VU Network Security Advanced
Lecture 06

Hidden Communication

Tanja Zseby
TU Wien

WS 2018/19

institute of telecommunications

TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Summary from Last Lecture

- Smart Grid security
  - AMI
  - WAMS

- Exam: Please register in TISS !!

- ! Lab Introduction:
  - only on Wednesday 28.11.
  - Room EI4!

Hidden Communication

Hidden Communication

- Steganography
  - Intentionally hiding information in content
- Covert Channels
  - Hiding information in the transmission (e.g., network protocol)
  - “Network Steganography”
- Subliminal Channels
  - Hiding information in crypto systems (e.g., signature)

- Encryption
  - ***Hides content*** against observers
- Steganography, Covert Channels, Subliminal channels
  - ***Hides*** existence of ***communication channel***

Steganography

- “concealed writing”
- Concealing the existence of a messages
- Information hidden in other (not suspicious looking) content

odd odd odd even even

The attackers had no luck

even odd odd odd

We are all okay!

. . . - - - . . .

S O S

Null Cipher

```
Apparently neutral's protest is
thoroughly discounted and ignored. Isman
hard hit. Blockade issue affects pretext
for embargo on byproducts, ejecting suets
and vegetable oils.
```

Source: Kahn, *The Codebreakers*, The Macmillan Company. New York, NY 1967.

Null Cipher

Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on byproducts, ejecting suets and vegetable oils.

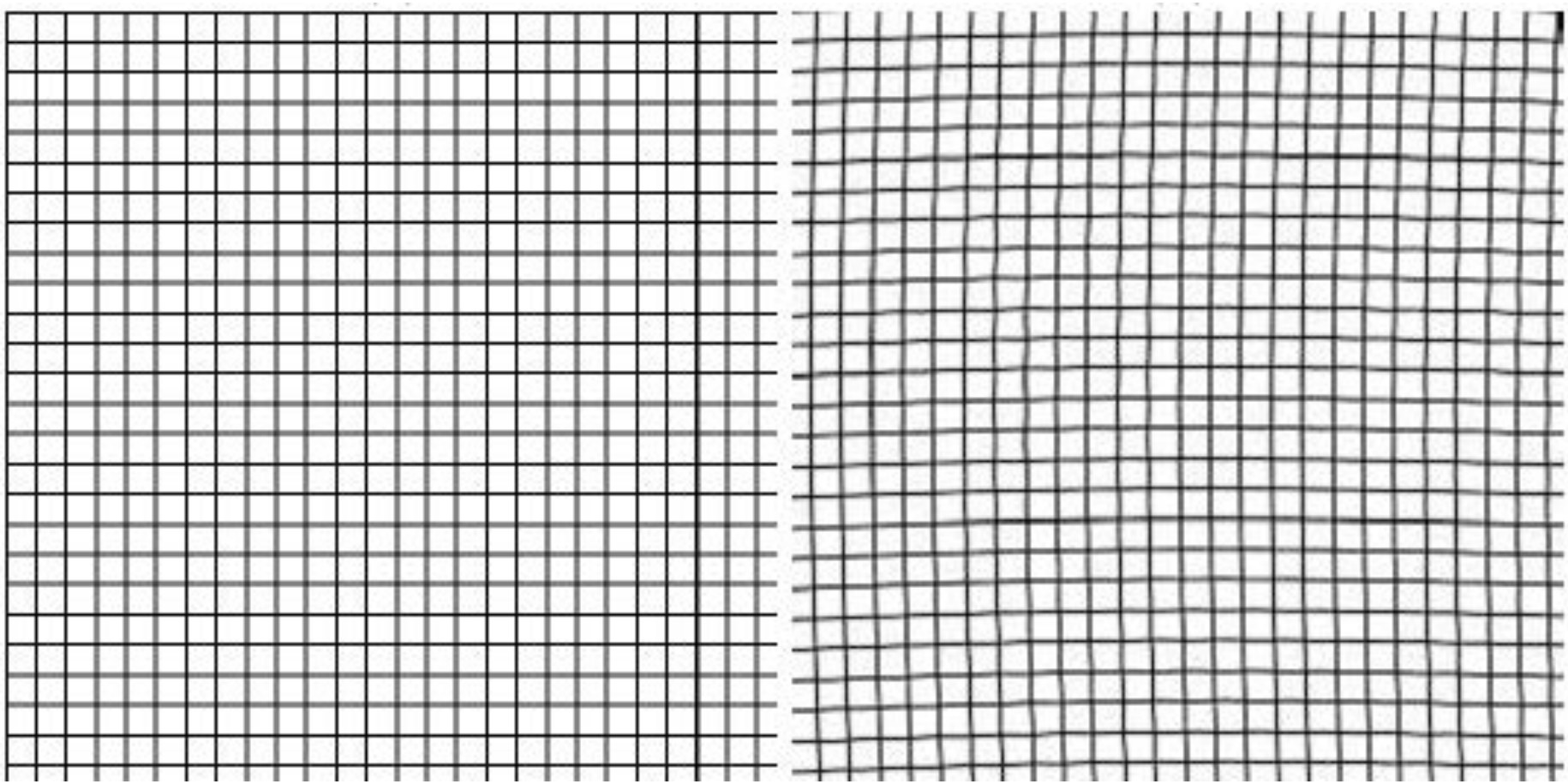
➔ Pershing sails from NY june 1

Source: Kahn, *The Codebreakers*, The Macmillan Company. New York, NY 1967.

Steganography

Source: Petitcolas, et al. “Information hiding-a survey,” Proceedings of the IEEE, vol. 87, no. 7, Jul. 1999.

Steganography



Source: Petitcolas, et al. “Information hiding-a survey,” Proceedings of the IEEE, vol. 87, no. 7, Jul. 1999.

Everything looks Suspicious

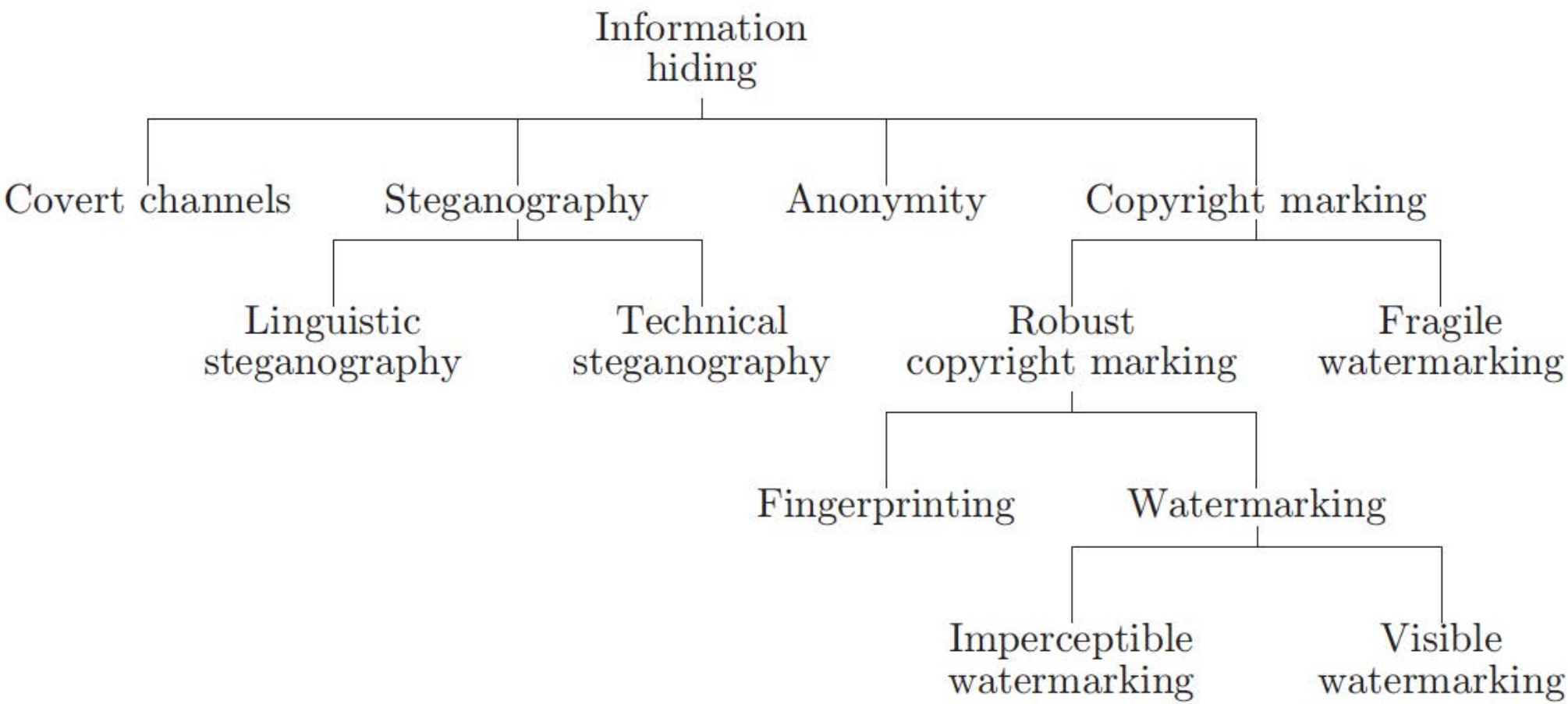
XOX

Still Important: Kerckhoffs's Principle

- Kerckhoffs's Principle

  Message must remain confidential even if *data-hiding technique* and encryption method are *known*
- Solution
  - Hiding ciphertext, not plaintext
  - Secrecy relies only in ***secret key***
  - Signal looks random even if hiding method known
  - Observer cannot decrypt content
- Why then steganography?
  - Encrypted communication may not be allowed
  - Conceal that communication takes place at all
  - Observer cannot prove that communication exists

Information Hiding



Source: Petitcolas, et al. “Information hiding-a survey,” Proceedings of the IEEE, vol. 87, no. 7, Jul. 1999.

Who wants to hide Communication?

