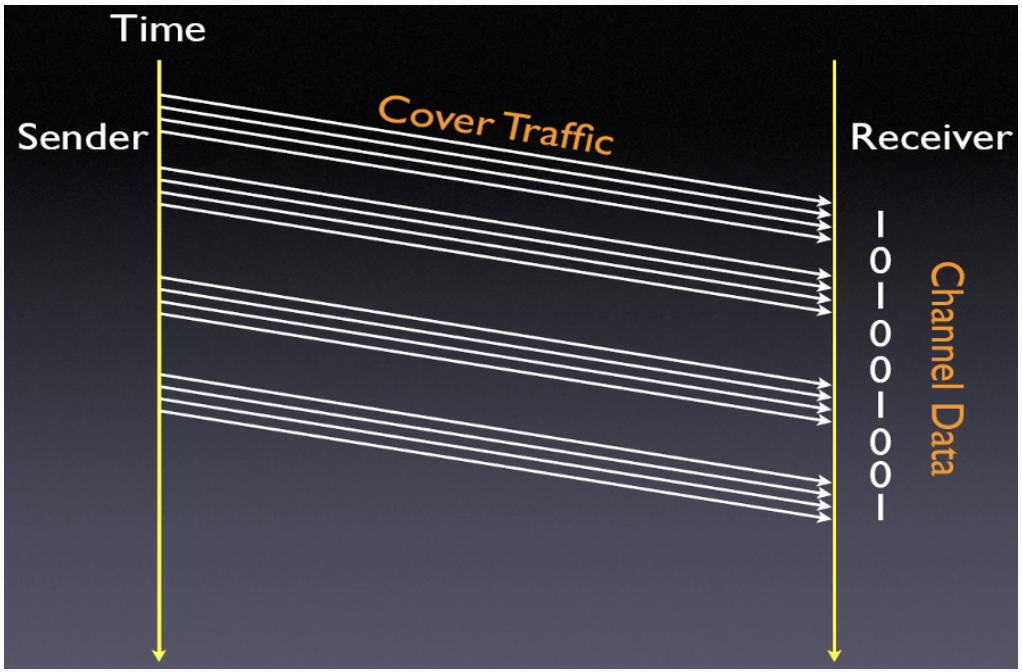
Fig. 4. OpenBSD ISN generator

- Several constraints make steganography easily detectable

Embedding Covert Channels into TCP/IP, Murdoch and Lewis, International Workshop on Information Hiding, 2005

