Intel PTT / Azure AIK Attestation Fix Guide

Fixing HTTP 400 (0x80190190) and HTTP 404 “authority does not exist” caused by stale Intel EK certificate state

Initial error	HTTP 400 / 0x80190190 — No valid TPM EK/Platform certificate provided
Second error	HTTP 404 — The authority INTC-KeyId-… does not exist
Root cause	Windows cached an old Intel EK intermediate chain that did not match the chain stored in TPM NV
Final fix	Install the real TPM chain, remove the stale branch, clear only the stale TPM WMI intermediate-cache entries, reboot, and let Windows rebuild them from TPM NV

Important: Do not blindly copy the certificate thumbprints or Intel product-CA number from this guide. Your TPM may use a different Intel certificate branch. The goal is to make Windows match the certificates physically stored in your own TPM NV indexes.

What this fixes

This guide is for systems where Intel PTT / TPM 2.0 and Secure Boot appear healthy, but Windows AIK enrollment or a TPM-backed attestation workflow fails. In this case, the machine was healthy at the TPM level, but Windows had stale Intel intermediate certificates cached locally.

- TPM 2.0 present and ready
- Secure Boot enabled
- Ready For Attestation = True
- AIK enrollment initially fails with HTTP 400 / 0x80190190
- After partially fixing the chain, enrollment may instead fail with HTTP 404 / authority does not exist

Before you start

Run the commands as Administrator. This guide modifies certificate stores and a specific TPM WMI registry cache. Backups are included before any destructive registry step.

- Do not clear the TPM just for this guide unless you have another reason to do so.
- Do not delete EKCertStore, EKCertStoreECC, or the entire TPM WMI tree.
- Do not remove certificates merely because their subject names look similar; compare actual thumbprints and issuer chains.

1. Install TPM Diagnostics

Open PowerShell as Administrator:

```
tpmtool oc add
```

Verify that TPM Diagnostics is available:

```
TPMDiagnostics.exe /?
```

2. Confirm the TPM is healthy

```
Get-Tpm
```

You should normally see:

```
TpmPresent   : True
TpmReady     : True
TpmEnabled   : True
TpmActivated : True

tpmtool getdeviceinformation
```

Look for:

```
Ready For Attestation: True
Is Capable For Attestation: True
```

3. Reproduce the current AIK enrollment error

Run this from an Administrator Command Prompt (cmd.exe):

```
certreq.exe -EnrollAIK -machine -config ""
```

The initial failure in this case was:

```
HTTP 400 Bad Request
0x80190190
No valid TPM EK/Platform certificate provided in the TPM identity request message.
```

4. Read the real Intel intermediate chain from TPM NV

On this machine, Intel stored the intermediate CA bundle in NV index 0x1C00100. Dump it:

```
TPMDiagnostics.exe ReadNVIndex 0x1C00100 -file C:\temp\Intel-NV-Chain.bin
```

Create an output folder:

```
New-Item -ItemType Directory -Force C:\temp\Intel-NV-All | Out-Null
```

Extract every DER certificate from the NV blob with PowerShell:

```
$data = [System.IO.File]::ReadAllBytes("C:\temp\Intel-NV-Chain.bin")
$out  = "C:\temp\Intel-NV-All"
$count = 0

for ($i = 0; $i -lt ($data.Length - 4); $i++) {
    if ($data[$i] -eq 0x30 -and $data[$i + 1] -eq 0x82) {
        $len = ([int]$data[$i + 2] * 256) + [int]$data[$i + 3] + 4

        if ($len -ge 400 -and ($i + $len) -le $data.Length) {
            $bytes = New-Object byte[] $len
            [Array]::Copy($data, $i, $bytes, 0, $len)

            try {
                $cert = [System.Security.Cryptography.X509Certificates.X509Certificate2]::new($bytes)
                $count++
                $file = "$out\cert-$count.cer"
                [System.IO.File]::WriteAllBytes($file, $bytes)

                Write-Host ""
                Write-Host "CERT $count"
                Write-Host "Subject: $($cert.Subject)"
                Write-Host "Issuer : $($cert.Issuer)"
                Write-Host "SHA1   : $($cert.Thumbprint)"

                $i += ($len - 1)
            }
            catch {}
        }
    }
}

Write-Host ""
Write-Host "Certificates extracted: $count"
```

