CSE 565 Computer Security
Spring 2019

Lecture 22: Anonymous Communication

Department of Computer Science and Engineering
University at Buffalo
• Anonymous communication
  – mixes
  – anonymizing proxies
  – onion routing

• Other anonymity services
  – anonymous digital money
  – anonymous access control
• Often if we don’t specify the name or other personal information, our communication seems anonymous.

• Normally, however, this is not the case:
  – if we read a web page, the web server knows from what address the request is coming.
  – if we connect to a chat channel, the server knows from what address we are coming.
  – if you send an encrypted email, the endpoints still can be recovered.

• But does it really matter?
Anonymous Communication

• Internet surveillance techniques are known as **traffic analysis**
  – it can be used to infer who is talking to whom over a public network

• Knowing the source and destination of our traffic allows others to **track your behavior and interests**

• This can lead to various **consequences**
  – an e-commerce website can use price discrimination based on your country or institution of origin
  – this can even threaten your job and physical safety by revealing who and where you are
    • e.g., you are traveling abroad and connect to your employer’s computers to check mail
Anonymous Communication

• **Consequences** of traffic analysis
  – when abroad, you can inadvertently reveal your national origin and professional affiliation to anyone observing the network
  – this holds even if the connection is encrypted

• **How does traffic analysis work?**
  – it examines packet header information
  – it applies to payload of any type (email message, web page, an audio file)
  – even if the payload is encrypted, traffic analysis still reveals a lot about what you are doing (and possibly what you are saying)
• Traffic analysis uses header information that discloses source, destination, size, timing, etc.

• The basic problem is that the recipient of your communications can see that you sent it
  - so can authorized intermediaries (i.e., Internet service providers) and sometimes unauthorized intermediaries

• A very simple form of traffic analysis might involve someone sitting between the sender and recipient on the network looking at headers

• More powerful types include:
  - spying on multiple parts of the Internet and using sophisticated statistical techniques to track the communication patterns
Benefits of Anonymous Communication

• Say, we can build anonymous communication channels, what does it enable us to do?
  
  – the basic line is that it allows organizations and individuals to share information over public networks without compromising privacy
  
  – individuals can keep websites from tracking them
  
  – individuals can connect to news sites, instant messaging services, and the like when these are blocked by their local Internet providers
  
  – individuals can publish websites and other services without needing to reveal the location of the site
  
  – individuals can conduct socially sensitive communication
    • e.g., chat rooms and web forums for rape and abuse survivors or people with illnesses
Benefits of Anonymous Communication

- What else do anonymous channels enable us to do?
  - journalists can communicate more safely with whistleblowers and dissidents
  - organizations can enable their workers to connect to their home websites while in foreign countries without letting others know for whom they are working
  - activist groups recommend anonymous communication as a mechanism for maintaining civil liberties online
  - corporations can perform competitive analysis and protect sensitive procurement patterns from eavesdroppers
  - law enforcement can visit and surveil websites without leaving government IP addresses in their logs
• Anonymity likes company
  – you cannot be anonymous by yourself
    • but can you have confidentiality by yourself?
  – a network that protects only Department of Defense (DoD) network
    users won’t hide that connections from that network are from DoD
  – you can be anonymous by hiding in the crowd
• There are several technical approaches to achieve anonymity
• The most popular are mixes and proxies
• What does a **mix** do?
  
  – it receives encrypted messages
  
  – it then randomly permutes and decrypts inputs

```
Mix

message 1
message 2
message 3
message 4
```
• The **key property** is that an adversary cannot tell which ciphertext corresponds to a given message.
The basic mix was introduced by Chaum in 1981

- there is a number of servers each with its own public key $p_{ki}$
- to send a message $m$ through servers 1, 2, and 3, envelope it using all of the servers’ keys

$$c = E_{p_{k1}}(E_{p_{k2}}(E_{p_{k3}}(m)))$$
Mixes

Server 1

\[ m_1 \]

\[ \text{decrypt and permute} \]

Server 2

\[ m_2 \]

\[ \text{decrypt and permute} \]

Server 3

\[ m_2 \]

\[ \text{decrypt and permute} \]

\[ m_3 \]

\[ m_1 \]

\[ m_2 \]

\[ m_3 \]
Mixes

- Each server on the way knows only which server gave it data and which server it is giving data to.

- No individual server ever knows the complete path that a data packet has taken.

- **One honest server preserves privacy**.

- Mixnets were introduced for email and other high latency applications:
  - each layer of message requires expensive public-key cryptography
  - sufficient number of messages needs to be accumulated to defeat timing attacks.

- But what if you need quick interaction?
  - web browsing, remote login, chat, etc.
- Anonymizing proxy

- communications appear to come from the proxy, not true senders
- it can use low-cost symmetric encryption (or no encryption)
- it thus is appropriate for web connections, SSL/TLS, ssh, etc.
• **Anonymizing proxy**

  – **advantages**: simple, focuses a lot of traffic for more anonymity

  – **disadvantages**: a single point of failure, compromise, attack

  – **example**: the Anonymizer and others

  – **risks** of using anonymizing HTTP proxies

    • all data you send to the service must first go through the proxy

    • a malicious proxy server can record everything you send to it, including unencrypted logins and passwords

      – don’t use proxy servers of unknown integrity

      – if there is no choice, do not pass any sensitive information through the proxy unencrypted
Onion Routing can be used to build traffic analysis resistant infrastructure.

