

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 and demo
- Discuss what’s left to do, and convince you to join us in world domination!

CUTLASS Rules!

- Questions Whenever (Unless we run short, we will let you know)

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 “unique” 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 2395 1242">Traffic is dependent on central authentication server (CALEA?)</p> <p data-bbox="1385 1355 2326 1518">Crypto is questionable and closed</p> <p data-bbox="1385 1641 2395 1712">Only 5-way conference, max.</p>

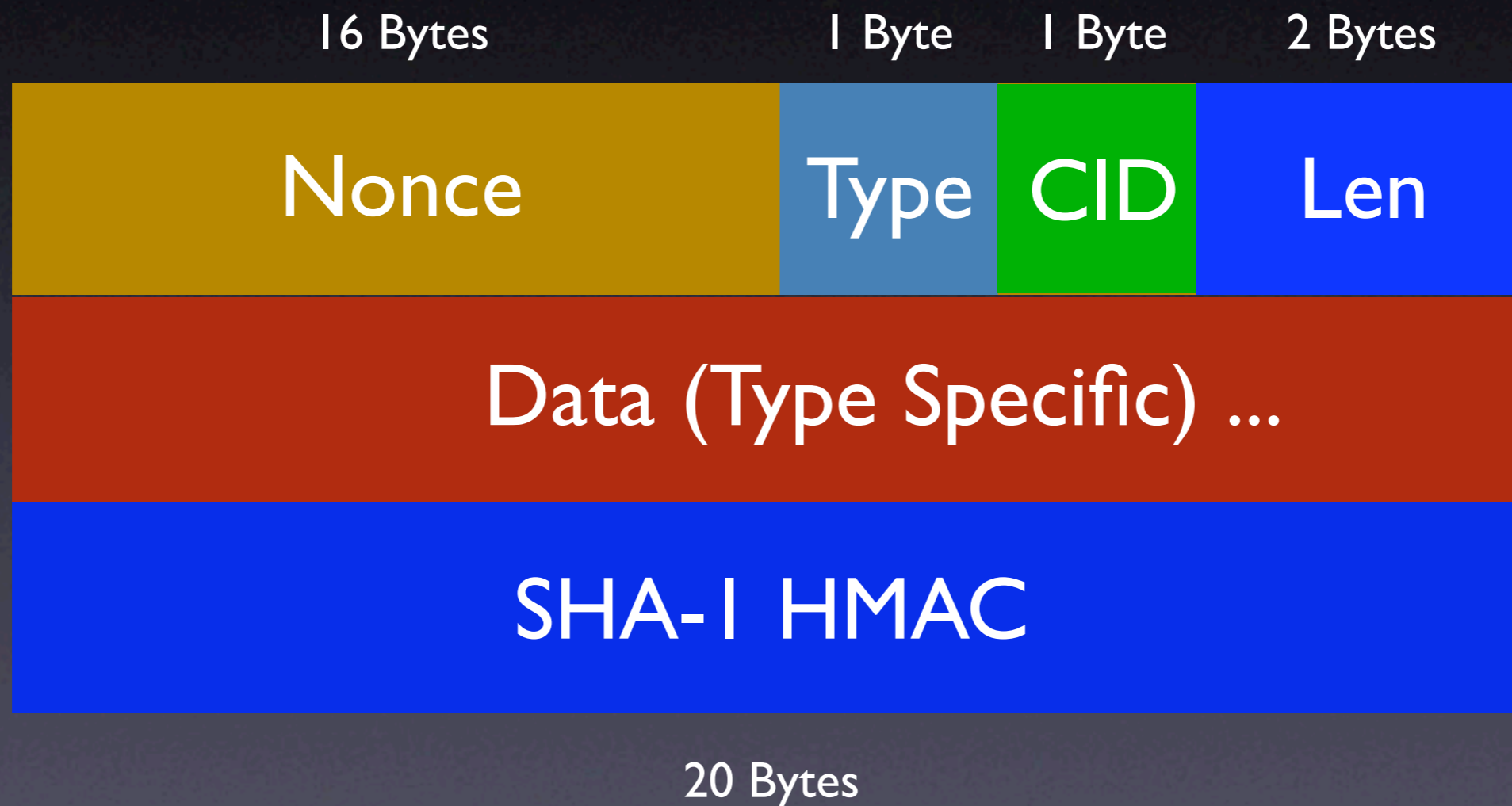
WASTE

The Good	The Bad
<p data-bbox="323 868 1292 1038">Encrypted, peer-to-peer file transfer</p> <p data-bbox="323 1154 858 1242">Cross-platform</p> <p data-bbox="323 1344 1190 1432">Code is broadly available</p>	<p data-bbox="1385 868 2354 1038">Licensing issues are fuzzy, at best</p> <p data-bbox="1385 1154 2184 1336">No way of removing someone from a group</p> <p data-bbox="1385 1441 2184 1528">Key exchange is painful</p>

