Cryptography (5830)

jr qrirybc cvbarrevat yrnqref sbe gur qvtvgny ntr

Cryptography: “Hidden writing”

- Study and practice of building security protocols that resist adversarial behavior
- Blend of mathematics, engineering, computer science

- Cryptanalysis: breaking cryptography

Cryptography use cases

Crypto is foundational

Failures highlight dependency on good crypto:

- WWII
- Government sabotage
- Playstation 3 crack
- Wireless keys for cars
- Password cracking
- WEP attacks
- Many TLS vulnerabilities
- …

Crypto and society

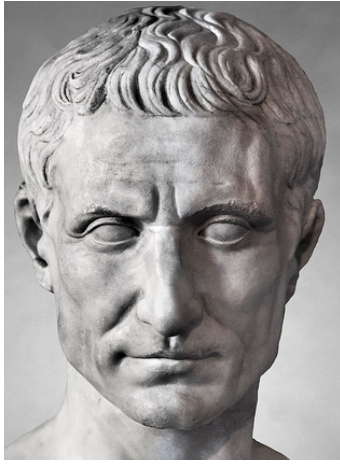
Google:
“report on smartphone encryption and public safety”

“keys under doormats”

Symmetric encryption

- Symmetric = secret key shared between sender and recipient
  - Three algorithms: Key generation (Kg), Encryption (Enc), Decryption (Dec)
  - Functionality (correctness)
  - Security
    - Capabilities of attacker
    - Attacker goals

Substition cipher (Caeser cipher)



Kg():
K <-$ {a,b,c,…,z}

Pick a random English letter from [a-z]

Enc(K,M):
Split M into characters $M_1,M_2,\ldots,M_m$
For i = 1 to m do
    C_i <- M_i + K mod 26
Return $C_1||C_2||\ldots||C_m$

Assume M is string of English lower-case letters

Dec(K,C):
Split C into characters $C_1,C_2,\ldots,C_m$
For i = 1 to m do
    C_i <- M_i - K mod 26
Return $M_1||M_2||\ldots||M_m$

Assume C is string of English lower-case letters

Let’s do some cryptanalysis

jr qrirybc cvbarrevat yrnqref sbe gur qvtvgny ntr

Auguste Kerckhoffs' (Second) Principle

(circa 1883)

"The system must not require secrecy and can be stolen by the enemy without causing trouble"

A cryptosystem should be secure even if its algorithms, implementations, configuration, etc. is made public --- the only secret should be a key

Why?

Some attacker capabilities

- **Unknown plaintext**
  - attacker only sees ciphertext(s)
- **Known plaintext**
  - attacker knows some plaintext-ciphertext pairs
- **Chosen plaintext**
  - attacker can choose some plaintexts and receive encryptions of them
- **Chosen ciphertext**
  - Attacker can see encryptions of chosen plaintexts and decryptions of chosen ciphertexts
- **Side-channels** such as timing, length, partial bits of keys, knowledge of bad key distribution, etc.

Attacker goals?

