Internet Privacy

Markus Klein, Joshua Juen

University of Illinois

Fall 2011

“I have nothing to hide!“

“...they accept the premise that privacy is about hiding a wrong. It's not." Rather, privacy is most often an issue of accountability and trust.”

Bruce Schneier

Who Needs Privacy?

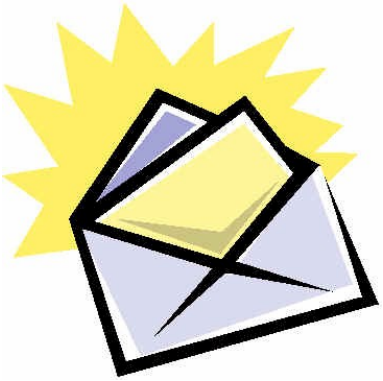
- Consider the following scenario:
  - Alice is an employee at XYZ inc. who suspects that her CFO Charlie is conducting insider trading.
  - Bob from the FBI is investigating Charlie's suspected insider trading.
  - Bob interviews Alice and decides to keep in contact with Alice.
- Should Alice be afraid to send information to Bob through the company's intranet?

Who Needs Privacy? Cont.

- Bob is a resident of Iran who wants to access international news.
  - Bob's government deems this material objectionable and limits access to Bob
  - The punishment to accessing such content is severe including possible imprisonment and fines
- This is a freedom of speech issue.
- Broad question, should we try to help Bob circumvent censorship?

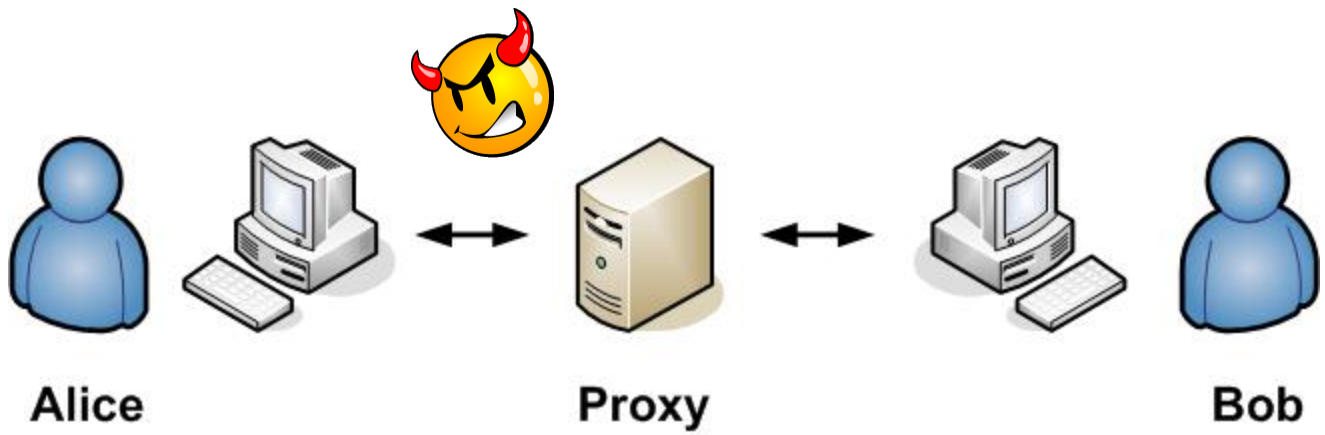
Why is Privacy a Problem?

- Every packet has a source/destination IP address
- Packets traverse unknown/untrusted networks
- Difficult to determine who is looking at our packets!!!!

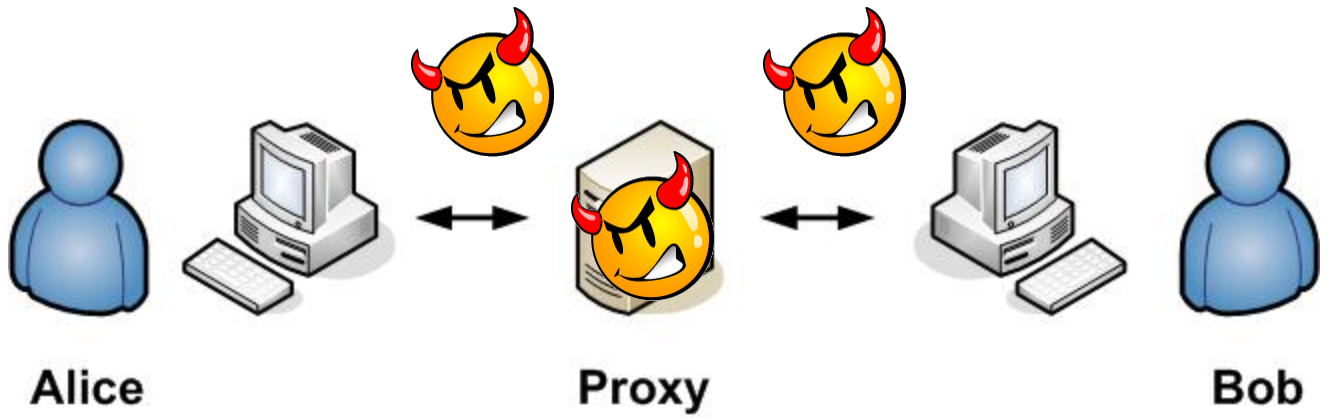


Privacy the Naïve Approach

- In the beginning there was the proxy:
  - Encrypt the destination address in the packet header and send to a trusted proxy.
  - Proxy forwards traffic on to the destination
  - CGIProxy and others….



