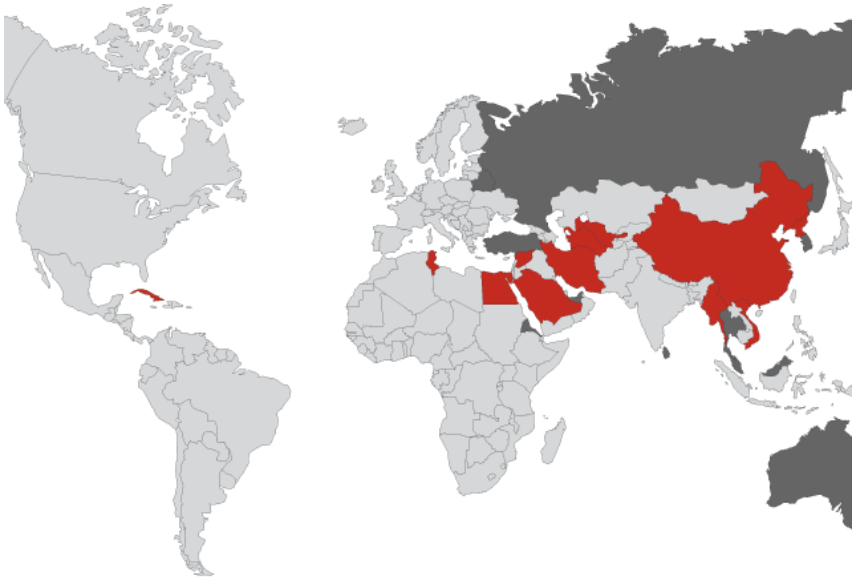
Chipping Away at Censorship with User-Generated Content

Sam Burnett, Nick Feamster and Santosh Vempala

Georgia Tech College of Computing

Internet Censorship is a Problem



- 12 censors
- 11 monitors
- More on the way
- Some censors have fastest growth in Internet usage

See http://rsf.org for more

It’s Not Only China…at Home, Too

Censorship: Labor's hidden policy

By Nick Ross

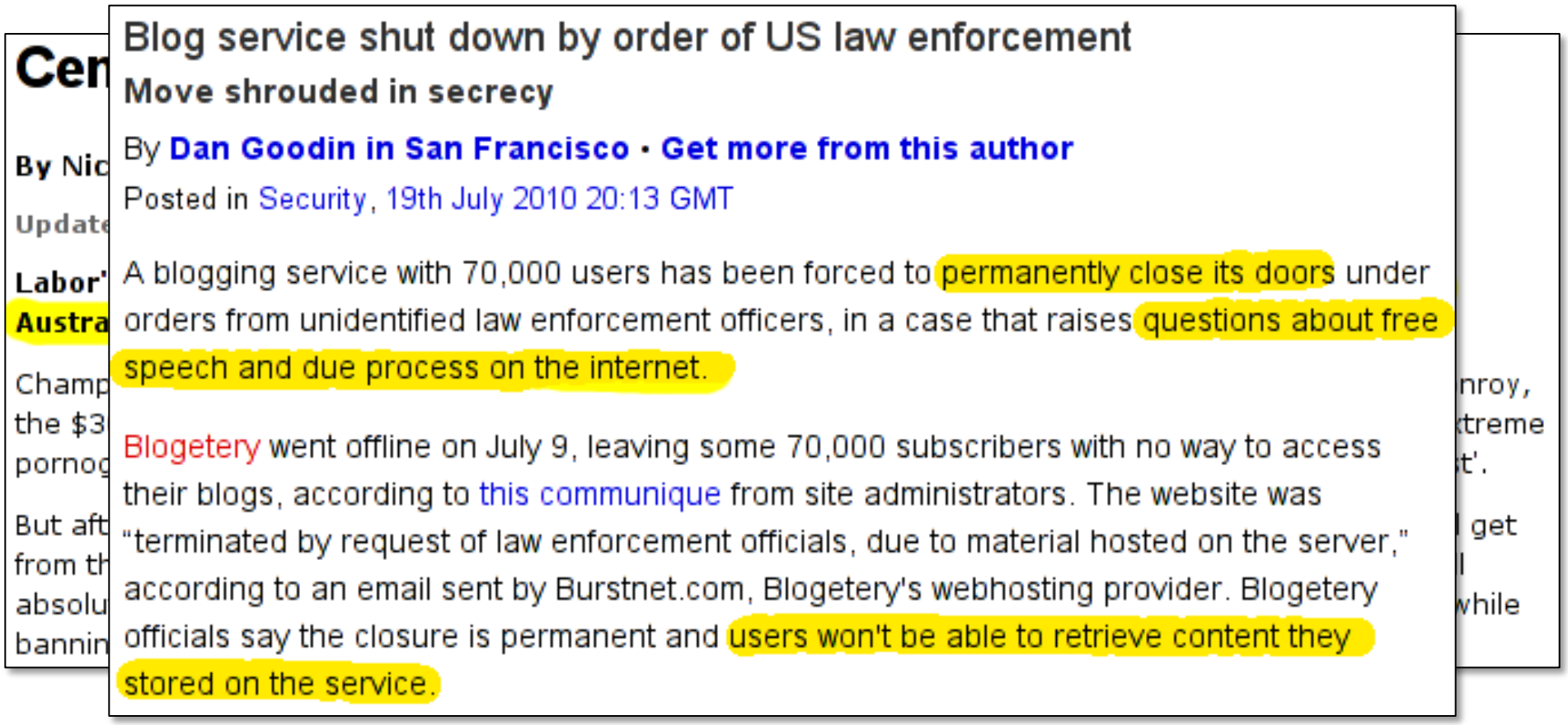
Updated Wed Jul 21, 2010 4:17pm AEST

Labor's internet filtering policy isn't being discussed in the run-up to the election but its impact on Australia is significant.

Championed by Minister for Broadband, Communications and the Digital Economy, Senator Stephen Conroy, the $30million+ filter is being sold by Labor as an internet block for child pornography, bestiality and extreme pornography with 'wide ranging support from the Australian public' and 'only minimal opposition against'.

But after a new, lengthy investigation it transpires that virtually none of this is true. What Australia will get from this internet filter is a framework for censorship that doesn't stop "the worst of the worst" but will absolutely curtail discussion on politically incorrect topics like euthanasia, safe drug taking and graffiti while banning relatively-tame adult content.

It’s Not Only China…at Home, Too



Blog service shut down by order of US law enforcement

Move shrouded in secrecy

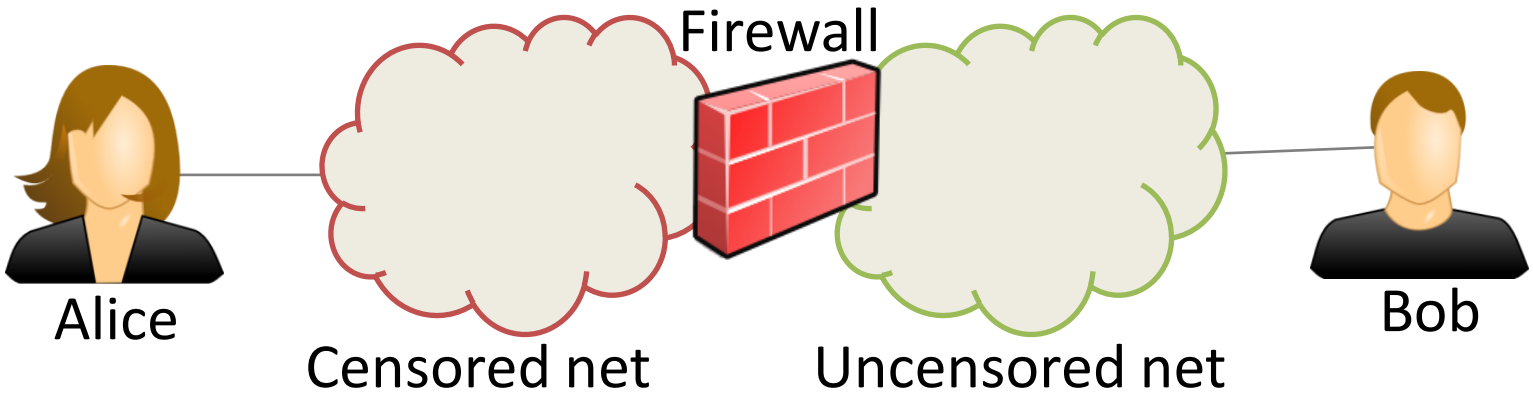
By **Dan Goodin in San Francisco** • **Get more from this author**

