

CUTLASS - Encrypted Communications for Everyone

Todd MacDermid

Jack Lloyd

Kathy Wang

(John Schweitzer)

(Nash Foster)

CUTLASS Talk

Overview

- What we want CUTLASS to be (Goals)
- Convince you of the need for CUTLASS (aka “The Status Quo Sucks”)
- Discuss the design
- Discuss what we’ve done
- Discuss what’s left, and (maybe) convince you to join us in world domination!

CUTLASS Rules!

- Questions Whenever (Unless we run short, we will let you know)
- No object throwing at speakers until end of talk

What Is CUTLASS?

- A peer to peer method of exchanging files, voice, text, and eventually video communication, all over secure channels
- Cross-platform, easy to use
- “Free.” BSD license

The Dream...







How Much of Your
Traffic do
YOU
Encrypt?

CUTLASS Design Goals

- Easy enough to use for broad adoption
- Cross-Platform - Avoid tweaky APIs
- Secure by default, encrypted channels, resistant to traffic analysis
- Useful with small network effect
- Extendable (both functionality paranoia)
- Not dependent on central servers

CUTLASS Anti-Goals

- Not a strong anonymity system
- Not restricted to existing standard protocols
- Not necessarily required to be completely meshed

“The Competition”

- Skype
- WASTE
- Haxial KDX (Formerly Netfone)
- Jabber
- GnomeMeeting and other Free VoIP
- GNUNet, etc.

Skype

The Good	The Bad
<p data-bbox="323 874 1163 1038">Encrypted, peer-to-peer voice</p> <p data-bbox="323 1160 1256 1324">UI is a marvel of simplicity, both in install and use</p>	<p data-bbox="1385 782 2376 854">Licensing terms are onerous</p> <p data-bbox="1385 966 2390 1140">Traffic is dependent on a few supernodes</p> <p data-bbox="1385 1252 2326 1426">Crypto is questionable and closed</p> <p data-bbox="1385 1539 2390 1610">Only 5-way conference, max.</p>

WASTE

The Good	The Bad
<p data-bbox="323 870 1292 1038">Encrypted, peer-to-peer file transfer</p> <p data-bbox="323 1156 858 1242">Cross-platform</p> <p data-bbox="323 1351 1190 1436">Code is broadly available</p>	<p data-bbox="1385 870 2354 1038">Licensing issues are fuzzy, at best</p> <p data-bbox="1385 1156 2184 1336">No way of removing someone from a group</p> <p data-bbox="1385 1447 2184 1533">Key exchange is painful</p>

Haxial KDX

The Good	The Bad
Encrypted, peer-to-peer voice, file transfer, and text	Windows-only Not free software (Annoyware) Key management. (All symmetric, OOB management). Voice is over TCP, and laggy

Jabber

The Good	The Bad
<p data-bbox="323 770 812 844">Free software</p> <p data-bbox="323 962 1210 1044">standards-based protocol</p> <p data-bbox="323 1152 902 1330">Multiplatform implementations</p> <p data-bbox="323 1438 1141 1616">Cryptographic controls available via SSL</p>	<p data-bbox="1385 770 2329 948">No real structure provided for voice transmission</p> <p data-bbox="1385 1056 2395 1418">Focused primarily on IM/chat. File transfer is stubbed out, but dependent on HTTP connection</p> <p data-bbox="1385 1535 2329 1712">SSL is only supported trust model</p>

GNUNet

The Good	The Bad
<p data-bbox="323 868 812 942">Free software</p> <p data-bbox="323 1058 1300 1228">Provides both datagram and stream channels</p> <p data-bbox="323 1344 1251 1432">Provides strong anonymity</p>	<p data-bbox="1385 868 2175 1038">Strongly dependent on network effect</p> <p data-bbox="1385 1154 1978 1242">Very high latency</p> <p data-bbox="1385 1359 2348 1528">Currently primarily focused on file transfer</p>

Free VoIP Implementations

The Good	The Bad
<p data-bbox="323 870 812 942">Free software</p> <p data-bbox="323 1064 1037 1136">Standards-compliant</p> <p data-bbox="323 1259 1256 1422">Can interface to the POTS network</p>	<p data-bbox="1385 870 2088 1044">Cryptography is not supported</p> <p data-bbox="1385 1156 2390 1524">Due to standards requirements, it will be difficult to add cryptographic support</p>

