CPSC 467b: Cryptography and Computer Security

Instructor: Michael Fischer
Lecture by Ewa Syta

Lecture 25
April 25, 2013

Anonymous Communication

DISSENT- Accountable Anonymous Group Communication

Anonymous Authentication

Anonymous Communication

Anonymity[1]

Anonymity is the state of being not identifiable within a set of subjects, the anonymity set.

Anonymity set is the set of all possible participants in a system that could have been the sender or recipient of a particular message.

Sender anonymity is the state of being not identifiable within the sender anonymity set as the sender of a particular message.

Receiver anonymity is the state of being not identifiable within the receiver anonymity set as the receiver of a particular message.

[1] A. Pfitzmann and M. Hansen, *Anonymity, unobservability, and pseudonymity: A consolidated proposal for terminology*, 2008

Goal of anonymity systems[1]

In general, anonymity systems seek to provide *unlinkability* between sent messages and their true recipients (receiver anonymity), and between received messages and their true senders (sender anonymity).

Benefits of anonymous communication[2]

- Investigative journalism
- Whistleblowing
- Law enforcement
- Self-help
- Personal privacy protection
- Avoiding persecution

[2] R. Kling, Y-C. Lee, and A. Teich, *Assessing Anonymous Communication on the Internet: Policy Deliberations*, Journal of Information Society, 1999

Harms of anonymous communication[2]

- Spamming
- Deception
- Hate mail
- Impersonation and misrepresentation
- Online financial fraud
- Other illegal activities

Approaches to anonymous communication

There are two main approach to anonymous communication:

- DC-nets
- Mix networks

Dining cryptographers problem[3]

Three cryptographers are sitting down to dinner at their favorite three-star restaurant. The waiter informs them that the meal has been paid anonymously by one of the cryptographers *or* the National Security Agency. The cryptographers respect each other's right to make an anonymous payment, but they wonder if the NSA paid.

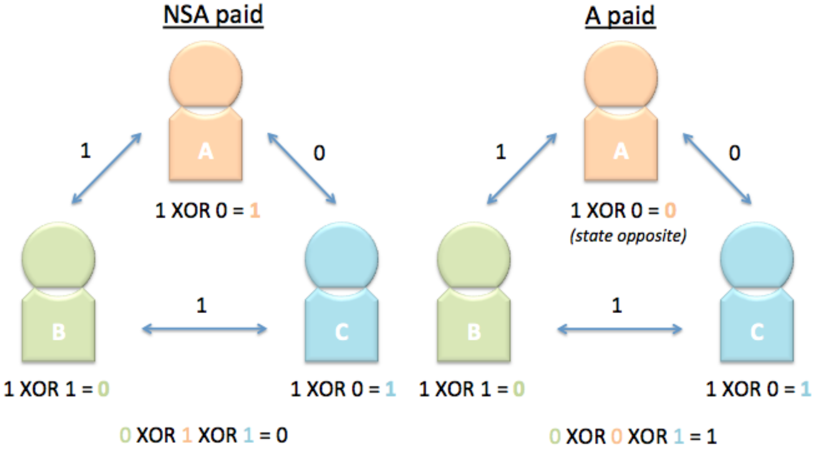
They find out by executing a two-stage protocol.

[3] D. Chaum, *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*, Communications of the ACM, 1981

DC-nets

1. Establish a secret.
   - Each cryptographer secretly flips an unbiased coin with the cryptographer on his right.
2. Reveal the secret.
   - Each cryptographer says whether the two coins he sees fell on the same side or on different sides.
   - If one of the cryptographers is the payer, he states the opposite of what he sees.

If the result is 0, then NSA paid. If it is 1, then one of the cryptographer paid. However, cryptographers who did not pay do not know which one did.


NSA paid
A
1
0
1 XOR 0 = 1
B
1
C
1 XOR 1 = 0
1 XOR 0 = 1
0 XOR 1 XOR 1 = 0
A paid
A
1
0
1 XOR 0 = 0
(state opposite)
B
1
C
1 XOR 1 = 0
1 XOR 0 = 1
0 XOR 0 XOR 1 = 1

DC-nets

The protocol is simple, elegant and unconditionally secure (assuming a secure channel) if carried out faithfully.

However, there are two major issues:

- Collisions – if two cryptographers paid for the dinner, their messages will cancel each other out. It means that only one participant can transmit at the same time.
- Disruptions - the last cryptographer can change the final result.