- Hackers
  - Malware distribution (e.g., worm spreading)
  - Convey sensitive information (e.g., Trojan horses)
  - Hide command structures (e.g. Botnet C&C)
- Criminals
  - Hide communication structures and content
- Military
  - Hide communication structure and content
- Citizens
  - Evade Censorship
  - Prevent pervasive monitoring
- Network administrators
  - Hide network management operations

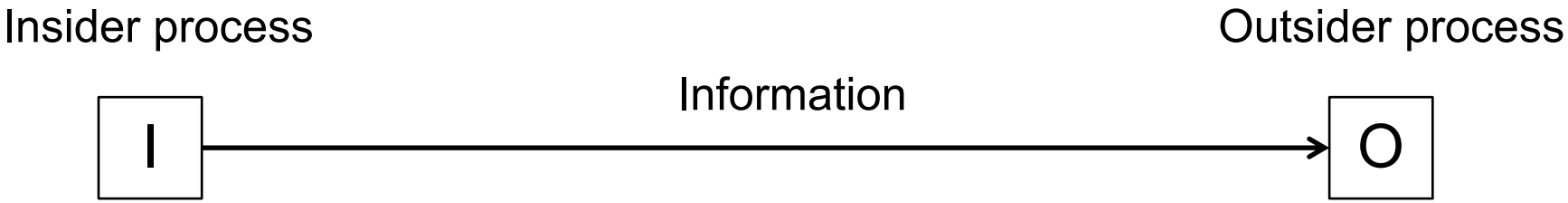
Side Channels vs. Covert Channels

- Side Channel
  - ***Unintentional*** information leakage due to implementation of algorithms
- Side Channel Attacks
  - Cryptanalysis based on side channel observations
- Covert Channels
  - ***Intentional*** use of a channel not intended for communication
  - Hiding communication
  - Conceal the existence of a message

- Side channels may be used as covert channels

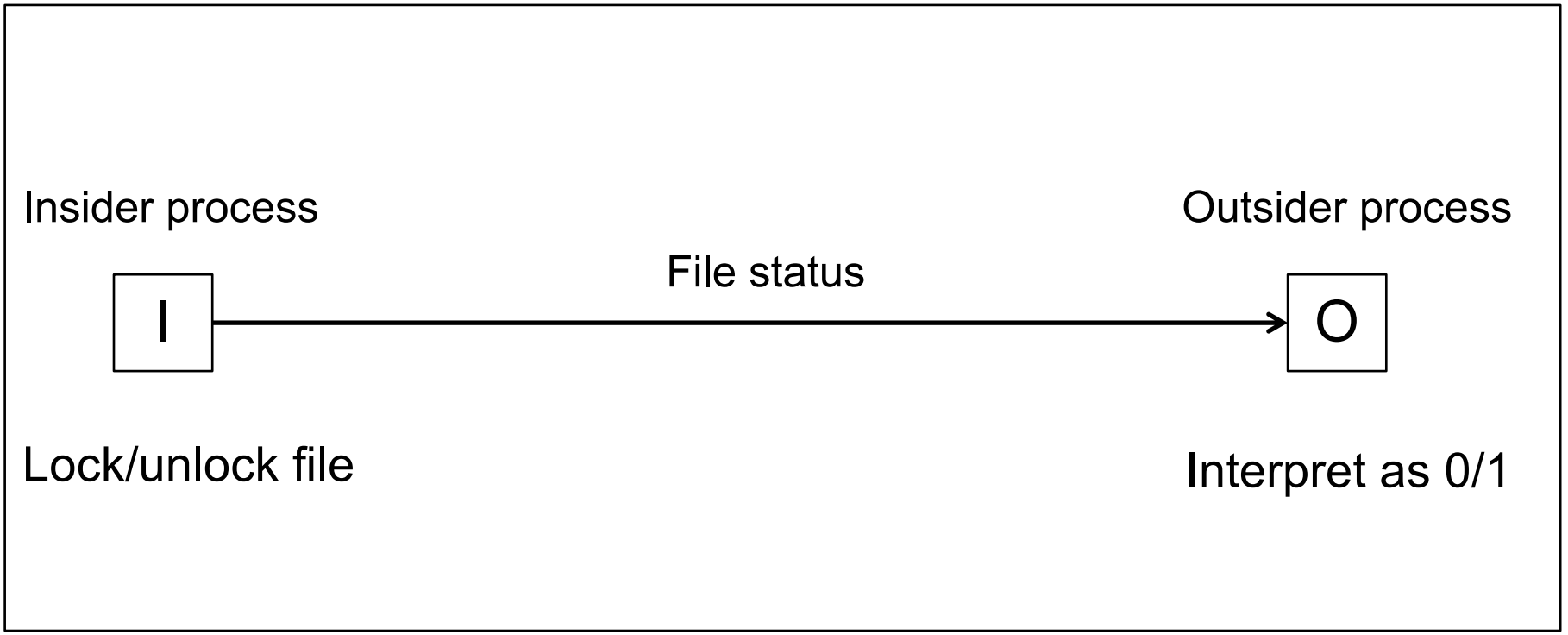
Covert Channels

Covert Channel

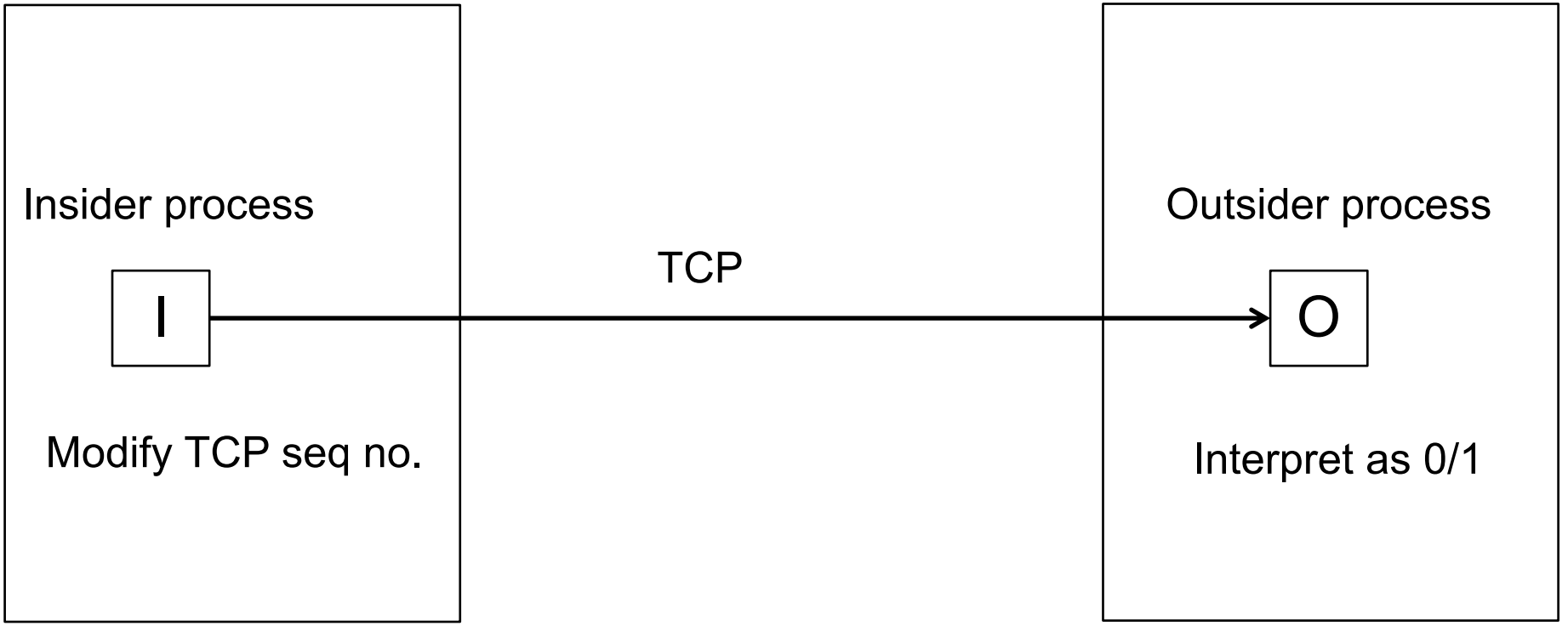


Intentionally establish channel for stealth communication

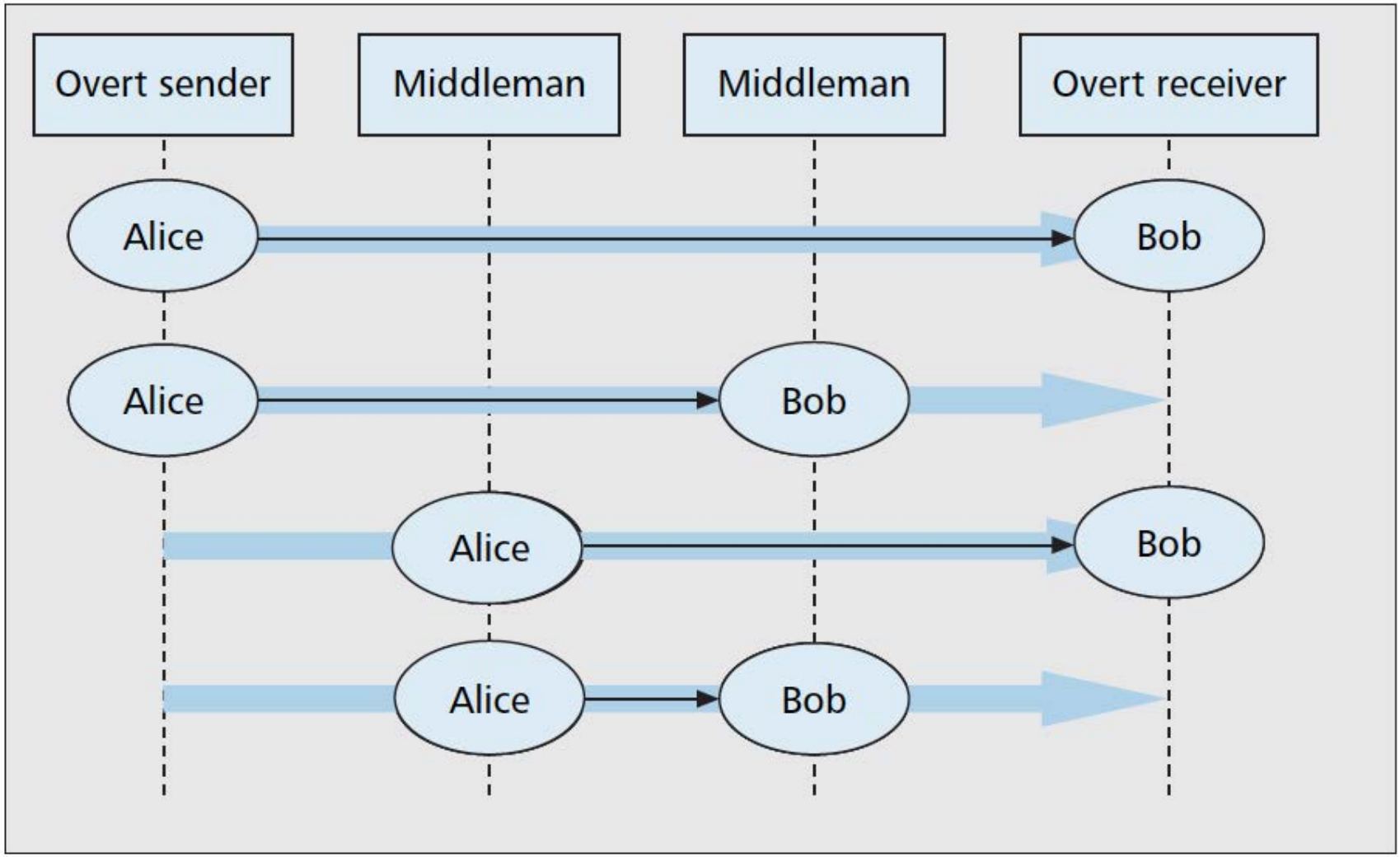
Example (inside a single host)