Posted in Security, 19th July 2010 20:13 GMT

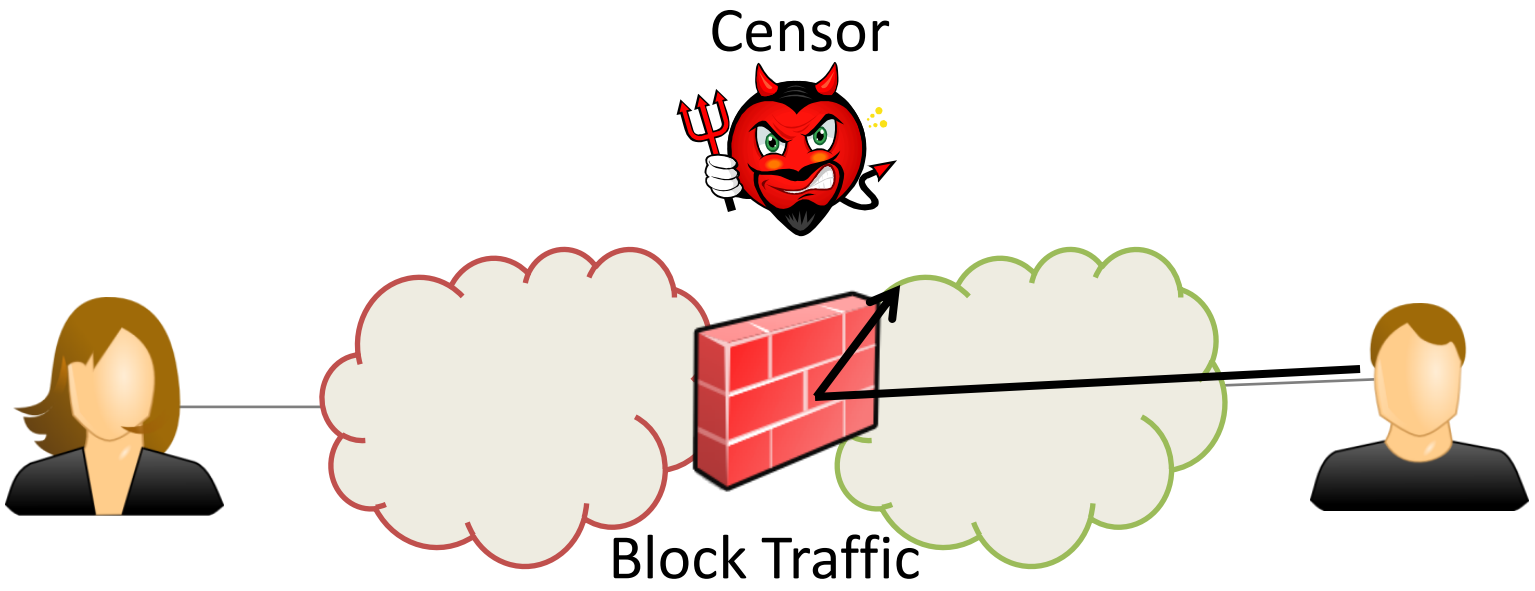
A blogging service with 70,000 users has been forced to permanently close its doors under orders from unidentified law enforcement officers, in a case that raises questions about free speech and due process on the internet.

Blogetery went offline on July 9, leaving some 70,000 subscribers with no way to access their blogs, according to this communique from site administrators. The website was "terminated by request of law enforcement officials, due to material hosted on the server," according to an email sent by Burstnet.com, Blogetery's webhosting provider. Blogetery officials say the closure is permanent and users won't be able to retrieve content they stored on the service.

