

Qcord privacy notice

What the mod sends, what our relay keeps.

Published at <https://qwu.world/minecraft/qcord/privacy> · Updated 2026-09-06 · Applies to Qcord 1.0.0.

Who runs this

Qcord and its relay are run by **QwU World**. Contact: it.qwuworld@gmail.com, or `#bugs-suggestions` in the Minecraft section of our Discord, <https://discord.gg/sSENQuz6a7>.

Qcord is a client-side Minecraft mod. Video, screen shares and voice travel between party members through one relay we rent:

LOCATION	ADDRESS	PROVIDER
Warsaw, Poland (EU)	<code>qwu.world:8797</code>	OVH Sp. z o.o.

There is no second relay. While this one is down, nothing on the default settings works.

You do not have to use it. In `config/qcord-client.toml`, `network.relayAddress` points parties at a relay of your own, and `serverLobby.autoJoin = false` stops the lobby check described below. `network.lanMode` hosts a party from your own game, but does not stop the lobby check on its own.

Nothing happens until you agree

On first launch Qcord shows a welcome screen that says what it captures, where it goes and what the relay sees, and asks whether server lobbies may be joined automatically. Until you press **Get started**, the mod opens no connection. `/qcord privacy` reopens the screen; the answer is stored as `privacy.consentVersion` in the config file, and 0 makes the mod ask again.

The disclosure is at version 2. Anybody who answered version 1 is asked again, because two things changed: media is now encrypted end to end, and invitations go through the relay instead of Minecraft chat, which lets the relay see who invited whom.

What reaches the relay

When you join a multiplayer world. About three seconds after joining, and only if the lobby check is on, your client asks the relay whether that server has a lobby. The question carries one value: the first 16 hex characters of a SHA-256 of the server address. That is not reversible on its own, but server addresses are guessable, so treat it as "which server you play on". No name or UUID is sent with it. On a server running Qcord-SL the value is derived from the server's signing key instead, so it is stable across address changes. In single player the mod never contacts the relay on its own. The welcome screen draws the lobby toggle **on**; pressing Get started without touching it turns the check on.

When you join a party or a lobby. The relay logs your **IP address truncated to its network** (an IPv4 address loses its last octet, an IPv6 address everything after the first 48 bits, the port is dropped; the full

address exists only while the connection is open), your **Minecraft username**, the **Qcord version**, and the **room code**. Your **account UUID** and **two public keys** (one per connection, one long-term) go to the other members of the room so their clients can tell whose stream is whose and encrypt to you; none of the three is logged. On a Qcord-SL server your client also presents a 238-byte statement signed by that server, saying you may join its lobby; it names your UUID and nothing else about you.

What else the relay writes, with your username and room code: when you leave, with counters of packets sent and dropped; the first time you share sound or speak in a room; when you are kicked, muted, restricted or promoted, and by whom; when a room or lobby is created and when a host toggles voice; a host action of yours that was denied; when you exceed a rate limit for media, key messages or control messages; a refused connection and why (version, full room, wrong token, duplicate name), with your truncated IP; a reconnect that took over a stale session; a message that broke the protocol. Connections that never reach a room, including every lobby check, and a connection closed because the client stopped reading, are logged with the truncated IP and the reason they closed. Once a minute the relay writes counts per room of key messages, and relay-wide counts of invitations; those lines carry no names. Who invited whom is held in memory for the two minutes an invitation lives and is never written down.

Your camera, screen, share sound and voice. They are encrypted on your computer and decrypted on the computers of the other people in your room. In a private party the keys come from your invite code, which the relay never receives: it gets a token derived from the code by a one-way function, enough to check that you were invited. There is no unencrypted mode; the mod cannot send or play media in the clear. The relay keeps the most recent video keyframe of each stream in memory, as ciphertext, so a new viewer sees a picture sooner. In a private party a joiner cannot read it, because a join changes the keys. In a server lobby keys do not change on a join, so a joiner can read that frame and the moment before their arrival.

Capture runs only while you switch it on and stops when you turn it off, leave the party or leave the world. The microphone opens only in a private party whose host has switched voice on. A server lobby has no voice.

What the relay still sees, because it has to route. Who is in a room, their names, who is host, who was muted or kicked, and when each of them came and went. For every packet: sender, stream (camera, screen, voice), codec, whether it is a keyframe, sequence number, timestamp, resolution and encrypted size, and which set of keys it was sent under. From those: your bitrate and, because voice is sent only while you speak, when you are speaking. Which cameras your client asked for, which in a lobby follows who is near you. When keys changed, and so when somebody joined or left. Who you invited and when, for every invitation, and which lobby it was addressed into; whether that person was there. Nothing is padded and there is no cover traffic.

Server lobbies on disk. Opening a lobby writes one record to `lobbies.json` on the relay: server key, room code, join token, policy, the UUIDs of its creator and hosts, creation and last-seen times, and up to 16 hashed alias addresses. A lobby claimed through Qcord-SL also stores that server's public key, its rotation count and when it was pinned. The record lives until a host deletes it or until 60 days pass with nobody using it. Private parties are not stored; a room is deleted 60 seconds after its last member leaves.