- Partial information about plaintext
- Plaintext recovery
- Secret key recovery

Known-plaintext attack

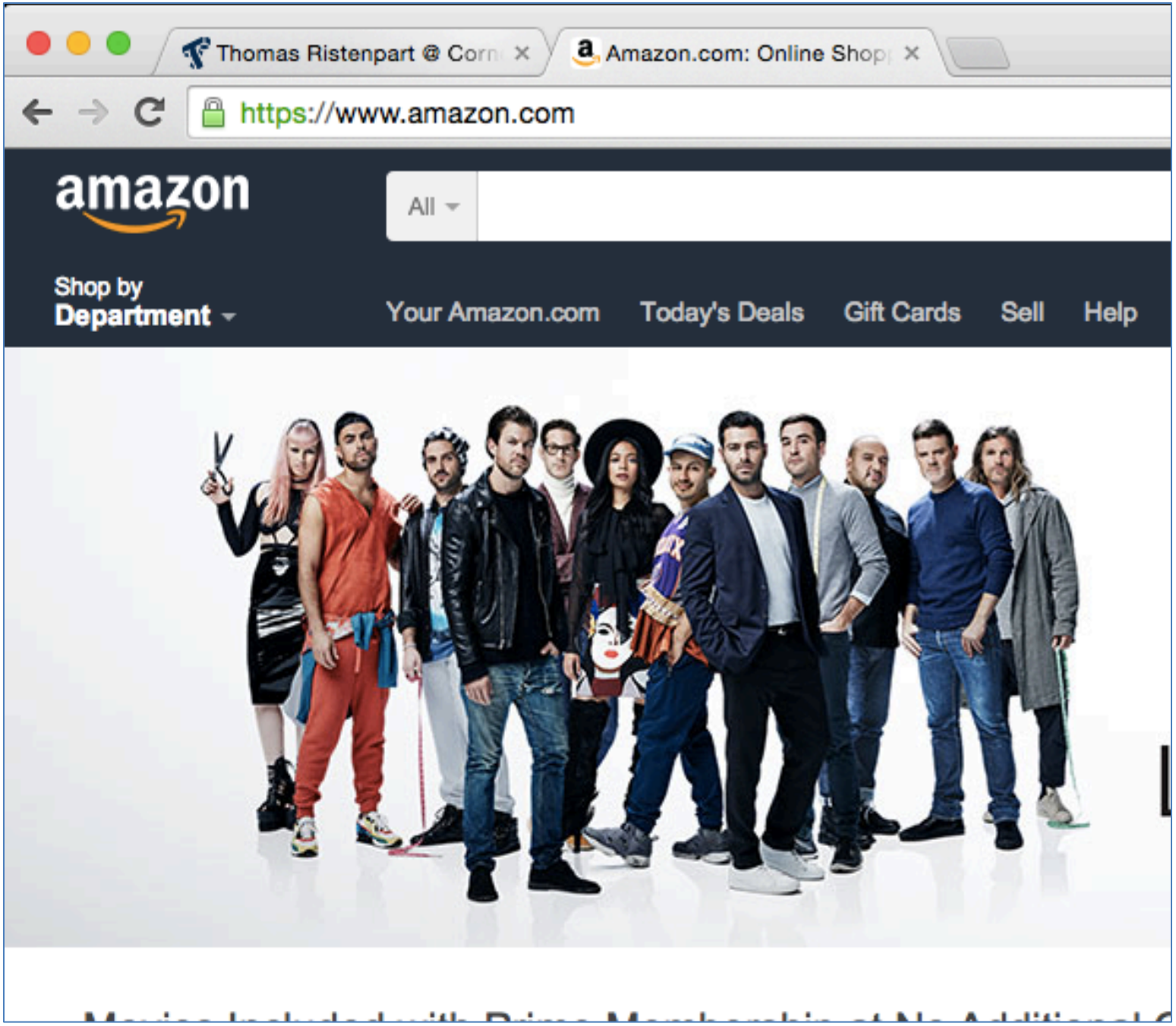
Say we know one plaintext, ciphertext pair

M = hello world
C = mjqqt btwqi

And we have a target ciphertext adversary wants to decrypt:

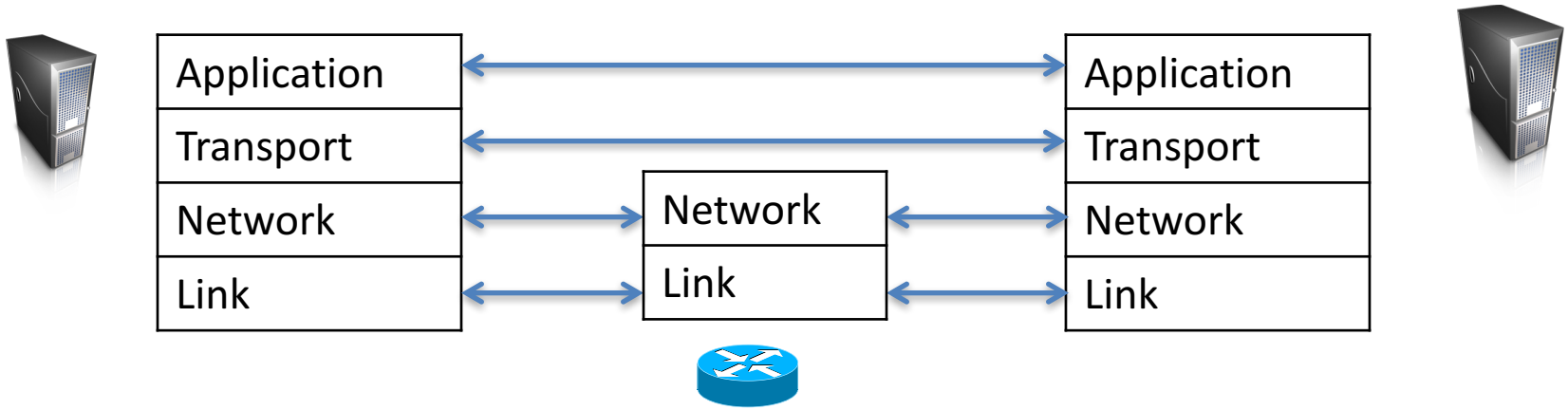
ymnx nx xzujw xjhwjy

How could an attacker get known plaintexts in practice?