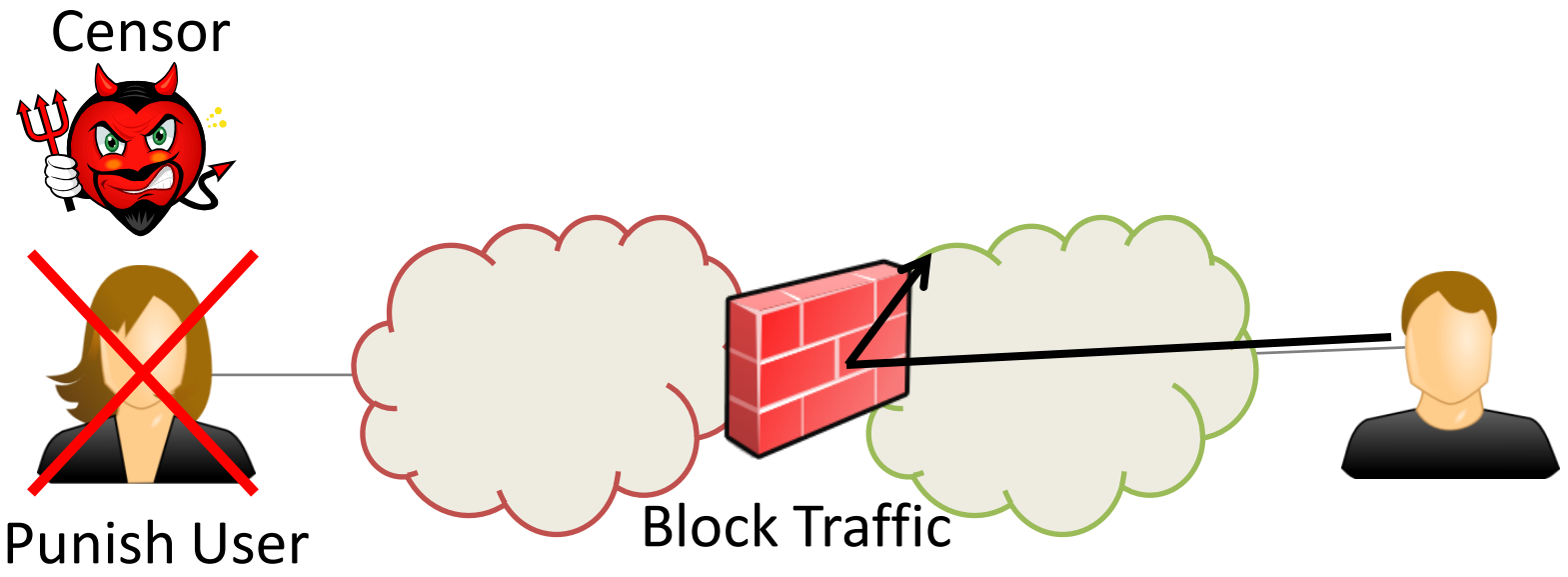
Intro to Internet Censorship



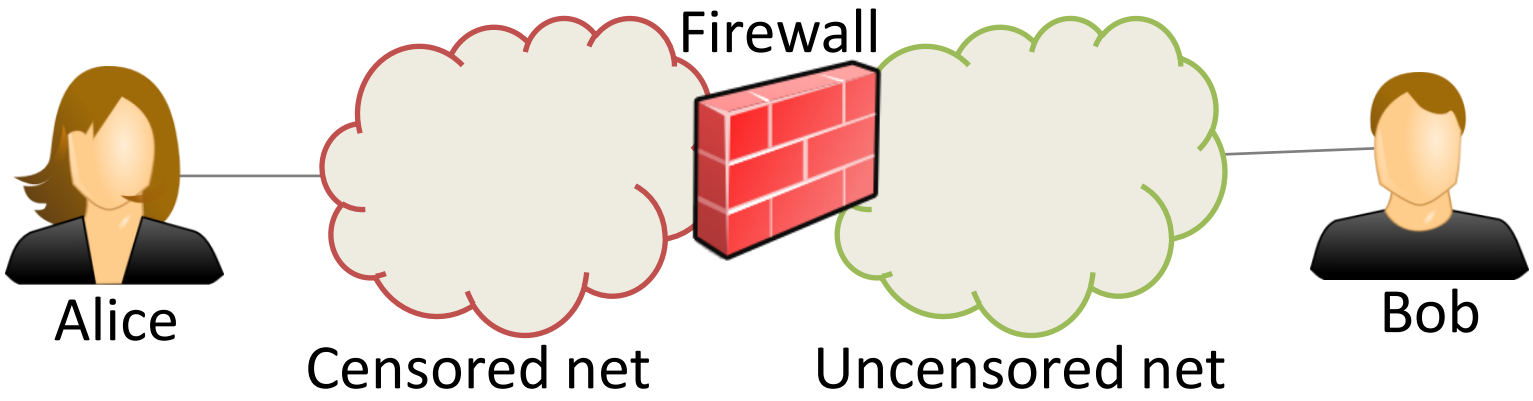
Intro to Internet Censorship



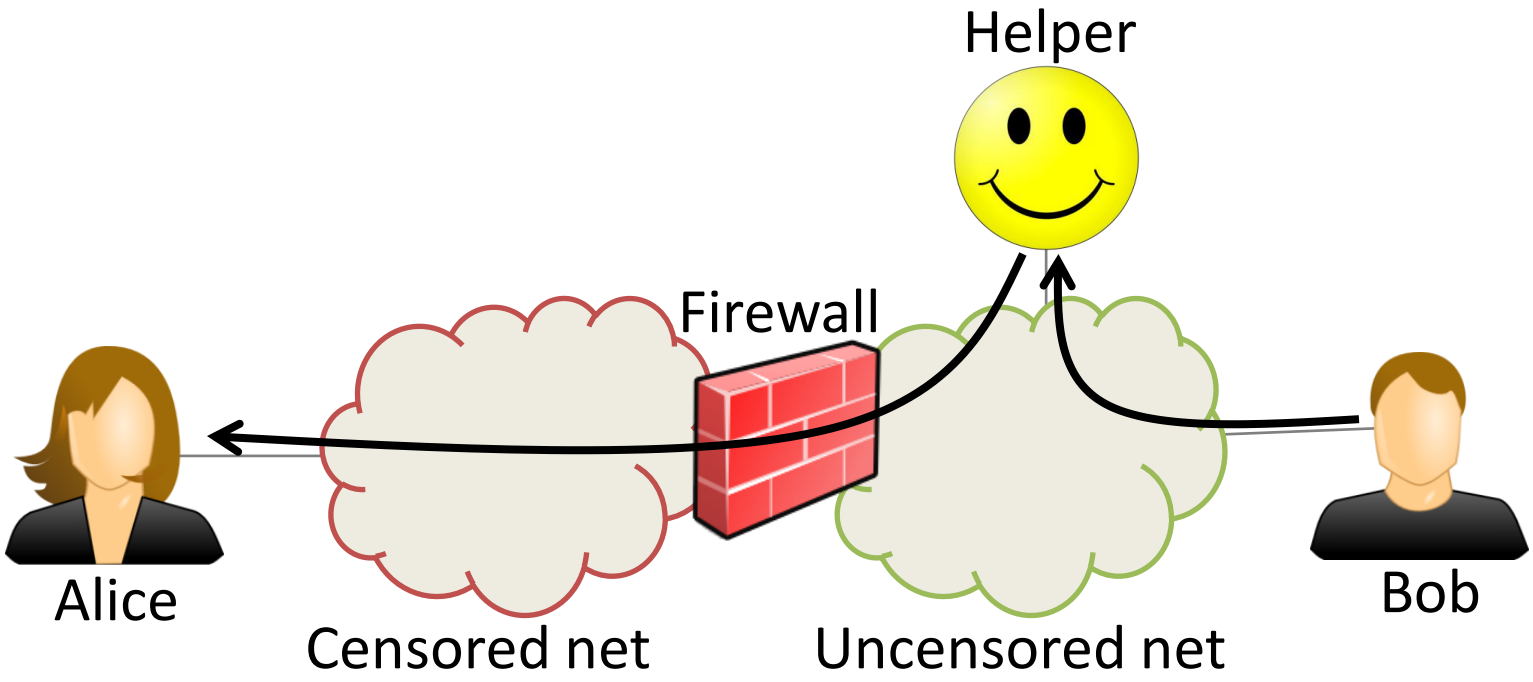
Intro to Internet Censorship



Solution: Use a Helper



Solution: Use a Helper

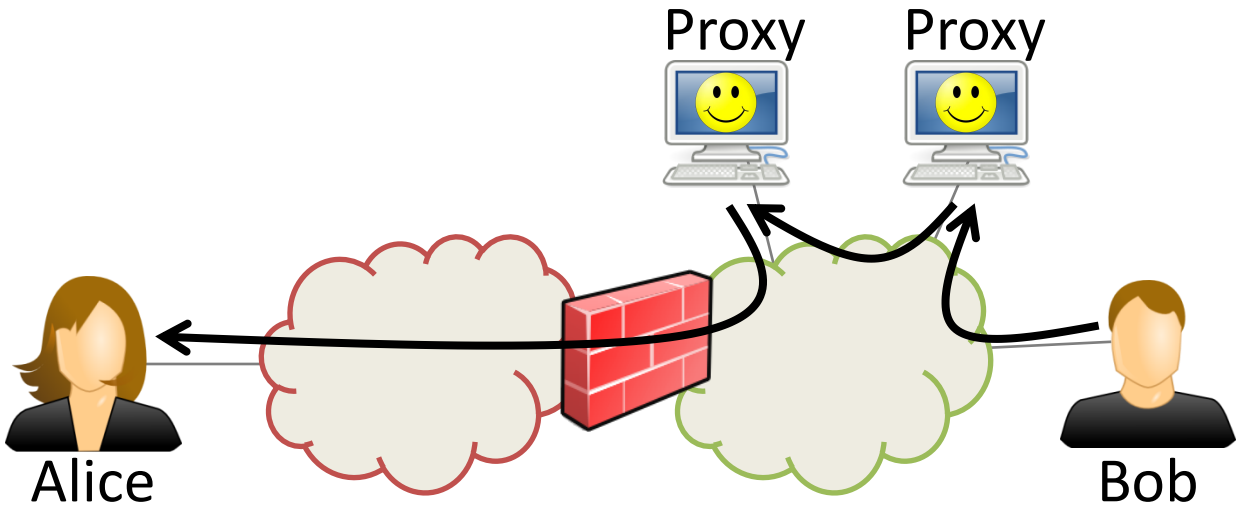


The helper sends messages to and from blocked hosts on your behalf

Design Goals for the Helper

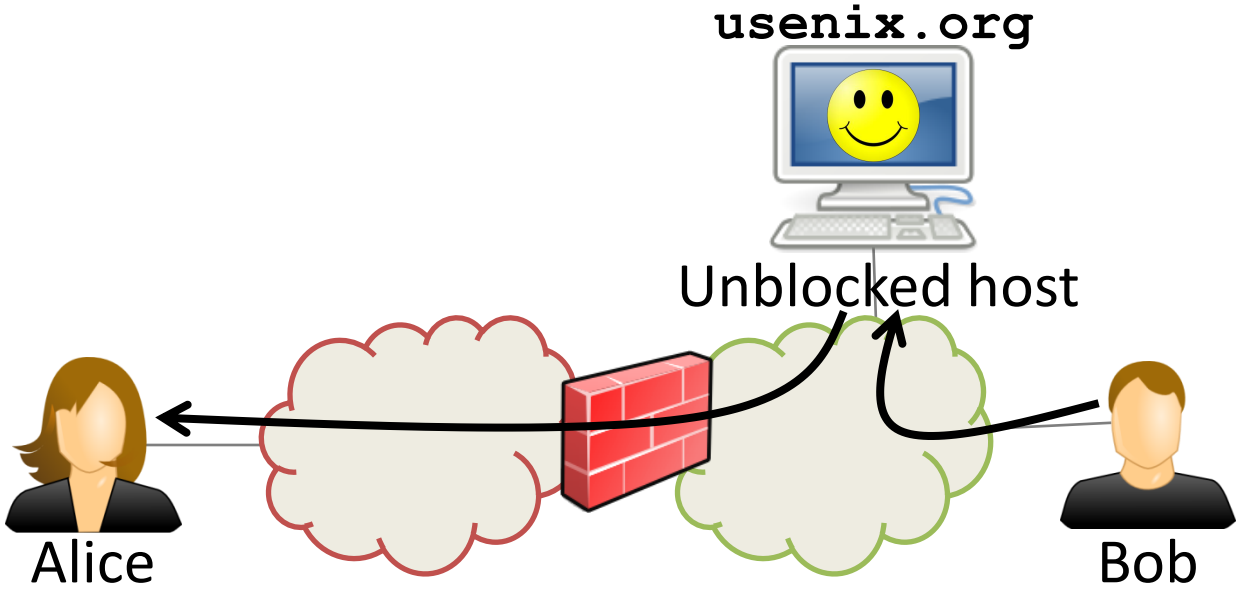
- Be **robust** against blocking
- Be **deniable** against user identification
- Require **no dedicated infrastructure**

What about Proxies and Mixnets?
(e.g., Tor)



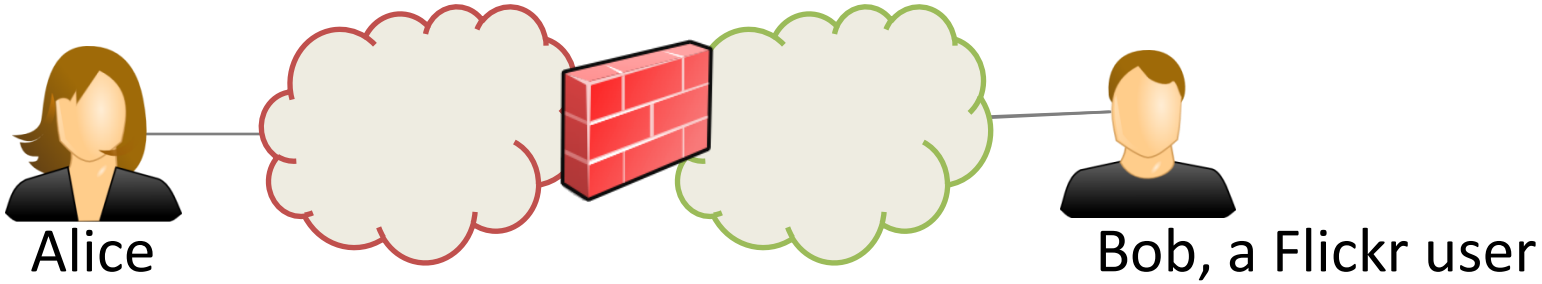
- Censors can **block proxies** if the proxy list is public
- **Not deniable** if encryption is incriminating
- **Requires dedicated infrastructure** (network of proxies)

What About Covert Channels?
(e.g., Infranet)

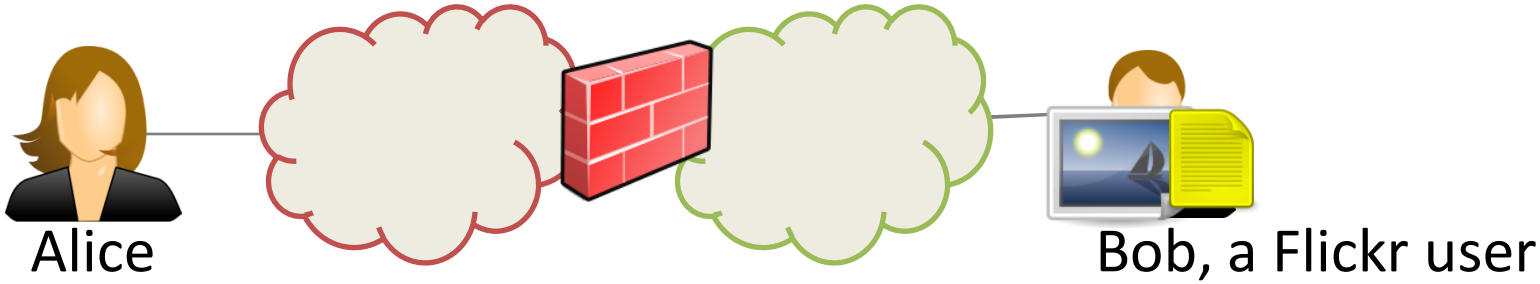


- **Not entirely robust** against blocking
- **More deniable** because messages are hidden
- **Requires dedicated infrastructure** (Web servers)