Privacy the Naïve Approach cont.

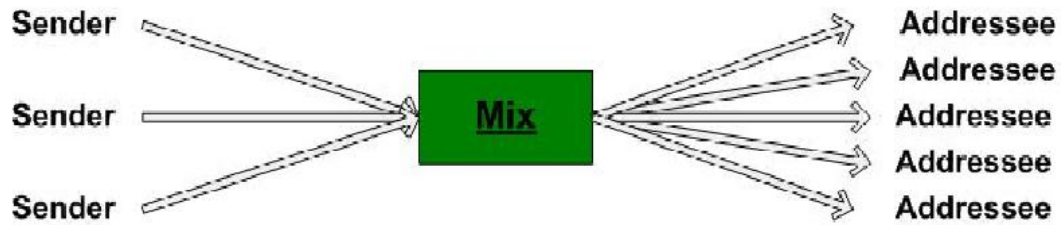


- Problems:
  - What if the attacker can see the outbound traffic from the proxy? (Global Passive adversary)
  - What if the proxy is controlled by the attacker?
  - Anonymity set is non-existent
- Goal we want a system to defeat a global passive adversary that is difficult for the adversary to control

High Latency Mix Nets

- Goal: Send high latency messages anonymously
  - Ideal for E-mail
  - Mixmaster, Mixminion, Babel, and many others…
- Threat model and design goals:
  - Resistant against a global passive adversary (can see all traffic)
  - Distributed network of volunteers (some may be controlled by an adversary)
  - Assume that the adversary can add, drop, reorder, and delay packets

