

The eMudhra Intermediate CA Compromise: Why CRLs Broke First

TechRounder PDF Edition

Live article: <https://www.techrounder.com/security/emudhra-crl-signature-failure-since-may/>

By Vipin PG | Published October 21, 2026 | Updated October 21, 2026 | Format: Deep Dive | 14 min read

In brief

Eight eMudhra CA keys were small integers anyone could recover. Six CRLs for the same TLS intermediates had stopped verifying seventeen months earlier.

Eight eMudhra certificate authority private keys were found to be so weak that researchers could recover them on ordinary computers. The keys used P-384 elliptic-curve cryptography, but their generation left them with almost none of the randomness needed to make that protection effective.

The weakness was reported on 17 October 2026. Six of the affected intermediate certificate authorities had also been involved in a certificate revocation failure seventeen months earlier, raising questions about what the original investigation had established.

A researcher identifying himself as Wayne demonstrated control of all eight private keys and reported the findings to Apple, Chrome, Microsoft, Mozilla and eMudhra. Chrome blocked the affected TLS certificates at 19:45 UTC that day. Mozilla completed its OneCRL block at 22:59 UTC. Three days later, Mozilla announced the emergency removal of trust in an affected eMudhra root certificate.

Both incidents are discussed in Mozilla's public incident record. The earlier CRL failure had been investigated and closed as a signing-configuration error. Archived monitoring data suggests the problem was visible before the date given in eMudhra's 2025 incident report.

The eMudhra intermediate CA compromise: one ceremony, eight keys

All eight affected certificates originated from a key-generation ceremony held on 30 April 2025. According to the incident record, the ceremony produced 24 intermediate certificate authorities across three series, using RSA-4096 and P-384 keys for four certificate purposes.

The compromised group consisted of every P-384 intermediate in the B and C series: six intended for TLS and two for code signing.

The TLS intermediates chained to emSign Root TLS CA - G3 and were trusted by the four major browser and operating-system root programs. The code-signing intermediates chained to emSign Root CS CA - G3 and were trusted by Microsoft.

Each affected intermediate also had a cross-sign from Sectigo's AAA Certificate Services. That gave the same public key two certificate chains, an important consideration when the browser vendors began blocking the compromised intermediates.

Image: The eight affected intermediate CAs and their revocation times, reported by eMudhra in Mozilla's incident record on 18 October 2026.

How P-384 keys lost their expected security

P-384 is designed to provide approximately 192 bits of security against conventional attacks. Its protection depends on choosing private keys from a sufficiently large and unpredictable range.

That did not happen here. The affected keys contained critically little entropy. Mozilla described them as effectively consisting of zeroes apart from the final byte, leaving only a tiny fraction of the expected private-key space to examine.

Researchers independently confirmed that the keys could be recovered with negligible computing effort. Mozilla completed its own verification at 21:48 UTC on 17 October, without knowing how the original reporter had recovered them.

Security researcher Scott Helme repeated the analysis and reported recovering the compromised keys almost immediately on his laptop.

The certificates could still pass ordinary structural checks. A public key derived from a weak private key can be mathematically valid and correctly encoded. A certificate parser has no way to establish from the encoding alone that the private key was generated using adequate randomness.

The failure occurred during private-key generation. The resulting certificates could appear technically valid while their signing keys offered almost no practical protection.

The CRLs that stopped verifying seventeen months earlier

On 5 June 2025, Sectigo reported to Mozilla that eMudhra was publishing six certificate revocation lists with invalid ECDSA signatures, one for each TLS intermediate later found to have a weak private key.

A certificate revocation list identifies certificates withdrawn by an issuer. Its digital signature allows software to verify that the information came from the issuing CA and has not been changed. A CRL with an invalid signature cannot be trusted as authenticated revocation information.

eMudhra's 2025 incident report dated the non-compliance from 29 May 2025 at 09:46 UTC to 6 June at 07:30 UTC, when correctly signed CRLs became available.