The Current CUTLASS Team



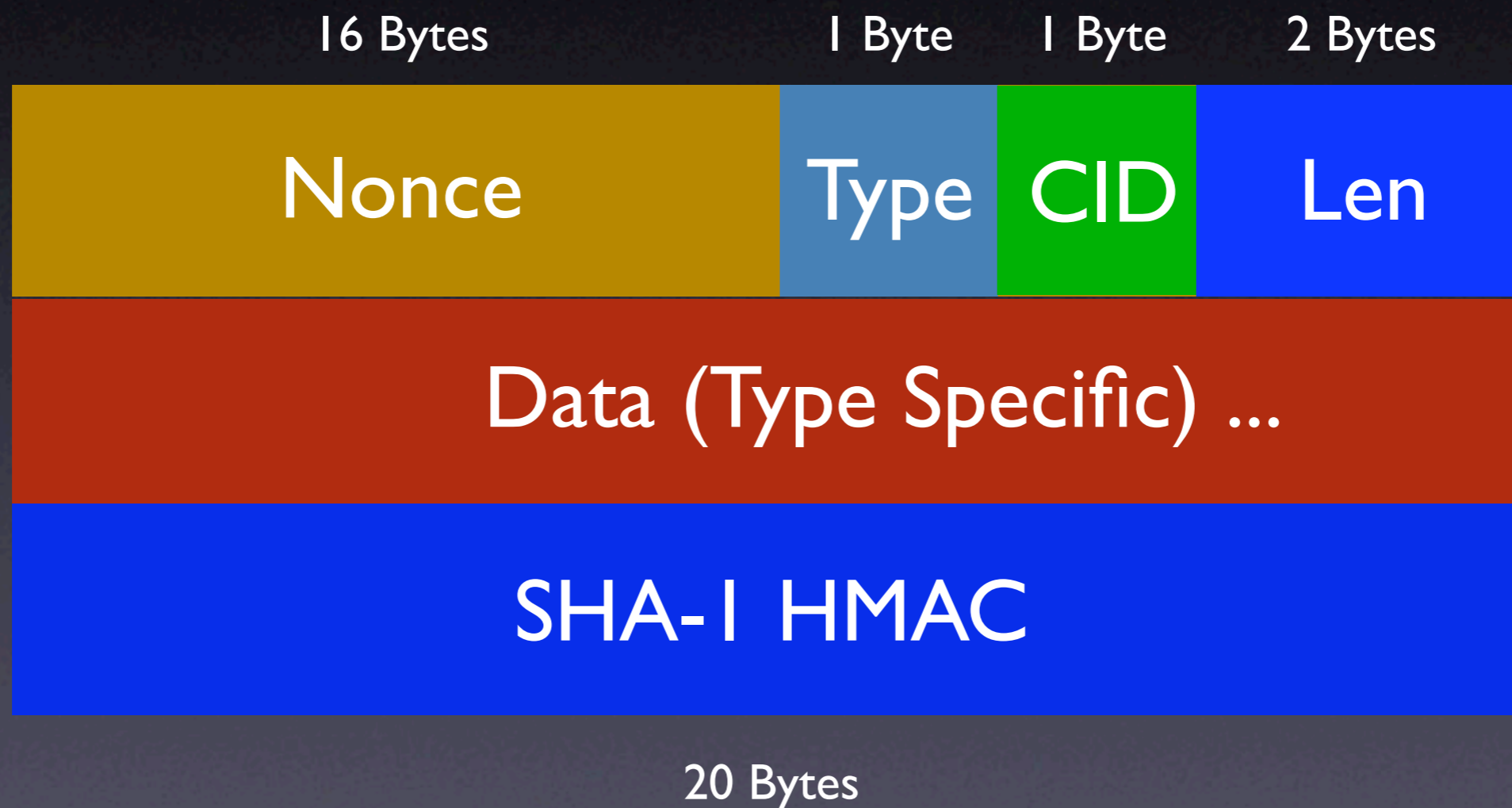
The Current CUTLASS Team

- Todd MacDermid - Plumbing, Glue, API
- Jack Lloyd - Crypto
- Kathy Wang - Portability
- John Schweitzer - GTK client
- Nash Foster - Audio