What we do not do

No accounts, no analytics, no crash reporting, no advertising, no update check, nothing downloaded at run time, nothing sold or shared. The relay speaks only its own protocol, over TLS, to the address in your config. The bundled FFmpeg and WebRTC libraries contain network code Qcord never calls.

To list screens, windows and devices the mod runs tools already on your system and reads their output: PowerShell on Windows; `osascript`, `system_profiler` and `screencapture` on macOS; `xrandr`, `wmctrl` and `pactl` on Linux. On macOS, one button runs `/usr/bin/open` to show the Screen Recording settings pane. The only other programs it starts are its own two capture helpers, `qcord-sck` (macOS) and `qcord-wasapi.exe` (Windows), unpacked from the jar and run only while you capture.

Why, and for how long

Legal basis: our legitimate interest in running the service (GDPR Article 6(1)(f)). Relay logs are kept **14 days**. `lobbies.json` is kept until the lobby is deleted or expires. An invitation in flight lives at most two minutes, in memory.

You can ask for a copy of what we hold, and ask us to correct, delete or restrict it, or object: write to it.qwuworld@gmail.com with your Minecraft username. You can also complain to your data protection authority.

Before you use it

- **The invite code is the key to the room.** Anyone holding it can join and can also decrypt the room's traffic if they can get at it. A party is protected from the relay operator; it is not protected from someone who has the code and controls or colludes with the relay.
- **Invitations travel through the relay, sealed for one person.** Both sides see a 10-digit safety code to read aloud. An invitation from a sender whose key does not match the one your computer knows is refused, and by default only people on your Minecraft player list can invite you. When somebody cannot be reached that way, the code goes to your clipboard and the mod says why. Do not paste it into server chat: a code seen by the server is a code the operator has. If one turns up in chat anyway, Qcord warns you and offers a new code (`/qcord newcode`), which reopens the room under a new key and disconnects everybody so you can invite them back.
- **A server lobby without Qcord-SL is encrypted but not authenticated.** Its keys derive from a value anybody can compute. The network cannot read it; a relay willing to substitute keys could. The badge says *Encrypted, membership not verified*. Assume anyone who knows the server can walk in and see the roster and cameras.
- **A server lobby with Qcord-SL is keyed by that server.** The relay gets a token and a signature, never the secret, and cannot read the room. The badge says *Encrypted end to end, verified by this server*; your client refuses to join if the relay reports a different key than the server handed you. The server owner holds the key and is the party responsible for that lobby; their notice is at <https://qwu.world/minecraft/qcord-sl/privacy>.
- **Screen sharing shows what is on your screen**, notifications included, to everyone in the room.

- **What encryption does not cover:** the people in your room, who can record; a modified Qcord on any member's machine; your own computer, where `identity.bin` and `known_peers.txt` are plain files and Java cannot reliably erase secrets from memory; a copy of `identity.bin`, which lets its holder appear as you (delete the file and a new key is made, with one warning to your friends); a relay that substitutes keys during an invitation to somebody new, which the safety code is there to catch; a removed member who kept the code (use a new code); traffic analysis; and us refusing to carry your traffic.
 - **The design has not been reviewed by an outside cryptographer.** It uses standard primitives from the Java platform (X25519, AES-GCM, HKDF-SHA-256, Ed25519) in an arrangement of our own, specified in `docs/SPEC-E2EE.md`, `docs/SPEC-LOBBY-V2.md` and `docs/SPEC-COMPANION.md` in the repository. It has been measured under load: a full 48-seat lobby with cameras on is about 206 Mbit/s leaving our single relay, and a 16-seat party about 60 Mbit/s (`docs/CHURN-RESULTS.md`).
-

The optional server mod

Qcord-SL is a separate mod a server owner may install on a dedicated server. Qcord never requires it. A private party shuts out both the Minecraft server and the relay. A lobby with Qcord-SL shuts out the relay operator; the server owner, acting together with the relay, could read it. A lobby without it can be read by a relay that substitutes keys.

What a Minecraft server can see

No video, no sound, no invitation, nothing about a party. The server can see that the mod registered two plugin-message channels, as with any mod that registers one. On a server running Qcord-SL, once you have agreed to the welcome screen, your client asks that server for the lobby key over those channels and receives it; the server therefore knows you are a candidate for its lobby.

What stays on your computer

- `config/qcord-client.toml`: settings, the UUIDs of players whose volume or camera size you adjusted, and up to 128 UUIDs of people you blocked from inviting you.
 - `config/qcord/identity.bin`: your long-term key. Its public half goes to the room; the secret half never leaves. Deleting it makes a new key and warns your friends once.
 - `config/qcord/known_peers.txt`: for each player you shared a room with, their UUID and the key they were first seen with, up to 4096 entries. It is never sent anywhere. A changed key shows an amber dot until you press **This is them**.
 - `qcord-natives/` in your game directory: the extracted libraries and capture helpers.
 - `latest.log` records the names of the devices and windows you captured.
-

The source

<https://github.com/QwU-World/qcord> holds the mod, the relay and Qcord-SL. Source-available: read, audit, build and run your own; do not redistribute.

Age

Qcord is not intended for children under 13.

Changes

When this notice changes, the date at the top changes with it.

© 2026 QwU World