

Secure System Design: Conference System Security



Zoom and the Like

- We've all used Zoom for the last few years
- Many of us have used other systems: Cisco's WebEx, Microsoft Teams, Google Meet, Skype, and more
- Are these secure?

Note: I'll be spending more time on Zoom, since that's the one most of us know best. But many of the comments are generic.

What is “Security” for a Conferencing System?

The usual questions:

- What are we trying to protect?
- Against whom?

The list of answers is longer than usual

- Audio privacy
- Video privacy
- Ability to participate
- How people can participate
- Participant list
- Metadata:
 - How much does each person talk?
 - Where is each participant?
 - Where is the participant who talks the most?
 - More?
- User privacy
- User device security
- More?

Attackers

- Possibly intelligence agencies—it depends on the meeting
- Possibly industrial espionage agents—again, it depends on the meeting
- Conference system operator
- Would-be unauthorized participants
- Zoom-bombers
- Hackers
- Voyeurs

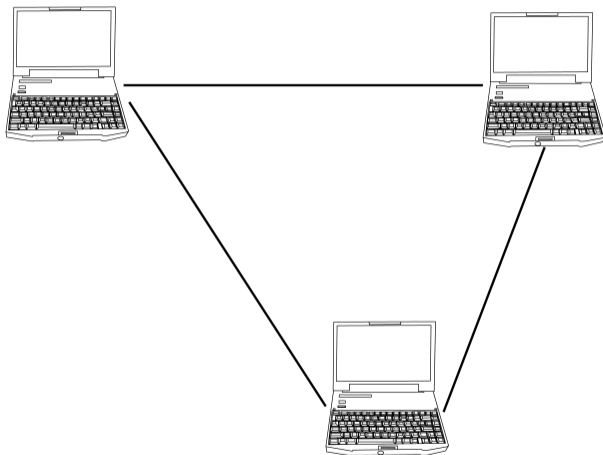
Disclaimer

There are many possible ways to do things. I *strongly* urge you to read the Zoom white paper for a detailed description of a real-world cryptographic architecture—they present far more detail than I will, especially because I'll be showing other alternatives.

Obvious First Steps

- Encrypt all links
- Authenticate all participants
- We have problems. . .

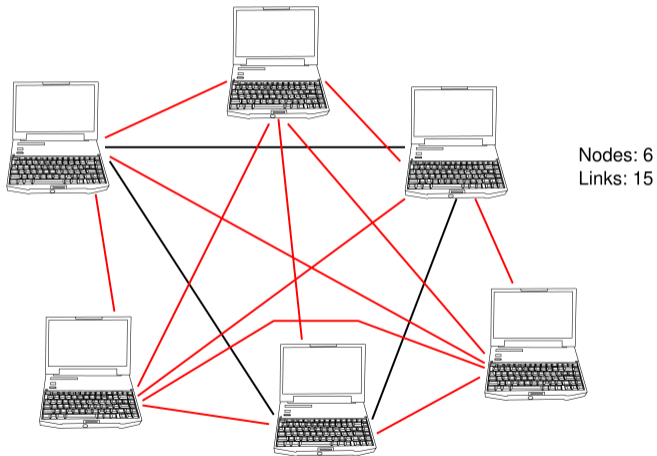
Architecture: Point-to-Point



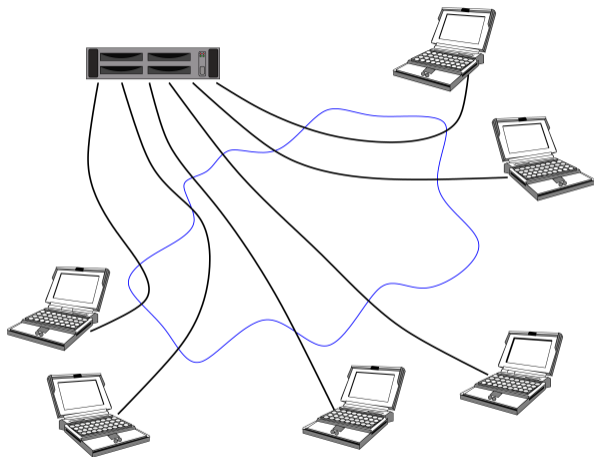
Nodes: 3
Links: 3

Do encryption from each node to every other

Point-to-Point Doesn't Scale



We Need a Star



To whom do we encrypt? How do we set up a session key?

Encryption and Authentication

- If we authenticate and encrypt to the central server, key setup is easy: the server creates and distributes a session key
- But that means that encryption isn't end-to-end—the operator sees all conversations in the clear
- Should the meeting host authenticate all users and distribute keys?
- Does the meeting host have enough CPU power and bandwidth to do that?
- How many participants can a meeting have?

