

CYBER THREAT ADVISORY

COMPROMISED eMUDHRA CA KEYS: RISK OF FORGED CERTIFICATES

Implications and Recommended Actions for the Indian Industry



OCTOBER 2026

Overview

The private keys of eight eMudhra intermediate CA certificates (six TLS, two code-signing) can be recovered by anyone in under a second. Each key also appears in a certificate cross-signed by Sectigo, so 16 certificates are affected. Anyone holding one of the TLS keys could issue a certificate for any website that clients trusting the chain would accept; the code-signing keys could forge code signatures on Microsoft platforms.

A researcher, Wayne, reported the issue privately on 17 October 2026 to Apple, Chrome, Microsoft, Mozilla and eMudhra, proving key control by signing a test message with each key. Mozilla, which received the report via Chrome, recovered the keys independently without knowing the reporter's method. Others have since recovered and published them, so the keys should be treated as public. Firefox and Chrome blocked the certificates within hours and Apple and Microsoft within two to three days, so residual risk for up-to-date browsers is low. It is not low for command-line tools, language runtimes, containers and embedded devices that do not check revocation and stay exposed until their CA bundle is updated.

Root Cause

eMudhra's own team generated the keys on Utimaco hardware security modules (HSMs) in a key-generation ceremony on 30 April 2025. The ceremony produced 24 intermediate keys in three series (A, B, C), two types (RSA-4096, P-384) and four purposes (DV, OV, EV, code signing). The affected keys are the P-384 keys of the B and C series.

A P-384 private key should be a random integer up to roughly 2^{384} (about 192 bits of strength). All eight affected keys are below 256, leaving at most about 8 bits. Computing the public keys for values 1 to 255 (or using the baby-step giant-step algorithm) and matching them to the certificate recovers a key in under a second on an ordinary laptop, with no special skill. The flaw is in key generation, not in the P-384 curve or signature algorithm, and eMudhra's account does not point to the HSM hardware.

eMudhra's preliminary root-cause findings:

- The keys were meant for export to eMudhra's US and European data centers for data-sovereignty reasons, so the ceremony script included an approved deviation to test key wrapping and unwrapping (export).
- A coding flaw created a second, weaker key pair in the HSM. Because the key labels were similar, that pair was used by mistake for the certificate signing requests, which eMudhra signed and also sent to Sectigo for cross-signing.
- Human errors in several steps and gaps in key-lifecycle controls contributed. The exact coding fault has not been published.

eMudhra says its RSA intermediates and its 'A' series (Entrust HSMs, India-only; replaced by an 'A-1' series in May 2025) are unaffected. Mozilla could not recover the 'A' or 'A-1' P-384 keys and had not yet examined the RSA keys. There was an earlier warning sign: in May-June 2025 the CRLs of the same six TLS intermediates failed signature verification (Mozilla Bug 1970728). eMudhra says

they were first signed with the original keys, which produced invalid signatures, and were then re-signed with the flawed keys. All findings are preliminary.

Affected Certificates

- **eMudhra copies (8):** eMudhra signed the eight keys under its G3 hierarchy. The six TLS intermediates chain to emSign ECC Root CA - G3, which Mozilla removed from NSS on 20 October 2026, and Chrome is removing in Chrome Root Store v43.
- **Sectigo cross-signs (8):** the same public keys, with a different issuer and serial numbers, cross-signed under Sectigo's legacy "AAA Certificate Services" (Comodo AAA) root. Sectigo only cross-signed them; it did not generate the keys.

Because the weakness is in the key, every certificate carrying it is compromised and each copy must be revoked separately. Chrome and Mozilla had already removed the Sectigo root (and Chrome's block matches the public key, covering both copies). Apple and Microsoft still trust it, so on their platforms the cross-signs were a second trust path until they blocked the certificates on 19 and 20 October respectively.

Incident Timeline

- **17 Oct:** Report emailed to Apple, Chrome, Microsoft, Mozilla and eMudhra at 18:05 UTC (23:35 IST). Chrome's CRLSet blocks the six TLS intermediates by 19:45 UTC. Mozilla receives the report at 20:06 UTC (01:36 IST, 18 Oct), confirms the proof and recovers the keys itself at 21:48 UTC. Firefox's OneCRL block is complete by 22:59 UTC.
- **18 Oct:** eMudhra revokes all eight intermediates between 09:24 and 09:38 UTC (reason: CA compromise). Mozilla opens the public incident report at 12:06 UTC.
- **19 Oct:** Apple blocks the certificates. Sectigo revokes the cross-signed copies. Downstream trust-store operators are notified.
- **20 Oct:** Microsoft blocks the certificates. Mozilla discloses publicly and ships emergency Firefox and NSS 3.131.1 releases. With the keys independently recovered and published, Mozilla removes trust in emSign ECC Root CA - G3 and Chrome commits to the same. eMudhra stops issuing from every intermediate under that root, with about 30,000 certificates to migrate.