Herbivore[4]

Herbivore is a distributed anonymous communication system, providing private file sharing and messaging over the Internet.

It lets people anonymously publish and retrieve documents, and guarantees that even the most resourceful adversary cannot compromise this anonymity.

Built to be self-organizing, Herbivore relies on neither central servers nor trusted parties, and ultimately provides anonymity by drawing on its community of users.

[4] http://www.cs.cornell.edu/people/egs/herbivore/

Mix-networks

A *mix* is a process that accepts encrypted messages as input, groups several messages together into a batch, and then decrypts and forwards some or all of the messages in the batch.

A mix network consists of mix routers. Mix routers forward the message along the specified path. Some mix routers may intentionally delay sending messages, send them in batches of specified size, etc.

The goal is to obscure the associations between incoming and outgoing messages.

Onion routing[5]

Onion routing is based on the idea of mix networks.

A message is iteratively wrapped with layers of encryption to form “an onion” that specifies properties of the connection along the route.

Each onion router along the route uses its public key to decrypt the entire onion. It exposes the identity of the next onion router, cryptographic information used, and the embedded onion. The router pads the onions and forwards to the next hop.

The last node in the chain removes the remaining layer of encryption and forwards the message to its destination.

[5] http://www.onion-router.net/

Tor (The Onion Router)[6]

Tor is a low latency anonymity network based on onion routing. It consists of a number of relays constantly encrypting and then randomly bouncing messages.

The random path packets take is extended one hop at a time, and each relay along the way knows only the previous and the next relay. A separate set of encryption keys is negotiated for each hop along the circuit.

Once a circuit has been established, it is used to exchanged different kinds of data. For efficiency, Tor uses the same circuit for connections that happen within the same ten minutes or so.

[6] The Tor Project `https://www.torproject.org/`

Tor

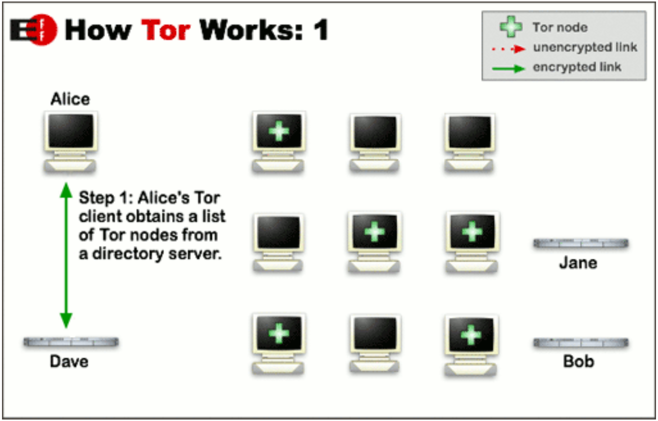
The goal is to protect users' from network surveillance.

No individual relay ever knows the complete path.

Neither an eavesdropper nor a compromised relay can link the connection's source and destination.

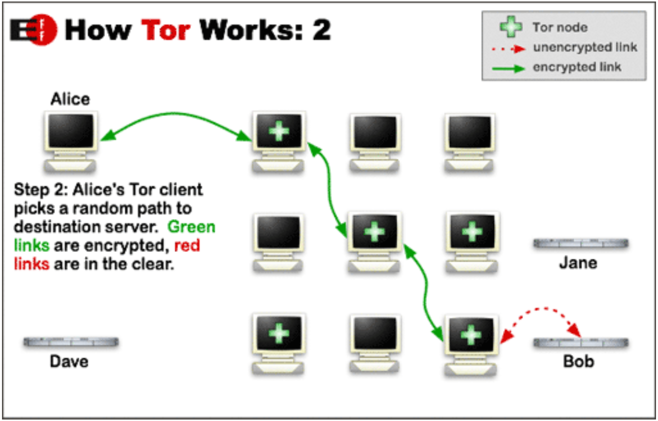
However, Tor cannot (and does not attempt to) protect the traffic entering and exiting the network.

How Tor works?



Step 1: Alice's Tor client obtains a list of Tor nodes from a directory server.