Mix Net Proxy Design

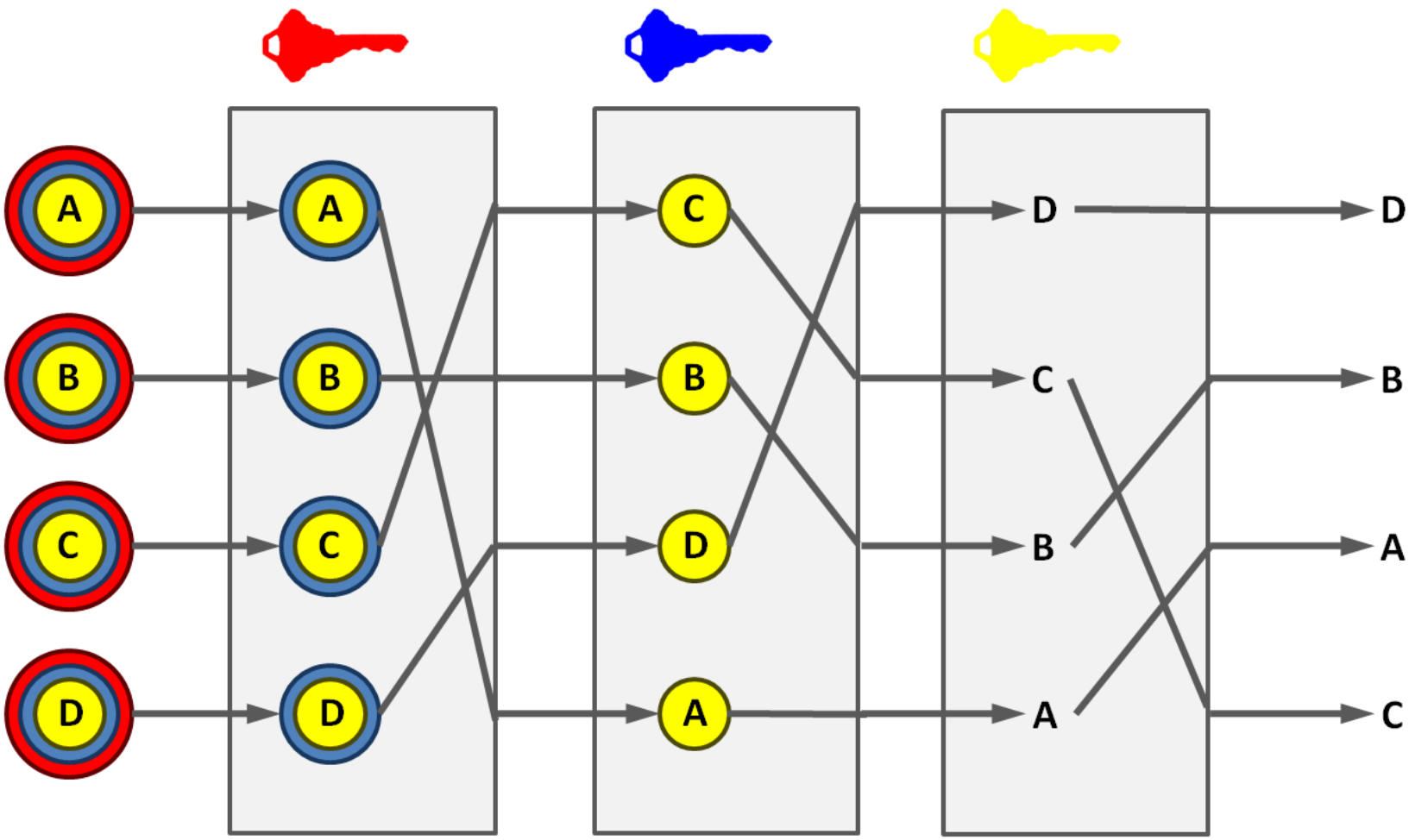


- Make all traffic the same size and look similar by using encrypted packets
- Make the input/output analysis more difficult by batching and adding cover traffic
- At each node every T unit of time:
  - Wait for N messages before forwarding
  - Forward 60% of the messages
  - Add dummy traffic that goes 1-4 hops
- Goal: even if the adversary observes all packets in a round, we want him to have 1/(Total Messages) chance to guess the sender/recipient pair

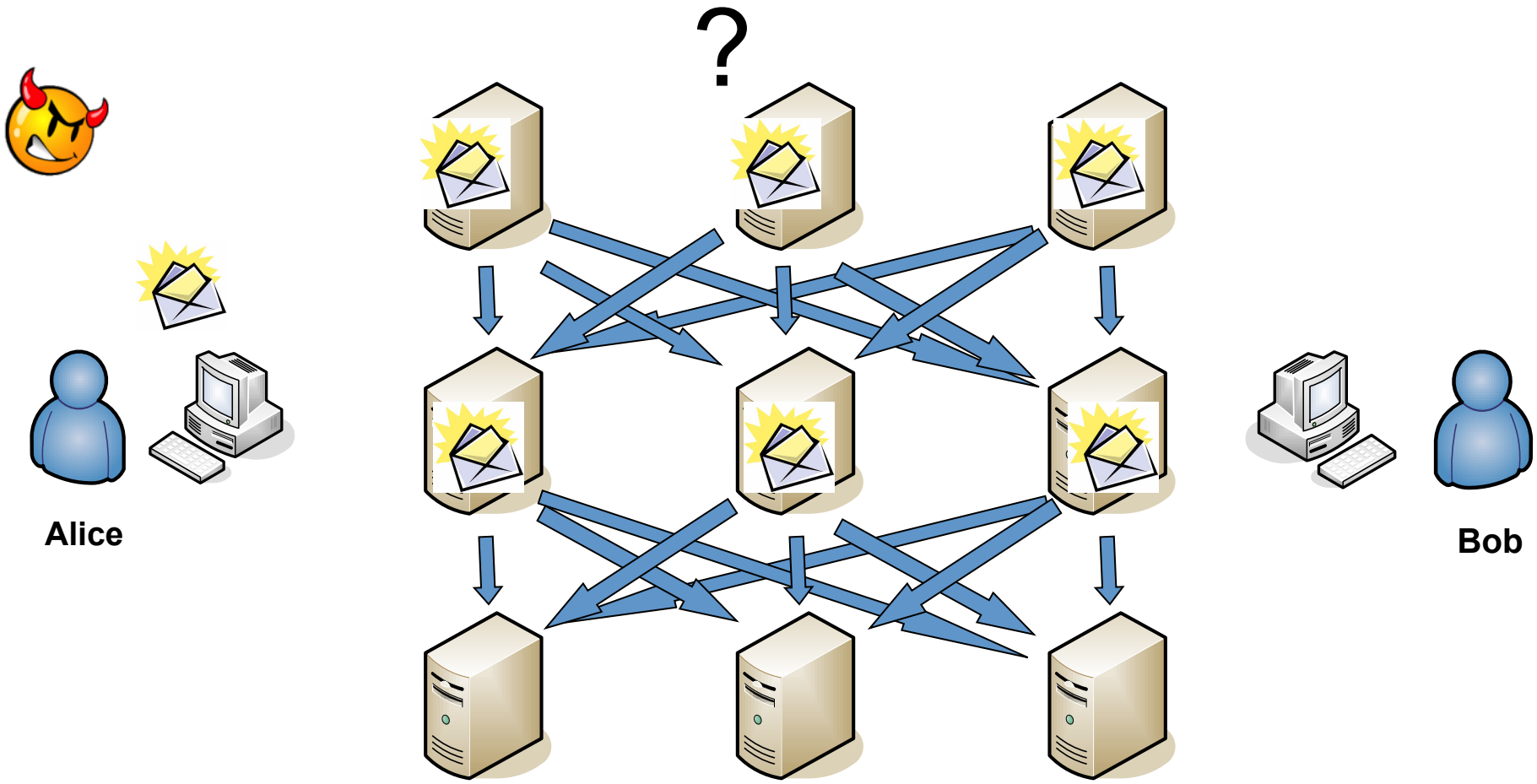
Mix Nets

- Increase the anonymity set:
  - Use multiple routers as proxies to forward traffic each with their own public/private key.
  - Source route traffic through the network with layered encryption of the headers.
  - Each client maintains a full view of the network and chooses proxies uniformly at random from the set.

Mix Nets Overview



Mix Nets cont.