Example (beween hosts)

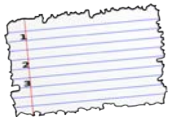


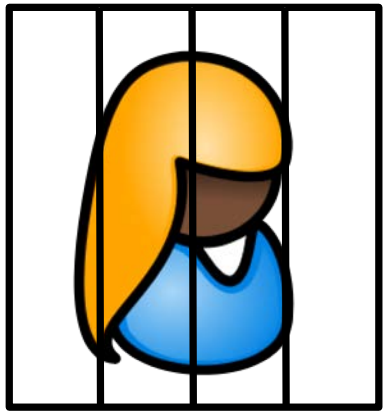
Covert Channel Possibilities

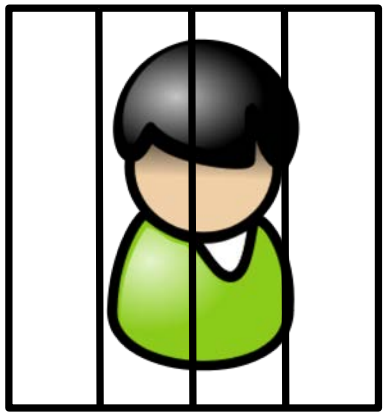


Source: Zander, et al. “A survey of covert channels and countermeasures in computer network protocols,” IEEE Communications Surveys Tutorials, vol. 9, no. 3, 2007.

Warden Model

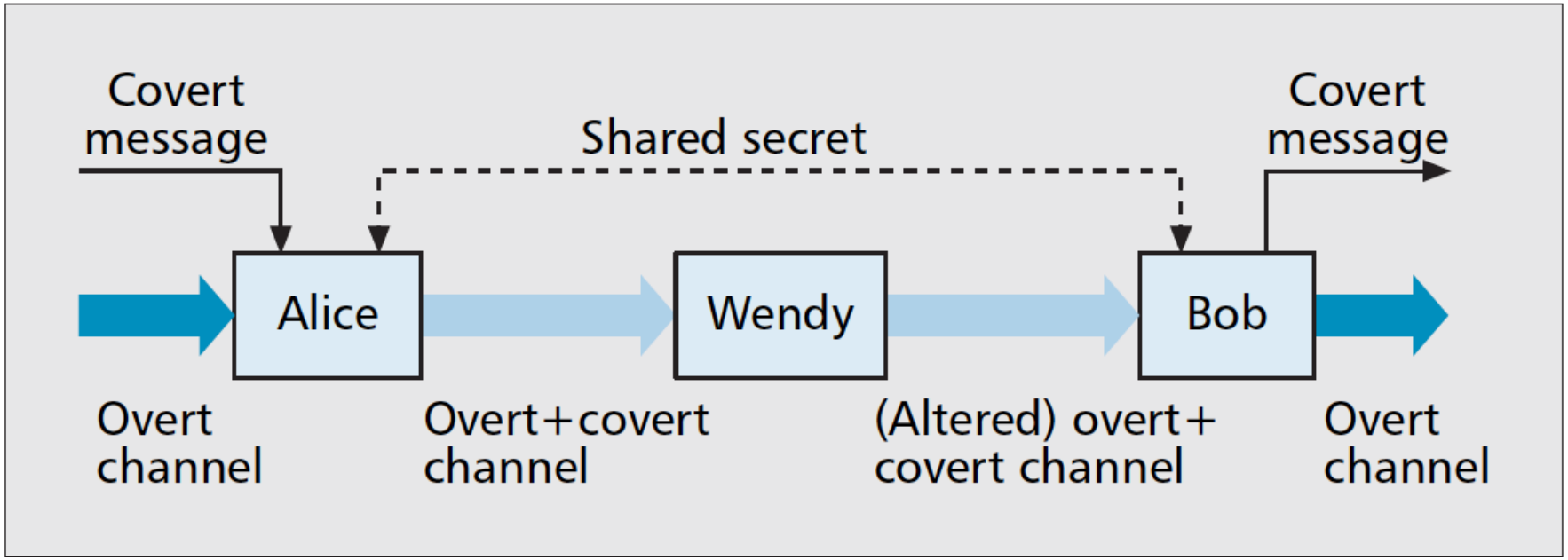






Simmons, “The Prisoners’ Problem and the Subliminal Channel,” in Advances of Cryptology, 1983.

Warden Model



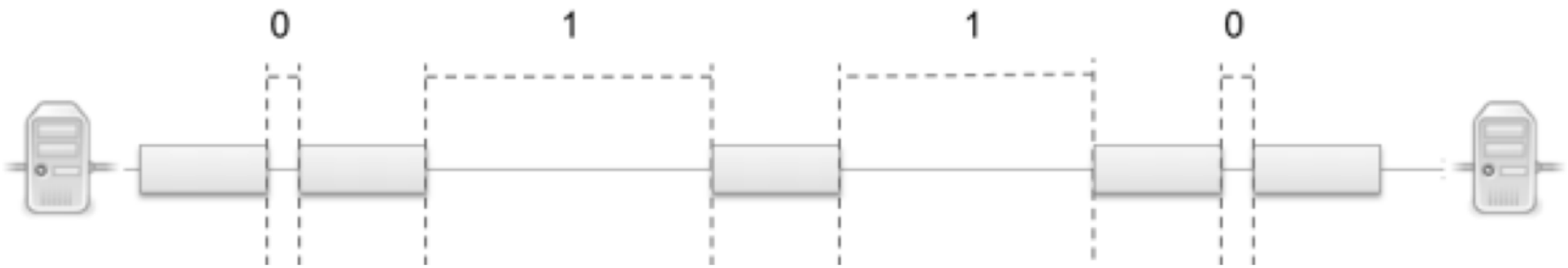
- Warden on the path
  - May inspect messages (passive warden)
  - May alter message (active warden)

Source: Zander, et al. “A survey of covert channels and countermeasures in computer network protocols,” IEEE Communications Surveys Tutorials, vol. 9, no. 3, 2007.

Covert Channels in Network Protocols

Covert Channels in Network Protocols

- Hiding information in network protocols
  - Header field modification
  - Varying packet sizes
  - Timing Variations
  - Influencing packet loss, packet order