Group Key Management

- A well-studied problem in the cryptographic literature
- All sorts of issues
 - Do nodes self-organize into a tree, to avoid bottlenecks?
 - If someone leaves the conference, do you rekey?
 - If someone is forcibly expelled from the conference, do you rekey?
 - How do you rebuild the tree, if you use one, after someone leaves?
 - (Zoom limits conferences to 1000 users, to permit centralized key distribution)
 - Etc...

Group Key Management

- A well-studied problem in the cryptographic literature
- All sorts of issues
 - Do nodes self-organize into a tree, to avoid bottlenecks?
 - If someone leaves the conference, do you rekey?
 - If someone is forcibly expelled from the conference, do you rekey?
 - How do you rebuild the tree, if you use one, after someone leaves?
 - (Zoom limits conferences to 1000 users, to permit centralized key distribution)
 - Etc...
- Of such questions are applied cryptography papers made...

Authentication

- How do you authenticate users?
- Authentication *must* be done by the party distributing the keys

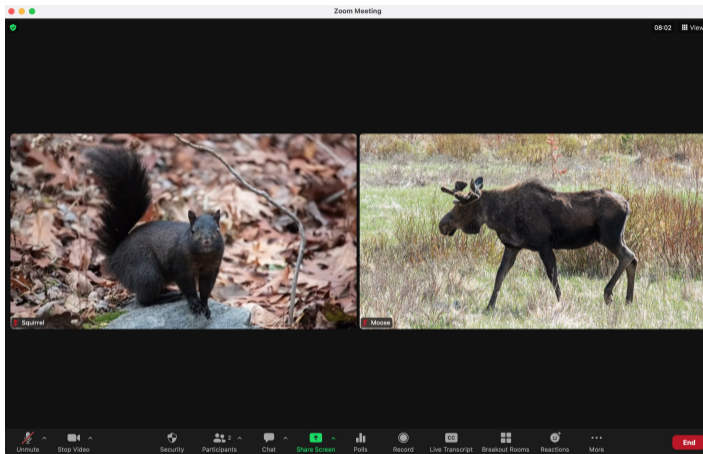
Authentication

- How do you authenticate users?
- Authentication *must* be done by the party distributing the keys
- Why?

Authentication

- How do you authenticate users?
- Authentication *must* be done by the party distributing the keys
- Why?
- How else do you know who is authorized to receive the session key?
- What if Andromedan intelligence tries to join?
- Zoom supports outsourced authentication, e.g., to enterprise customers
- Zoom also supports free accounts
- How do you do authentication?

The Need for Authentication




Is Squirrel really talking to Moose, or is it Boris or Natasha impersonating Moose?

Authentication and Certificates

- The only scalable way to do key distribution involves public key cryptography and hence certificates
- Where are the certificates stored?
- Where are the private keys stored, and how are they protected?
- Assume: multiple devices and enterprise authentication
- (When I lecture, I log on to a classroom computer that doesn't retain any state)
- Most likely answer: a new key pair is generated at authentication time, and the public key is passed up

What Are the Flaws?

What Are the Flaws?

- To whom is the public key passed?
 - If it's passed only to the conference server and not to the enterprise identity server, it isn't authenticated
 - A corrupted conference server could claim that some random node had been properly authenticated
-  End-to-end encryption requires end-to-end identity verification

- With end-to-end encryption, only the conference host can hand out session keys
- Only the conference server does authentication
- So: the conference server has to pass new login requests to the host
- Pass their name or other userID, too, to put into the participants list
- That name could be forged—but you'd see an extra, unexpected participant
- Security is a systems problem. . .

More Sophisticated Authentication

- Assume that users have certificates
- Well, they don't—but clients can generate key pairs and their enterprise identity server can sign the certificate on first use
- User devices must have secure storage for private keys
- Note: must make provision for multiple certificates per user, since many users have multiple devices; alternatively, must *securely* pass around the private key
- Also note: must make provision for certificate revocation
- (Zoom also has a complex mechanism for tracking devices owned by each user)

Complications

- Record to cloud—how is this protected?
- Encrypted? To what public key? (And how is the data decryption key to be distributed?)
- Phone clients
- “Guest” users

Displayed Usernames

- Do you trust the user?
- Maybe users can reset their names, as I did to create the Moose and Squirrel screenshot
- Or: rely on the name provided by the enterprise

- Partly a matter of the user interface—who can share screen, send text messages, unmute, turn on video, etc.?
- Conference leader should be able to expel misbehaving users
- How do you keep them from returning?
- If they're authenticated, it's easy
- If not, make sure that connection includes a hardware device ID, e.g., a MAC address or motherboard serial number


Is Hardware ID Secure?