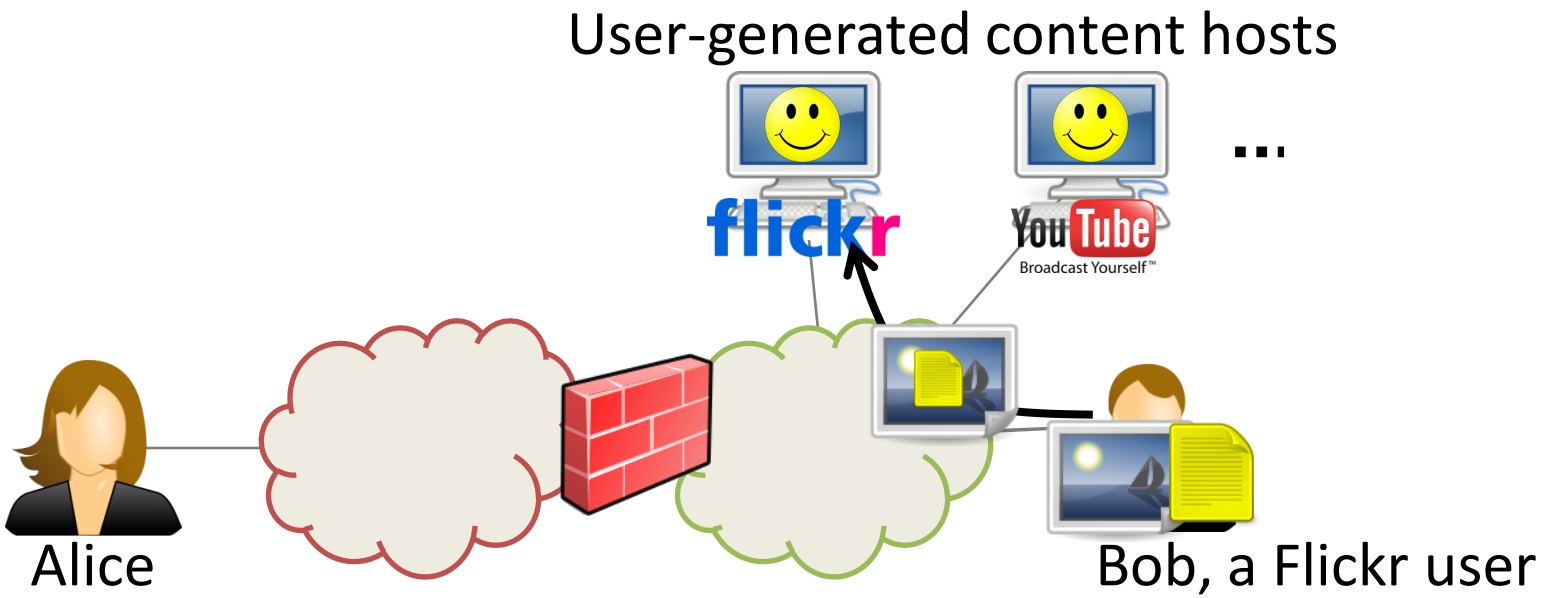
Collage: Let User-Generated Content Help Defeat Censorship



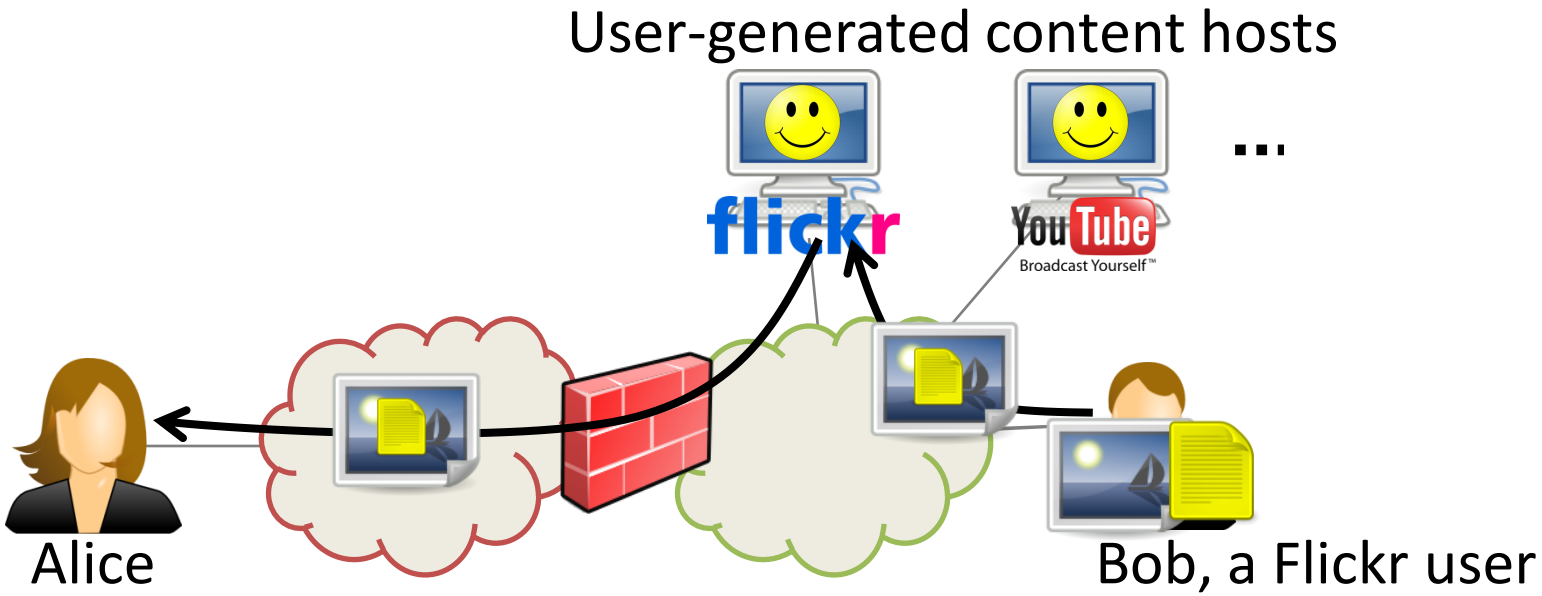
Collage: Let User-Generated Content Help Defeat Censorship



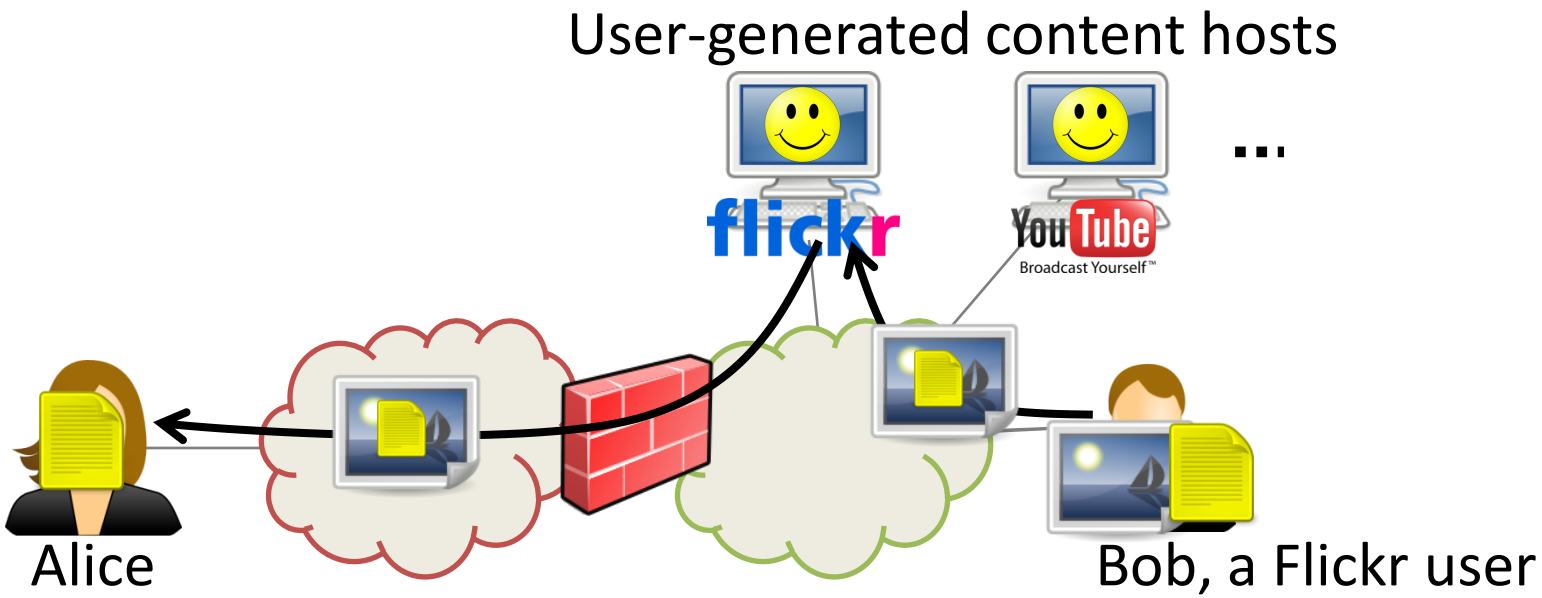
Collage: Let User-Generated Content Help Defeat Censorship