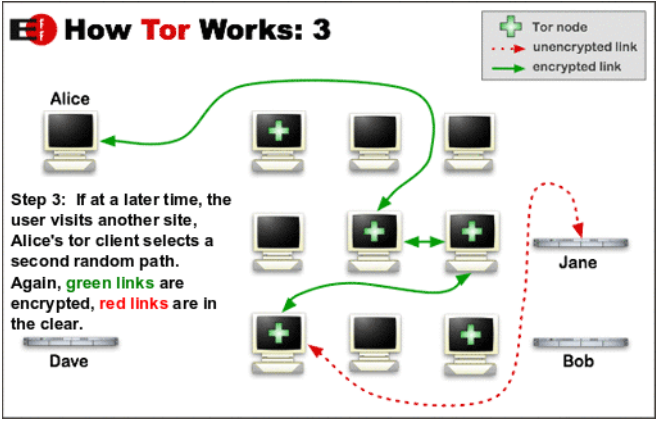
How Tor works?



How Tor Works: 2

Step 2: Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.

How Tor works?



How Tor Works: 3

Step 3: If at a later time, the user visits another site, Alice's tor client selects a second random path. Again, green links are encrypted, red links are in the clear.

Limitations of existing schemes

Method	Weakness
Mix Nets, Tor	Traffic analysis attacks
Group and Ring Signatures	Traffic analysis attacks
Voting Protocols	Short, fixed-length messages
DC Nets	Anonymous DoS attacks

DISSENT- Accountable Anonymous Group Communication

Dissent[8]

DISSENT stands for Dining cryptographers Shuffled Send Network. It provides accountability by adding a *blame* phase.

DISSENT is suitable for latency-tolerant applications because its scalability is limited.[7]

DISSENT consists of two sub-protocols: a *shuffle* protocol and a *bulk* protocol.

[7] The version presented here. There are other scalable versions of DISSENT.

[8] H. Corrigan-Gibbs and B. Ford, Dissent: Accountable Group Anonymity, 17th ACM Conference on Computer and Communications Security (CCS 2010)

Security Goals

Anonymity - the adversary cannot guess the sources of the messages from honest users with probability significantly greater than random guessing.

Integrity - every honest member for whom the protocol completes successfully has the same output and receives the messages of all the other honest members.

Accountability - every honest member for whom the protocol failed obtains proof of some member's misbehavior, and the adversary cannot produce a valid proof of misbehavior by an honest member.

Possible outcomes

Each protocol run will either succeed or fail.

1. Success.
   - All members faithfully follow the protocol.
   - Secret messages are delivered.
   - Secret permutation are unrecoverable.
2. Failure
   - Some member(s) deviates from the protocol.
   - Messages are unrecoverable.
   - At least one dishonest member exposed.

The Shuffle Protocol

The shuffle protocol builds on a data mining protocol by Brickell and Shmatikov. [9] Main idea: encrypt and shuffle.

DISSENT adds protection against DoS attacks by malicious group members by adding *go/no-go* and *blame* phases.

[9] J. Brickell and V. Shmatikov, *Efficient Anonymity-Preserving Data Collection*, KDD 2006

The Bulk Protocol

The bulk protocol builds on DC-nets and uses the shuffle in place of the DoS-prone slot reservation systems to prearrange the DC-nets transmissions.

A set of message descriptors that "describes" each member's DC-net transmission is anonymously shuffled. The shuffled order of the descriptors indicates the order in which the anonymous senders should transmit their secret messages. This guarantees each member exactly one message slot per round.

Message descriptors include information to verify each member's behavior. If misbehavior occurs, then members receive anonymous accusations that they can themselves verify.

Performance

A DISSENT prototype was tested under Emulab on groups of 44 nodes connected via simulated wide-area links.

- It incurs 1.4-minute latency when distributing messages up to 16MB among 16 nodes with 100mn inter-node delays.
- It handles large message loads, both balanced and unbalanced, in about 3.5× the time required for non-anonymized communication.

A 1MB message can be sent anonymously in:

- less than 1 minute in a 4-node group
- 4 minutes in a 20-node group
- 14 minutes in a 40-node group

Anonymous Authentication

What is anonymous authentication?

Regular authentication refers to verifying one's claim of identity.

Informally, *anonymous authentication* is the process of authenticating without revealing one's identity.

In case of anonymous authentication, there is no claim of identity! So what does it mean to "authenticate without revealing identity"?

Anonymous Authentication

Let's think of authentication differently. Instead of verifying one's identity, we can verify *any* property of a user, for example, their age.

Allowed users: anyone who is a resident of Connecticut.
Authentication claim: I am a resident of Connecticut.

Mostly, anonymous authentication schemes allow to verify whether a user belongs to a certain group. Anonymity comes from the fact that there are many users in the group.