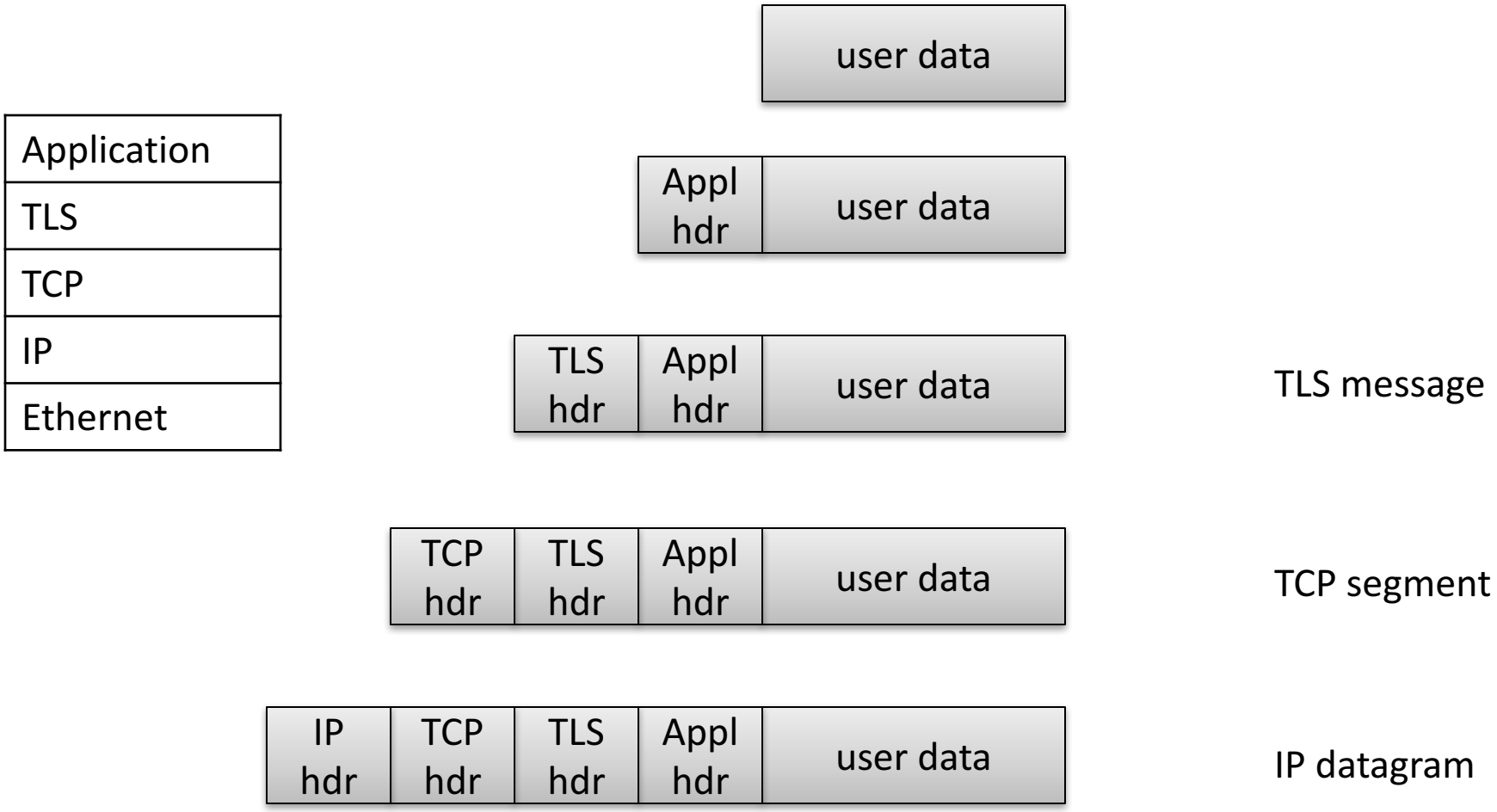

Thomas Ristenpart @ Corn
Amazon.com: Online Shop
https://www.amazon.com
amazon
All
Shop by
Department
Your Amazon.com
Today's Deals
Gift Cards
Sell
Help

Internet protocol stack

Application	HTTP, FTP, SMTP, SSH, etc.
Transport	TCP, UDP
Network	IP, ICMP, IGMP
Link	802x (802.11, Ethernet)