The company attributed the error to a manual process that referenced the wrong CA key configuration. Its impact assessment said no end-entity certificates had been issued by those intermediates. eMudhra reported completing automated CRL signing and post-signing validation on 20 June 2025.

An archived monitoring record suggests an earlier start. A crt.sh dashboard capture from 3 June 2025 listed all six CRLs with ECDSA verification failures. The dashboard recorded a "listed here since" date of 23 May 2025.

That was six days before the start of the affected period stated in eMudhra's report. The monitoring timestamp establishes when the problem appeared in the dashboard, although it does not settle when the underlying configuration error first occurred.

Mozilla's reviewer also reported that several RSA CRLs from the B and C series were unavailable during the earlier investigation.

The 2025 report documented the signing failure and its repair. It left questions about how the incorrect key had been selected.

Two explanations of the CRL failure

During the October 2026 investigation, Mozilla asked eMudhra to explain how the compromised keys related to the CRL-signing problems reported in 2025.

The company's later response described an approved deviation during the original ceremony to test key wrapping and unwrapping. According to eMudhra, a coding flaw caused a second set of weaker keys to be created within the hardware security module.

Similar labels assigned to the key pairs then contributed to the weaker keys being used to generate certificate signing requests, including those submitted to Sectigo. eMudhra acknowledged human errors at several stages of the process.

Its 2026 account also described problems involving the originally intended keys and subsequent CRL signing. The 2025 report had attributed the invalid signatures to an incorrectly referenced CA key configuration.

Those explanations could describe different stages of the same failure, but the public record does not establish the complete sequence. In particular, the cited material does not independently confirm which key pair signed the invalid CRLs published in 2025.

Mozilla's reviewer questioned whether the appearance of the same intermediate set in both incidents could be coincidental. The overlap deserves examination, although it has not been demonstrated that the weak-key generation directly caused the earlier CRL failure.

The October discussion raised another issue involving emSign Root TLS CA - G1. A researcher reported that CRL number 11, published on 14 October 2026, failed signature verification in repeated checks.

The cited thread contained no conclusive explanation from eMudhra by 21 October. An invalid CRL signature is a serious reliability and compliance concern, but it does not establish that the G1 root private key was compromised or that the RSA hierarchy suffers from the same weak-key problem.

The response timeline: rapid certificate blocking

The researcher and Mozilla recorded slightly different starting points for the response, depending on whether the clock begins with discovery, notification or receipt of the report.

The researcher described finding the weakness on 17 October and distributing notifications by 18:05 UTC. Mozilla received the forwarded report at 20:06, acknowledged it two minutes later, independently confirmed the compromised keys at 21:48 and completed its OneCRL block at 22:59.

Chrome acted earlier. The Chrome Root Program received the report at 18:05 UTC and shipped a CRLSet update at 19:45.

The Chrome block targeted the subject public key information (SPKI) of the six TLS intermediates. Blocking the public keys also covered Sectigo's cross-signed versions, which carried the same keys under a different certificate chain.

Chrome could distribute that protection through its security update mechanism without requiring users to install a new browser version or restart, provided their installations received the updated CRLSet.

Image: Response actions reported by the Chrome Root Program, Mozilla and participants in the public incident discussion through 21 October 2026. Update availability and installation may differ between devices.

eMudhra marked all eight intermediates revoked on 18 October, recording the reason as 'cACompromise'.

Chrome did not include the two code-signing intermediates in its TLS-specific CRLSet block. Their certificate purposes prevent their use in ordinary TLS validation paths. The compromised code-signing keys remained a separate concern for software-signing systems.

Why certificate revocation was not considered sufficient

Certificate authorities publish revocation information so applications can reject certificates that should no longer be trusted. That protection depends on correctly signed revocation data reaching clients that actually enforce it.

The six TLS intermediates had already produced invalid CRLs in May 2025. The October disclosure exposed a separate problem: their private keys were recoverable, allowing unauthorized certificates to be signed under the affected chains.

Even a correctly signed CRL offers little protection to an application that never retrieves or checks it.