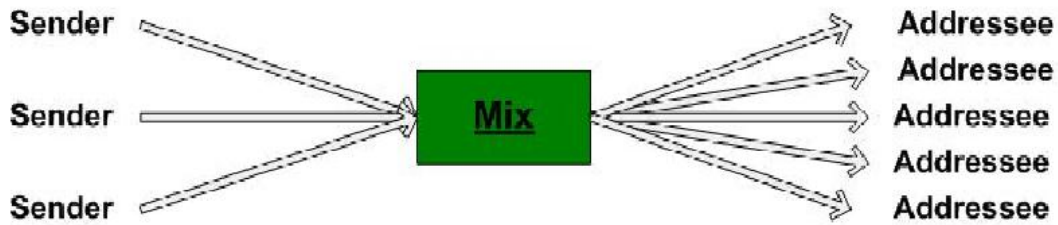
Mix Nets: Conclusions

- Mix Nets are great because:
  - Resistant to global passive adversary.
  - Resistant to adversary controlling nodes.
  - Impossible to correlate traffic because all traffic is identical.
- Mix Nets are bad because:
  - They are high latency
    - Only send one batch every time period T.
    - Do not send if less than N messages.
    - Do not send with approximately 40% probability.

Tor

- Mixnets provide high privacy guarantees but are unusable for many applications
- Goal: Design a low-latency anonymous communication system based on onion routing
- TOR: The Onion Router

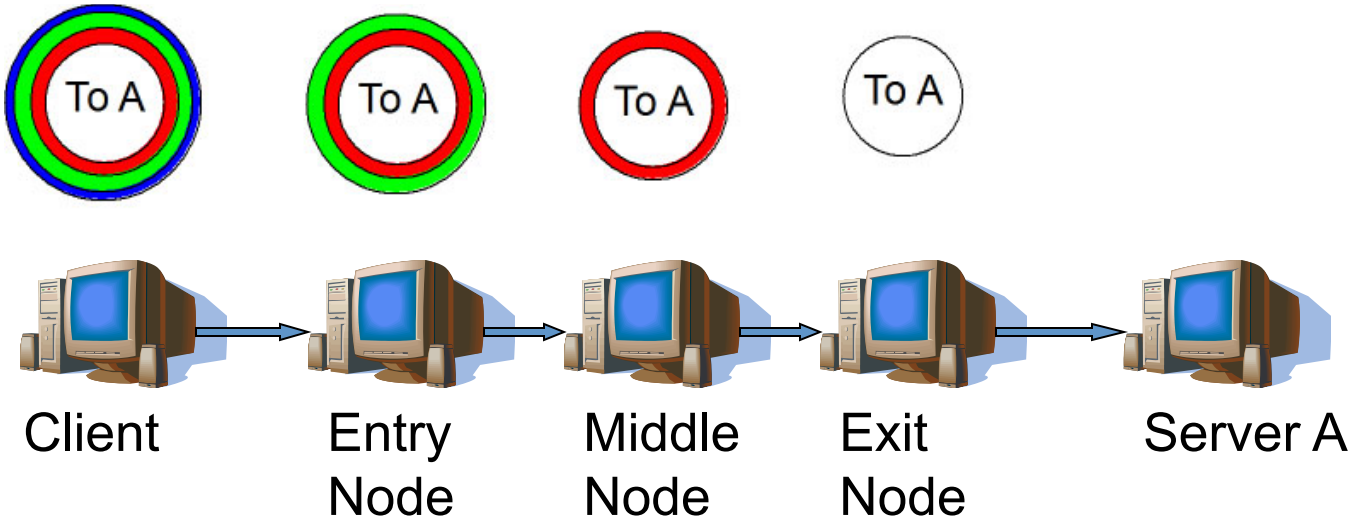
Tor: Low Latency Proxy Design



- To maintain low latency:
  - Nodes send the packets along the circuit as quickly as possible.
  - No batching and no rounds.
  - No cover traffic is added.
  - While traffic is sent in 512 B cells, the number of cells is unique depending on the load.