On this machine, the three extracted certificates were:

```
cert-1 = CSME ADL ROM CA
cert-2 = CSME ADL SVN01 Kernel CA
cert-3 = CSME ADL PTT 01SVN
```

5. Inspect the certificates and identify your real Intel branch

```
Get-ChildItem C:\temp\Intel-NV-All\*.cer | ForEach-Object {
    Write-Host "`n================ $($_.Name) ================"
    certutil -dump $_.FullName | Select-String `
        "Subject:","Issuer:","Subject Key Identifier","KeyID="
}
```

The chain discovered on this specific machine was:

```
EK
  -> CSME ADL PTT 01SVN
  -> CSME ADL SVN01 Kernel CA
  -> CSME ADL ROM CA
  -> ODCA 2 CSME P_ADL 00002983 Issuing CA
  -> ODCA CA2 CSME Intermediate CA
  -> Intel OnDie Root
```

Do not assume 00002983 is yours. Use the AIA URL embedded in your own ROM certificate.

6. Find and download the correct Intel product CA

Inspect the AIA URL in the ROM certificate:

```
certutil -dump C:\temp\Intel-NV-All\cert-1.cer |
    Select-String -Pattern "http","KeyID","Authority Key Identifier","Subject Key Identifier" -Context
2,2
```

On this machine the URL was:

```
https://tsci.intel.com/content/OnDieCA/certs/ADL_00002983_ODCA_CA2.cer
```

Download the URL shown on your own machine. Example for this machine:

```
Invoke-WebRequest `
  "https://tsci.intel.com/content/OnDieCA/certs/ADL_00002983_ODCA_CA2.cer" `
  -OutFile "C:\temp\ADL_00002983_ODCA_CA2.cer"

certutil -dump C:\temp\ADL_00002983_ODCA_CA2.cer
```

7. Download the upper Intel intermediate and root

```
Invoke-WebRequest `
  "https://tsci.intel.com/content/OnDieCA/certs/ODCA_CA2_CSME_Intermediate.cer" `
  -OutFile "C:\temp\ODCA_CA2_CSME_Intermediate.cer"

Invoke-WebRequest `
  "https://tsci.intel.com/content/OnDieCA/certs/OnDie_CA_RootCA_Certificate.cer" `
  -OutFile "C:\temp\Intel-OnDie-Root.cer"
```

8. Import the correct certificate chain

Import the three certificates extracted from TPM NV:

```
certutil -addstore -f CA C:\temp\Intel-NV-All\cert-1.cer
certutil -addstore -f CA C:\temp\Intel-NV-All\cert-2.cer
certutil -addstore -f CA C:\temp\Intel-NV-All\cert-3.cer
```

Import your Intel product CA (example shown for the 00002983 branch):

```
certutil -addstore -f CA C:\temp\ADL_00002983_ODCA_CA2.cer
```

Import the upper Intel intermediate and OnDie root:

```
certutil -addstore -f CA C:\temp\ODCA_CA2_CSME_Intermediate.cer
certutil -addstore -f Root C:\temp\Intel-OnDie-Root.cer
```

9. Dump and verify the RSA EK certificate

Inspect TPM NV indexes first:

```
TPMDiagnostics.exe NVSummary
```

On this machine the RSA EK certificate was at 0x1C00002:

```
TPMDiagnostics.exe ReadNVIndex 0x1C00002 -file C:\temp\EK-1C00002.cer
```

Verify the complete EK chain:

```
certutil -verify -urlfetch C:\temp\EK-1C00002.cer
```

Before the chain was fixed, the important failure was:

```
CERT_TRUST_IS_NOT_SIGNATURE_VALID
TRUST_E_CERT_SIGNATURE
The signature of the certificate cannot be verified.
```

After importing the correct chain, the command completed successfully. A revocation-offline warning may still be shown if the CRL endpoint cannot be reached; in this case certutil still ended with a successful verification.