Applications

Anonymous authentication has many applications:

1. Whistleblowing
2. Anonymous elections, surveys, and data collection
3. Anonymous access to resources

Anonymous Signatures

A *group signature*[10] scheme allows a member of a pre-arranged group to anonymously sign a message on behalf of the group. It requires a trusted third party and a fixed group roster.

A *ring signatures*[11] scheme allows a member of ad-hoc group to to anonymously sign a message on behalf of the group. No trusted party is needed.

[10] D. Chaum and E. Van Heyst, *Group signatures*, EUROCRYPT 1991

[11] R. Rivest, A. Shamir and Y. Tauman, *How to leak a secret*, ASIACRYPT 2001

Anonymous Authentication Scenario[12]

Victor has a great collection of e-books. He is happy to share them with Polly and Peggy but no one else. Each time they want to read one of his e-books, they need to authenticate to Victor.

However, Polly and Peggy do not want Victor to know who is reading which e-book. Therefore, they would like to be able to prove that one of them (either Polly or Peggy) are authenticating in a way that prevents Victor from learning which one.

Assume that Polly and Peggy have a public/private key pair $(y_1 = g^{x_1}, x_1)$ and $(y_2 = g^{x_2}, x_2)$ respectively.

[12] The scenario makes more sense if there are many Pollys and Peggys.

“Either/Or” Proof

An “either/or” proof is a proof of knowledge of a solution of problem 1 *or* problem 2. For example, we can prove the knowledge of the discrete logarithm of $y_1 = g^{x_1}$ or of $y_2 = g^{x_2}$.

When Polly or Peggy wishes to authenticate, she can prepare a proof that says "I know either x_1 or x_2” which translates to "I am either Polly or Peggy”.

Victor will be satisfied because he knows it is either Polly or Peggy. Polly and Peggy will be happy as well because Victor does not learn their individual taste in e-books.

"Either/Or" Proof (adapted from[13])

	Peggy		Victor
1.	Choose random v_1, v_2 and w Compute $t_1 = y_1^w g^{v_1}$, $t_2 = y_2^w g^{v_2}$	$\xrightarrow{t_1, t_2}$	
2.		$\xleftarrow{c}$	Choose a random c
3.	Calculate $c_1 = w$, $c_2 = c - c_1$, $r_1 = v_1$ and $r_2 = v_2 - x_2 c_2 + x_2 c_1$	$\xrightarrow{c_1, c_2, r_1, r_2}$	Check $t_1 \overset{?}{=} y_1^{c_1} g^{r_1}$, $t_2 \overset{?}{=} y_2^{c_2} g^{r_2}$ $c_1 + c_2 \overset{?}{=} c$ Accept if checks succeed.

[13] J. Camenisch and M. Stadler,*Proof Systems for General Statements about Discrete Logarithms*, ETH Zürich, Institut für Theoretische Informatik,1997

Proof Verification

$$t_1 \stackrel{?}{=} y_1^{c_1} g^{r_1} \qquad t_2 \stackrel{?}{=} y_2^{c_2} g^{r_2} \qquad c \stackrel{?}{=} c_1 + c_2$$

$$\begin{aligned} y_1^{w} g^{v_1} &= y_1^{c_1} g^{r_1} \\ y_1^{w} g^{v_1} &= y_1^{w} g^{v_1} \end{aligned}$$

$$\begin{aligned} y_2^{w} g^{v_2} &= y_2^{c_2} g^{r_2} \\ g^{x_2 w} g^{v_2} &= g^{x_2 c_2} g^{v_2 - x_2 c_2 + x_2 c_1} \\ g^{x_2 w} g^{v_2} &= g^{x_2 c_2} g^{v_2} g^{-x_2 c_2} g^{x_2 w} \\ g^{x_2 w} g^{v_2} &= g^{v_2} g^{x_2 w} \end{aligned}$$

$$\begin{aligned} c &= c_1 + c - c_1 \\ c &= c \end{aligned}$$

“Either/Or” Proof

There are some questions that need to be answered:

1. Will Polly and Peggy always be able to prepare a valid proof?
2. Why Victor does not learn who prepared the proof?
3. Why Mallory cannot prepare a valid proof?

Additional Resources

- The Tor Project, `https://www.torproject.org/`
- How Tor works? by Artist Molly Crabapple and Writer John Leavitt `https://www.eff.org/whatistor`