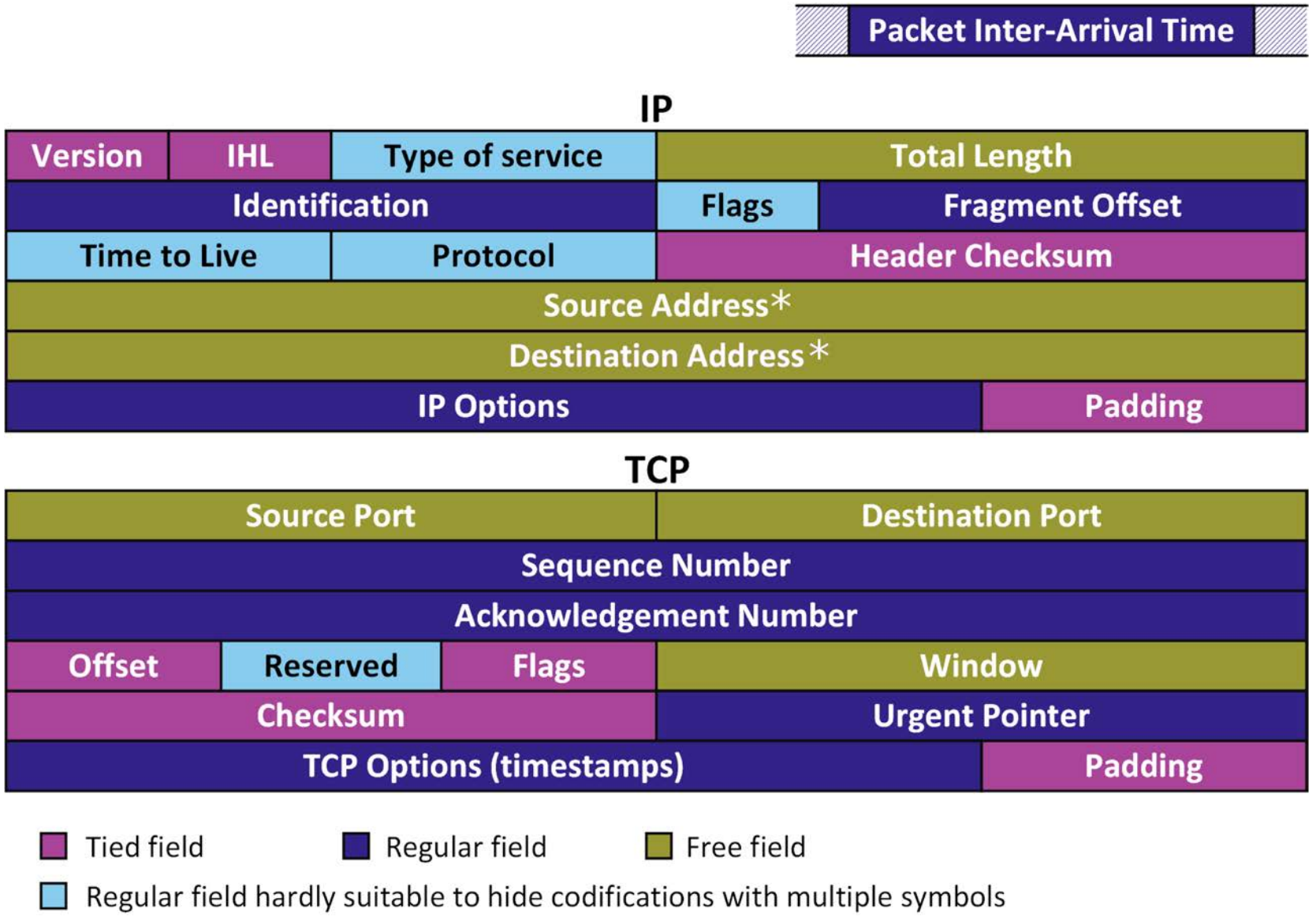
Covert Channels in Network Protocols

- Cover Channels possible at all layers, but
- Physical/Link layer ➔ not routed
  - Communication partner needs to be on same LAN
  - Often access to Hardware required
- Application Layer ➔ limited infrastructure
  - Skype, Gaming, P2P Applications
  - Infrastructure exist but not everywhere
  - Firewalls/Filters may block specific applications
- TCP/IP ➔ broadly deployed
  - Infrastructure throughout whole Internet
  - Used in many different environments (cloud, smart grid,car2car,…)

Why do Covert Channels Exist?

- Several header fields can have different values
  - Which all look plausible (e.g., to a warden)
  - Padding with arbitrary bits
  - Random numbers in protocols
- Some fields less suitable
  - Fields with predictable outcome (due to context)
  - Fields that change on the path
- Bandwidth of a covert channel
  - Number of bits usable for transmitting hidden information
- Task for warden
  - Distinguish whether header generated by unmodified TCP/IP stack or by covert channel

TCP/IP Covert Channels



TCP/IP Covert Channels

- Tied fields ➔ easy to detect misuse
  - Fixed values (version, padding)
  - Fields with specific meaning for traffic configuration (TCP offset, flags)
  - Inferred from other fields (IHL, Checksums)
- Regular Fields ➔ may be used, but
  - Common values (Options usually not used)
  - Behavioral patterns (sequence numbers, ID)
- Regular Fields with restricted range ➔ low capacity
  - Few common values (Protocol, ToS, flags)
  - Few common ranges (TTL)
- Free fields ➔ hard to detect misuse
  - Free to choose by sender (packet length, addresses, ports)

Fragmentation Fields

- Identification (16 bits)
  - Arbitrary value ➔ may be suitable
  - But not always random (different algorithms)
- Flags (3 bits) ➔ predictable from context
  - Don't Fragment (DF), More Fragments (MF)
  - If datagram ≤ MTU: setting DF has no effect
- Fragment Offset ➔ predictable from context
  - Needs to fit to other fragments
- And: Fragmentation is rare
  - 0.67% [ShannM02], 0.06% [JohnT07]
  - 91.3% DF set [JohnT07]

[ShannM02] Shannon, Moore, Claffy, "Beyond folklore: observations on fragmented traffic,"IEEE/ACM Transactions on Networking, vol. 10, no. 6, pp. 709–20, 2002.
[JohnT07] John, Tafvelin: Analysis of internet backbone traffic and header anomalies observed. *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement* (IMC '07), 2007.

IP ID Calculation Possibilities

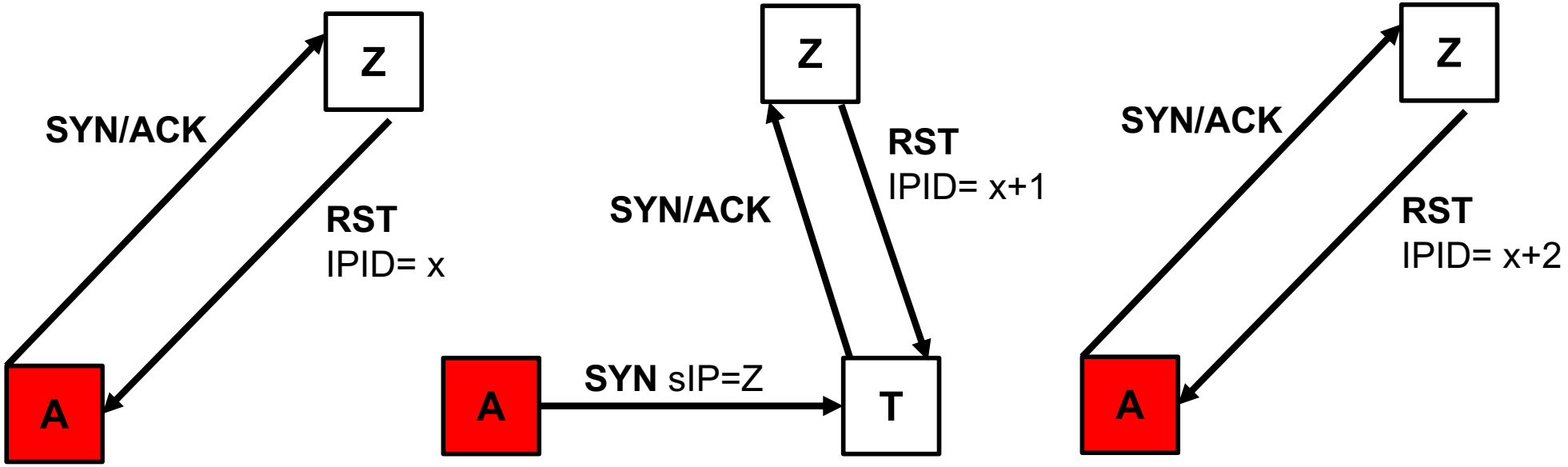
- Sequential Global ID
  - Global counter
  - IDs to different hosts are increasing
- Sequential per-host ID
  - Counter per host
  - IDs to same host are increasing
  - IDs to different hosts unrelated

- But: IP ID should not be predictable

Source: Murdoch, Lewis “Embedding Covert Channels into TCP/IP,” in Proceedings of the 7th International Conference on Information Hiding, Berlin, Heidelberg, 2005, pp. 247–261.

Why IP ID should not be predictable

- Idle Scan
  - Scan for open ports on a victim
  - Without sending packets with real source IP to target
    ➔ no trace at target



Attacker checks current IPID at **Zombie**
Zombie didn't expect SYN/ACK
➔ sends RST

Attacker sends SYN to **Target** with spoofed sIP (of Zombie)
If port open: Target sends SYN/ACK to Zombie (spoofed srcIP)
Zombie sends RST

Attacker sends SYN/ACK to Zombie to check IPID
If IPID = x+2 ➔ port at target open

IP ID Calculation: Desired Features

- Unpredictable
  - Include random data in calculation
- Unique
  - Unique within given time interval
  - Prevent collisions (packets with same ID)
- Simple to generate
  - But: Generation of real random data high effort
  - Use key as seed for pseudo random generator
- Unpredictability of IP ID
  - Prevents idle scanning attacks
  - But: increases usability as covert channel

New Proposed Standard (RFC6864, 2013)

RFC 791, RFC1122: demand ***uniqueness*** of IPID for

- all packets in a flow (sIP,dIP,sPort,dPort, protocol)
- max lifetime of packet

- But: IPID is16 bit ➔ uniqueness usually does not hold
- But: Some devices keep IPID constant (cell phones)

RFC6864: Use IP ID only in packets that

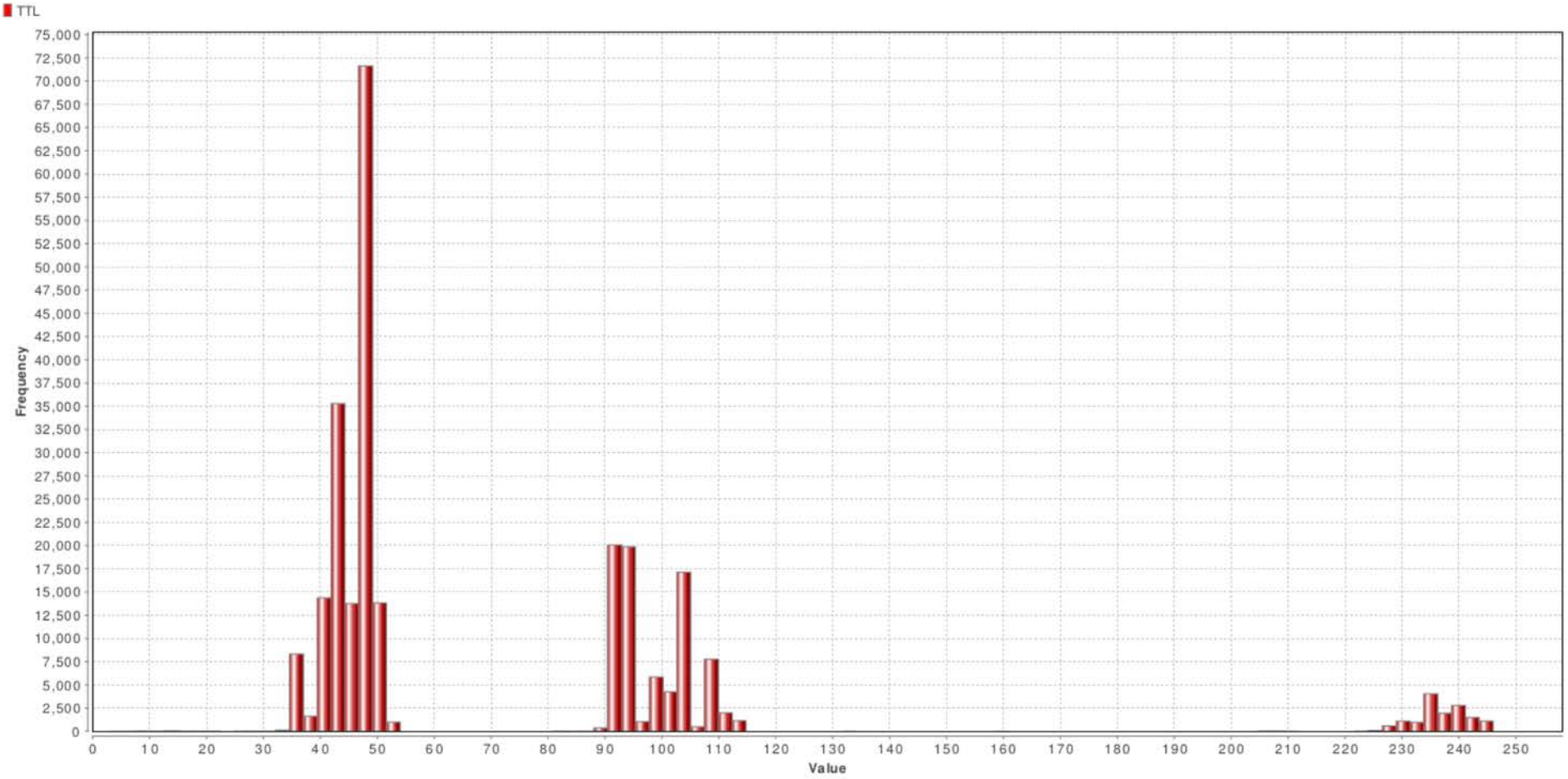
- are fragmented or
- can get fragmented

