Anonymous Communications
Today’s program

• Cryptography background/refresher
• Motivation: why anonymity?
• What is anonymity?
• How to communicate anonymously
• Tor: overview, details, metrics
• Attacks on anonymous communications
Cryptography background/refresher
Symmetric key cryptography, public key cryptography, certificates
Encryption and decryption

- Fundamental question in information security: who is the adversary?
- Goal: send message without the adversary learning the content
- Encryption and decryption as functions: $Dec(k, Enc(k, m)) = m$
• Goal: send message *without the adversary being able to modify it*
• Message Authentication Code (MAC) computed over the message
• The other end point uses the symmetric key to verify the MAC
Secure channel: encryption + authentication

- In a secure channel, all messages are encrypted and authenticated
- Properties: confidentiality and integrity
Asymmetric encryption/decryption

- All parties who know the public key can encrypt a message
- Only the private key owner can decrypt the message
- Can be used to establish a shared symmetric key:
  - Sender chooses key, encrypts it with the destination’s public key
Digital signatures

- Goal: send message *without the adversary being able to modify it*
- Only the private key owner can create a signature
- All the other parties can verify the signature with the public key
Certificates

- Alice wants to communicate with Bob, *but*:
  - They don’t share a symmetric key
  - Alice doesn’t know Bob’s public key
- Digital signatures can be used
  - A trusted party (*Certification Authority*) can certify Bob’s public key
- TLS/HTTPS is based on this mechanism
A note on computational costs

- Public key cryptography is *expensive*!
- Symmetric encryption/MAC of 100 byte message: ~ 1μs
- Asymmetric encryption/signature of 100 byte message: ~ 1ms
- Usually, asymmetric crypto is used to establish a symmetric key
Motivation: why anonymity?
Why anonymity?

- IP addresses still leak *metadata* information:
  - Who talks to whom, at what time, for how long, how frequently…
  - NSA can log connection metadata, and later incriminate Snowden
- We may want to hide from the destination itself (to avoid retaliation)
Why anonymity?

• It is not just for (alleged) criminals who want to act with impunity!

• **Military applications**
  • Covert intelligence gathering
  • Covert attacks
  • Penetration testing on own infrastructure
  • Undercover agents communicating out of a monitored country

• **Trade, industrial R&D**
  • Detect price discrimination
  • Hide revealing patent searches in an untrusted database
Why anonymity?

• Anonymous reporting
  • Tips regarding criminal activity, but also accidents

• Human rights
  • Free speech, whistleblowing, censorship avoidance

• Building block for other technologies:
  • Crypto-currencies (Bitcoin, Ethereum)
  • Electronic Voting
What is anonymity?

(Sender/receiver) anonymity, threat models
Magic box abstraction

- The Magic box can do anything, and is fully trusted
- Useful to abstract away unimportant details
- We will see real world instantiation later
Terminology: “anonymity”

- **Sender anonymity**
  - Adversary knows receiver
  - Adversary may learn message
  - Sender is unknown

- **Sender anonymity set**
  - Set of all possible senders
  - Small set → little anonymity
Terminology: “anonymity”

- **Receiver anonymity**
  - Adversary knows sender
  - Adversary may choose message
  - Receiver is unknown
- **Hidden service** (pseudonym known)
- **Return address**
  - Token provided by receiver
- **Receiver anonymity set**

(Magic box)
Threat model(s)

- There are various types of adversary that can be considered.
- Degree of control: *local* or *global*.
- Type of control: *network* or *compromised infrastructure*.
  - Various combinations are possible.
  - The infrastructure is never fully compromised.
- Type of behavior: *passive* or *active*.
- Often not clearly specified → unclear guarantees.
How to communicate anonymously

How can we realize the magic box with real-world components?
What mechanism can we use?

- Wireless communication: *broadcast*
- Receiver anonymity is guaranteed
- Sender can still be de-anonymized
  - Localization through triangulation
  - Sender can move to avoid detection
- Alternative: hijacked connection
  - stolen SIM card, hacked WiFi
  - network ID ≠ personal sender ID
What mechanism can we use?