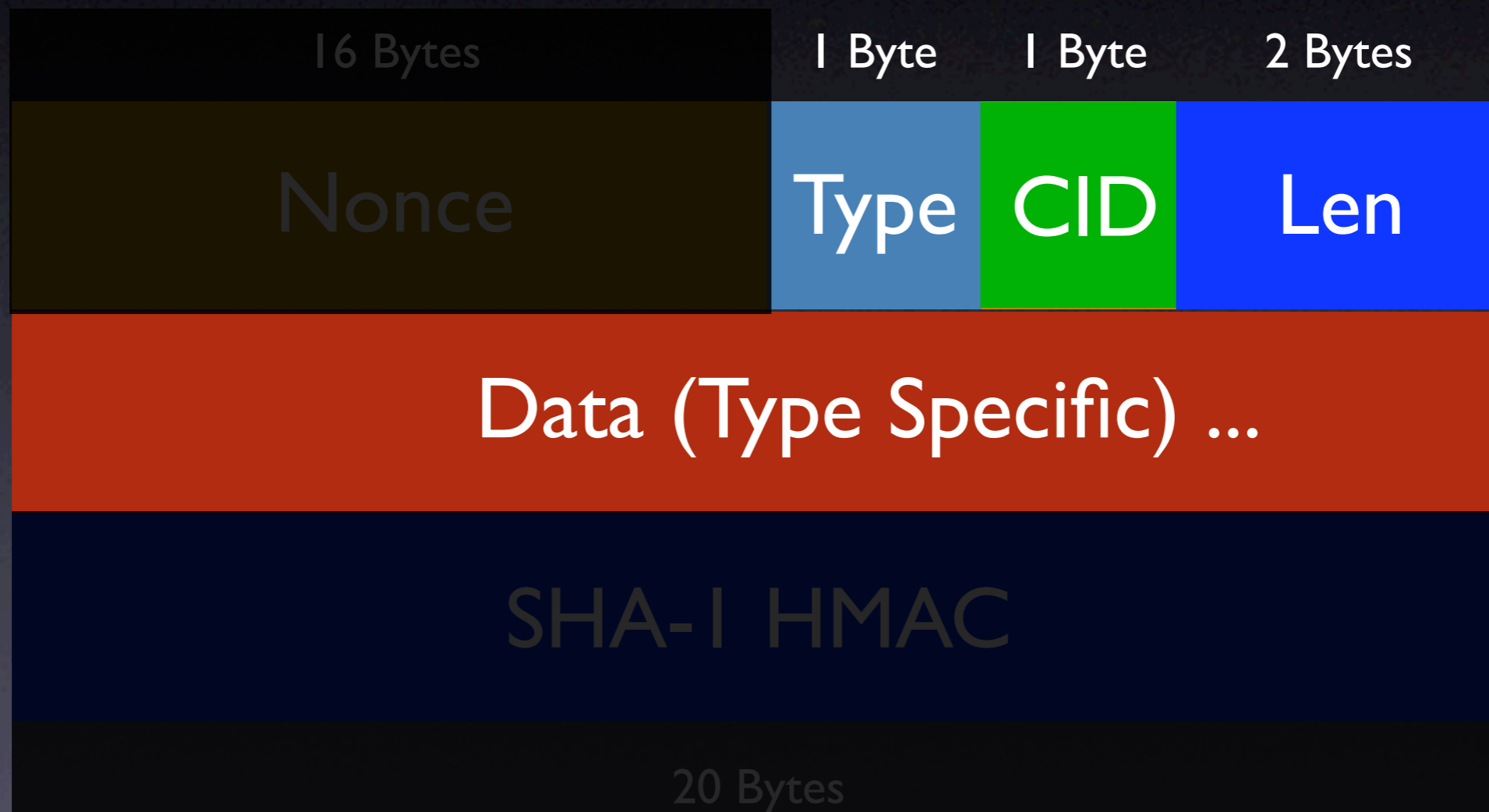
CUTLASS Protocol Overview

- Default of UDP for all traffic
- Allows anonymizing and traffic analysis defeating measures, but not enabled by default
- Allows individual and group messages/transfers
- Allows sound, file, and text transfers

CUTLASS Packet Structure



CUTLASS Packet Encrypted Portions



CUTLASS Packet Types

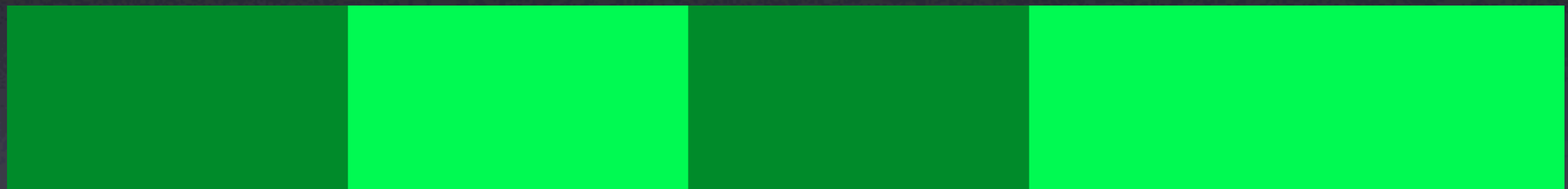
- Key Exchange
- Ping/Pong
- Connection Information Req/Resp
- Audio
- Reliable Transport
- Please Forward

CUTLASS Transport Layer

“Gap”-based requests

0

4500



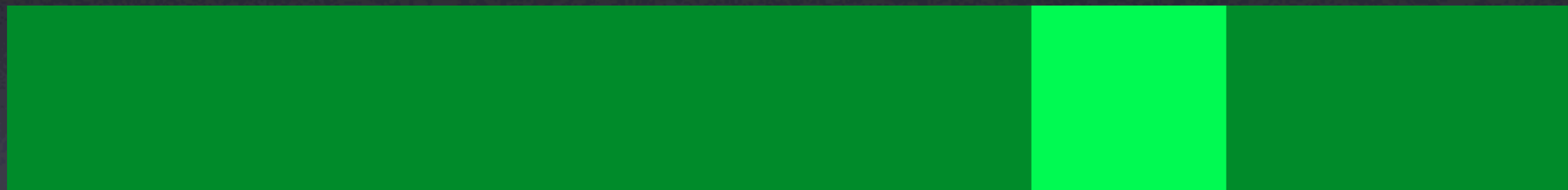
Request: 0-4500

CUTLASS Transport Layer

“Gap”-based requests

0

4500



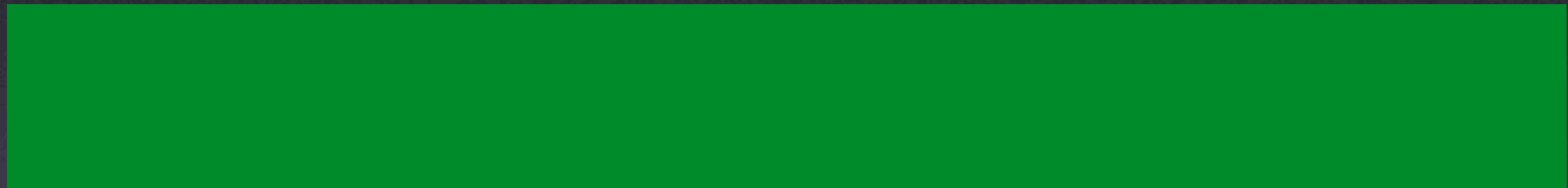
Request: 1000-2000,3000-4500

CUTLASS Transport Layer

“Gap”-based requests

0

4500



Request: 3500-4000

CUTLASS Transport Rate-Limiting

- Requests immediately get one response
- Successful request/response pair increases unsolicited rate by one PPS
- Periodically send unsolicited data according to rate
- If number of gaps increases, decrease unsolicited rate

CUTLASS Transport Stats

Copying 34 MB file over 10Mbps local link:

- SCP: 45 seconds
- CUTLASS: 53 seconds

Simultaneous copy bandwidth consumption:

- 75% of bandwidth used by SCP
- 25% of bandwidth used by CUTLASS

CUTLASS Transport Layer Advantages

- Unrestricted by window size
- Easy to turn into Bittorrent-style requests
- Easy recovery from halted transfers
- Potentially good performance across high-latency networks (not yet tested, insert salt here)

Existing Trust Models

- SSL - “One key to rule them all”
- PGP - “Trust no one”
- SSH - “First time’s free, kid”

Welcome to McCrypto, Can I Take Your Order?

