CS 6262 - Network Security: Privacy/Anonymity

Professor Patrick Traynor
4/16/13
Reminders

• One week from today - final poster presentations!
  ‣ You should be just about finished running your experiments.
  ‣ Remember, you must turn in your code in addition to your poster, with instructions to build and run your experiments.
When Confidentiality is Insufficient
Privacy != Confidentiality

• Confidentiality refers to the property of the content being unreadable from unauthorized readers.
  ‣ A man-in-the-middle can see ciphertexts fly by, but their contents are indistinguishable from random bits.

• Privacy refers to the *awareness of the existence of communication between two or more parties*.
  ‣ Do Alice and Bob talk to each other? How often? Are the messages indicative of their content?

• Note that these two are often used interchangeably in the vernacular, but should not be.
Questions...

WHAT IF THE MAYAN CALENDAR ENDS IN 5105
AND WE'VE JUST BEEN HOLDING IT UPSIDE DOWN
Questions...

- Can we have confidentiality without privacy?
Questions...

• Can we have confidentiality without privacy?

• Can we have privacy/anonymity without confidentiality?
Anonymity

• The purpose of anonymity is to protect identity.
  ‣ e.g., An anonymous poster on a website wants you to read their comments.
  ‣ Their intended goal is to expose content without letting you know who revealed the content itself.

• You *do not* have to be anonymous to have privacy. You *must* maintain privacy to achieve anonymity.

• Ok, great. Can we do any better than just not logging into a webpage when posting contents?
Anonymous Publishing

- **Goal**: Publish “The Graduate Student’s Manifesto”, a subversive guidebook to completing your Ph.D. without exposing your identity.

- **Publius**: Encrypted content is posted across multiple servers, readers must assemble a threshold number of key pieces to recover plaintext.

- Published content is cryptographically tied to the URL, meaning that changes can instantly be detected.
Private Browsing

• Most major browsers now provide “Private Browsing” Modes, that allow you to visit webpages while reducing the state you expose to the world.
  ‣ Does not record visited pages in your browsing history.
  ‣ Stores cookies while on a single site and deletes them when you leave that site.

• What protections are provided by these mechanisms?
  ‣ Who is the adversary?

• Try Panopticlick to see if you can be fingerprinted: https://panopticlick.eff.org/
Anonymous Proxies

• Simplest architecture - redirect all traffic via an encrypted tunnel to some proxy in the Internet, which in turn forwards your traffic to its intended destination.
  ‣ e.g., YouHide.com, Proxify.com, The Anonymizer, Anonymouse.org, etc, etc...

• In their terms of service, many of these services note that they will not sell your information to third-parties.

• What protections are provided by these services?
  ‣ Who is the adversary?
Mixes

• Originally proposed by Chaum, a client selects a series of *mix nodes* called a *cascade* through which each message should pass.

• Messages are encrypted in reverse order of the cascade using the public key of each mix node.

• Messages are decrypted in each mix, which reveals the next hop along the cascade.
  
  ‣ Note that messages are *stored*, interleaved and eventually *forwarded* in a mix.
Mixes in Action
Mixes in Action
Mixes in Action
Mixes in Action
Mixes in Action
Mixes in Action
Mixes in Action
Mixes in Action
Mixes in Action
Mixes: Limitations

• A simple, mechanism for providing privacy (and potentially anonymity) for store and forward-based communications.

• Where does that leave everything else?
  ▸ HTTP? SMTP? IMAP? SSH?
• Extends the mix concept to “real-time” traffic.
  ‣ Note that real-time is somewhat of a misnomer.

• Like in mix networks, Tor wraps each message in multiple layers of encryption, from last to first hop.
  ‣ Tor specifically mandates three layers. Why three?

• Upon receipt, each message is decrypted, placed into the outgoing queue and sent out as quickly as possible.
Tor in Action
Tor in Action
Tor in Action
Tor in Action

circID = 100
Tor in Action

circID = 100

circID = 867
Tor in Action

circID = 100

circID = 867
Tor in Action

circID = 100

circID = 867

circID = 5309

file
Tor in Action

circID = 100

circID = 867

circID = 5309

file
Tor in Action

circID = 100

circID = 867

circID = 5309
Tor: Details

- Mix networks are very much a uni-directional process.
  - How does Tor get responses back to their sender?

- Tor relies on circuits, pre-established identifiers and keys to return such information.
  - The “exit node” receives a response from a webpage and, knowing the ID of the previous hop (circID), encrypts the message.
  - The previous node receives the message, looks up the corresponding circID for the next hop, encrypts and forwards.
  - The originator eventually receives a thrice encrypted packet.
Tor: Hidden Services

• Tor also allows users to access “hidden services”.
  ‣ Services within the Tor network that do not want their identities revealed.
  ‣ Tor includes a rendezvous service to allow users to find registered services.