Mozilla expressed concern about clients outside the major-browser ecosystem that lack effective revocation mechanisms. Blocking the compromised intermediates through remotely distributed security lists would protect many browsers, but could leave independently managed applications exposed.

Removing the affected root from trust stores addresses more of those clients once they receive the update. Applications using outdated CA bundles or separate trust decisions still need attention from their maintainers.

Why Firefox issued a point release

Firefox ordinarily receives OneCRL blocklist changes through Mozilla's RemoteSettings infrastructure. Installations with RemoteSettings disabled by policy, or unable to reach it, might not receive the new entries through that channel.

Mozilla therefore issued browser updates containing the relevant blocklist data directly.

Mozilla explained the decision in its 20 October security announcement. The Firefox 158.0.1 release notes confirmed that the update covered installations where RemoteSettings was unavailable.

Firefox 158.0.1 and the corresponding ESR releases, including ESR 115.43.1 and ESR 153.5.1, extended protection to those configurations.

The release also fixed unrelated vulnerabilities. Mozilla's advisory documented the certificate-blocking issue alongside problems involving SVG handling, uninitialized memory in the cubeb audio backend and a PDF viewer denial of service.

Root removal and the migration of 30,000 certificates

On 20 October, Mozilla announced the emergency removal of trust in emSign ECC Root CA - G3.

The decision affected more than the eight compromised intermediates. Other certificates chaining to that ECC root could also lose trust, even when their own private keys were not known to be weak.

Mozilla asked eMudhra to stop issuing certificates beneath the affected root. It did not announce removal of the separate emSign Root CA - G1 RSA root.

eMudhra agreed and estimated that approximately 30,000 certificates needed migration. About 97% had been issued through ACME automation, according to the company.

The planned replacement path used intermediate CAs chaining to the remaining G1 RSA root. eMudhra subsequently confirmed that issuance under the ECC root had stopped.

Google announced a corresponding Chrome Root Store change. In its security guidance, Chrome advised operators with certificates beneath the affected root to obtain replacements from a trusted PKI hierarchy.

Website operators could therefore face certificate errors even when their own certificates were never directly compromised. Once a client distrusts the root, any chain depending on that root can fail validation.

AI-assisted recovery exposed the extent of the weakness

Mozilla's incident discussion described tests in which language-model agents recovered compromised private-key material from information already present in public certificates.

Helme reported a similar result using a hosted language model. The tests reinforced the researchers' finding that recovering the keys required little specialist knowledge or computing power.

Mozilla's concern extended beyond the researchers who independently reproduced the results. Anyone who obtained the affected public certificates could potentially have discovered the weakness before the October report.

eMudhra also described using AI-assisted mathematical analysis when examining other issuers. The company has not published a complete independently audited methodology for that assessment in the cited incident material.

The compromised keys have been independently demonstrated. Claims about the security of other intermediates remain dependent on the evidence supporting eMudhra's wider review.

What Certificate Transparency can verify

Certificate Transparency (CT) provides public records of issued TLS certificates. Major browsers require qualifying certificates to be logged, making undisclosed issuance harder to use against clients that enforce those requirements.

Google reported that no TLS certificates chaining to the six compromised intermediates had appeared in CT logs before Chrome applied its block.

Based on that finding, Google concluded that fraudulent certificates issued through those intermediates could not have been used successfully against clients enforcing the relevant CT requirements during that period.

Many applications outside major browsers do not enforce CT. A certificate absent from public logs may still be accepted by a client that trusts the issuing chain and does not require logging.

The researcher's disclosure website identifies that limitation. Mozilla likewise cautioned that the absence of logged certificates cannot rule out attacks against clients that do not enforce CT.

Cloudflare's investigation found no active exploitation affecting its services. That finding covers Cloudflare's own visibility and does not establish whether exploitation occurred elsewhere.

The two code-signing intermediates present a different problem. Authenticode certificates are not subject to the same public CT logging requirements as browser-trusted TLS certificates.

eMudhra stated that the affected code-signing intermediates had issued no end-entity certificates. The cited public record contains no independent audit confirming that assertion.