- SSL/TLS: Great for TCP, UDP not so much
- PGP, S/MIME: Message based; very inefficient for Cutlass-style protocols
- IPsec: Admit it, IPsec sucks
- SRTP: Tied too strongly to RTP to be useful
- DTLS: No real world implementation; not yet standardized

CUTLASS Key Exchange

- Similar to a TLS key exchange
- First Exchange RSA public keys and per-session nonces
- Follow up with signed Diffie-Hellman parameters
- Now start talking
- Server responses are optional based on client knowledge of server key

Whizbang Features and Non-Features

- Perfect forward secrecy
- RSA key authentication, with password-based authentication coming soon
- Confidentiality and integrity
- No replay protection on a per-packet basis; higher CUTLASS protocols must handle repeats

Welcome... to the World of Tomorrow!

- DTLS with tweaked parameters
- Require ephemeral Diffie-Hellman
- No export ciphers
- Client authentication
- Pervert the X.509 trust model into the CUTLASS trust scheme
- We will have flying cars!

What's Done?

- Direct Connections
- Key Exchange
- Ping/Pong checks
- Text Messages
- Reliable Transport Layer

LibCUTLASS

- CUTLASS is currently divided into libcutlass and clients
- API docs in tarball
- Action handling functions registered by clients
- Existing clients: text-cutlass, gtk-cutlass

Portability Status

- Currently runs under Cygwin
- Cygwin binaries not really portable
- Currently following GAIM model (MinGW)
- Need to write ALSA driver replacement

What's Left to Do?

- Audio (Real Soon Now!)
- Group management
- Windows, Mac OS X, and PocketPC clients
- Directory servers
- Connection Forwarding
- Video

CUTLASS Audio

- Using Speex and ALSA
- Anyone willing to write other audio drivers, please join us!

CUTLASS Groups

- Groups can be authenticated or unauthenticated
- Groups can be advertised or hidden
- Group communication is still point-to-point
- Group members are a consensus reality

CUTLASS Group Management

- SuperOps have copy of private group key
- SuperOps cannot be revoked
- Regular Ops can be designated by SuperOps, will not have private key
- Ops may authenticate users, kick, ban, etc.
- These are effectively suggested local policies