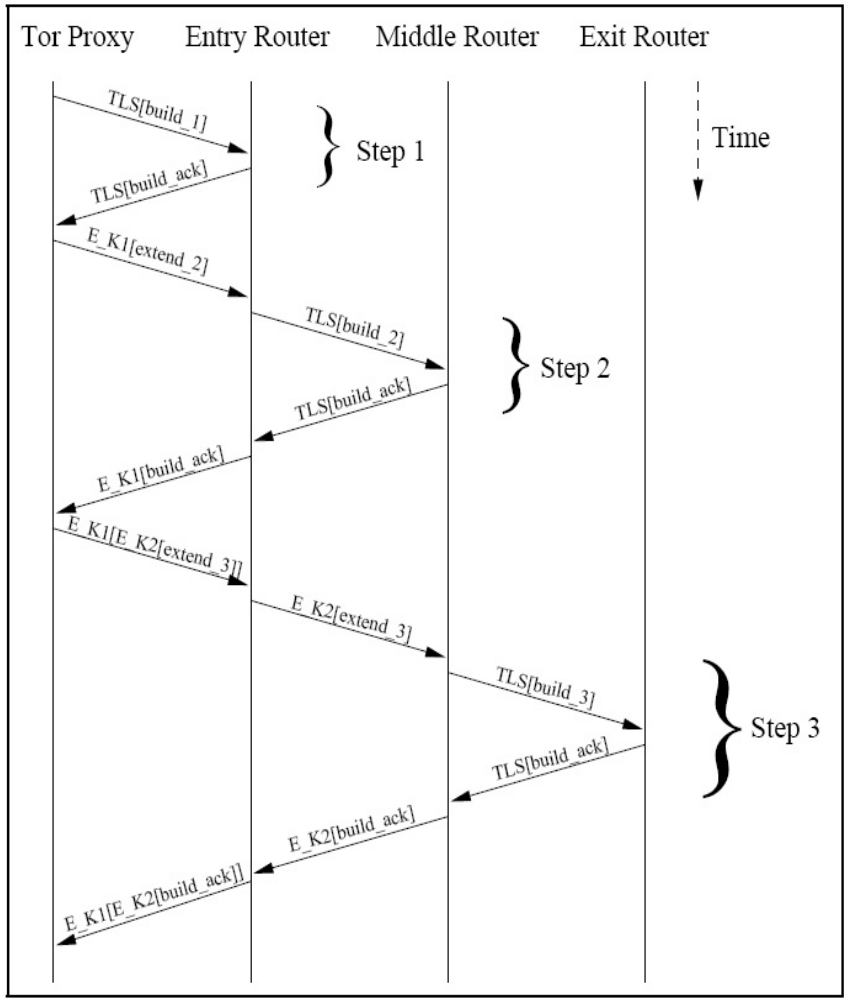
Tor: Onion Routing

- Client chooses three nodes to establish a circuit to send/ receive traffic every ten minutes.
  - Encrypt traffic so each onion router only sees the previous and next hop address
  - Only the entry router knows the client
  - Only the exit router knows the destination
  - Example: $E_G(M||E_M(EX||E_{EX}(A||ToA)))$



Tor: Security Problem

- Susceptible to end-to-end traffic analysis
  - If the attacker controls entry and exit nodes
  - Correlate traffic coming into entry and exiting exit node to identify the client and server
- Thus Tor assumes a limited global passive adversary!!!!



Tor: Conclusions

- Tor is Popular
  - The most widely deployed overlay network implementing onion routing
  - Currently* 2,500 volunteer nodes running forwarding onion routers
  - Currently* an estimated 250,000 users
- Protects end-to-end privacy as long as the adversary cannot compromise traffic going into the system and traffic coming out of the system.

*As of January 2010

Papers

- George Danezis, Roger Dingledine, Nick Mathewson, "Mixminion: Design of a Type III Anonymous Remailer Protocol," Security and Privacy, IEEE Symposium on, p. 2, 2003 IEEE Symposium on Security and Privacy, 2003
- C. Gülcü and G. Tsudik. Mixing E-mail with Babel. In *Network and Distributed Security Symposium (NDSS 96)*, pages 2-16. IEEE, February 1996.
- Kevin Bauer , Damon McCoy , Dirk Grunwald , Tadayoshi Kohno , Douglas Sicker, Low-resource routing attacks against tor, Proceedings of the 2007 ACM workshop on Privacy in electronic society, October 29-29, 2007, Alexandria, Virginia, USA
- George Danezis and Claudia Diaz. A survey of anonymous communication channels. Technical Report MSR-TR-2008-35, Microsoft Research, January 2008.

Chipping Away at Censorship Firewalls

Background

- As of 2010, 59 countries filter Internet traffic China, Saudi Arabia, Iran, … But also France and Australia have recently garnered attention with controversial filtering practices

- Problem: Present systems use proxies which can be detected and blocked by censors

Requirements

- **Availability**
  Even if a censor might block parts of the resources, the system should keep going

- **Deniability**
  Avoid to a certain degree that users of the system get identified by censors