The main idea is to combine advantages of mixes and proxies:
- use (expensive) public-key crypto to establish circuits
- use (cheaper) symmetric-key crypto to move data

Trust is distributed like in mixes.

Onion routers form an overlay network.

There are proxy interfaces between client machines and onion routing network.
• The Onion Routing (TOR) network
• Tor establishes routing connections called **circuits**
  – during circuit setup session keys are negotiated using servers’ public keys
  – after some time session keys used in a circuit are refreshed to limit the impact of key compromise

• **Tor circuit setup**
  – the client chooses a set of onion routers to tunnel packets through
  – the client’s proxy establishes a session key and circuit with the first onion router on the list
  – proxy tunnels through that circuit to extend to the second router on the list, etc.
- Client applications connect and communicate over the Tor circuit
  - many applications can share it to communicate with various destinations

- Directory servers maintain a list of onion routers, their status, location, current keys, etc.
  - they also control which nodes can join the networks (helps prevent certain attacks and abuse)

- The current implementation of Tor works for TCP streams and can be used by any application with SOCKS support
  - see http://www.torproject.org for more detail
• Tor setup in more detail
  – each user runs local software called an onion proxy to fetch directories, establish circuits, and handle connections from user applications
  – each onion router maintains a long-term identity key and a short-term onion key
    • the identity key is used to sign TLS certificates, router descriptor information (address, bandwidth, etc.), and directories
    • the onion key is used to decrypt requests from users to setup a circuit and negotiate session keys
  – the TLS protocol establishes a short-term link key when communicating between onion routers
    • these keys are rotated periodically and independently
• Tor circuit setup
  
  – the client’s onion proxy (OP) chooses routers OR₁, OR₂, …
  
  – OP engages in a Diffie-Hellman key establishment with OR₁:
    
    • OP sends $g^{a_1}$ encrypted under OR₁’s key:
      
      $$\text{Enc}_{pk_1}(g^{a_1})$$

      OP $\xrightarrow{}$ OR₁

    • OR₁ responds with $g^{b_1}$ and a hash of $k_1 = g^{a_1 b_1}$:
      
      $$g^{b_1}, h(k_1 || \text{“handshake”})$$

      OP $\xleftarrow{}$ OR₁

    • the hash tells OP that OR₁ indeed computed $g^{b_1}$
• Tor circuit setup (cont.)

• OP then uses OR\(_1\) to extend the circuit to OR\(_2\):
  
  – OP tunnels through OR\(_1\) key exchange negotiation for OR\(_2\):
    
    \[ \text{Enc}_{pk_2}(g^{a_2}) \]
    
    \[
    \begin{array}{c}
    \text{OP} \\
    \downarrow
    \end{array}
    \quad
    \begin{array}{c}
    \text{OR}_1
    \end{array}
    \]

  – OR\(_1\) relays the request to OR\(_2\) and forwards OR\(_2\)’s reply to OP:
    
    \[
    \begin{array}{c}
    \text{OP} \\
    \downarrow \quad \text{Enc}_{pk_2}(g^{a_2})
    \end{array}
    \quad
    \begin{array}{c}
    \text{OR}_1
    \\
    \downarrow \quad \text{OR}_2
    \end{array}
    \]

    \[ g^{b_2}, h(k_2||"handshake") \]

    \[ g^{b_2}, h(k_2||"handshake") \]

  – here \( k_2 = g^{a_2b_2} \) is a session key shared between OP and OR\(_2\)
• Tor circuit setup (cont.)
  – the process continues until session keys with all of the routers on the path are established

• Established circuits use layered encryption as in mixes, but now decryption is fast

• As before, each router randomly permutes the packets

• Session keys are re-negotiated after a short period of time (e.g., one minute)
• Tor properties
  – replay attacks are not effective
    • replayed circuit setup will result in a new session key at an honest onion router
  – perfect forward secrecy is achieved
    • recording all traffic sent to a node and later breaking its public key will not reveal encrypted content
  – it can adapt to network dynamics
    • if one router becomes unusable, building a whole new circuit is not required
• Tor makes it possible for users to hide their locations while offering services
  – such services include web publishing, instant messaging servers, etc.
  – for example, a Tor user can setup a website where people publish material without worrying about censorship
  – nobody is able to determine who is offering the site and nobody know who is posting to it

• These services are called hidden services, and setting up a hidden service includes
  – selecting a few onion routers as introduction points
  – advertising these points on the lookup service
  – building a circuit from each introduction point to the service
• Anonymous communication has many motivations for use by individuals, organizations, and the government

• Early proposals include mixes and proxies

• The onion routing (Tor) project provides a real-life system for achieving anonymous communications
  – http://www.torproject.org