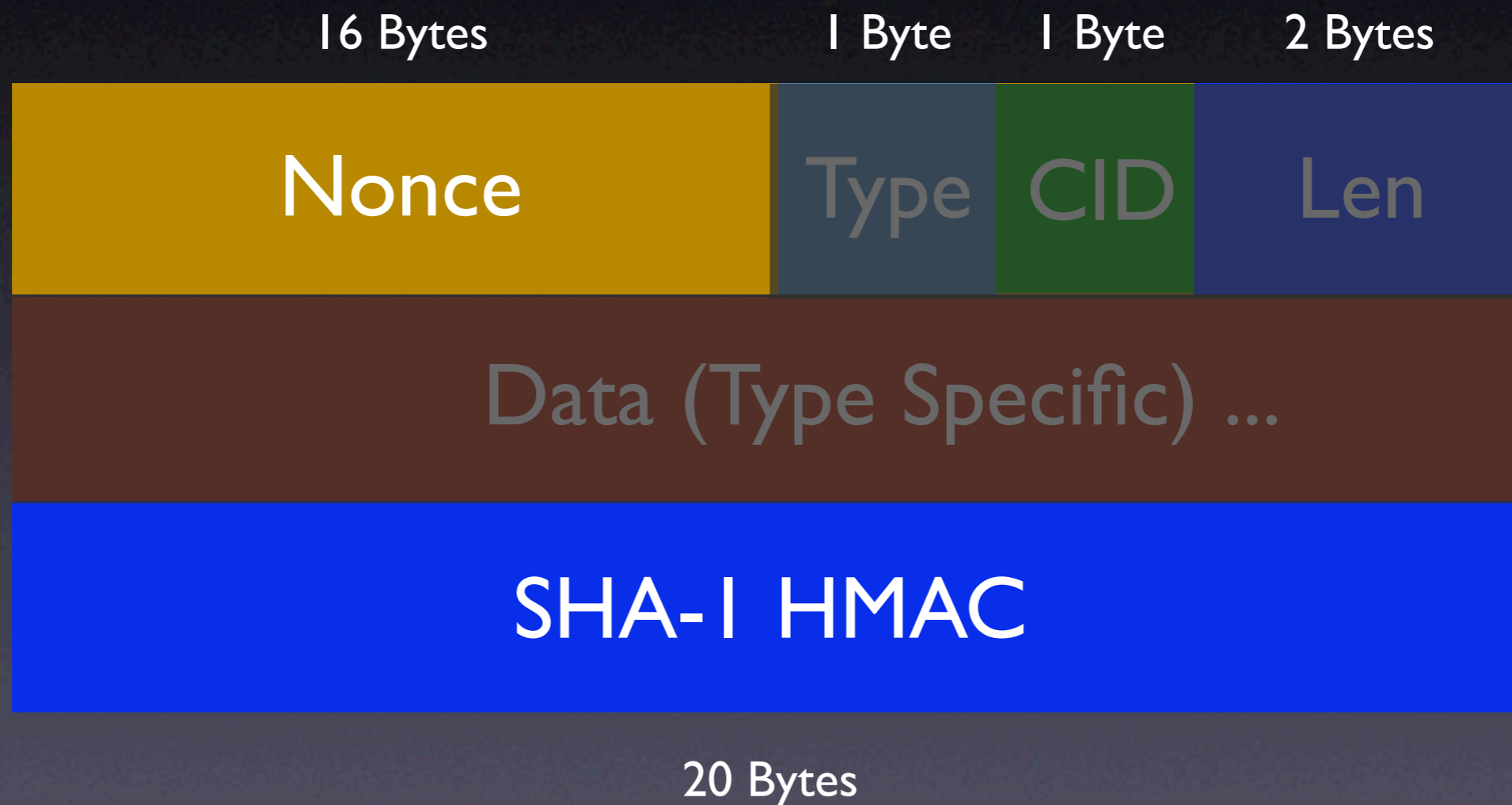
CUTLASS Protocol Overview

- UDP-based
- Allows anonymizing and traffic analysis defeating measures, but not enabled by default
- Allows individual and group messages/transfers
- Allows voice, file, and text transfers