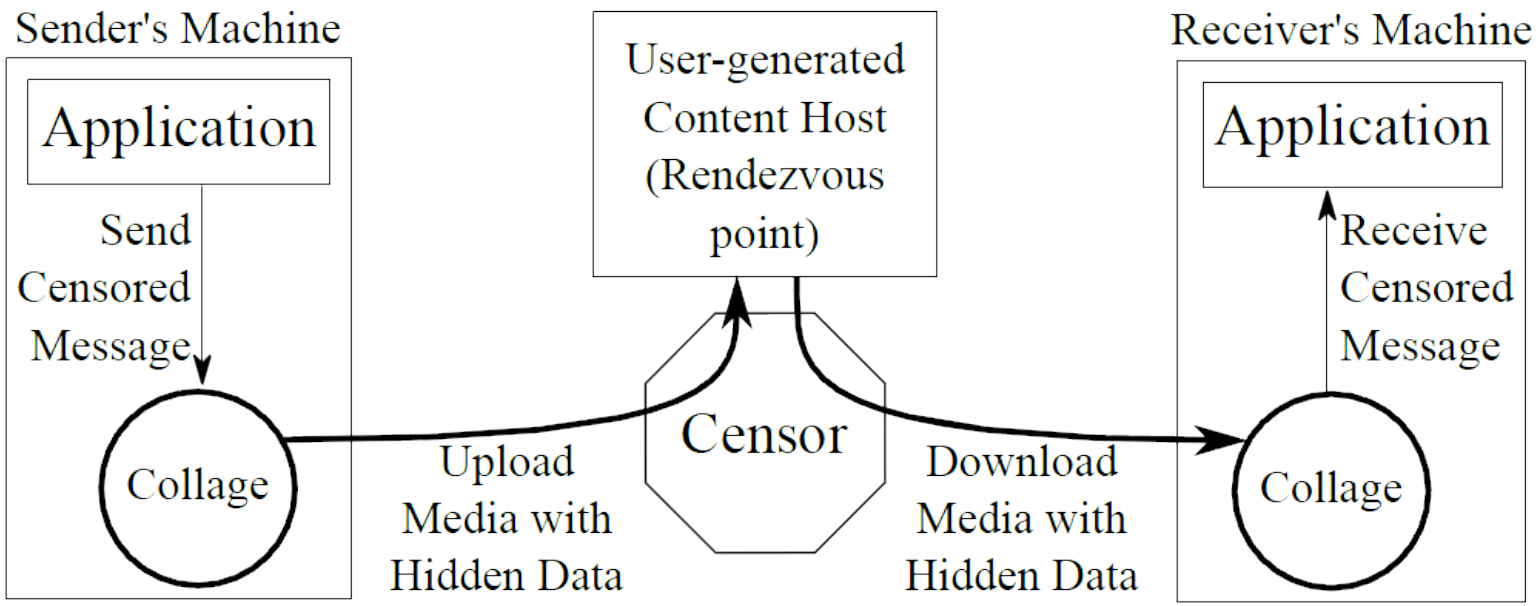
Idea

Make use of user-generated content

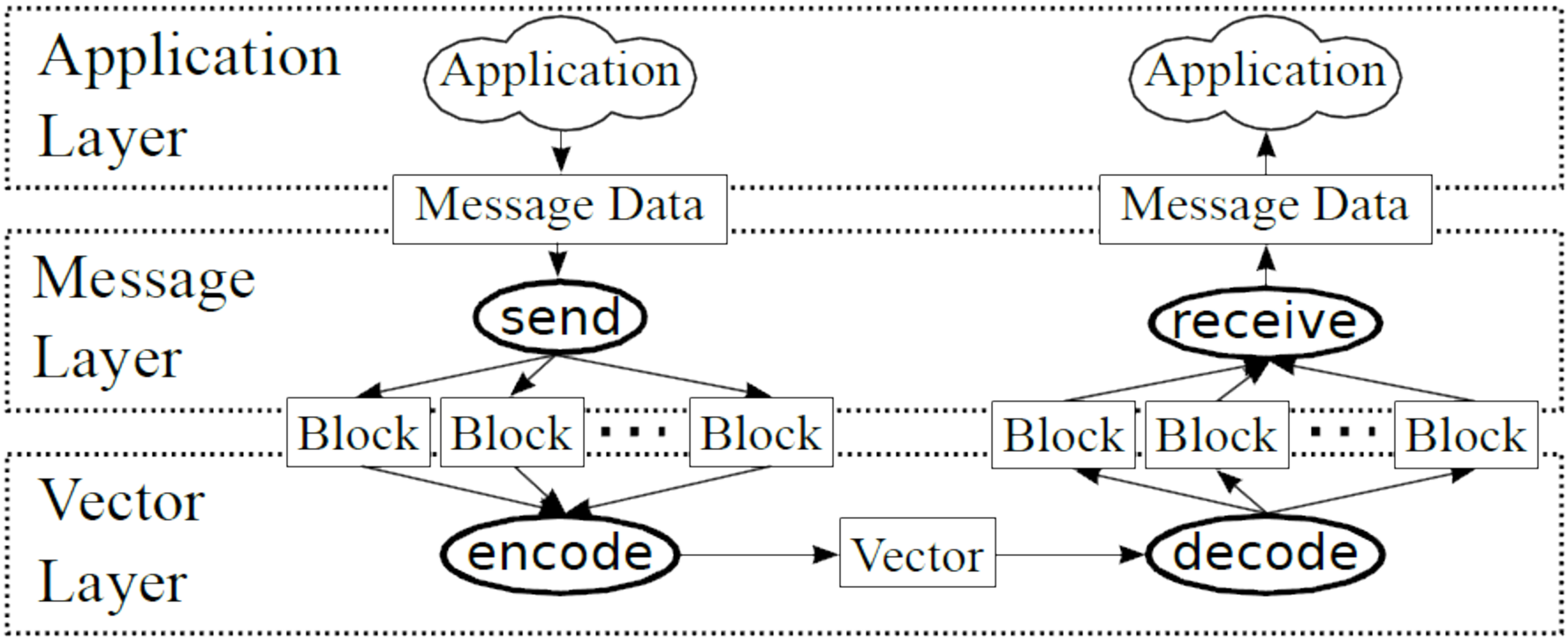
- Embed communication into user-generated content
  i.e. hide data in image, texts or videos

- Use existing infrastructure of the Internet
  i.e. content hosts like Flickr, Picasa, YouTube

