Resolve an Arma 3 mod update or signature mismatch without bypassing checks

“Wrong signature”, missing-addon errors and a mission returning to the lobby point to different stages of an Arma 3 mod problem. Keep the intended security settings in place and find the exact disagreement before changing the community's mod set.

1. Identify the first failing stage

Record the client message, UTC time, mission and server build. Note whether all players fail or only one preset does. Save the current RPT excerpt around that attempt; an old report from a different profile can lead you toward the wrong launch configuration.

If the process never becomes ready, examine the first dependency or path error. If players reach the lobby but a mission cannot start, check the mission's requirements. If a player is rejected during joining, identify the named addon or signature rather than changing unrelated gameplay settings.

2. Compare intended and loaded content

Read the active `-mod` and `-serverMod` lists and compare them with the directories actually staged for this service. Verify folder spelling and dependency instructions. A downloaded Workshop item that is absent from the relevant launch list may never load.

Check the client preset against the ordinary mod list. Server-only mods are not broadcast as client requirements, so do not move a required client package to `-serverMod` just to hide the mismatch. Use the author's role requirements for each component.

3. Check the signature chain

For the named addon, compare the trusted publisher's public `.bikey`, the supplied `.bisign` and the PBO from the same release. A correctly named file is not evidence that it came from the correct publisher. A stale key, mismatched artifact or modified signed PBO needs a supported release, not a new exception for every rejected player.

Review the configured key directories; installations may use additional key-folder options. Do not disable signature verification, bypass BattlEye or manufacture a replacement signature for somebody else's release. Keep full configuration dumps private because they can contain administrator passwords.

4. Coordinate the update

Close the affected player's game and launcher before allowing Steam to finish applicable updates, then check the selected preset again. On the server, schedule a normal stop and preserve mission files, profiles, configuration and framework databases before replacing artifacts through the supported workflow.

Stage risky updates separately. A mission framework can migrate stored campaign state, making a simple binary downgrade insufficient. Retain the pre-update matching dataset and permitted artifacts; a Workshop ID list alone may not retrieve an older release.

5. Isolate and verify

Use an isolated copy to test the named dependency group with an ordinary player account. Change one variable at a time and retain logs from each run. Do not remove production content mods and let the active campaign save without them.

Success means the intended client joins, the mission actually runs, required features work and persistent progress survives a normal restart. Send support or the mod author the named error, affected source IDs, role lists, server/client build comparison and redacted RPT excerpt if the issue remains. Recover the compatible mission/database/configuration checkpoint together when rollback is required.

Related CloudExa guides

- Diagnose Arma 3 connection failures and mission loading errors
- Schedule game server backups and check retention

Official references

- Bohemia Interactive: active mod roles and startup options
- Bohemia Interactive: addon signature file relationships
- Bohemia Interactive: Workshop update and Launcher behaviour
- Bohemia Interactive: server configuration and security

Revision #1
Created 2026-11-10 11:05:47 UTC by CloudExa Hosting Admin
Updated 2026-11-10 11:05:47 UTC by CloudExa Hosting Admin