Is Hardware ID Secure?

- In theory, no—it's client-side, and hence is spoofable
- Changing a MAC address on a computer is easy for knowledgeable users
- But—most people don't know how to do that
- Most people don't know what hardware ID is used by their client
- It's harder to do for phones and tablets
- As a *practical* matter, it may be good enough
- Failing that, lock the conference

Is Hardware ID Secure?

- In theory, no—it's client-side, and hence is spoofable
- Changing a MAC address on a computer is easy for knowledgeable users
- But—most people don't know how to do that
- Most people don't know what hardware ID is used by their client
- It's harder to do for phones and tablets
- As a *practical* matter, it may be good enough
- Failing that, lock the conference

 As always, note that security is a *systems property*—it's not just cryptography

Thumbnails and Large Images

- Conferencing systems generally have thumbnail displays for some users but full-screen for the speaker
- If you have multiple screens, Zoom will let you see up to 49 thumbnails *and* a full-screen image of the speaker
- How is this done?
- Two ways. . .

Two Ways

- If there's no end-to-end encryption: the conference server generates thumbnails of each stream and sends whichever is desired to the client
- Or: clients upload two video streams, one high-resolution and one low-resolution for thumbnails; other clients request the ones they want
- Most end users will not have enough bandwidth for circa 50 high-resolution images
- Or maybe they will—Zoom isn't used much for real full-motion video (except when cats crash the meeting)
- (Of course, most users don't have their cameras on most of the time. . .)

Intermediate Resolutions?

- Maybe someone is on a lower-bandwidth link and can't handle a full-res feed
- Maybe someone wants intermediate-size “thumbnails”
- (Actually, Zoom's “thumbnails” do enlarge when there are only a few participants)
- If there's no end-to-end encryption, the conference server (which is likely a distributed system in its own right) can rescale or even transcode video

Do We Actually Need End-to-End Encryption?

- As noted last week, few phone calls are encrypted
- Do you trust the phone company? You do for most phone calls
- Is your software vendor the same company as the conference operator?
- 👉 Why do you trust the software?
 - And—was Elon Musk in your threat model? Should he be?

Are Internet Services Different?

- If a computer is hacked, it's hacked, and everything on it is available to the attacker
- With phone switches and Internet routers, the control plane is separate from the data plane—the control plane sets up a forwarding path in the hardware, but doesn't see the actual data
- However...

It's Not That Simple

- By law, in the US and many other jurisdictions, communications services have to provide for “lawful access”
- That is, there has to be a way to silently send a copy of all data in a conversation to law enforcement
- (In the US, that law is known as CALEA—Communications Assistance to Law Enforcement Act)
- However, in most places providers don't have to turn over encryption keys (caution: *very oversimplified!*)
- In other words, hacking the control plane is enough
- Again: what is your threat model?

Who is Talking?

- Perhaps the speaker's identity is sensitive
- (Imagine Volodymyr Zelenskyy holding a distributed cabinet meeting—and who might be interested in which IP address is his)
- Do you send encrypted “silence packets” continuously?
- That chews up bandwidth
- They may also compress differently
- Similar issues for camera on/off
- N.B. Researchers have shown that they can do recognition of language in encrypted VoIP by looking at packet size distribution and timing

User Interface/User Experience (UI/UX) and Security

- User interfaces matter
- Defaults matter
- If you get them wrong, security can suffer—but sometimes, you have to impose inconvenience to achieve security

Zoom URLs

- Zoom URLs frequently include a password
- Is this insecure?

- Zoom URLs frequently include a password
- Is this insecure?
- How will the password be distributed if not in the URL?
- If it's via the same medium—email, a restricted web site, etc.—there's little security benefit to separating the two
- You have hurt the user experience without reaping any benefit

- When I click on a Zoom URL, I have to give my browser permission to open Zoom. I then have to start video and audio manually. Why?

- When I click on a Zoom URL, I have to give my browser permission to open Zoom. I then have to start video and audio manually. Why?
- Think IFRAMEs—if Zoom auto-opened the client and always fired up the camera and mic, any web site you visited could spy on you while their page was open
- It's a simpler user experience if all sites are honest, but. . .
- (Zoom made a similar mistake at first, in the name of better UX)

- Three years ago, there was a lot of concern about Zoom and China
- A lot of the code was developed in China
- Some of the servers were in China
- The concern, of course, was the Chinese government
- Were the issues real?
- If so, are they fixed?