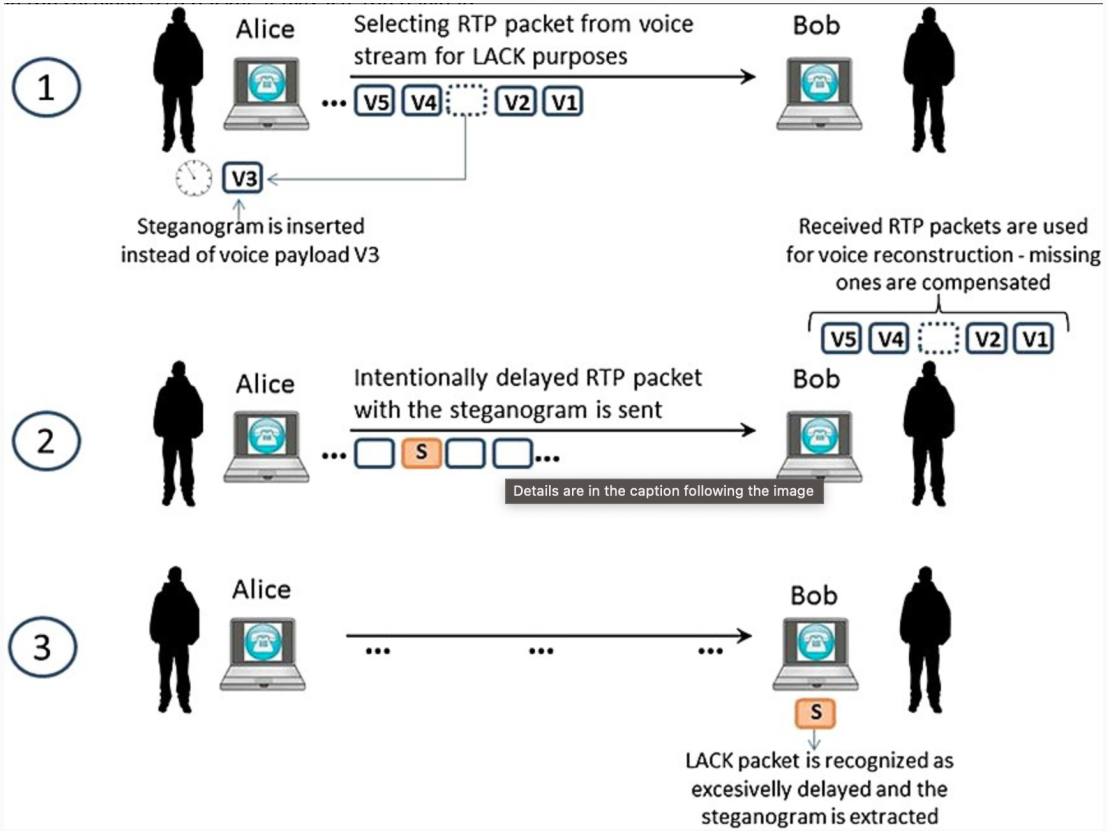
Covert timing channels on TCP/IP

- These typically propagate covert information by crafting delays between certain events
  - e.g., modify usual inter-packet delay, introduce intentional losses by skipping sequence numbers



Covert storage & timing channels on TCP/IP

- We may also have hybrids of storage and timing (e.g., LACK)
  - Replace encrypted packet contents with covert data and use timing delays for signalling the receiver about specific packets



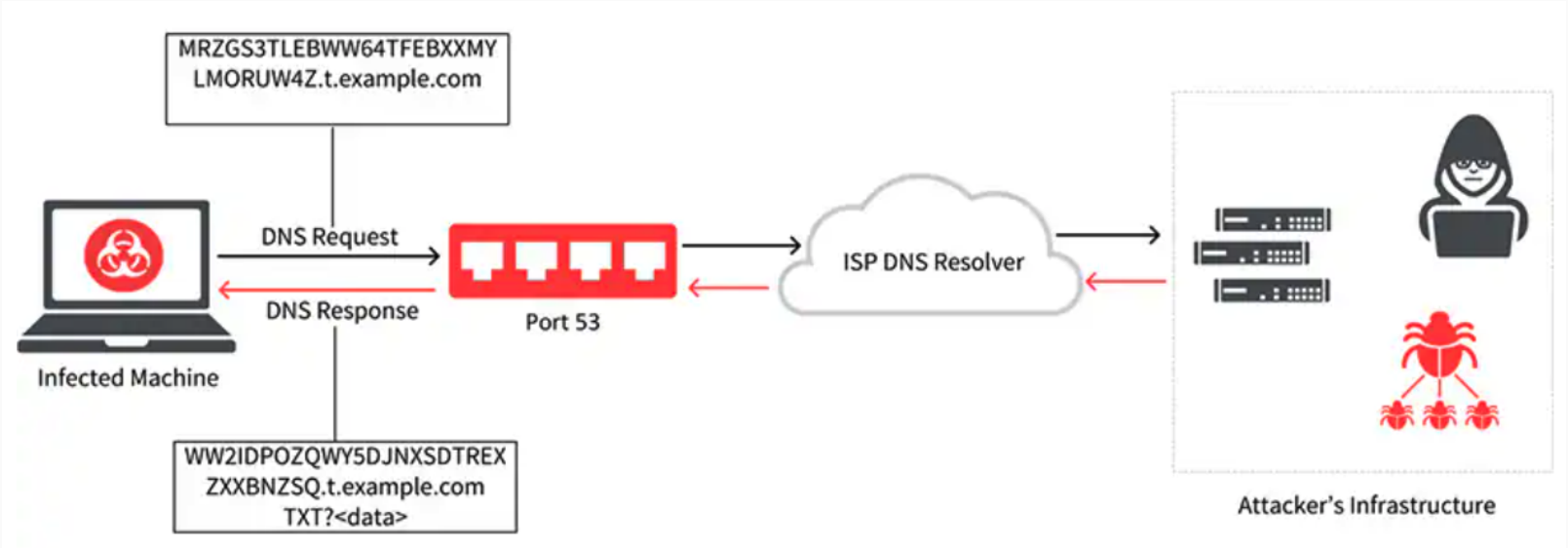
Lost audio packets steganography: the first practical evaluation, Mazurczyk, Security Comm. Networks, 5

Covert channels at the application level

- Many examples:
  - HTTP
  - DNS
  - Games
  - VoIP/video traffic
  - Push notifications
  - ...