- Don't use IP ID Field in "atomic datagrams"
  - Atomic: (DF==1)&&(MF==0)&&(frag_offset==0)
  - Source MAY set ID field to ***any value*** [RFC6864]
  - Perfect for covert channels

Time-to-Live (TTL)

- Decreased by each router
  - Small variations on path
- Initial value depends on operating system
  - Typical initial values: 64,128, 255
- Potential initial value known
  - Value range on path and at receiver predictable

TTL Distribution



Using TTL as Covert Channel

- Values should be in typical range
  - Typical for operating system
  - If warden close to sender ➔ values close to initial value expected

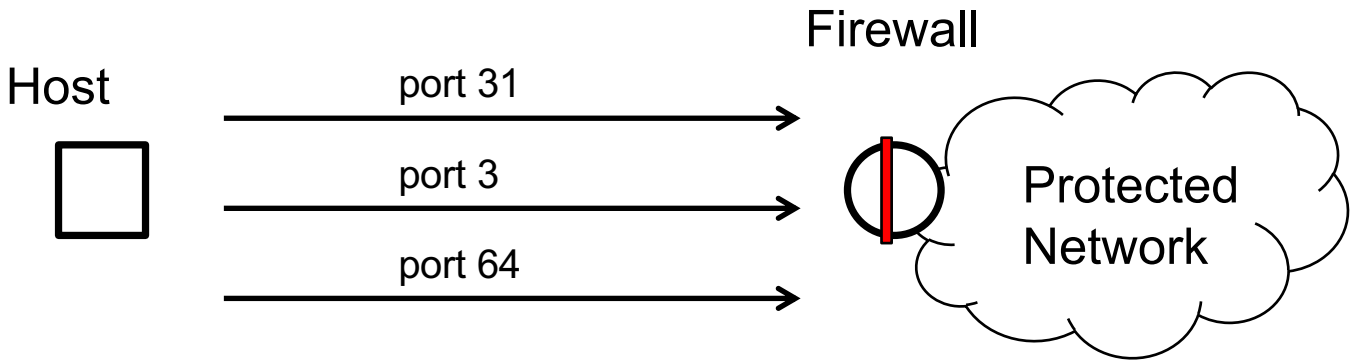
- Final distribution should look similar
  - Typical variation on the path

- If used as middleman
  - High increase ➔ packets may loop
  - High decrease ➔ packets may be discarded

S. Zander, G. Armitage, and P. Branch, “An Empirical Evaluation of IP Time To Live Covert Channels,” in *15th IEEE International Conference on Networks, 2007. ICON 2007*, 2007, pp. 42–47

Port Numbers

- Source and destination port numbers
  - Port numbers observable on host
- Simple Example: port knocking
  - Firewall monitors sequence of ports in requests
  - Used to enable services for machine that “knocks”



- But: security by obscurity
  - Log file contains the “key”
  - Easy to crack with replay attack

TCP Header Fields

- TCP Initial sequence number (ISN)
  - Should be random
  - But some constraints (overlapping, uniqueness)
  - Suited for covert channels

- Used in tools: Covert_TCP, Nushu

TCP ISN Generation

- RFC 793: prevent repeating ISNs
  - global 32-bit counter
  - incremented by 1 roughly every 4 microseconds
  - ➔ But: predictable
- RFC 6528: unpredictable ISN
  - Separate counter per socket (srcIP, srcPort, destIP, destPort)
  - Used as input for a pseudorandom function (PRF)
  - Timer value added (4 microsecond timer) ➔ M

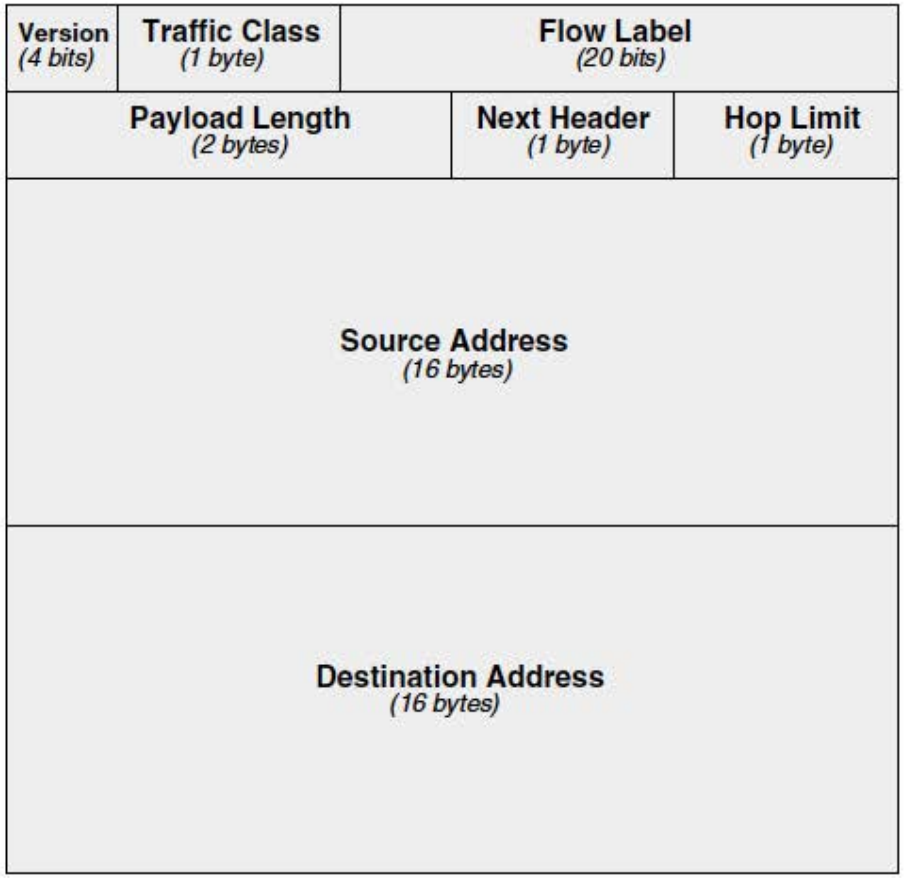
ISN = M + PRF(srcIP, srcPort, destIP, destPort, secretkey)

Using TCP ISN as Covert Channel

- Assumptions for Warden
  - Knows operating system used
  - Knows TCP ISN generation method
  - Can check for unusual distributions
- Unpredictable ISNs
  - Bad for idle scanning attacks
  - But: good for hiding covert channel

- Goal (of attacker): distribution of values for TCP ISN with CC should be ***indistinguishable*** from original TCP ISN
  - Mimic output of original TCP ISN generation

Covert Channels in IPv6



ID	Field	Covert Channel	Bandwidth
α	*Traffic Class*	Set a false traffic class	8 bits/packet
β	*Flow Label*	Set a false flow label	20 bits/packet
γ	*Payload Length*	Increase value to insert extra data[3]	Varies
δ	*Next Header*	Set a valid value to add an extra extension header[3]	Varies
ϵ	*Hop Limit*	Increase/decrease value	$\approx$ 1 bit/packet
ζ	*Source Address*	Set a false source address	16 bytes/packet

AND: Extension Headers!

Source: N. Lucena, G. Lewandowski, S. Chapin: Covert Channels in IPv6

Suitable Fields for Covert Channels

- Fields not predictable from context
  - Not depending on packet content, size, previous packets etc.
  - Not depending on other header fields
- Goal: Similar Distributions regardless if
  - Field generated by standard TCP/IP stack
  - Field generated by covert channel
- Kerkhoffs' Principle: hiding ciphertext not plaintext
  - Ciphertext looks random
- Ideal: Random numbers in protocols
  - Standard field looks random
  - Ciphertext looks also random ➔ hard to detect

Additional Methods: Use Flow Characteristics

- Packet Sizes
  - Use different packet sizes to signal message
  - Need to correspond to content
  - Typical packet sizes
- Packet Rates
  - Signal bit by rate (on/off, high/low rate)
  - Low capacity channel
  - Need to fit to context (application, content,…)
  - Inter-packet delays
- Packet Loss
  - Requires protocol with sequence numbers
  - Omitting sequence numbers at sender (“phantom packets”)
- Packet Order
  - Example: send packets to different destinations (Increasing destinations ➔ 0, Decreasing destinations➔ 1)

Prevention of Covert Channels

- Host and Network Security
  - Prevent installation of malicious software
  - Block suspicious protocols (ICMP)
- Protocol Design
  - Check specification for potential CCs
  - Clearly define the use of fields
- Traffic normalization ➔ modify packets
  - Set unused bits to zero
  - Remove unkown header extensions
  - Rewrite fields (e.g. set IPID=0 if DF=1)

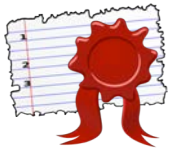
Source: Zander, et al. “A survey of covert channels and countermeasures in computer network protocols,” IEEE Communications Surveys Tutorials, vol. 9, no. 3, 2007.