• Simple idea: use a proxy or VPN
• The proxy has to be trusted
• To avoid fully trusting the proxy:
  • Layered encryption
• Still, a malicious proxy is a problem
• Improvement: use a VPN to connect to another VPN!
Layered encryption (onion routing)

Servers (*relays*) typically run by volunteers
How to establish the keys?

- **Per-flow**: establish a *virtual circuit* (keys) once per flow, reuse it for all packets in the flow using only *symmetric key crypto*
  - *Efficient*, since symmetric key cryptography is faster
  - *Lower anonymity*: adversary can see the packet flow
  - Used in most common existing systems

- **Per-message**: establish new keys on the fly for each message
  - *Inefficient*, but higher anonymity
  - Used in so-called *mix-nets* (mostly for email)

  similar to *virtual circuit* vs. *datagram* in routing
Parallel with networking

- In a way, the relays form a network
- Differences w.r.t. the Internet:
  - Overlay network (on the Internet)
  - Fully connected topology
  - Source-controlled routing
  - Main goal of routing is anonymity, not reaching the destination
Tor: the second generation onion router
Tor basics

- Virtual circuits over 3 relays
- *Per-hop TCP*, established on the fly
- *Per-hop TLS*, except on the last hop
- Supports SOCKS proxy (TCP appl.)
- Comes with Tor browser (Firefox) for cleaning HTTP/HTTPS traffic
  - Dingledine, Mathewson, Murdoch, and Syverson, Tor: the second-generation onion router, 2014 draft
Life-cycle of a Tor circuit

• **Circuit setup**
  - The sender negotiates shared keys with all relays on the path (this requires expensive *asymmetric key cryptography*)
  - The relays store the necessary state

• **Data forwarding**
  - Packets for one or more flows are forwarded along the circuit
  - Only *symmetric key cryptography* is used

• **Circuit tear-down**
  - The circuit is destroyed to free state on relays or to prevent attacks
Telescopic circuit setup

• Keys are negotiated one relay at a time
• The circuit is “extended” by one hop at a time
  • For each hop, a shared key and an ID are established
  • That’s why it is called telescopic
Data forwarding

• The sender has established a circuit (keys and IDs)
• A data packet is encrypted as usual (layered encryption)
• The ID of the next relay is added in clear text
  • To protect against network adversaries, links are encrypted (TLS)
Circuit tear-down

• Can be initiated both by sender and by intermediate relays
  • The sender communicates the tear-down to one relay at a time, starting from the furthest away
  • The exit relay may tear down the circuit if a corrupt packet is detected, or some other attack
• Circuits have a limited lifetime, so they will eventually be destroyed
Details of the Tor protocol
Tor details and advanced features

- End-to-end integrity checking
  - Establishes a secure channel between client and exit relay
- Exit policies (exit can restrict the destinations they connect to)
- Multiple streams per circuit
- Censorship resistance (bridges)
- Hidden services
  - Provides receiver anonymity
Cells

- Basic unit is the cell (512 bytes)
- It contains a circuit ID and a command field (in cleartext)
- Same for cells in both directions
- A relay cell is decrypted, and its digest is checked:
  - If correct: check command
  - Otherwise replace circuit ID and forward cell along
Example circuit communication

create cl, $E(g^x_1)$

created cl, $g^y_1, H(g^{x_1 \cdot y_1})$

relay cl, $\{\text{extend, OR}_2, E(g^{x_2})\}$

created c2, $g^y_2, H(g^{x_2 \cdot y_2})$

relay c2, $\{\text{begin, bob:80}\}$

TCP handshake

relay c2, $\{\text{connected}\}$

GET ...

relay c2, $\{\text{data, “GET...”}\}$

relay c2, $\{\text{data, response}\}$

response

relay c2, $\{\text{data, response}\}$

relay c1, $\{\text{begin, bob:80}\}$

relay c1, $\{\text{connected}\}$

relay c1, $\{\text{data, “GET...”}\}$

relay c1, $\{\text{data, response}\}$

relay c1, $\{\text{data, response}\}$

relay c1, $\{\text{data, response}\}$

Image credits: Tor paper
Hidden services

- The hash of Bob’s public key is the identifier of his hidden service.
- Bob has connections to a set of *introduction points* (IP).
- To communicate, Alice connects to an IP and suggests a *rendezvous*.
- Bob can connect to the rendezvous and start the communication.