10. Find stale Intel certificates in LocalMachine\CA

```
Get-ChildItem Cert:\LocalMachine\CA |
Where-Object {
    $_.Subject -like '*CSME*' -or
    $_.Subject -like '*ODCA*'
} |
Format-List Subject,Issuer,Thumbprint
```

Compare these thumbprints and issuers against the certificates extracted from your TPM NV bundle. Remove only certificates that you have positively identified as belonging to the stale/wrong branch.

Example: on this machine the stale branch was:

```
0A9A7B50641A20BE005718405762D977B3331B9E
F1F86D8C4F36E69E34F01A670E2785E84F8A5B71
2CFDBE2878BF7D24E20BF1854F9382E11C9B4551
1165BCED7658A41851E9382FECF3FA4F372FABED
```

These exact commands were used on that machine:

```
certutil -delstore CA 0A9A7B50641A20BE005718405762D977B3331B9E
certutil -delstore CA F1F86D8C4F36E69E34F01A670E2785E84F8A5B71
certutil -delstore CA 2CFDBE2878BF7D24E20BF1854F9382E11C9B4551
certutil -delstore CA 1165BCED7658A41851E9382FECF3FA4F372FABED
```

Warning: Do not run those deletion commands unless those exact certificates are stale on your own machine.

11. Retry AIK enrollment — the 400 may become a 404

Run from Administrator Command Prompt:

```
certreq.exe -EnrollAIK -machine -config ""
```

On this machine, fixing the cryptographic chain changed the failure from HTTP 400 to:

```
HTTP 404 Not Found
The authority "INTC-KeyId-..." does not exist.
```

That turned out to be a clue that a second stale local cache still existed.

12. Inspect the TPM WMI intermediate certificate cache

```
TPMDiagnostics.exe EkCertStoreRegistry
```

Then inspect the actual registry-backed cache:

```
reg query
"HKLM\SYSTEM\CurrentControlSet\Services\TPM\WMI\Endorsement\IntermediateCACertStore\Certificates" /s
```

Compare the subkey thumbprints with the SHA1 thumbprints of the cert-1/cert-2/cert-3 files extracted directly from TPM NV.

13. Back up the TPM WMI intermediate cache

Before deleting anything, export the exact subtree:

```
reg export "HKLM\SYSTEM\CurrentControlSet\Services\TPM\WMI\Endorsement\IntermediateCACertStore"
C:\temp\TPM-IntermediateCACertStore-backup.reg /y
```

If you ever need to restore it:

```
reg import C:\temp\TPM-IntermediateCACertStore-backup.reg
```

14. List the cached thumbprints and compare them with TPM NV

```
Get-ChildItem
"HKLM:\SYSTEM\CurrentControlSet\Services\TPM\WMI\Endorsement\IntermediateCACertStore\Certificates" |
    Select-Object PSChildName
```

Get the actual TPM NV certificate thumbprints again:

```
Get-ChildItem C:\temp\Intel-NV-All\*.cer | ForEach-Object {
    $cert = [System.Security.Cryptography.X509Certificates.X509Certificate2]::new($_.FullName)

    [PSCustomObject]@{
        File       = $_.Name
        Subject    = $cert.Subject
        Issuer     = $cert.Issuer
        Thumbprint = $cert.Thumbprint
    }
}
```

If the registry subkeys do not match the certificates in TPM NV, the WMI cache is stale.

15. Delete only the stale WMI intermediate certificate entries

Do not delete EKCertStore, EKCertStoreECC, the parent IntermediateCACertStore key, or the entire TPM WMI tree. Delete only the stale certificate subkeys under IntermediateCACertStore\Certificates.

Example from this machine:

```
$store =
"HKLM:\SYSTEM\CurrentControlSet\Services\TPM\WMI\Endorsement\IntermediateCACertStore\Certificates"

Remove-Item "$store\0A9A7B50641A20BE005718405762D977B3331B9E" -Recurse -Force
Remove-Item "$store\2CFDBE2878BF7D24E20BF1854F9382E11C9B4551" -Recurse -Force
Remove-Item "$store\F1F86D8C4F36E69E34F01A670E2785E84F8A5B71" -Recurse -Force
```

Again: replace those hashes with the stale entries you identified on your own system. Do not blindly copy them.