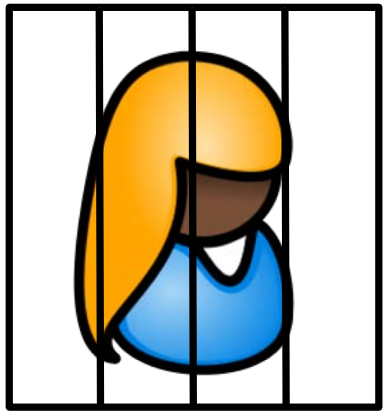
Subliminal Channels

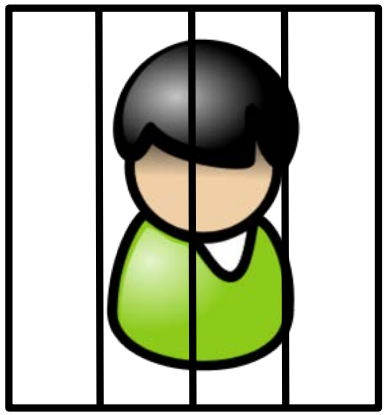
Subliminal Channels

- Cryptographic protocols
  - Often use random numbers
- Ideal for hiding information
  - Ciphertext looks random
  - Easy to hide ciphertext in random number
- Hard to detect even if hiding technique known
  - Distributions look similar
  - Decryption only possible with additional information (key)
- Subliminal Channel
  - A special form of covert channel
  - Exploit randomness used in crypto systems
  - Hide information in data exchange in crypto systems

Scenario: Message Authentication without Secrecy







Simmons, “The Prisoners’ Problem and the Subliminal Channel,” in Advances of Cryptology, 1983.

Message Authentication without Secrecy

- Message is ***authenticated***
  - Bob can verify that message was not changed on the way (e.g., by guard)
- Message is ***not secret***
  - Guard can inspect message content

- Authentication requires key
  - Otherwise Guard can forge message

- Guard suspects that key is used to transmit encrypted information

  ➔ Guard demands access to authentication key

Message Authentication without Secrecy

- How to provide key to guard?
- With symmetric key
  - Guard gets the key ***after*** Bob received message
  - Guard cannot alter the message
  - But can inspect content and verify if message authentication works with the key

- With asymmetric keys
  - Guard gets the ***public key***
  - Guard cannot alter the message (needs private key to generate valid signature)
  - But can inspect content and verify signature

A Historic Example

Vienna, June 18, 1979

Strategic Arms Limitation Talks (SALT II)

- Goal: Reduce nuclear weapons
  - in US and Soviet Union
- Technical challenge:
  - Allow Soviet Union any time to ***check amount*** of missiles in US (and vice versa)
- But: ***not revealing location*** of the missiles
  - Prevent that enemy knows which silos to attack
- Solution: Install ***tamper proof sensors*** at silos
  - Sensors equipped with crypto keys
  - Get unique ID (but not related to location)

G. J. Simmons: The History of Subliminal Channels,
IEEE Journal on Selected Areas in Communications, vol. 16, no. 4, pp. 452–462, May 1998.

Requirements

- Soviet Union can ask any time for current ***amount***
  - How many silos are occupied?
  - But not infinite queries
  - ➔ limited number of queries
  - Queries authenticated with ***digital signature*** (using SU secret key)
  - Only validated queries are answered
- Not revealing ***location*** of missiles
  - Not: Which silos are occupied?
  - Missiles are transported among silos frequently
  - Real and fake transports (missile “shell game”)

Crypto System

- Each side can choose own crypto system
  - Didn't trust each other
  - Hoped to get insight in technology progress of other side

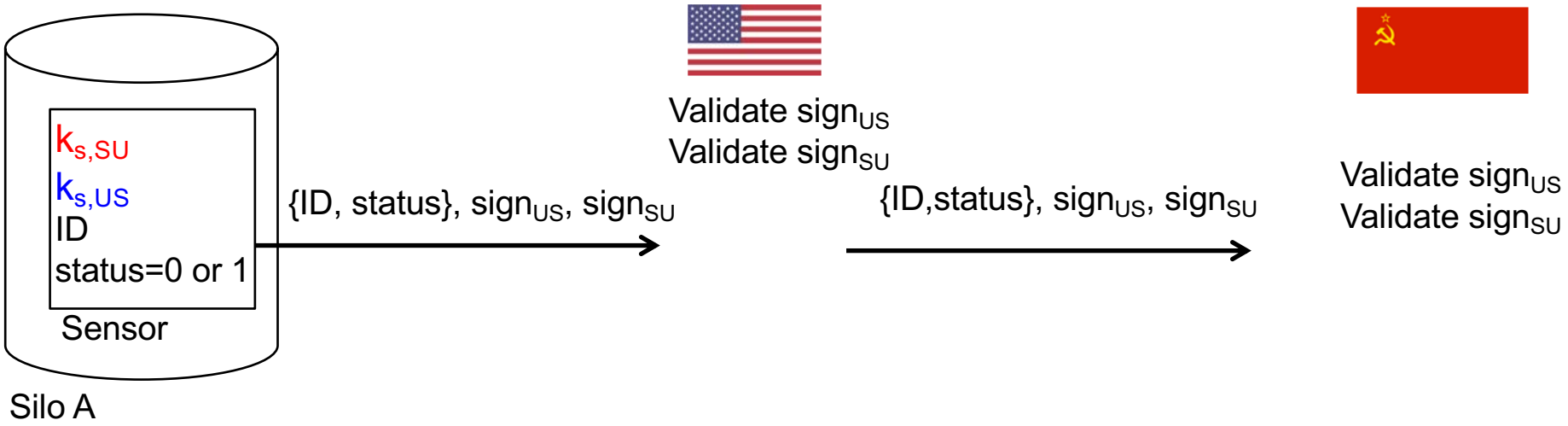
- Full crypto specification known to other side
  - Conform to Kerckhoffs' Principle

Requirements

- Prevent ***forging of reports*** from sensors
  - US: fake report to show empty US silo
  - US: send report if none requested (exhaust number of allowed reports for SU)
  - SU: fake report to say US has too many full silos

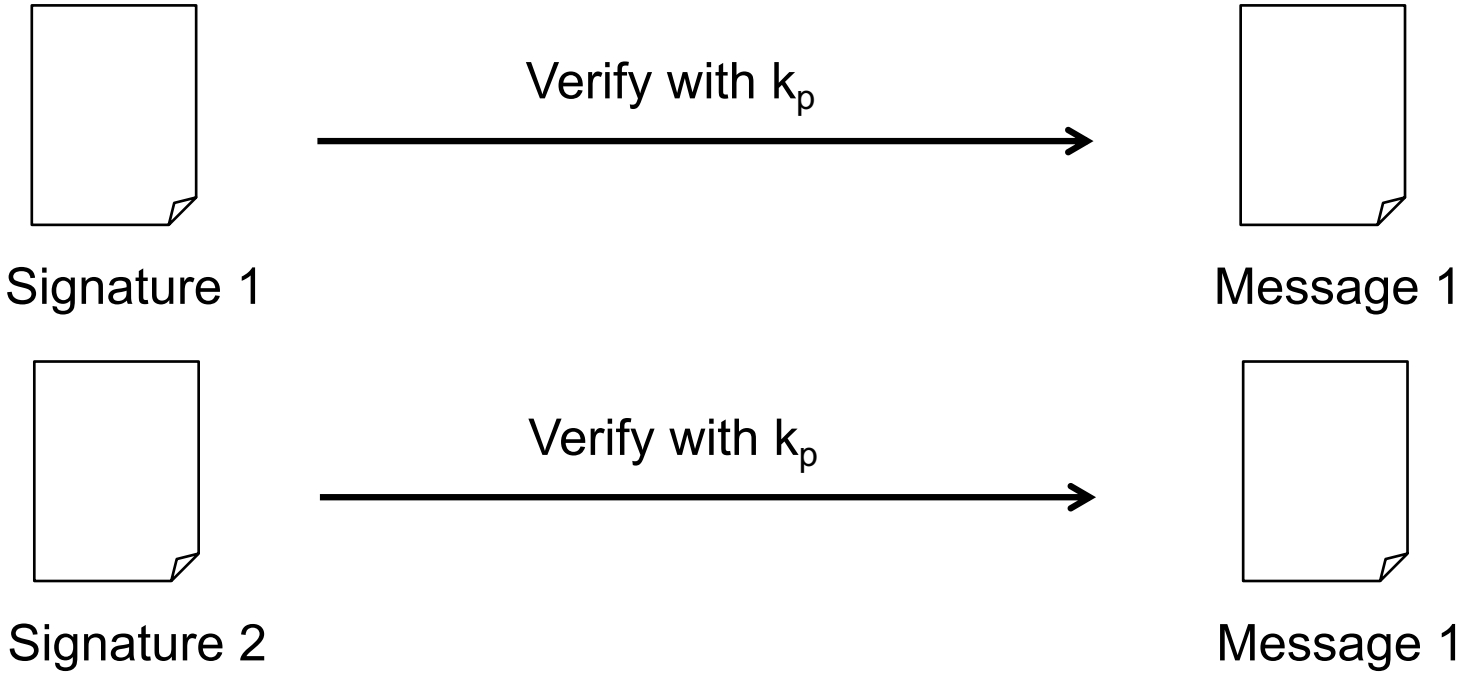
- Ensure that reports from sensors can not be forged
  - ***Unique ID*** per Silo ➔ prevent that US resends report from empty silo several times
  - ***Signature from US***
  - ***Signature from SU***
- SU does not know which ID is at which location