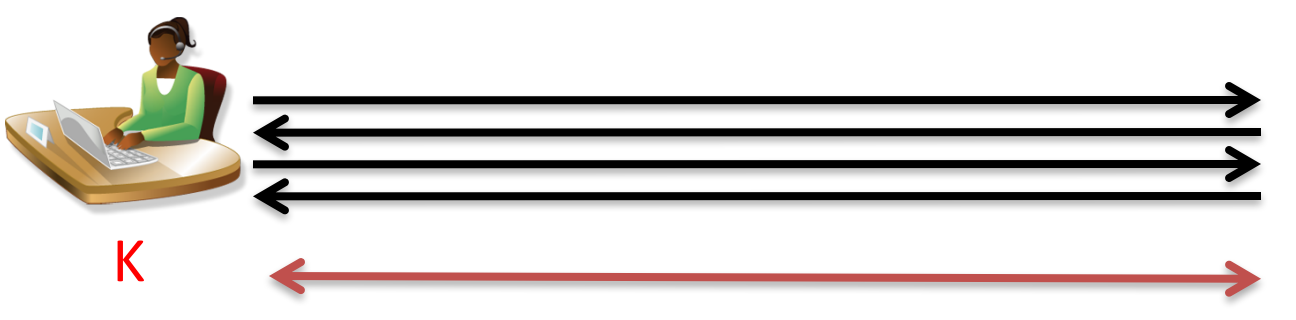
TLS sits between application and TCP



Places TLS is used

- HTTPS (the web)
  - HTTP messages but over TLS, not TCP
- Email connections
  - When getting information from your email server (not the email contents themselves)
- Virtual private networks (VPNs)
  - Tunnel other internet connections over a TLS connection

How TLS works (high level view)



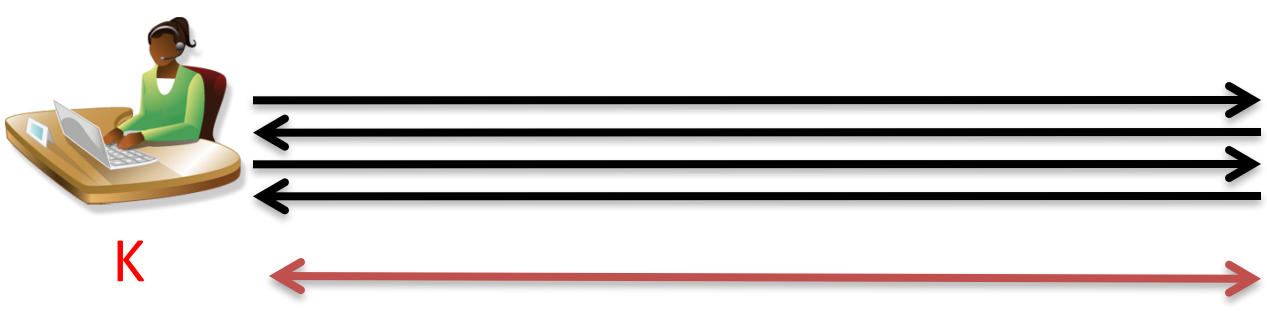
Step 1:
Key exchange protocol to share secret K

Step 2:
Send data via secure channel

Goals of handshake (key exchange protocol):

- Negotiate version
- Negotiate parameters (crypto to use)
- Authenticate server (Is server actually Amazon.com?)
  - Digital signatures and certificates
- Establish shared secret
  - Asymmetric encryption primitives

How TLS works (high level view)



Step 1:
Key exchange protocol to share secret K

Step 2:
Send data via secure channel

Goals of secure channel (record layer protocol):

- Confidentiality
  - Only sender/recipient can learn information about plaintext
- Integrity
  - Only sender/recipient can generate valid ciphertext

Some goals for course

- **Learning to speak crypto**
  - What primitives are for
  - What are the security targets associated with them
  - How do cryptographers formalize & evaluate
  - Ability to reason about policy from technical standpoint
- **Current best practices** for cryptographic constructions you are likely to encounter
  - Understand why they are considered best
  - Know how to break some inferior choices
- You should be able to **build crypto libraries** exposing clean, easy-to-use interfaces

Administrative

- Course website:
  https://github.com/cornelltech/CS5830-Spring2017
- Rough schedule is there, it will evolve
  - Topics you'd really like to hear about? Let me know
- Slides will be available after lectures on website
- Big things:
  - Homework assignments (Python coding, short answer)
  - Midterm
  - Final
  - Extra credit opportunities
- TA: Paul Grubbs

Startup project involve crypto?

- Come talk to me
- Extra credit potential