The compromised private keys could be derived from information available since the certificates were created. Investigators therefore cannot establish from the October disclosure alone when the keys were first recovered or by whom.

No confirmed malicious exploitation had been established in the cited evidence by 21 October 2026.

What administrators should check on their systems

Browser updates do not necessarily protect command-line tools, backend services or applications with their own certificate stores.

Some TLS clients use separate CA bundles and perform limited revocation checking. Others do not enforce Certificate Transparency. An application can consequently continue trusting an affected certificate chain after a browser on the same machine has begun rejecting it.

Administrators should identify applications using eMudhra certificate chains, check the trust stores those applications read, and review updates from the maintainers of their TLS libraries.

The researcher's disclosure website provides test endpoints for checking whether clients still accept the affected chains. Administrators can use those documented tests in authorized environments while leaving normal TLS verification enabled.

Remediation had been reported by 21 October for several trust stores and libraries, including NSS, Go's certificate verification code, Python certifi, rustls-related components, CFSSL and Linux CA certificate packages.

Availability varies by platform and distribution. An upstream fix may not have reached every packaged application, container image or system installation.

Two implementation details deserve particular attention.

- macOS command-line tools: Some curl installations use certificate bundles or validation paths separate from Safari and other Apple applications. Administrators should check the TLS backend, certificate-store source and revocation behavior of the curl build actually installed.

- Go applications: The Go crypto/x509 mitigation changes certificate-chain verification behavior. Applications compiled using affected Go toolchains may require rebuilding with a patched release, depending on how they perform certificate verification.

Helme's tests showed that some unpatched TLS clients accepted certificate chains signed using the compromised intermediate keys. The results depended on each client's trust store and revocation configuration.

Cloudflare warned administrators against disabling certificate verification to work around connection failures. Systems that still trust the affected chains need updated trust components or other vendor-supported remediation.

India's DSC infrastructure: what subscribers should ask

eMudhra operates international certificate authority services under the emSign brand and separate digital signature services in India.

iTnews described eMudhra as India's largest licensed certification authority. Its Indian certificate services include digital signature certificates used for electronic filing, business transactions and government-related processes.

The documented October 2026 compromise concerns specific TLS and code-signing intermediates under the emSign G3 hierarchy. The cited evidence does not identify the separate Indian DSC issuance infrastructure as compromised.

In its public incident statement, eMudhra said the problem was limited to the identified intermediate certificates and did not affect its root CA private keys or other intermediate CAs.

The company also said the removal of roots C1 and C3 had been planned separately and was unrelated to the October incident.

Those statements are relevant when assessing the scope of the incident, although root-program investigations into the company's remaining certificates were still underway.

Website operators should identify whether their certificates chain to the affected ECC root, one of the compromised intermediates, a Sectigo cross-sign or the separate G1 RSA hierarchy.

Indian DSC holders should check the issuing CA and certificate chain used by their signing applications, along with any notices from eMudhra or the applicable regulatory authority.

The reported emSign G3 compromise provides no evidence that all Indian DSC holders need to revoke or replace their certificates.

The HSM explanation and unresolved questions

eMudhra stated that the affected keys were generated using Utimaco hardware security modules described as FIPS 140-2 Level 3 compliant. The company said the hardware itself did not appear to be responsible.

That explanation prompted questions on Mozilla's public security policy discussion list.

Jeffrey Walton questioned how a certified cryptographic module could be involved in repeated weak-key generation. Peter Gutmann argued that FIPS certification does not guarantee security in every deployment, citing the limitations of certification and examples of certified products with security weaknesses.

A certified hardware module can still be used incorrectly. Its certification does not establish the security of surrounding software, key-selection logic or operational procedures.

Several statements about the incident's scope remain unverified by an independent public audit. These include eMudhra's claims that the code-signing intermediates were never used for production issuance, that the affected intermediates issued no end-entity certificates, and that other issuers were free of similar weaknesses.

The lack of a published audit does not make those claims false. It leaves their verification dependent on the company's investigation and any further evidence provided to root-program operators.

Chrome said it was continuing to examine eMudhra's remaining CA certificates for inclusion in the Chrome Root Store.