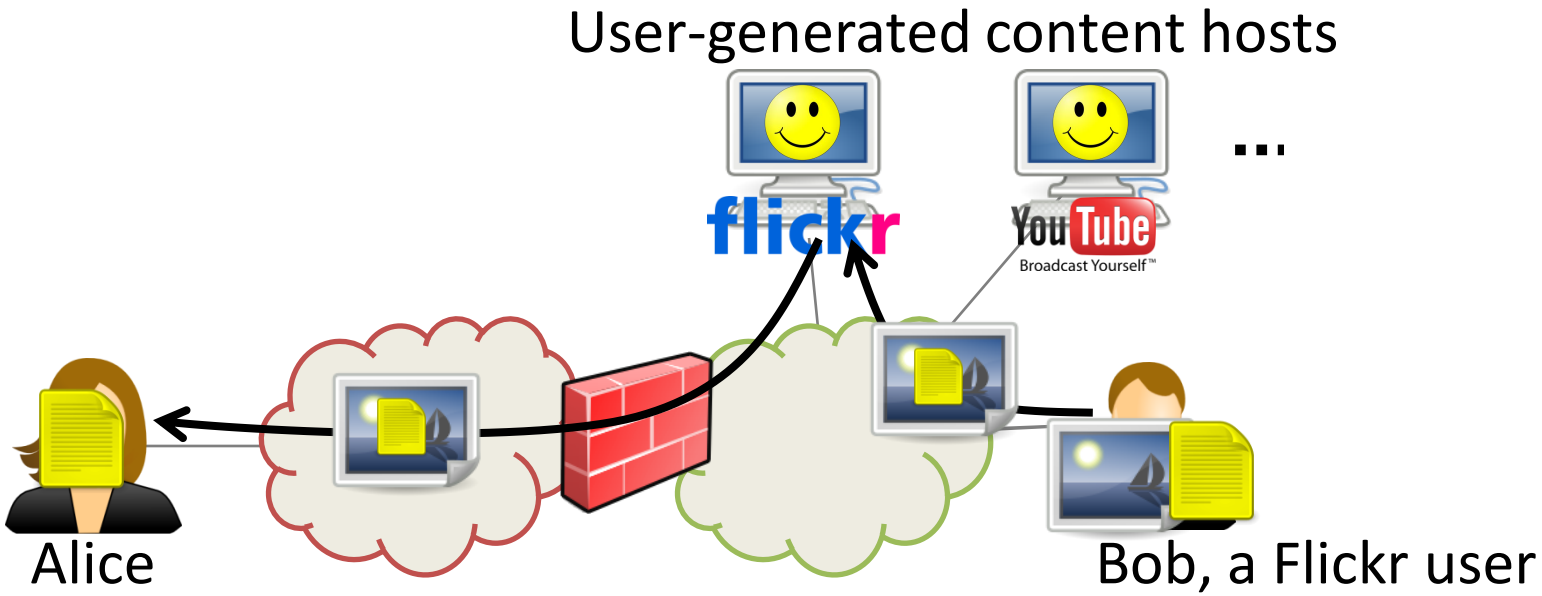
Collage: Let User-Generated Content Help Defeat Censorship



Collage: Let User-Generated Content Help Defeat Censorship



Collage: Let User-Generated Content Help Defeat Censorship



- **Robust** by using redundancy
- Users generate **innocuous-looking traffic**
- **No dedicated infrastructure** required

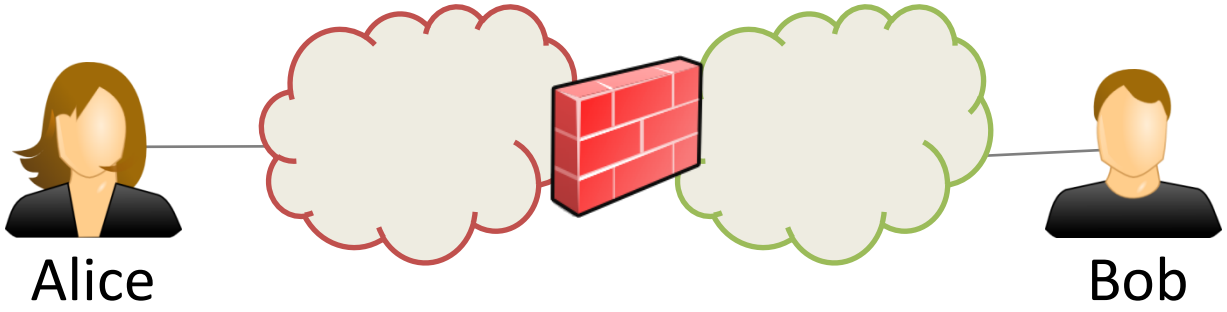
Why Might Collage Work?

- Lots of User-Generated Content (**UGC**)
  - More than 4 billion Flickr images
  - A day of video uploaded to YouTube every minute
- **Many** sites host UGC
- We have tools to **store censored data** in UGC
  - Steganography, watermarking

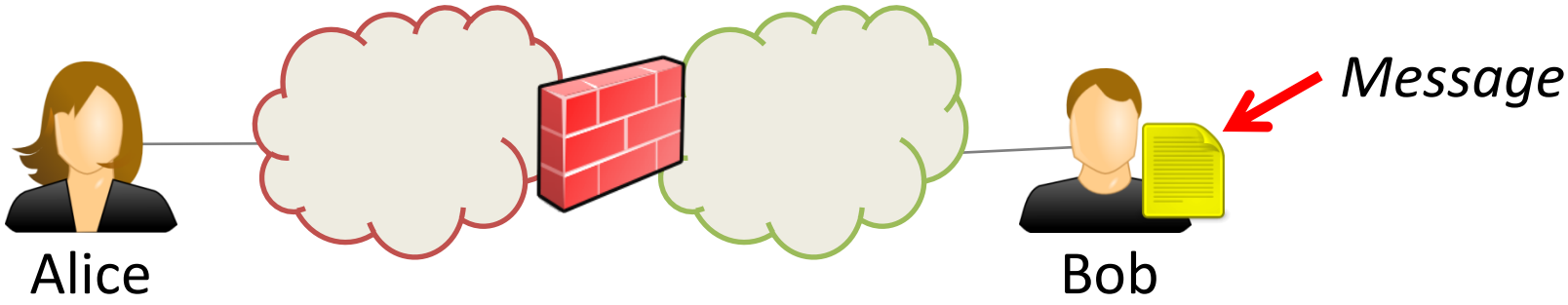
Outline

- Background and Design Goals
- **Collage Design**
- Performance and Demo

Collage, Step-by-Step



Collage, Step-by-Step



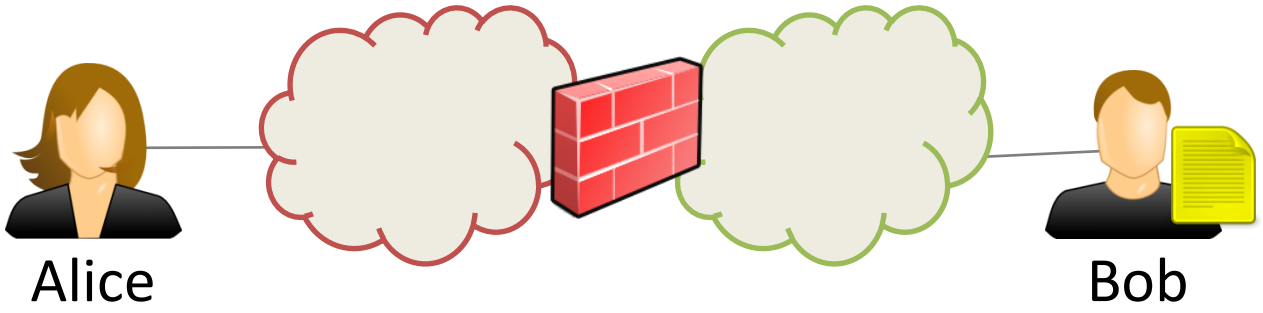
Collage steps:

1. Obtain message
2. Pick message identifier
3. Obtain cover media
4. Embed message in cover
5. Upload UGC to content host
6. Find and download UGC
7. Decode message from UGC

Step 1: Obtain message

- Application specific, not just Web sites

Collage, Step-by-Step



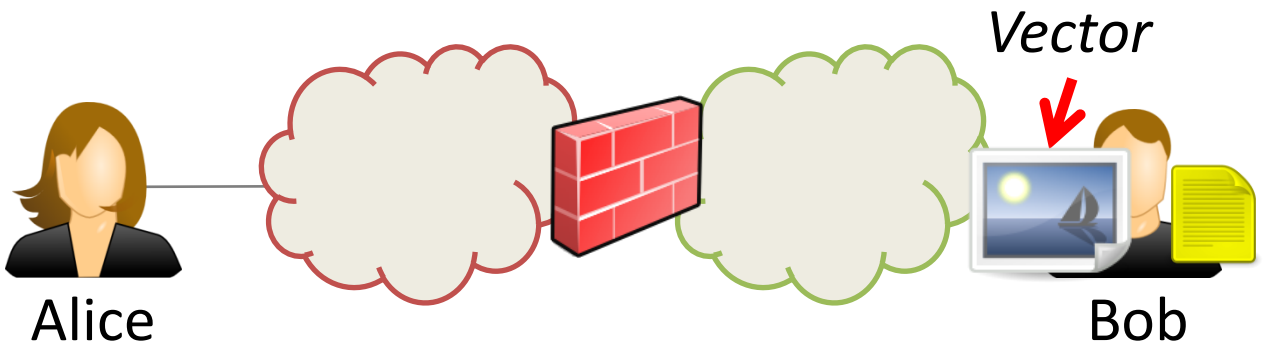
Collage steps:

1. **Obtain message**
2. **Pick message identifier**
3. **Obtain cover media**
4. **Embed message in cover**
5. **Upload UGC to content host**
6. **Find and download UGC**
7. **Decode message from UGC**