• Hidden services include:
  ‣ Anonymous publishing (think alternative to Publius)
  ‣ Black Markets (Silk Road)
  ‣ NGOs (Reporters Without Borders)
Tor: Limitations

• Tor is run on a series of nodes located throughout the world.
  ‣ The hope of this architecture is that not only can you pick a diverse route, but that you can also rely on servers in other countries if yours outlaws Tor.

• Problem: Everyone knows which nodes are running Tor, so if it is illegal, these nodes are already blocked.
Tor: Limitations

• Unlike mix networks, Tor’s lack of potentially infinite delay of packets makes it susceptible to timing attacks.

• Many of researchers have demonstrated the ability to add fingerprints to flows by changing the *inter-packet timing*. 
Tor: Limitations

- Unlike mix networks, Tor’s lack of potentially infinite delay of packets makes it susceptible to timing attacks.

- Many of researchers have demonstrated the ability to add fingerprints to flows by changing the *inter-packet timing*.
Additional Techniques

- **Crowds**: Clients join a “jondo”, a group that forwards messages to a random other member.
  - Each receiver gets a message, it flips a biased coin and if heads, it forwards the message to another random node. If tails, it sends the message to the final destination.

- **Hordes**: Similar to Crowds, but assumes that nodes share a multicast connection.
• Mix-based schemes are intuitive, and allow for relatively high throughput.

• Unfortunately, they do not offer strong, formally verifiable guarantees.
  ‣ How many mix nodes must you visit to achieve “anonymity”? What about insiders in each of these designs?
Dining Cryptographers

- Allows a sender to anonymously send a single bit

Alice

Bob

Charles
Dining Cryptographers

- Allows a sender to anonymously send a single bit

\[ \text{Flip}_{A,B} = 1 \]
Dining Cryptographers

- Allows a sender to anonymously send a single bit

![Diagram showing Alice, Bob, and Charles with flip \( \text{Flip}_{A,C} = 0 \).]
Dining Cryptographers

- Allows a sender to anonymously send a single bit

\[ \text{Flip}_{B,C} = 1 \]
Dining Cryptographers

- Allows a sender to anonymously send a single bit

**NSA**

**Alice:** $A, B \oplus A, C = 1 \oplus 0 = 1$

**Bob:** $A, B \oplus B, C = 1 \oplus 1 = 0$

**Charles:** $A, C \oplus B, C = 0 \oplus 1 = 1$

$$A \oplus B \oplus C = 0$$

Alice

Bob

Charles
Dining Cryptographers

- Allows a sender to anonymously send a single bit

**NSA**

- Alice: \(A, B \oplus A, C = 1 \oplus 0 = 1\)
- Bob: \(A, B \oplus B, C = 1 \oplus 1 = 0\)
- Charles: \(A, C \oplus B, C = 0 \oplus 1 = 1\)
  \[A \oplus B \oplus C = 0\]

**Bob**

- Alice: \(A, B \oplus A, C = 1 \oplus 0 = 1\)
- Bob: \(A, B \oplus B, C = \neg(1 \oplus 1) = 1\)
- Charles: \(A, C \oplus B, C = 0 \oplus 1 = 1\)
  \[A \oplus B \oplus C = 1\]
DC-net Protocols

• Various extensions to the basic DC-net model.
  ‣ e.g., Collision resistance, maliciousness, etc
  ‣ Take advantage of underlying broadcast or multicast network topologies.

• More recent schemes take advantage of emerging cryptographic primitives:
  ‣ pMixes (Melchor et al.) use Private Information Retrieval (PIR) to hide their queries.
  ‣ SFENets (Nipane et al.) use Secure Function Evaluation (SFE)
DC-nets: Limitations

• These systems have strong, provable properties.
  ‣ Based on certain assumptions (or varying strength), you can demonstrate that these systems provide certain properties.

• There is no such thing as a real-time DC-net.
  ‣ Some get close (SFENets show IM client working at practical speed), but operations are far too heavy for SSH, HTTP and VoIP.
  ‣ Most are significantly slower.
Other Applications Spaces

- **Wireless**
  - Spread spectrum techniques make communications “indistinguishable” from noise.

- **Voting**
  - Traditional ballots are “anonymous”. Cryptographic techniques make this (and many other properties) possible in electronic voting systems.

- **Money**
  - Cash is anonymous. Electronic forms of currency with similar features (e.g., eCash, BitCoin) are being investigated.
Conclusions

• Privacy and Anonymity are properties that go beyond confidentiality.

• Anonymous communications are generally broken down into two generally classes of solutions: Mixes and DC-nets
  ‣ One gives you quite strong guarantees, but at a cost. The other gives you “reasonable” performance, but with fuzzy guarantees.

• This is a very deep and complex field.
  ‣ There are many more techniques, and challenges facing them. Nothing yet provides us with everything that we need!