Example: DNS Tunneling

- **DNS Tunneling** is based on encoding the data of other programs or protocols in DNS queries and responses



Example: Games

- We can create covert channels by encoding information in games' virtual worlds which are shared by multiple users

How to detect/prevent network covert channels

- A **warden** inspects (and/or manipulates) traffic to detect (and/or break) covert channels
- Storage channels
  - Passive: Analyze transmitted data for anomalies.
  - Active: Normalize data in header fields
- Timing channels
  - Passive: Analyze packet timing for inconsistencies
  - Active: Shape traffic (e.g., constant rate)

Network information hiding

Traffic obfuscation

Information concealment in communication networks

- Timing and content anomalies may be an effective way to detect covert channels
- Are there better ways to hide the existence of covert data transmissions?

Information concealment in communication networks

- Timing and content anomalies may be an effective way to detect covert channels
- Are there better ways to hide the existence of covert data transmissions?

Well, there are!

- **Traffic obfuscation**: Hide the true characteristics of a covert data transmission by shaping the "look" of data exchanges
  - e.g., used to hide malware communication with a C&C server, shroud terrorist activities, etc.

Different techniques for traffic obfuscation

- **Randomize traffic**: Don't look like any particular protocol
- **Mimic traffic**: Attempt to look like some other protocol
- **Tunnel traffic**: Piggyback on another protocol's execution

Traffic randomization

- Network intrusion detection systems are typically configured to look for signatures of well-known protocols (e.g., ssh, Tor, etc.)
- **Idea**: evade inspection by generating traffic that does not conform to any known protocol specification
  - Randomize packet sizes and timings
  - Randomize packet contents (no signatures)
- Examples:
  - Shadowsocks
  - V2Ray
  - OutlineVPN

Issues with traffic randomization systems

- "Look-like-nothing" might be a signature in itself
- Does not work if wardens have protocol allowlists in place
- Can be detected through cryptographic flaws and entropy tests

Security Notions for Fully Encrypted Protocols, Fenske and Johnson, FOCI 2023
How the Great Firewall of China Detects and Blocks Fully Encrypted Traffic, Wu et al., USENIX Security 2023

Traffic mimicking

- **Idea**: Hide a protocol's execution by mimicking another innocuous protocol's characteristics (e.g., Skype)
- Leverage steganography or encrypted carrier protocols
  - Embed covert data in specific protocol fields
  - Mimic how an encrypted cover protocol sends its traffic
- Examples:
  - SkypeMorph
  - StegoTorus
  - CensorSpoofer

Issues with traffic mimicking systems

- It is very difficult to build a perfect imitation
  - Respond to network perturbations
  - Cover all corner cases and error conditions (and bugs!)
  - Mimic relationships between sub-protocols
  - Keep up with the cover protocol's updates
- Now believed to be a fundamentally flawed approach

The Parrot is Dead: Observing Unobservable Network Communications, Houmansadr et al., S&P 2013

Traffic tunneling

- **Idea**: Piggyback covert data on the execution of a protocol
- Send covert data as the protocol's application messages
  - Avoids mimicking issues
  - Still needs to ensure the cover protocol does not generate "weird" traffic patterns
- Examples:
  - VoIP/video: FreeWave, DeltaShaper, Protozoa
  - HTTPS: meek, decoy routing, Balboa
  - IM/e-mail: Camoufler, SWEET
  - Cellphones: Dolphin
  - ...

Issues with traffic tunneling systems

- Oftentimes, there is a disconnect between the usage patterns of the cover protocol and the covert protocol
  - Times of use, duration, etc.
- The "greedy" tunneling of covert data may change the cover protocol's typical traffic patterns
  - e.g., exchaging very large IMs very frequently on both directions
- Covert data embedding mechanisms may slow down the cover's protocol activity, leading to noticeable changes in traffic patterns
  - e.g., when replacing media data with covert content

Takeaways

- Covert channels allow for the surreptitious transfer of information, both within processes of a given machine or across machines
- Network covert channels are increasingly hard to detect, but can also be used for noble purposes (e.g., censorship evasion within repressive environments)

Pointers

- **Textbook:**
  - Mazurczyk – Chapters 2, 3, 5, 7, Johnson – Appendix A
- **Other resources:**
  - Embedding Covert Channels into TCP/IP. Murdoch and Lewis, International Workshop on Information Hiding, 2005