Step 2: Pick message identifier

- Application specific
- Only intended recipient should know it

Collage, Step-by-Step



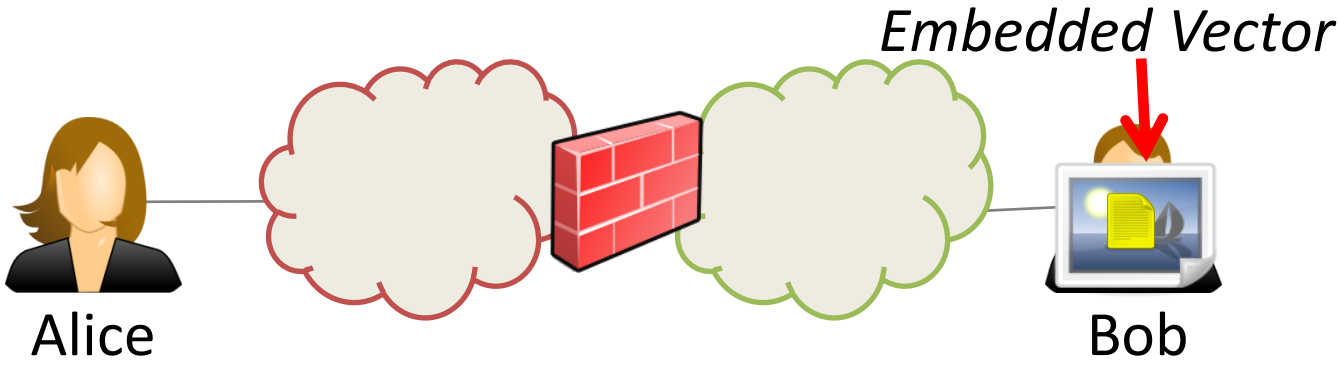
Collage steps:

1. **Obtain message**
2. **Pick message identifier**
3. **Obtain cover media**
4. **Embed message in cover**
5. **Upload UGC to content host**
6. **Find and download UGC**
7. **Decode message from UGC**

Step 3: Obtain cover media

- Your personal photos
- Generous users

Collage, Step-by-Step



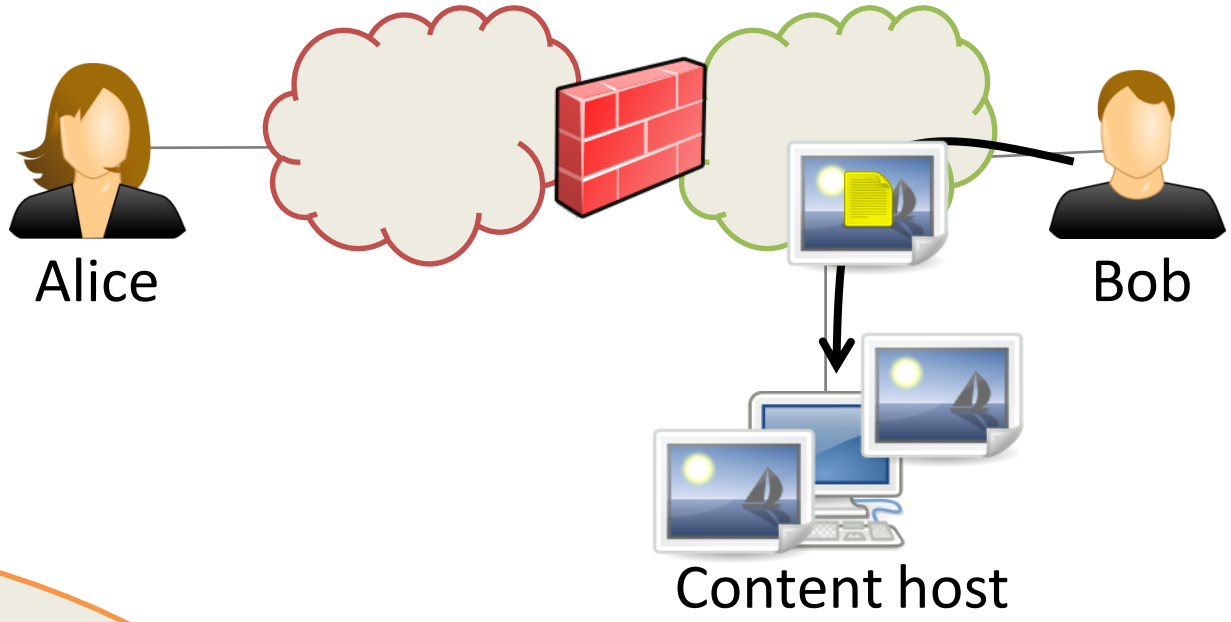
Collage steps:

1. **Obtain message**
2. **Pick message identifier**
3. **Obtain cover media**
4. **Embed message in cover**
5. **Upload UGC to content host**
6. **Find and download UGC**
7. **Decode message from UGC**

Step 4: Embed message in cover

- Encrypt, erasure code, and embed
- Discussed later

Collage, Step-by-Step



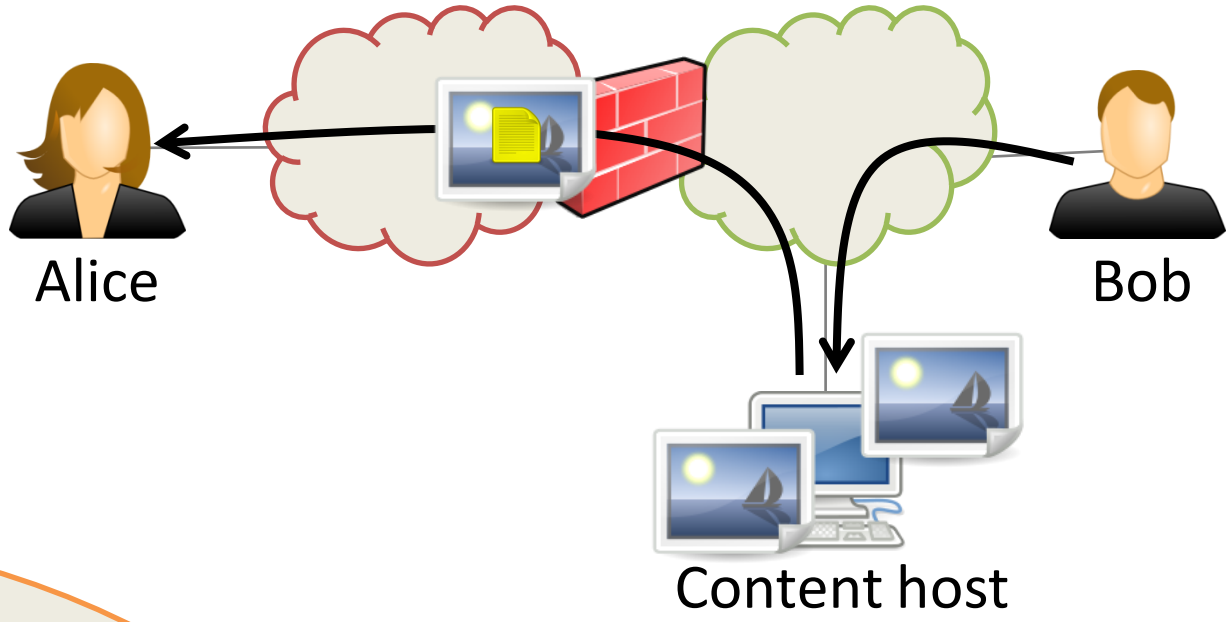
Collage steps:

1. **Obtain message**
2. **Pick message identifier**
3. **Obtain cover media**
4. **Embed message in cover**
5. **Upload UGC to content host**
6. **Find and download UGC**
7. **Decode message from UGC**

Step 5: Upload UGC to content host

- Discussed next

Collage, Step-by-Step



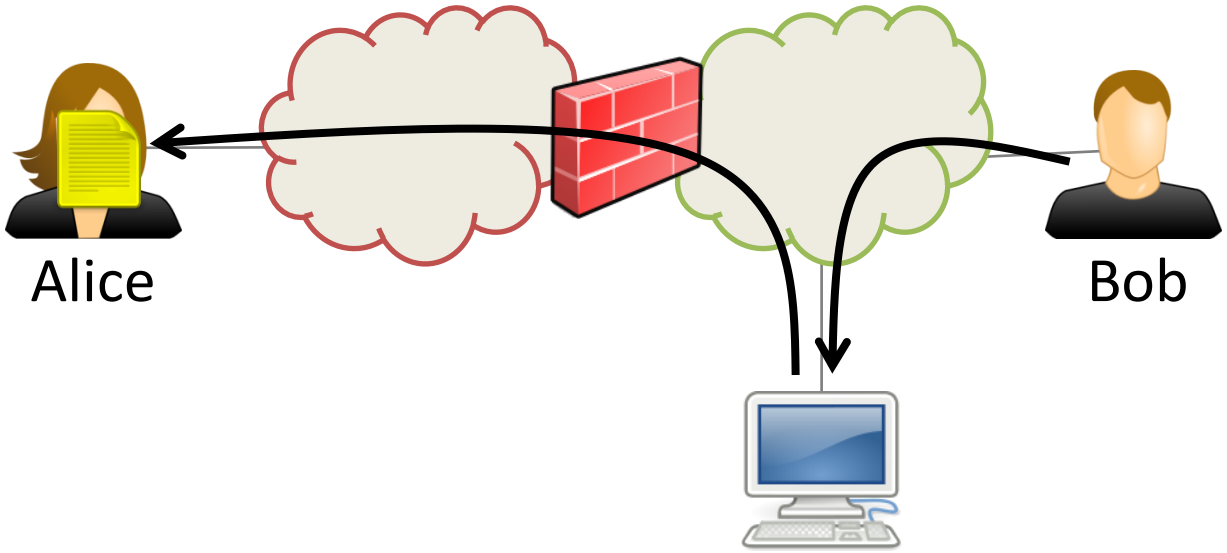
Collage steps:

1. **Obtain message**
2. **Pick message identifier**
3. **Obtain cover media**
4. **Embed message in cover**
5. **Upload UGC to content host**
6. **Find and download UGC**
7. **Decode message from UGC**

Step 6: Find and download UGC

- Discussed later

Collage, Step-by-Step



Collage steps:

1. **Obtain message**
2. **Pick message identifier**
3. **Obtain cover media**
4. **Embed message in cover**
5. **Upload UGC to content host**
6. **Find and download UGC**
7. **Decode message from UGC**

Step 7: Decode message from UGC

- Extract, decode, and decrypt

Embedding Messages in Vectors

- **Encrypt** the message using the identifier
- Generate chunks using **erasure coding**
  - Generate many chunks, recover from *any* k-subset
  - Allows splitting among many vectors, robustness
- **Embed** chunks into vectors

Collage steps:

1. Obtain message
2. Pick message identifier
3. Obtain cover media
4. **Embed message in cover**
5. Upload UGC to content host
6. Find and download UGC
7. **Decode message from UGC**

Steganography: hard to detect

Watermarking: hard to remove

Do the reverse to decode

Agreeing on Vector Locations

- Crawling all of Flickr is not an option
- Need to agree on a subset of the content host without any immediate communication

Solution: A predictable way of mapping message identifiers to subsets of content hosts

Collage steps:

1. Obtain message
2. Pick message identifier
3. Obtain cover media
4. Embed message in cover
5. **Upload UGC to content host**
6. **Find and download UGC**
7. Decode message from UGC

Solution: Task Mapping

Message Identifier

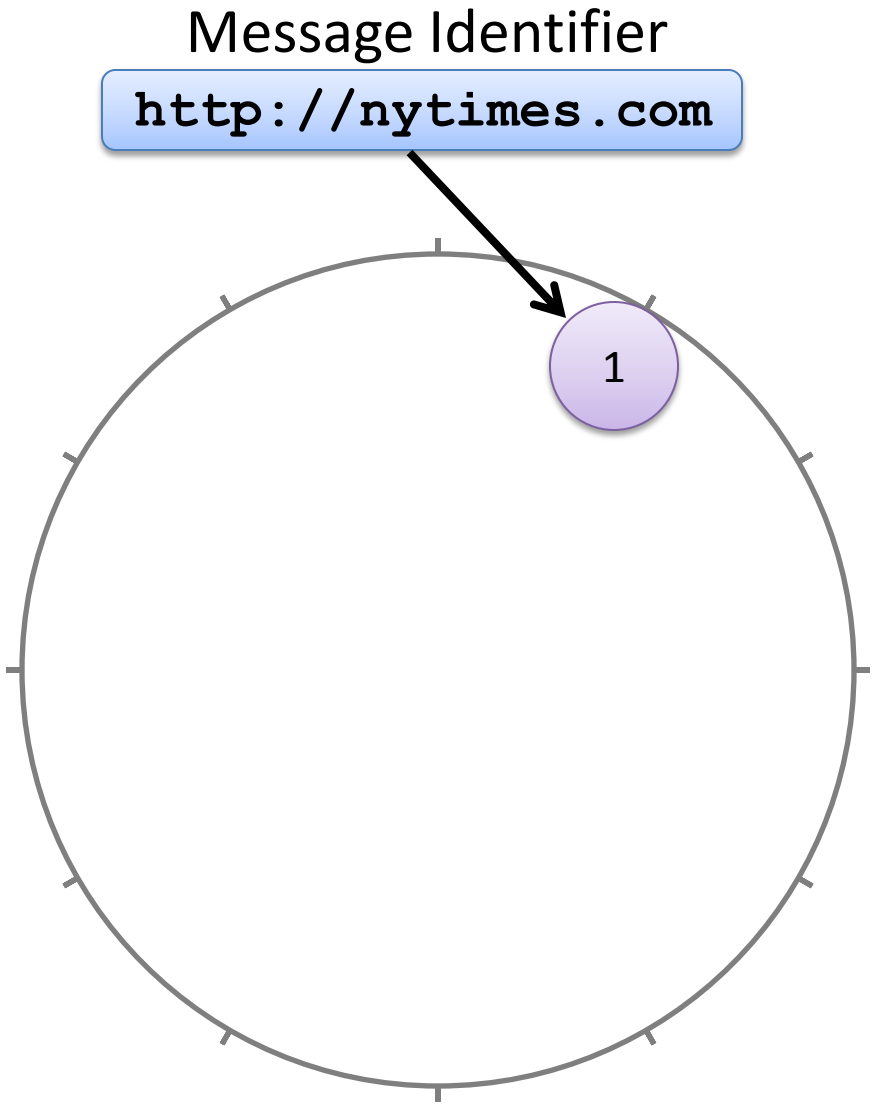
`http://nytimes.com`

Collage steps:

1. Obtain message
2. Pick message identifier
3. Obtain cover media
4. Embed message in cover
5. **Upload UGC to content host**
6. **Find and download UGC**
7. Decode message from UGC

Solution: Task Mapping

1. Hash the identifier
2. Hash the tasks
3. Map identifier to closest tasks

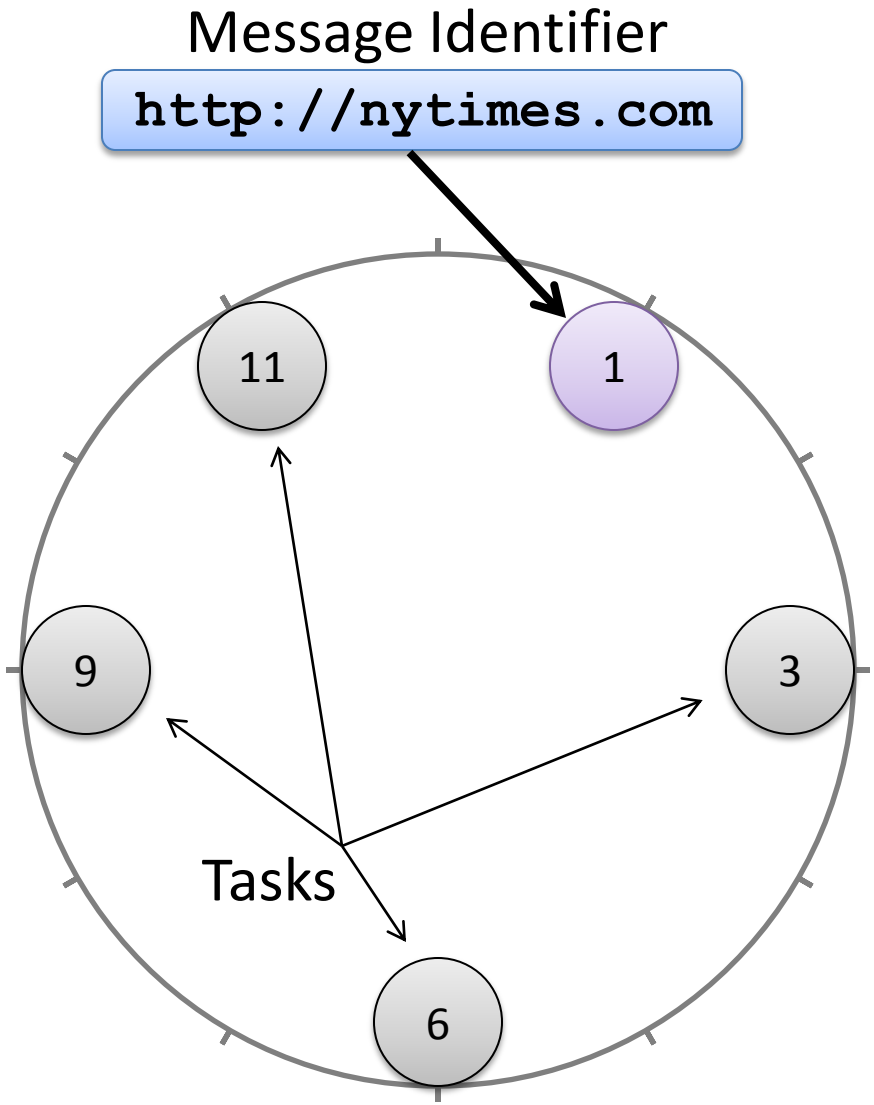


Collage steps:

1. Obtain message
2. Pick message identifier
3. Obtain cover media
4. Embed message in cover
5. **Upload UGC to content host**
6. **Find and download UGC**
7. Decode message from UGC