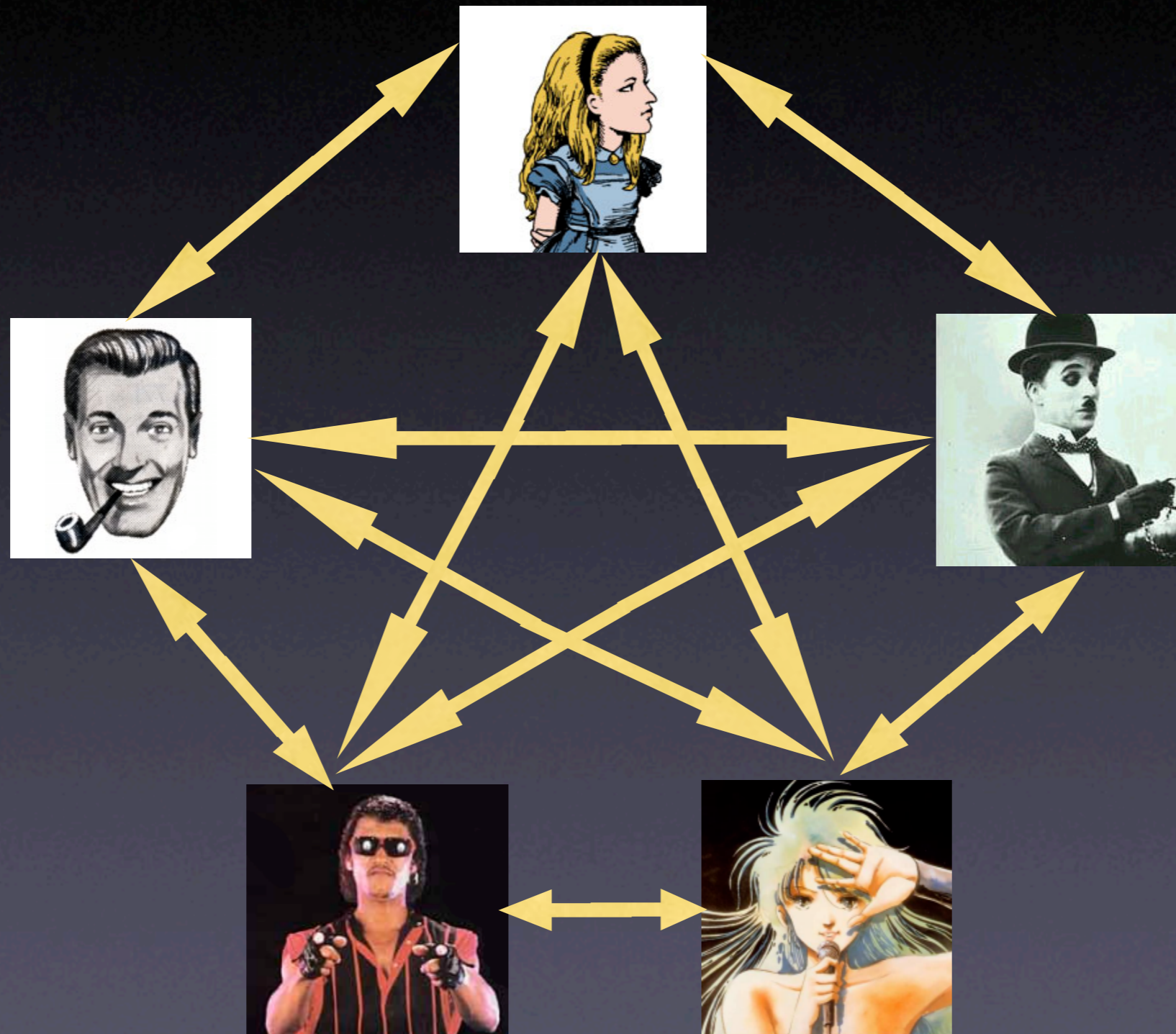
CUTLASS Directory Servers

- Anyone can be a directory server
- Store registered users, key fingerprints, and network locations
- Store advertised groups, group key fingerprints, and group operators
- Will NOT store file directories
- Will NOT be initially meshed, but is certainly a future desire

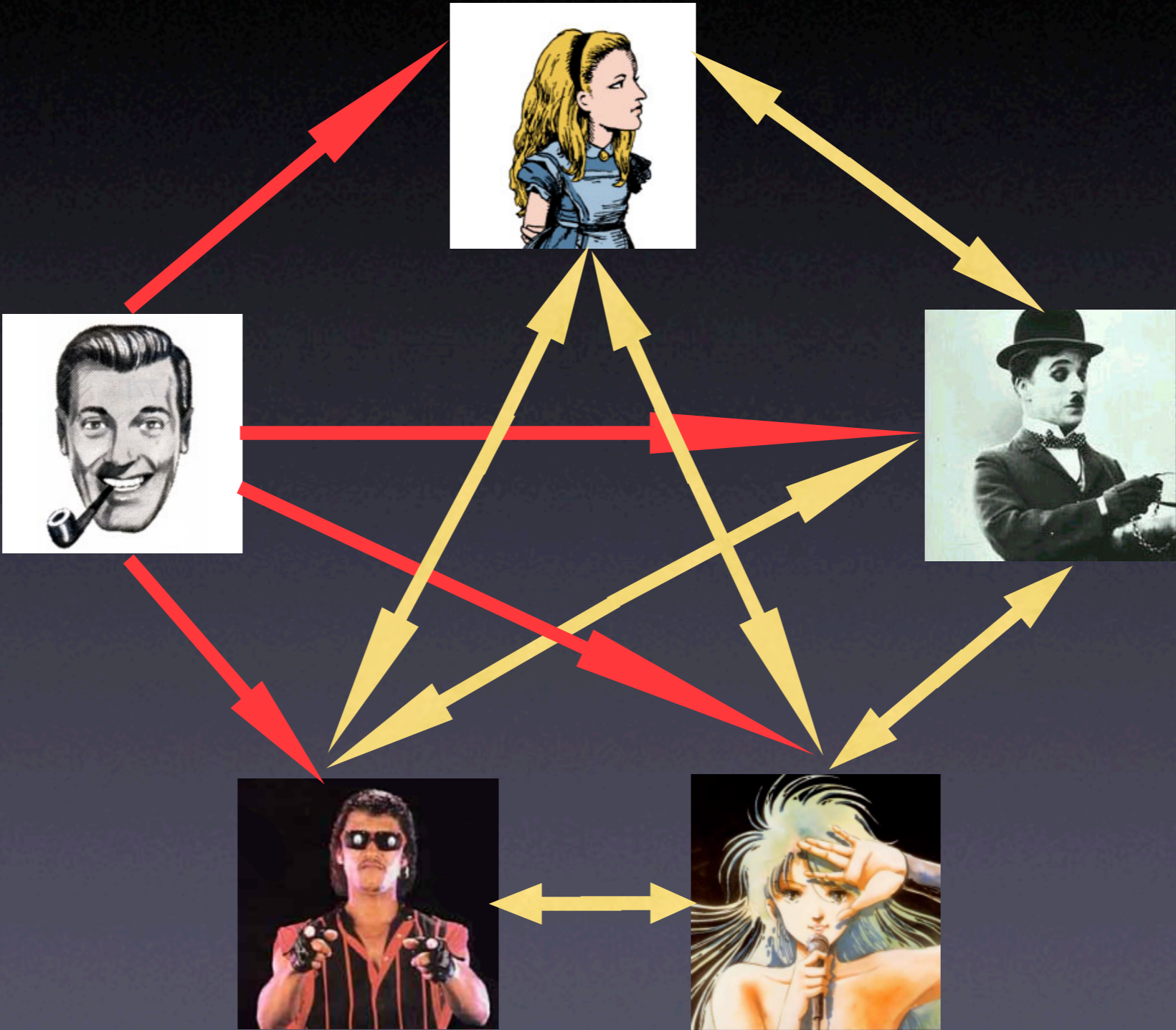
CUTLASS Connection Forwarding

- By default, clients will permit channel allocation between any two remote hosts
- Hosts may (and should) rate-limit between forwarded hosts
- Forwarded connections are opaque to forwarding host
- Peers may request peers to “meet” at a designated hosts