Possible Solution



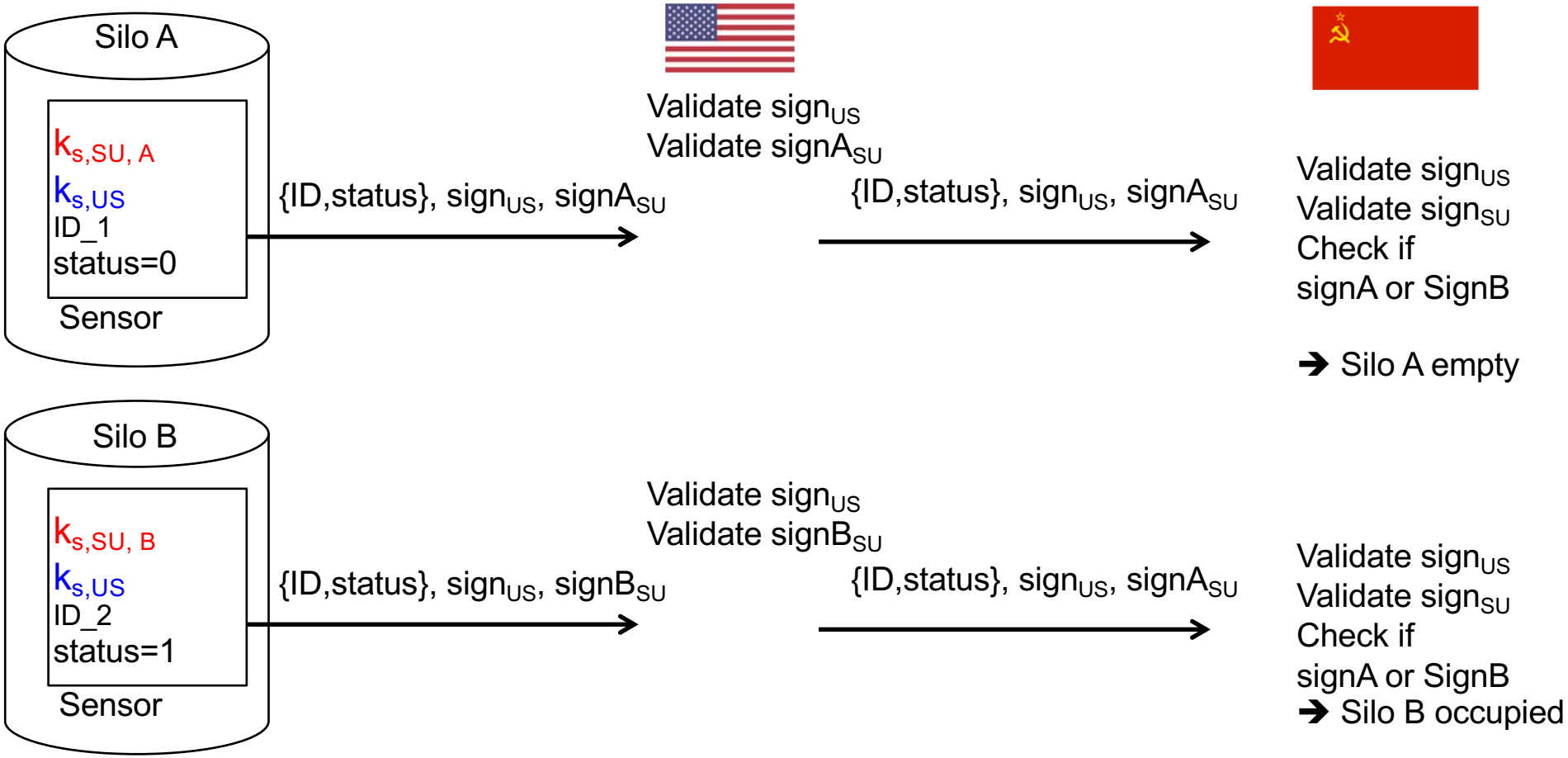
- US checks if
  - Report has not been modified ➔ using $k_{p,US}$
  - SU Signature is valid ➔ using $k_{p,SU}$
    - Prove that SU agrees that report content generated by sensor
- SU checks if
  - Report has not been modified ➔ using $k_{p,SU}$
  - US Signature is valid ➔ using $k_{p,SU}$
    - Prove that US agrees that report content generated by sensor

What if …



Different ***valid signatures*** for same message

Subliminal Bit



- Relates secret key to location (at installation)
  - For US signature looks ok
  - SU can find out location

Possible?

NSA (1979): *"Well, that was interesting, but there aren't any ciphers like that."* (according to [Simm98])

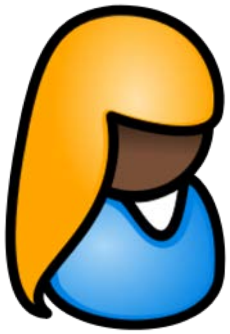
- But Rabin Cipher (1979)
  - Private key (p,q), public key: n with n=pq
- Encryption $c = m^2 \ mod \ n$
- Decryption: Find m for which $c \equiv m^2 \ mod \ n$
  - Hard if only n known, Simple if p,q known (using CRT)
  - But: Inverse not unique (e.g. *c=1, n=6* ➔ *m=1, m=5*)
- Signature*:
  - Find s that fulfills $s^2 \equiv m \ mod \ n$
  - Each solution is valid signature
- Other such ciphers exist

* In practice use padding to make m a QR

[Simm98] G. J. Simmons: The History of Subliminal Channels, *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 452–462, May 1998.

Establishing a Subliminal Channel

k_{s1}, k_{s2} ,k_p



Alice

k_{s1}, k_{s2} ,k_p

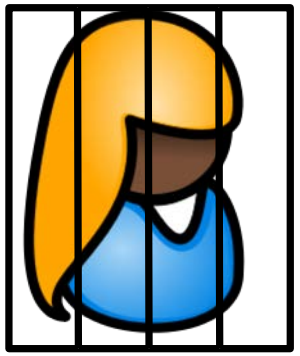
Bob

- Alice and Bob agree on secret keys
  - before they are captured
- Agree on convention that
  - Signing with k_{s1} means subliminal bit= 0
  - Signing with k_{s2} means subliminal bit= 1

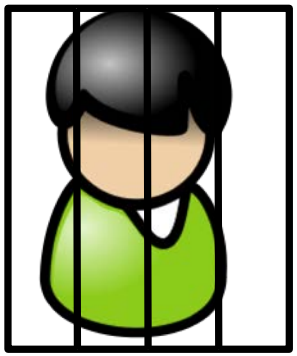
Establishing a Subliminal Channel

k_p

k_{s1}, k_{s2} ,k_p

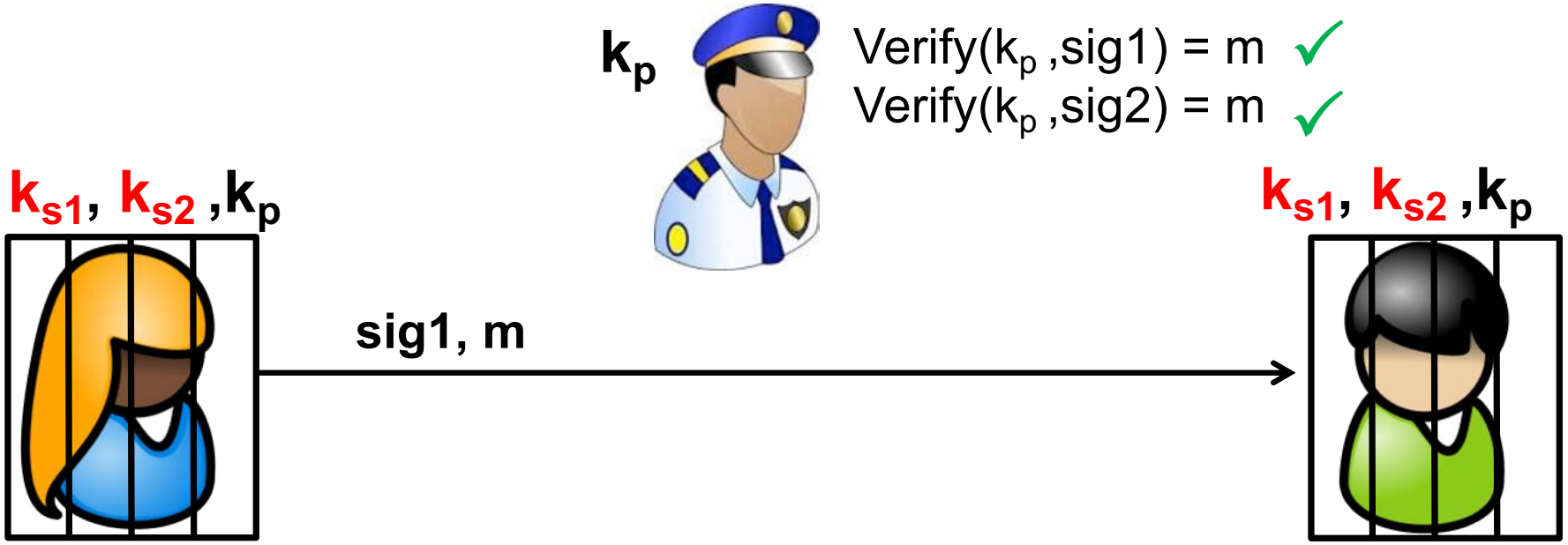


k_{s1}, k_{s2} ,k_p



- Guard only gets public key
  - To verify signature

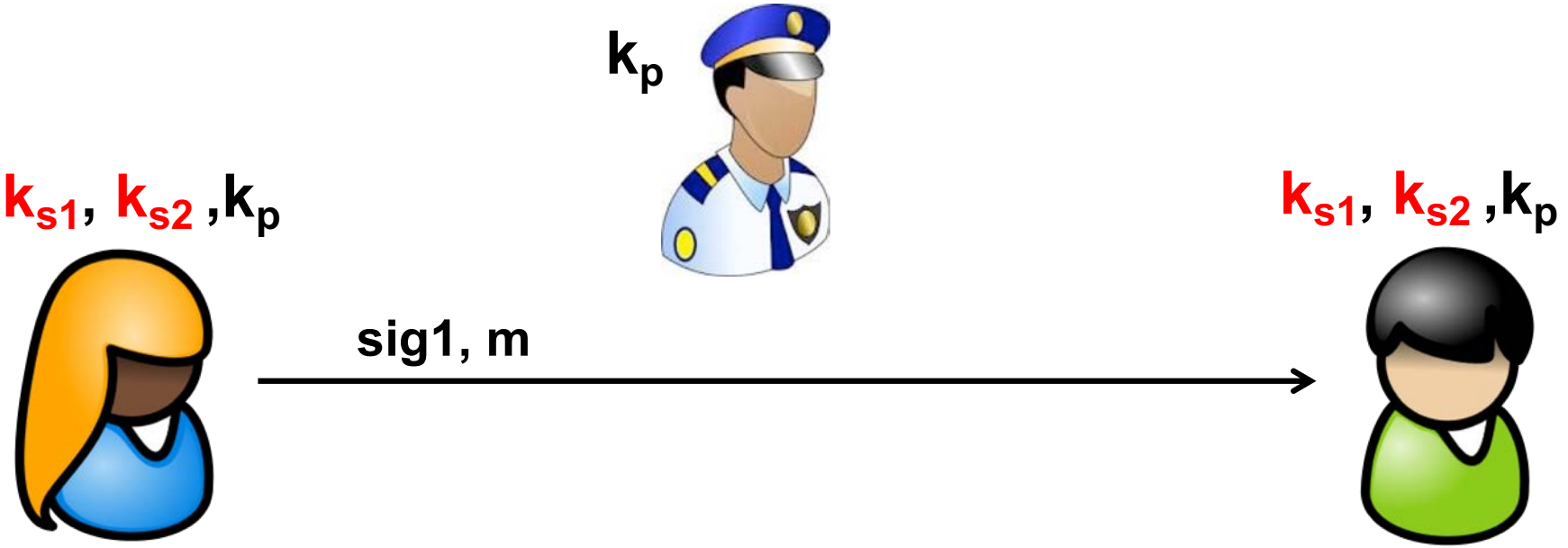
Establishing a Subliminal Channel



sig1=E(k_{s1} ,m)
sig2=E(k_{s2} ,m)