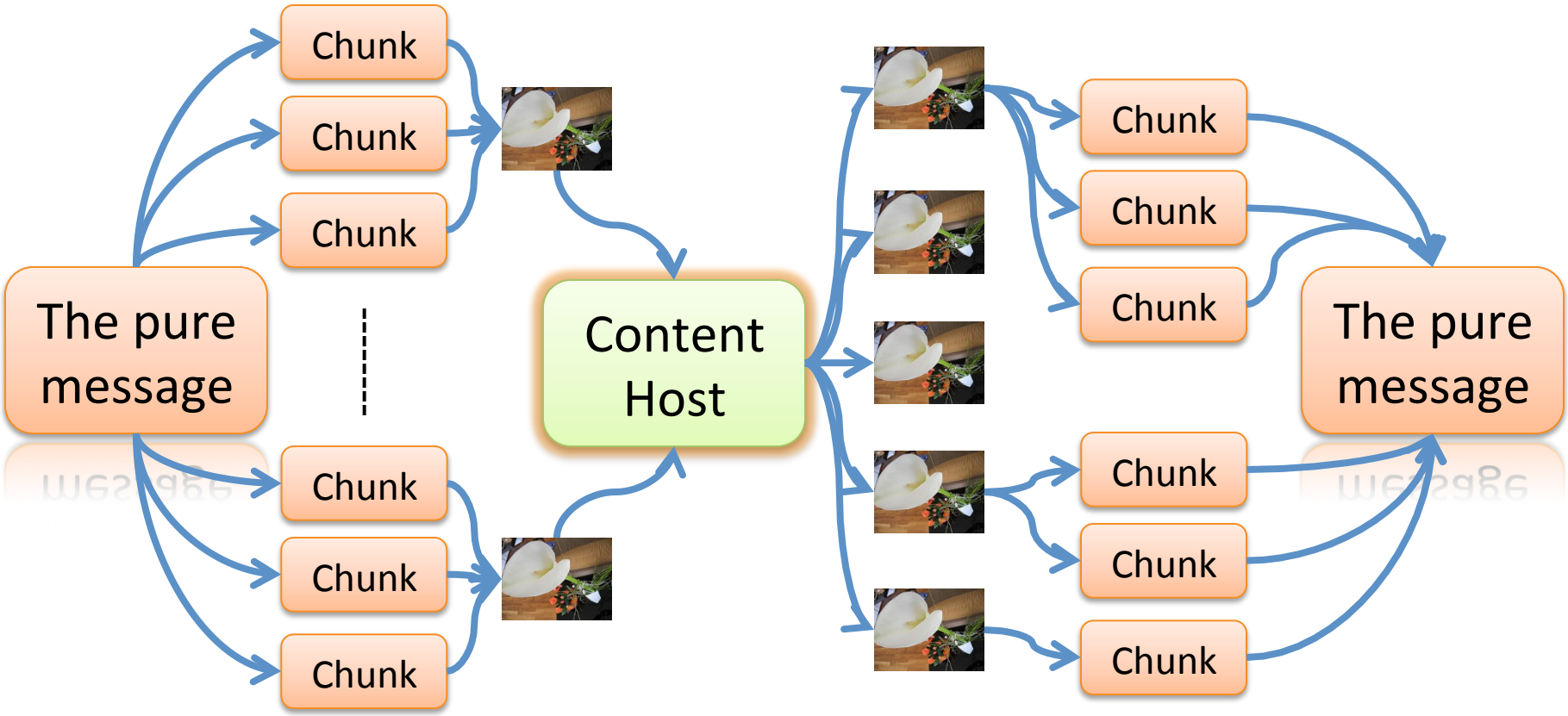
The Collage System



Design of Collage



Design of Collage



Design of Collage

Application Layer

- Message
- Vector *e.g. image, video, …*
- Message Identifier (a pre-shared secret)

- Task: a way to send/receive data using a public data sharing-service
  e.g. „publish a photo tagged with 'flowers'"

Design of Collage

Application Layer

- Task Database
  Contains pairs of tasks

Content host	Sender task	Receiver task
Flickr	PublishAsUser('User', Photo, MsgData)	FindPhotosOfFlickrUser('User')
Twitter	PostTweet('Watching the Olympics', MsgData)	SearchTwitter('Olympics')

Design of Collage

Message Layer

Goals:

- Split the message into smaller chunks
- Hide the chunks in user content
- Deliver/Receive the user content

Design of Collage

Message Layer

Send function:

- Uses a rateless erasure encoder to split a message into chunks
- Encrypts the blocks from the encoder using the message identifier
- Encodes the cipher text into a vector

Design of Collage

Message Layer

Receive function:

- Looks for possible vectors with hidden data in content hosts
- Decodes the cipher text from the user content
- Decrypts the cipher text using the message identifier
- When enough chunks have arrived, the rateless erasure decoder restores the original message

Design of Collage

Vector Layer

- Is responsible for en-/decoding small chunks of data into a *vector*
- Relies on information hiding technology *e.g. Steganography, Digital Watermarking*