The Servers

- You don't want servers in China for US-to-US sessions
- The reason has nothing to do with politics!
- What's the issue?

The Servers

- You don't want servers in China for US-to-US sessions
- The reason has nothing to do with politics!
- What's the issue?
- Latency—the speed of light is too slow
- The issue is conversational dynamics—it's hard to tell when it's ok to speak if packets have to go halfway around the world and back before you hear someone else starting to talk

The Servers and Security

- Was there a security issue?
- In theory, if you use end-to-end encryption, there isn't—the conference server never sees the content
- It does, however, see metadata, which can be very revealing
- Zoom over Tor would have even more latency. . .

- Software development is much more of an issue
- You don't know what code does
- It's *hard* to find deliberate back doors in code via an audit

Finding Back Doors

“At their first meeting, Gosler asked Morris Sr. the question that had been troubling him for some time now. ‘How complex can software be for you to have total knowledge of what it could do?’...

“Morris Sr. told Gosler that, of the top of his head, he would have ‘100 percent confidence’ in an application that contained 10,000 lines of code or less, and zero confidence in an application that contained more than 100,000 lines of code. . . Turns out that it was an application with fewer than 3,000 lines of code.

“Morris invited an elite NSA squad of PhDs, cryptographers, and electrical engineers to take a look. Not one discovered Gosler’s implant.”

Perlroth, *This is How They Tell Me the World Ends*

Can We Monitor?

- If an application steals data, it has to have some way to exfiltrate it
- Can that work here?

Can We Monitor?

- If an application steals data, it has to have some way to exfiltrate it
- Can that work here?
- Possibly not. . .
- Why not?

- It may be possible to detect exfiltration of actual content—audio and especially video streams are big
- However, what if an attacker joins the conversation surreptitiously?
- Maybe the code will never add user “Boris Badenov” to the visible participant list
- Maybe the code will simply exfiltrate the session key and let the SIGINT folks monitor links
- Or maybe we can find such implants—the Gosler/Morris story was from 1987

Exfiltrating Keys

- How would you surreptitiously exfiltrate a key?
- Steganography!
- Do innocuous-seeming queries of, say, the DNS, but embed key bits in the queryID?
- Modulate interpacket timing
- It's noisy, so use an error-correcting code
- And you only have to exfiltrate enough bits that brute force on the rest works

- Early on, Zoom had numerous errors
- Most of these appear to have been accidents, but they showed signs of lack or care of lack of knowledge
- Example: encrypting sessions with ECB mode
- *Many* coding errors in the Linux client

Coding Errors

- Calling untrusted functions, e.g., `strcpy()` and `popen()`
- Not using standard defenses, e.g., stack canaries
- Many more

Corporate Security Consciousness

- There was obviously no corporate consciousness about security
- The cryptography was laughably bad
- It seems probable, though not certain, that the Zoom server was equally bad
- There was likely a *cultural problem* at Zoom—they didn't take security seriously, beyond “check the box” issues like encryption (and they did that incorrectly)

Should Enterprises Have Abandoned Zoom Three Years Ago?

- We were at the start of a pandemic emergency
- None of the competing products were nearly as user-friendly
- There was a *business need* to run some sort of conferencing system
- Zoom was very familiar to some people (I'd taught a class via Zoom in 2018)
- For a university, at least, the risk was low compared with the benefits

Zoom's Response

- Luckily, Zoom listened to its critics and cleaned up its act
- They brought in outside experts to review their system
- The cryptography was reworked by top-notch people
- They fixed their errors, the egregious stuff quickly and other stuff as they could
- They were *not* defensive

Zoom's Response

- Luckily, Zoom listened to its critics and cleaned up its act
- They brought in outside experts to review their system
- The cryptography was reworked by top-notch people
- They fixed their errors, the egregious stuff quickly and other stuff as they could
- They were *not* defensive
- But could they review all of their legacy code?

Zoom's Response

- Luckily, Zoom listened to its critics and cleaned up its act
- They brought in outside experts to review their system
- The cryptography was reworked by top-notch people
- They fixed their errors, the egregious stuff quickly and other stuff as they could
- They were *not* defensive
- But could they review all of their legacy code?
- And—were other major conferencing platforms more secure?

Insecurity Isn't a Sin

- Running software carries a risk/benefit tradeoff
- You want stuff to be secure, but you don't always have a choice
- And if your analysis is wrong and you're hacked—again, it's not a sin

Questions?



(Prothonotary warbler Central Park, April 20, 2022)