- Alice sends sig1 or sig2
- Verification with public key succeeds in both cases
  - For signatures generated with k_{s1} or k_{s2}

Establishing a Subliminal Channel



sig1=E(k_{s1} ,m)
sig2=E(k_{s2} ,m)

Check if D(k_p ,sig1)=m ✓
If E(k_{s1} ,m)= sig1 ➔ bit=0
If E(k_{s2} ,m)= sig1 ➔ bit=1

- Bob verifies message with k_p
- Generates Signature with k_{s1} and k_{s2} to check which key was used ➔ deduces subliminal bit

Kerkhoffs?

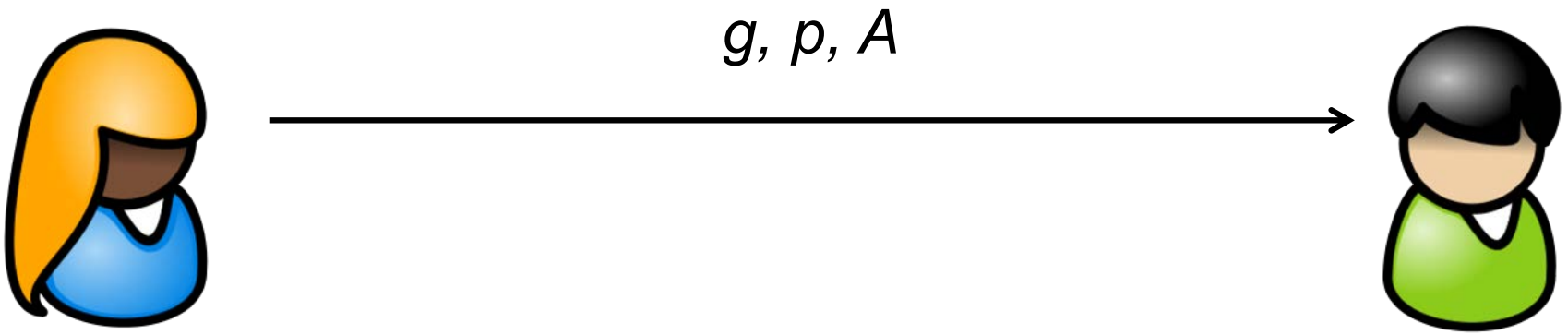
- Guard suspects that A and B hide information by using different signatures
  - But does not know k_{s1}, k_{s2}
  - Cannot find out which key was used
  - Cannot decrypt the subliminal bit

- Bandwidth of subliminal channel
  - 1 bit per message
  - Very low…

Subliminal Channel in Elgamal Signature

Recap: Elgamal Signature

Select a prime number p
Select a generator g
Generate random number a (secret)
Calculate $A=g^a \bmod p$

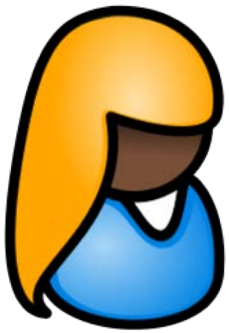


Recap: Elgamal Signature

Chose k (secret)
Compute $r=g^k \bmod p$
Find s with
$m=ar+ks \bmod p-1$

Compute $g^m \bmod p$
Compute $A^r r^s \bmod p$
Compare if

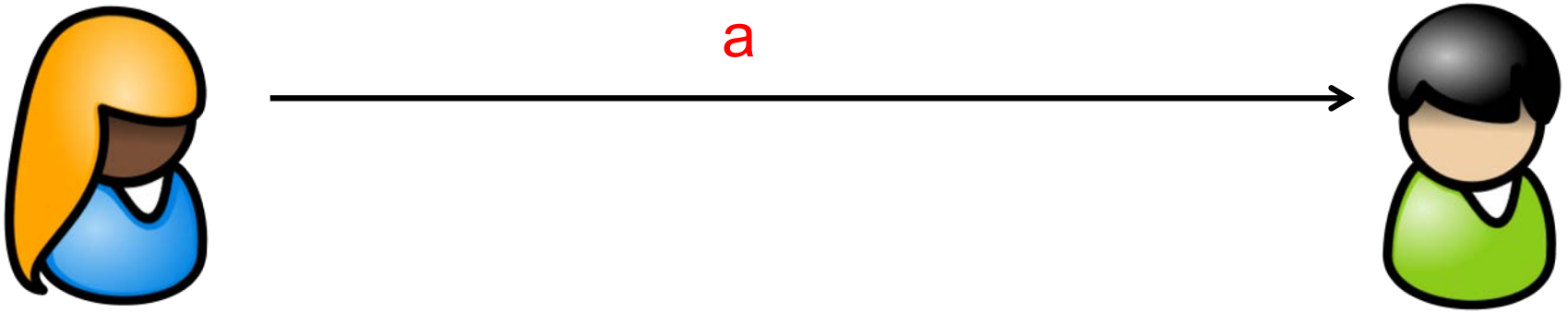
$$g^m \equiv A^r \cdot r^s \ mod\ p$$



m, r, s

Elgamal Subliminal Channel

Select a prime number p
Select a generator g
Generate random number a (secret)
Calculate $A=g^a \bmod p$

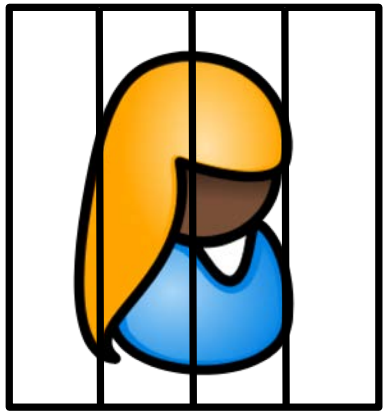


- Alices shares her private key *a* with Bob
  - Before they are captured
- Violation of public/private key concept
  - In normal operation: never share your private key!

Elgamal Subliminal Channel

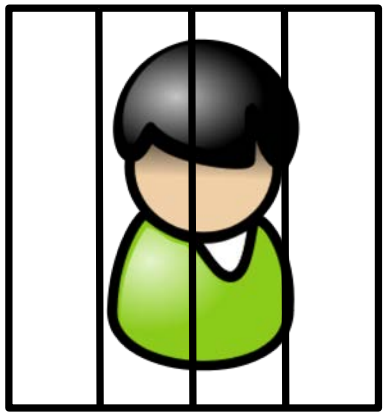
g, p, A

a



g, p, A

a



Elgamal Subliminal Channel

Chose $k=m_{sub}$ subliminal message
Compute $r=g^k \bmod p$
Find s with
$m=ar+ks \bmod p-1$

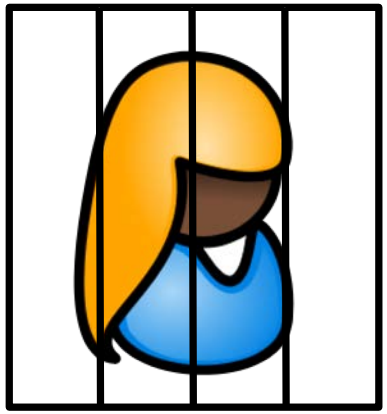
g, p, A

m, r, s

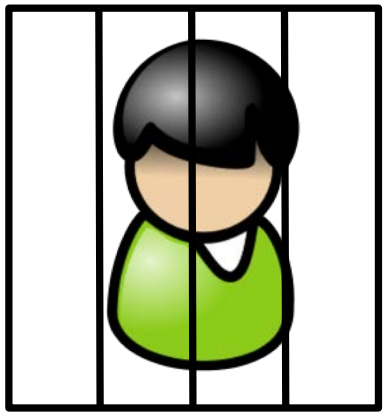
g, p, A

m, r, s a

a



m, r, s



Verification (by Guard or Bob)

- Guard and Bob have: g, p, A, m, r, s
- Compare if:

$$g^m \equiv A^r \cdot r^s \mod p$$

- Verification works because r and s fit to chosen k:

$$m \equiv ar + ks \mod p - 1$$

$$g^m = A^r \cdot r^s = g^{ar} \cdot g^{ks} = g^{ar+ks} = g^{m+i\cdot(p-1)}$$
$$= g^m \cdot g^{i\cdot(p-1)} = g^m \cdot \left(g^i\right)^{p-1}$$

$$g^m \equiv g^{ar+ks} \mod p$$

Extraction of Subliminal Message (by Bob)

- Bob knows secret key *a*

$$m \equiv ar + ks \mod p - 1$$

$$\boldsymbol{m} \equiv \boldsymbol{ar} + m_{sub} \cdot \boldsymbol{s} \mod \boldsymbol{p} - 1$$

- Bob knows m, a, r, s, p
- ➔ Bob can calculate k=m_{sub}

$$m_{sub} \equiv (m - ar) \cdot s^{-1} \mod p - 1$$

- Even if guard suspects subliminal channel
  - Guard does not know *a*
  - ➔ cannot calculate m_{sub}

Summary

- Covert Channel
  - Transmitting information on channels not intended for communication
- Covert Channels in Network Protocols
  - Header fields
  - Flow characteristics (packet sizes, timing)
- Subliminal Channels
  - Sending information out of a cryptosystem
  - Broadband subliminal channel in Elgamal Signature

Thank you!

Message received?