Design of Collage

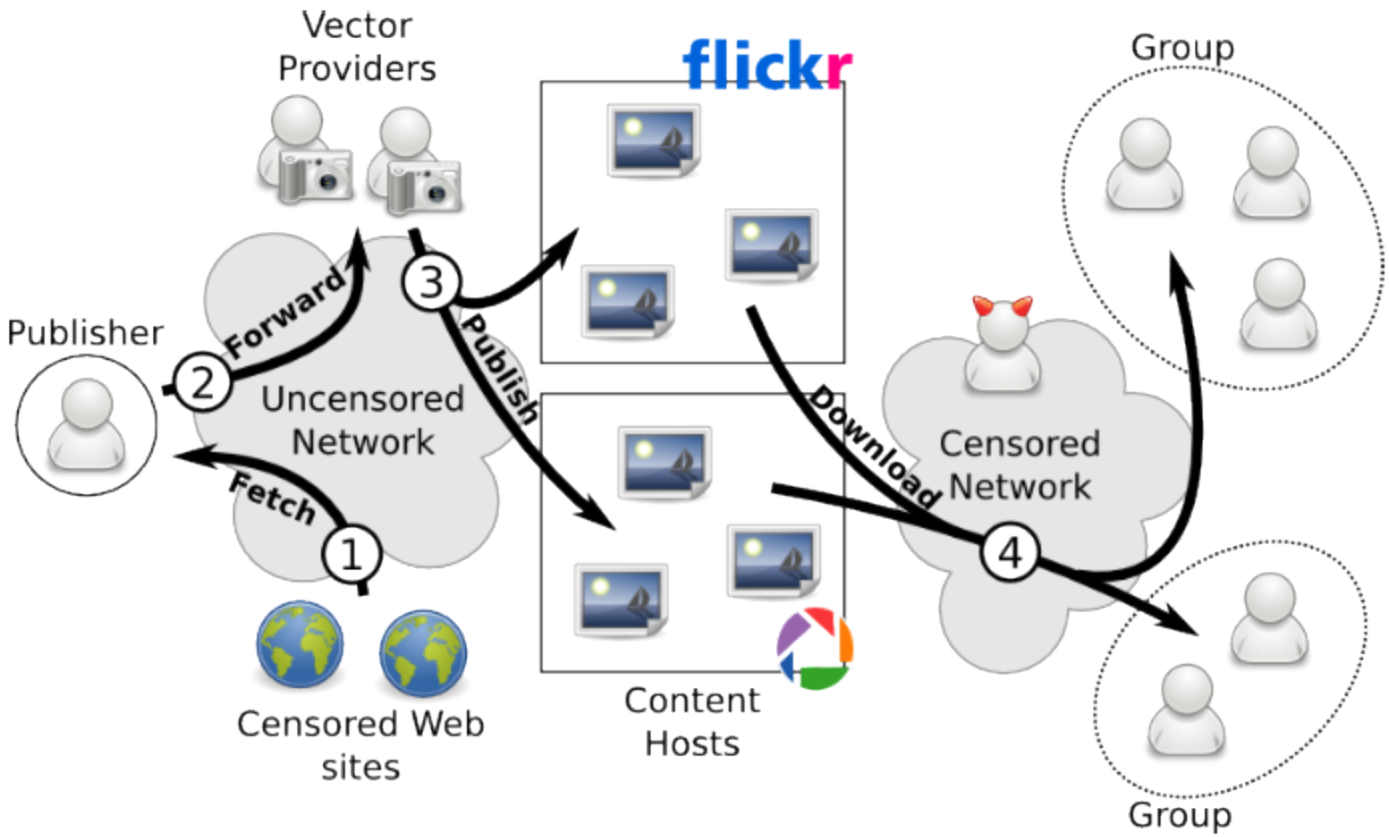
Rendezvous

- Use a hash-function
- Hash all tasks and store the resulting table

- To find the tasks to use for a given message, hash the message identifier and take *m* closest tasks
- The hash table has to be exchanged by all Collage members

Example Implementation

Web Content Proxy



Evaluation

- Big overhead in computation, storage and transfer time, depending on verious factors:
  - Type of user-generated content
  - Method of encoding vectors
  - Choosen Tasks

- For instance: Web Proxy requires 9 JPEGs to transmit 23 KB of news summary
  Transfer time approx. 1 minute

Guarantees

- None actually ☺

- Achieving *availability* and *deniability* are both best effort

Papers

- BURNETT, S., FEAMSTER, N., AND VEMPALA, S. Chipping away at censorship with user-generated content. USENIX Security Symposium (2010).
- George Danezis, Roger Dingledine, Nick Mathewson, "Mixminion: Design of a Type III Anonymous Remailer Protocol," Security and Privacy, IEEE Symposium on, p. 2, 2003 IEEE Symposium on Security and Privacy, 2003
- C. Gülcü and G. Tsudik. Mixing E-mail with Babel. In *Network and Distributed Security Symposium (NDSS 96)*, pages 2-16. IEEE, February 1996.
- Kevin Bauer , Damon McCoy , Dirk Grunwald , Tadayoshi Kohno , Douglas Sicker, Low-resource routing attacks against tor, Proceedings of the 2007 ACM workshop on Privacy in electronic society, October 29-29, 2007, Alexandria, Virginia, USA
- George Danezis and Claudia Diaz. A survey of anonymous communication channels. Technical Report MSR-TR-2008-35, Microsoft Research, January 2008.

Discussion

- Switching the perspective: Can you think of any good argument which may legitimate censorship?
- Tor sacrifices some anonymity for lower latency. Do you think that Tor's assumptions about the adversary are realistic with respect to users' need for privacy?
- Avoiding censorship requires hiding the act of communication. The method here utilized user generated content. What are some other "hidden channels" of communication available for such systems?