![Diagram showing the process of communicating through hidden services](bb111serv2ncblef.onion)
Censorship resistance in Tor

• The Tor network contains a number of bridge relays (or bridges)
• Not (all) publicly listed, instead distributed through friends networks
• Used to circumvent censors which black-list Tor relays
• Problem: deep packet inspection allows detection of Tor traffic
• Solution: obfuscate the traffic (pluggable transports)
• Obfuscation tries to hide Tor traffic features, but Censors improve their detection heuristics
  • This gives rise to the censorship arms race
Tor metrics – how Tor is doing today
• Over 2M daily users (estimated based on directory lookup frequency)
The exit relays and entry guards are a small fraction of all the relays.
Relays per country

81 countries with 6495 relays (3128 visible)
2017-11-20 12:00:00
Performance (file download)

Time in seconds to complete 1 MiB request to public server

Measured times on all sources per day

The Tor Project – https://metrics.torproject.org/
Bridges in the Tor network

The Tor Project – https://metrics.torproject.org/
Attacks on anonymous communications
Attacks

- A number of attacks has been proposed against these systems
- For many it is unclear whether they fit the stated threat models
  - Some of them are practical, requiring limited resources
  - Others are only achievable by state-level adversaries
- *Traffic analysis attacks*: flow fingerprinting, website fingerprinting
- *Higher-layer attacks*: stack fingerprinting
Passive traffic analysis

- The adversary observes the edges of the network, recording traffic patterns
  - Flow length, bandwidth pattern, inter-packet timings
- Real-time detection is challenging
  - Alternative is store and compare later → large amount of storage!
Active traffic analysis

- The adversary actively modifies packet timings
  - Inter-packet timings (delaying/reordering packets), packet drops
  - *Flow watermarking*: inject one bit of information (marked or not)
  - *Flow fingerprinting*: inject multiple bits (e.g., sender IP address!)
Website fingerprinting

- Adversary needs only one observation point (ISP, other WiFi user…)
- Adversary has built a database of fingerprints of websites
- Particularly effective for interactive applications (health/tax forms)
  - Chen et al., *Side-channel leaks in web applications: a reality today, a challenge tomorrow*, S&P 2010
Intersection attack

- Often, users only communicate with a small subset of other users
- Idea: every time a message is seen by the target, register the sets of destinations
- This is called intersection attack
  - Kesdogan et al., Limits of anonymity in open environments, IH, 2002
- More effective: statistical disclosure
  - Danezis and Serjantov, Statistical disclosure or intersection attacks on anonymity systems, IH, 2005
Higher-layer attacks

- OS Network stack fingerprinting
  - Compromised adversary can probe TCP stack
  - Solution: per-hop TCP
  - Still, HTTP layer may be identifiable!

End-to-end TCP

- IP tunnel
- IP tunnel
- IP tunnel
Higher-layer attacks

• Most de-anonymization is still done through other means:
  • Trick user into downloading malware
  • Trick user into downloading file that will access the Internet directly
• To achieve anonymity, all layers need to be anonymized:
  • Any gap will break anonymity
  • (This is unlike other security properties)
Thanks! Any questions?

daniele.asoni@inf.ethz.ch
References

• Danezis, Diaz, and Syverson, Systems for anonymous communications, Handbook of Financial Cryptography and Security, 2009

• Dingledine, Mathewson, Murdoch, and Syverson, Tor: the second-generation onion router, 2014 draft http://sec.cs.ucl.ac.uk/users/smurdoch/papers/tor14design.pdf

• Regarding the evolution of Tor: https://blog.torproject.org/top-changes-tor-2004-design-paper-part-1

• Kesdogan et al., Limits of anonymity in open environments, IH, 2002

• Danezis and Serjantov, Statistical disclosure or intersection attacks on anonymity systems, IH, 2005

• Danezis and Goldberg, Sphinx: A compact and provably secure mix format, S&P 2009

• Chen et al., Side-channel leaks in web applications: a reality today, a challenge tomorrow, S&P 2010

• Le Blond et al., Herd: a scalable, traffic analysis resistant anonymity network for VoIP systems, ACM SIGCOMM, 2015