Solution: Task Mapping

1. Hash the identifier
2. Hash the tasks
3. Map identifier to closest tasks

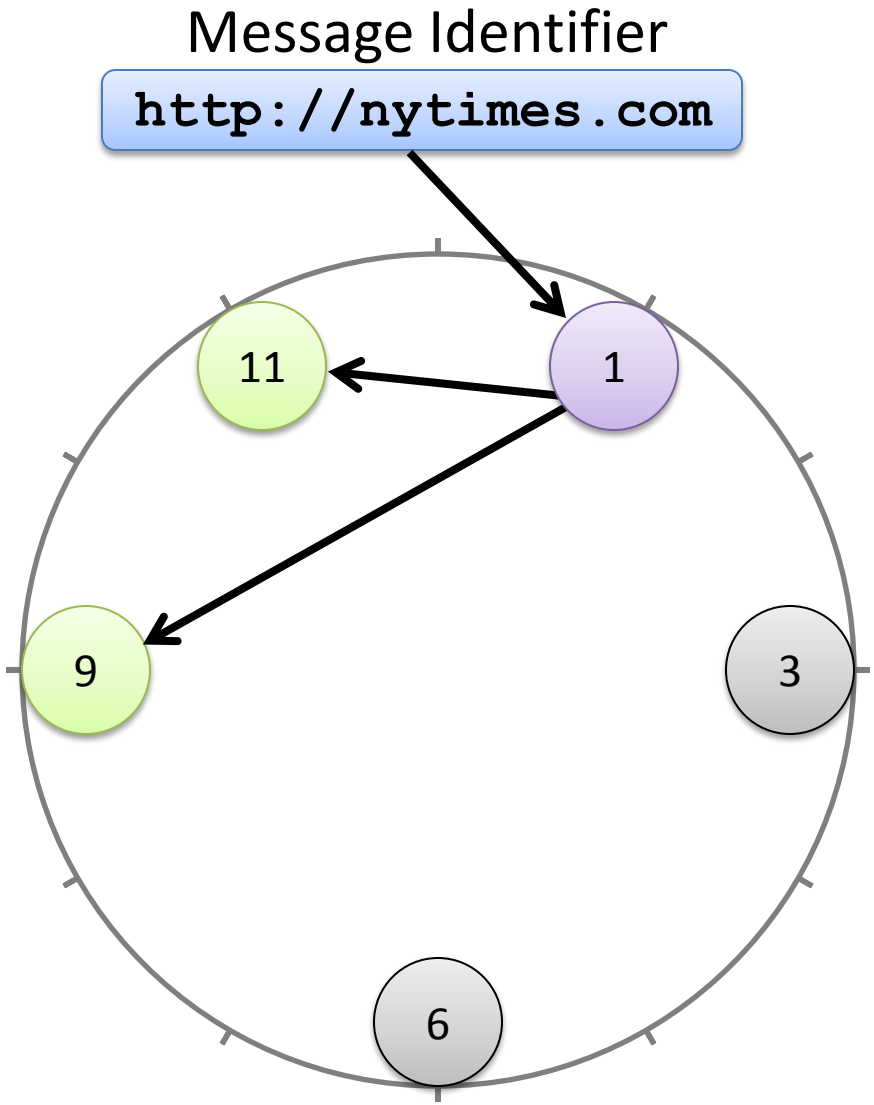


Collage steps:

1. Obtain message
2. Pick message identifier
3. Obtain cover media
4. Embed message in cover
5. **Upload UGC to content host**
6. **Find and download UGC**
7. Decode message from UGC

Solution: Task Mapping

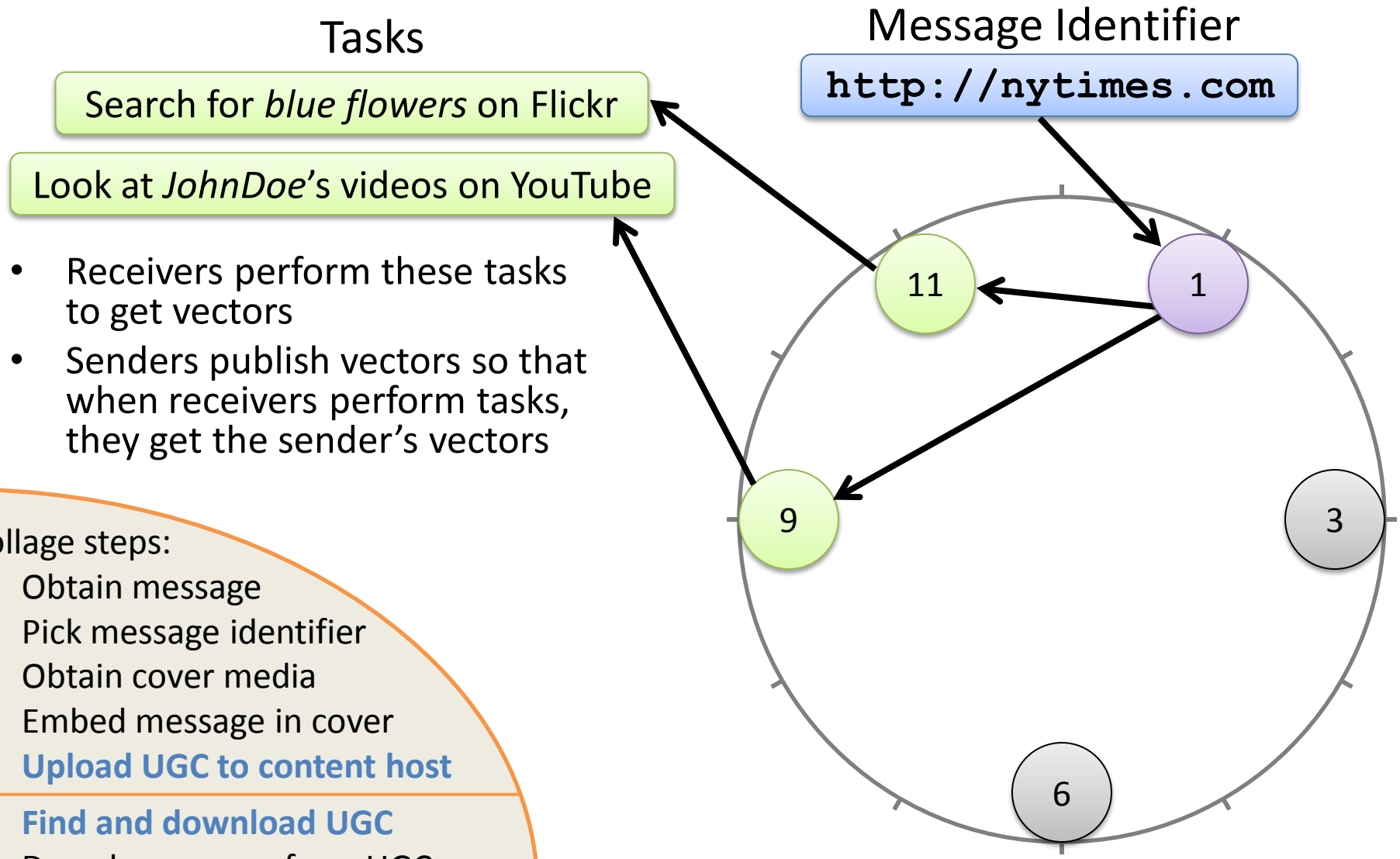
1. Hash the identifier
2. Hash the tasks
3. Map identifier to closest tasks



Collage steps:

1. Obtain message
2. Pick message identifier
3. Obtain cover media
4. Embed message in cover
5. **Upload UGC to content host**
6. **Find and download UGC**
7. Decode message from UGC

Solution: Task Mapping



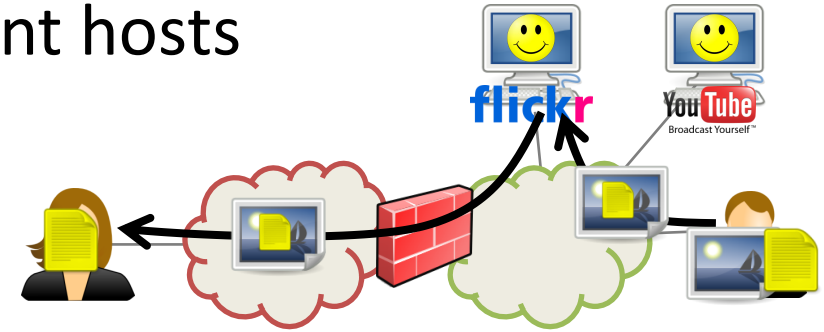
- Receivers perform these tasks to get vectors
- Senders publish vectors so that when receivers perform tasks, they get the sender's vectors

Collage steps:

1. Obtain message
2. Pick message identifier
3. Obtain cover media
4. Embed message in cover
5. **Upload UGC to content host**
6. **Find and download UGC**
7. Decode message from UGC

How Does Collage Meet the Design Goals?

- **Robust** against blocking
  - Erasure coding
  - Many content hosts
- **Deniable** against user identification
  - Traffic only to/from content hosts
  - Depends upon task construction
- Require **no dedicated infrastructure**
  - Messages stored on content hosts



How Do You Start Using Collage?

Send & Receive Messages

1. Distribute software
   - CDROM
   - Spam everyone
   - A secure network
2. Refresh task list
   - Receive using Collage
   - Online resource
3. Message identifier
   - Application specific

Help Censored Users

1. Donate your UGC vectors
   - Photos on Flickr
   - Tweets on Twitter
   - Etc.
2. Write Collage applications
   - http://gtnoise.net/collage

Outline

- Background and Design Goals
- Collage Design
- **Performance and Demo**

Performance Metrics

- Sender and receiver **traffic overhead**
- Sender and receiver **transfer time**
- **Storage** required on content hosts

These metrics can vary a lot:

- Different content hosts
- Different tasks

Case Study

	News Articles	Covert Tweets
Content host	Flickr	Twitter
Message size	30 KB	140 Bytes
Vectors needed	5	30
Storage needed	600 KB	4 KB
Sending traffic	1,200 KB	1,100 KB
Sending time	5 minutes	60 minutes
Receiving traffic	6,000 KB	600 KB
Receiving time	2 minutes	½ minute

Experiments performed on a 768/128 Kbps DSL connection

Demo of a Collage Application

What Should You Do Now?

- Try out the demo application
- Donate your photos
  - Right now, just for Flickr Pro users
  - Embeds news articles when you upload photos

Visit http://gtnoise.net/collage

Conclusion

- Collage evades Internet censorship by tunneling messages inside user-generated content
  - Robust against blocking
  - Deniable against user identification
  - Requires no dedicated infrastructure

- More work needed
  - Statistical deniability against traffic analysis
  - Learn timing behavior from users
  - Tor bridge discovery

http://gtnoise.net/collage