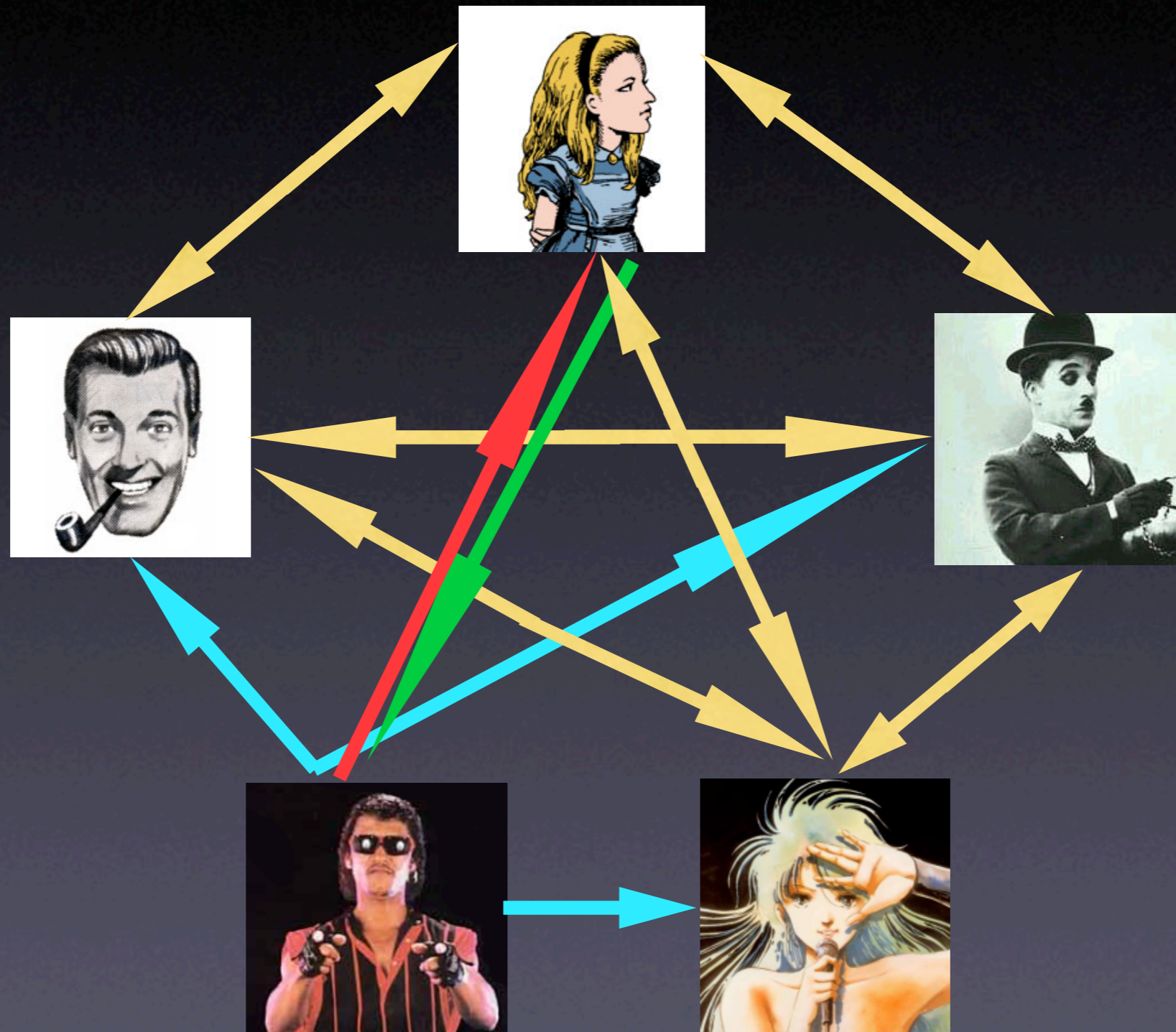
Best Case: Fully Meshed Communications



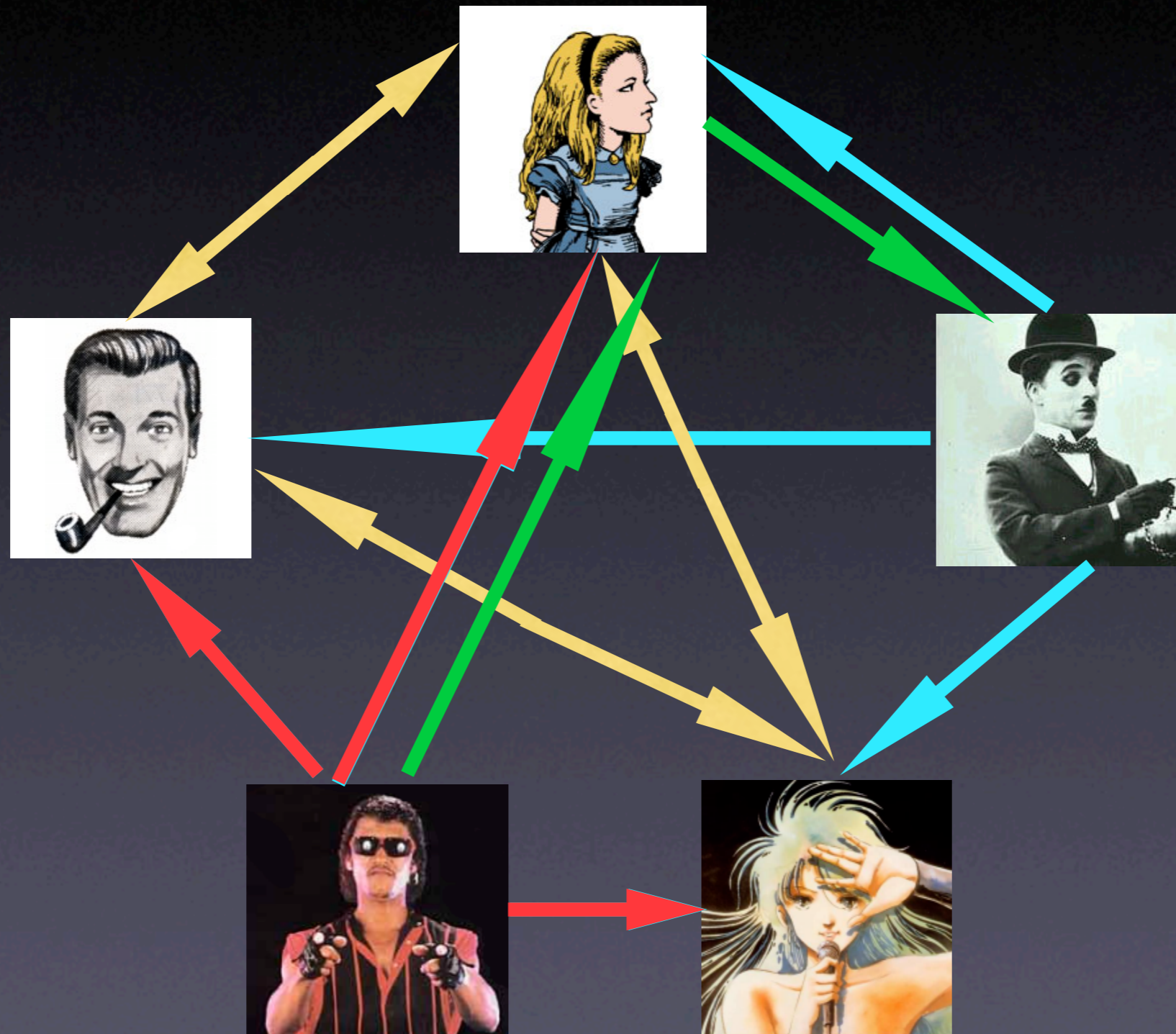
Bob Talks to the Group



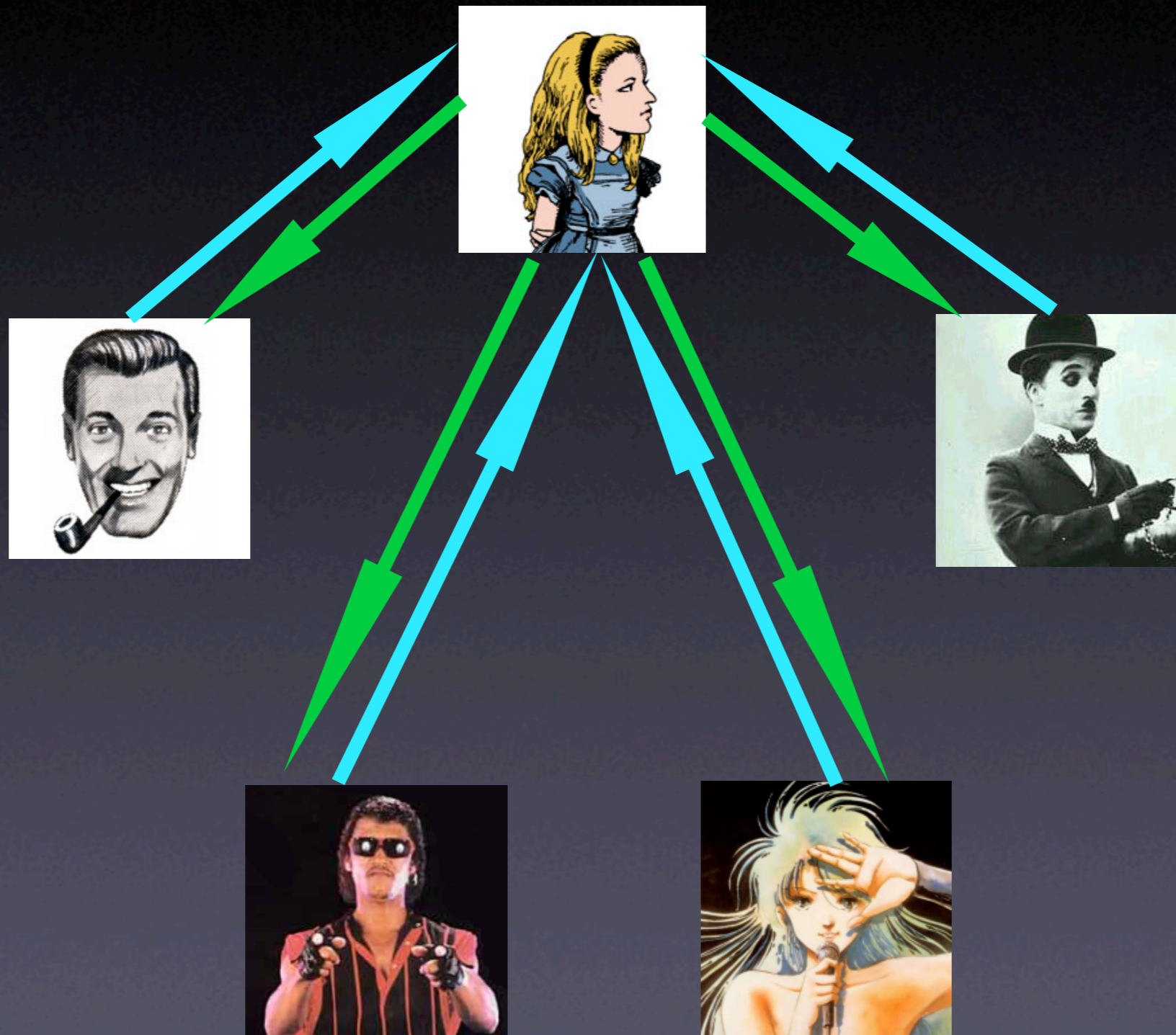