CUTLASS Packet Structure



CUTLASS Packet Encrypted Portions



CUTLASS Packet Types

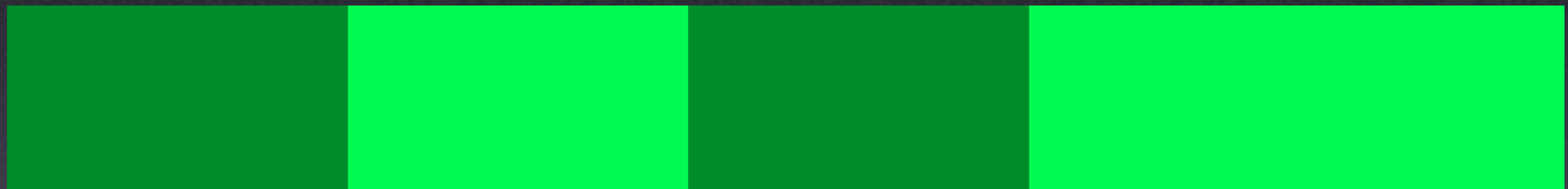
- Key Exchange
- Ping/Pong
- Connection Information Req/Resp
- Audio
- Reliable Transport

CUTLASS Transport Layer

“Gap”-based requests

0

4500



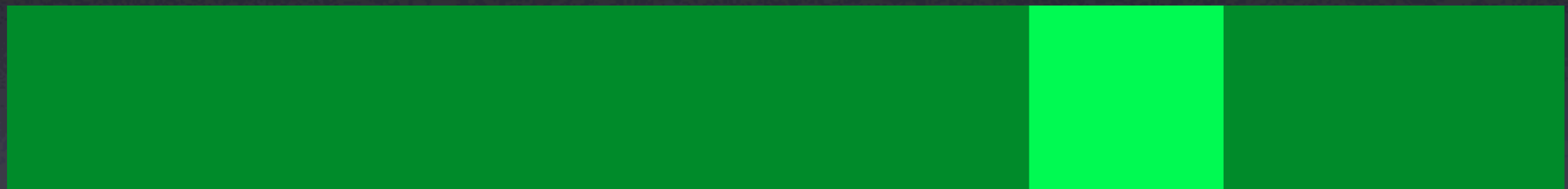
Request: 0-4500

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)

Reinventing the Wheel

- SSL/TLS - Requires TCP or equivalent
- PGP and S/MIME - Message-based; very inefficient with many packets
- IPsec - Admit it, IPsec sucks
- SRTP - Too strongly tied to RTP to be helpful

Key Exchange

Initiator / "Client"

----- $\text{nonce}_c, H(\text{nonce}_c, \text{RSA}_s), \text{RSA}_c$ ----->

<----- $\text{nonce}_s, \text{nonce}_c, \text{RSA}_s$ -----

----- $\text{DH}_c, \text{SIG}_c(\text{DH}_c)$ ----->

<----- $\text{DH}_s, \text{SIG}_s(\text{DH}_s)$ -----

Responder / "Server"

Random Crypto Goop

- Session keys from ephemeral Diffie-Hellman
- AES-256 in counter mode, with HMAC/SHA-1
- RSA signatures with PSS/SHA-256
- No replay protection at the crypto layer (but there will be!)

The Five Year Plan

- DTLS - TLS over datagram (IETF draft)
- OpenPGP and SRP authentication for TLS (IETF drafts)
- DTLS + SRP + OpenPGP = sweet
- The IETF isn't quick, and I code even slower

CUTLASS Voice

- Using Speex, with 8 KHz sample rate
- Phone quality, more or less
- Currently supports OSS
- Anyone willing to write other audio drivers, please join us!

What's Done?

- Direct Connections
- Key Exchange
- Text Messages
- File Transfer
- Audio
- GTK GUI

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

What's Left to Do?

- Group management
- Windows, Mac OS X, and PocketPC clients
- Directory servers
- Connection Forwarding
- Video
- Gaim plugin