16. Reboot — this was the critical final step

```
shutdown /r /t 0
```

After reboot, Windows should rebuild the WMI intermediate certificate cache from the TPM NV data.

```
Get-ChildItem
"HKLM:\SYSTEM\CurrentControlSet\Services\TPM\WMI\Endorsement\IntermediateCACertStore\Certificates" |
    Select-Object PSChildName
```

On the fixed machine, the stale cache changed from:

```
0A9A7B50641A20BE005718405762D977B3331B9E
2CFDBE2878BF7D24E20BF1854F9382E11C9B4551
F1F86D8C4F36E69E34F01A670E2785E84F8A5B71
```

to the actual certificates embedded in TPM NV:

```
5E9B63EAEC3447471F409B92AE679B982E1715D7
6DD1BA6052BDD47805534A8482E20CB9F7100EA7
72F23EB50029142DCFBBCE997C7906088FF35CDB
```

17. Verify the rebuilt cache and EK chain

```
TPMDiagnostics.exe EkCertStoreRegistry

certutil -verify -urlfetch C:\temp\EK-1C00002.cer
```

The EK should remain signature-valid. You should not see TRUST_E_CERT_SIGNATURE or CERT_TRUST_IS_NOT_SIGNATURE_VALID.

18. Final AIK enrollment test

Run from Administrator Command Prompt:

```
certreq.exe -EnrollAIK -machine -config ""
```

After Windows rebuilt the stale WMI intermediate certificate cache from TPM NV, AIK enrollment finally succeeded.

19. What actually went wrong

The key lesson is that the following three locations must agree:

```
TPM NV certificate chain
        ↓
Windows LocalMachine\CA certificate chain
        ↓
TPM WMI IntermediateCACertStore
```

On the affected machine, they did not agree. Windows had cached an older Intel PTT/CSME chain with almost identical certificate subject names. That made the problem easy to miss.

- With the wrong chain, Windows could build a name-compatible path but the EK signature did not verify, causing the HTTP 400 behavior.
- After correcting the certificate chain but before rebuilding the stale TPM WMI cache, enrollment moved to an HTTP 404 authority error.
- After clearing only the stale WMI intermediate-cache entries and rebooting, Windows repopulated the cache from TPM NV and AIK enrollment succeeded.

20. Reference system that was fixed

CPU	Intel Core i5-14600KF
Motherboard	Gigabyte B760 DS3H WIFI6E GEN5
TPM	Intel PTT / TPM 2.0
TPM firmware	600.18.1040.2765
TPM model	ADL
Intel branch found in TPM NV	ODCA 2 CSME P_ADL 00002983 Issuing CA

21. Troubleshooting already attempted before the actual fix

- BIOS update
- Intel ME update
- TPM clear/reprovision
- TPM maintenance
- EK reinstall from NVR
- Core Provisioning EK retrieval
- AIK recreation
- DISM /RestoreHealth and SFC
- Windows 11 in-place repair installation

None of those resolved the issue. The final piece was correcting the stale TPM WMI IntermediateCACertStore and rebooting so Windows rebuilt it from TPM NV.

22. Condensed checklist

1. Confirm TPM/Secure Boot are healthy.
2. Dump 0x1C00100 and extract the real TPM intermediate certificates.
3. Use the ROM certificate AIA to download the correct Intel product CA for your own branch.
4. Import the TPM intermediates, Intel product CA, upper ODCA intermediate, and OnDie root.
5. Dump the RSA EK (commonly 0x1C00002) and verify it with certutil -verify -urlfetch.
6. Remove only the stale/wrong Intel branch from LocalMachine\CA.
7. Retry AIK enrollment; if it becomes a 404, inspect the TPM WMI IntermediateCACertStore.
8. Back up that registry subtree.

9. Delete only stale certificate subkeys under IntermediateCACertStore\Certificates.
10. Reboot and verify Windows rebuilt the cache using the TPM NV certificate thumbprints.
11. Verify the EK again and retry certreq -EnrollAIK.

Documented from a successful real-world repair. Validate certificate identifiers against your own TPM before deleting anything.