Dave Behind NAT



Dave and Charlie Behind NAT

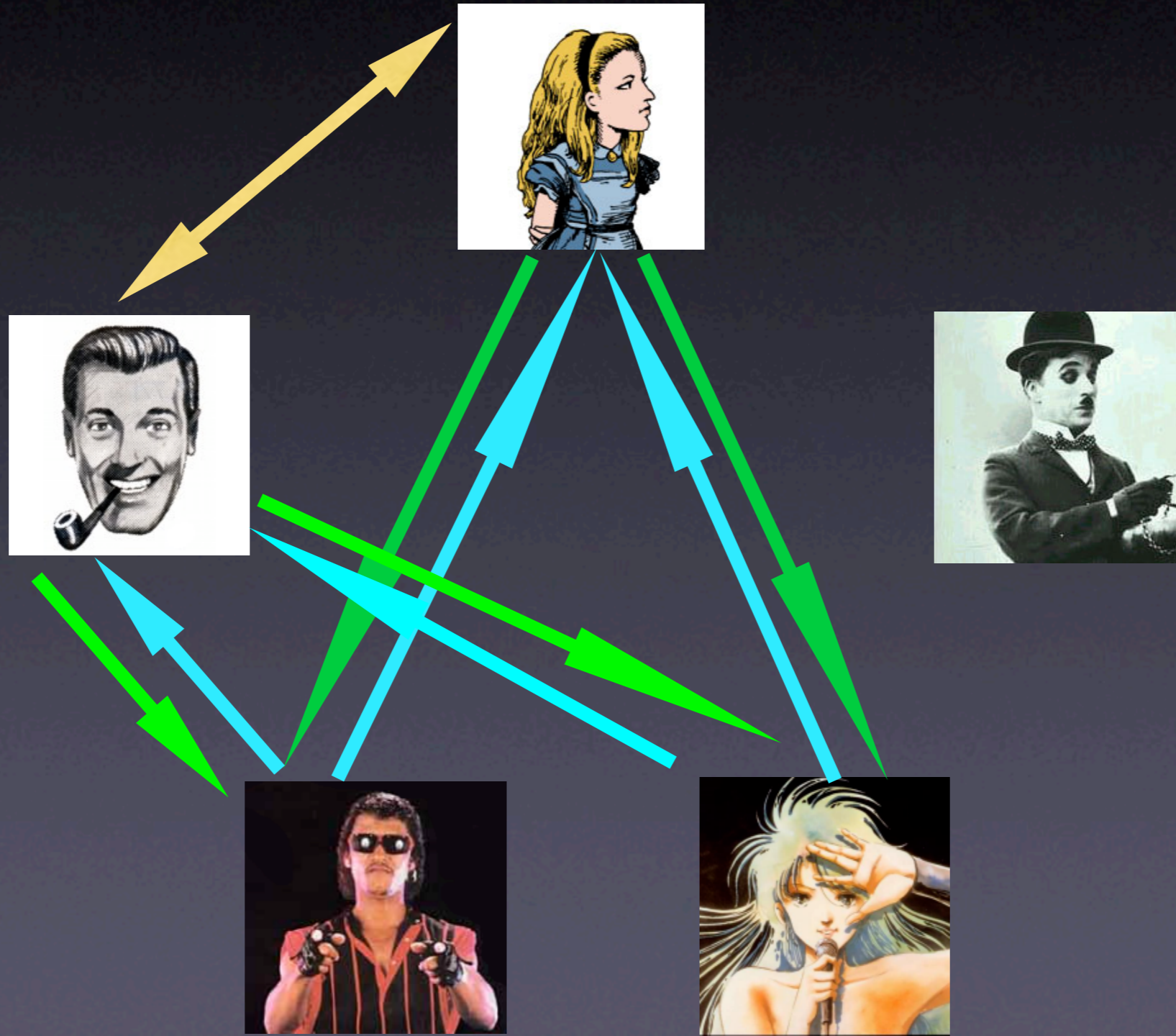


Everyone Behind NAT Except Alice



Alice Directory Server

Bob UnNATed



Defend The Bill of Rights!



Join the CUTLASS List

Send email to:

cutlass-subscribe@synacklabs.net

Operators are standing by
Please have your GPG/PGP key ready!