Mozilla's investigation also sought more detail about the original key-generation ceremony, the relationship between keys and hardware devices, other intermediates created during the ceremony, and the 2025 CRL-signature mismatch.

The separate report of an invalid G1 CRL on 14 October still needs a technical explanation. Nothing in the evidence cited here establishes that the G1 root private key suffers from the same weakness as the compromised P-384 intermediates.

Next steps after 27 October

Google expected Chrome Root Store version 43, removing trust in emSign ECC Root CA - G3, to reach devices using that root store on the morning of Tuesday, 27 October 2026, Pacific time. Canary and Beta releases were expected to receive the change earlier.

Operators whose TLS certificates depend on the affected root should replace them with certificates from a hierarchy trusted by their clients. Deployment schedules can differ between browsers, operating systems and other TLS applications.

Following the October disclosure, eMudhra committed to reviewing its previous key-generation ceremonies. The company also began planning the migration of approximately 30,000 certificates to the G1 RSA hierarchy, with most of the work expected to use ACME automation.

Google reported working with Certificate Transparency log operators to prevent new certificates chaining through the compromised intermediates from entering accepted CT logs.

The continuing investigations may establish whether the 2025 CRL failures and the weak-key generation shared an operational cause. They may also clarify whether other intermediates produced during the April 2025 ceremony require further action.

The invalid CRLs were visible to external monitoring in May 2025 and were addressed as a signing-configuration problem. In October 2026, researchers discovered that private keys belonging to some of those same intermediates had been weak since their creation.

What remains unexplained is whether the original investigation examined the key-generation and signing procedures closely enough to detect the weakness, and why subsequent checks did not uncover it before the October disclosure.

References

1. bugzilla.mozilla.org - show_bug.cgi - https://bugzilla.mozilla.org/show_bug.cgi?id=2083449
2. scotthelme.co.uk - eight-bits-of-trust-emudhras-ca-keys-cracked-in-milliseconds - <https://scotthelme.co.uk/eight-bits-of-trust-emudhras-ca-keys-cracked-in-milliseconds/>
3. bugzilla.mozilla.org - show_bug.cgi - https://bugzilla.mozilla.org/show_bug.cgi?id=1970728
4. web.archive.org - web / 20250603145523id_ - https://web.archive.org/web/20250603145523id_/https://crt.sh/mozilla-disclosures
5. blog.google - security / protecting-chrome-and-android-users-from-compromised-emudhra-intermediate-ca-certificates - <https://blog.google/security/protecting-chrome-and-android-users-from-compromised-emudhra-intermediate-ca-certificates/>
6. blog.mozilla.org - security / 2026 - <https://blog.mozilla.org/security/2026/10/20/protecting-firefox-users-following-the-compromise-of-emudhra-intermediate-certificates/>
7. firefox.com - en-US / firefox - <https://www.firefox.com/en-US/firefox/158.0.1/releasenotes/>
8. mozilla.org - en-US / security - <https://www.mozilla.org/en-US/security/advisories/mfsa2026-111/>
9. emudhra.we-have-a-problem.com - <https://emudhra.we-have-a-problem.com/>

10. blog.cloudflare.com - emudhra-certificate-compromise - <https://blog.cloudflare.com/emudhra-certificate-compromise>
11. go.dev - issue / 82345 - <https://go.dev/issue/82345>
12. itnews.com.au - news / cloudflare-warns-fixing-botched-emudhra-certs-goes-beyond-browsers-629804 - <https://www.itnews.com.au/news/cloudflare-warns-fixing-botched-emudhra-certs-goes-beyond-browsers-629804>
13. emudhra.com - en-in / faq - <https://emudhra.com/en-in/faq/certifying-authority-digital-signature-esign>
14. emudhra.com - en / post - <https://emudhra.com/en/post/certificate-security-incident>
15. groups.google.com - a / mozilla.org - <https://groups.google.com/a/mozilla.org/g/dev-security-policy/c/0R3XLJL6FuE/m/VAgSvefvAAAJ>