Impact and Risk Assessment

- **Browsers - low residual risk:** up-to-date Chrome, Firefox, Edge and Safari have the revocation and enforce Certificate Transparency (CT). No certificates from the affected intermediates appear in CT logs, so there is no sign of attack on CT-enforcing clients.
- **Non-browser clients - higher risk:** many clients enforce neither CT nor revocation. OneCRL, CRLite and CRLSets are browser-specific, so most other clients depend on their CA bundle alone. The keys have been recoverable since April 2025, and it is not known whether anyone abused them against such clients.

- **Potential impact:** man-in-the-middle interception or impersonation of any domain; forged code signatures on platforms that trust the code-signing intermediates (Microsoft).
- **eMudhra's account:** only one intermediate (EM DV TLS CA - G3C) went into production, the code-signing intermediates never did, and no end-entity certificates were issued from any of them.

Who Is Affected

The following should check their systems:

- **Servers, containers and cloud workloads:** Linux hosts, container base images and VM templates with an older ca-certificates package, most of which do not check revocation.
- **Software developers and vendors:** applications that ship their own CA bundle or verifier (Python certifi, Go, Node.js, Rust webpki-roots, Composer, Electron), until the bundle or runtime is updated and the software is rebuilt and redeployed.
- **API gateways, proxies and integrations:** partner APIs, payment and banking integrations, webhooks and SaaS connectors usually use non-browser TLS libraries; an attacker who can intercept this traffic could impersonate the remote service.
- **IoT, OT and embedded devices:** devices with fixed or rarely updated trust stores, until the manufacturer ships an update.
- **Windows code-signing environments:** systems without Microsoft's 20 October protection could accept software signed with the compromised code-signing keys.
- **Apple and Microsoft devices are not yet updated:** these platforms also trust Sectigo's AAA root, so devices without the 19-20 October protections can trust both the eMudhra and the Sectigo chains.
- **Firefox with Remote Settings disabled:** installations where Remote Settings is disabled by policy or blocked need Firefox 158.0.1, ESR 115.43.1 or ESR 153.5.1.
- **eMudhra (emSign) certificate customers:** about 30,000 certificates chain to emSign ECC Root CA - G3. They are not compromised, but will stop working on updated clients (NSS, certifi, webpki-roots and, after Root Store v43, Chrome) and must be reissued on another chain.

Immediate Actions

All users:

- Update browsers and operating systems to the latest versions and restart the browser.

IT and security teams:

- **Patch and rebuild:** Roll out OS, browser and CA-bundle updates across servers, containers and base images, and rebuild container images that bake in CA bundles.
- **Test non-browser clients:** Test clients that don't use browser trust stores. Both commands below should fail with a certificate error; if succeeds, that client still trusts a compromised chain. The first tests eMudhra's chain and the second the Sectigo cross-signed chain, and a client can pass one and fail the other.

```
curl https://dv-g3b-emudhra.we-have-a-problem.com/
```

```
curl https://dv-g3b-aaa-emudhra.we-have-a-problem.com/
```

- **Slow-to-update fleets:** Prioritize proxy, API-gateway and IoT/embedded fleets that cannot update quickly.

Developers and certificate owners:

- **Update CA bundles and runtimes:** Python certifi 2026.10.19, Go 1.27.3 / 1.26.10 (programs must be rebuilt), curl CA bundle 2026-10-20, NSS 3.131.1, Alpine ca-certificates 20261020-r0 and Rust webpki-roots 1.0.10. Ubuntu, Debian, Fedora, Arch and Gentoo have updated ca-certificates packages. Node.js, Bun and FreeBSD fixes were pending at the time of writing; the full list is on the disclosure site.
- **Migrate emSign certificates:** about 97% of the ~30,000 affected certificates were issued via ACME and must move from the ECC G3 chain to the RSA G1 chain. Make sure renewal onto the new chain works, monitor for failures and avoid pinning to intermediates. Chrome is also reviewing eMudhra's G1 root, so watch for further changes.
- **Manual distrust (experts only):** where no patched bundle exists, the disclosure site documents how to distrust the affected roots on macOS curl, Linux, NetBSD and FreeBSD. On most platforms this also distrusts Sectigo's widely used AAA root and can break legitimate connections, so it is not recommended for ordinary users.

Preventive Recommendations

Organisations relying on certificates:

- **Patch trust stores automatically:** include CA bundles, language runtimes and container base images in routine patching, and rebuild images on a schedule, not only when code changes.
- **Know where your trust stores are** keeping an inventory of applications, appliances and devices that carry their own CA bundle, so a distrust event can be applied everywhere within days.

- **Honour revocation and CT:** prefer the operating-system verifier or libraries that consume revocation data and enforce CT for high-value connections.
- **Build crypto-agility:** automate issuance and renewal (for example with ACME), keep an approved second CA ready, and avoid pinning to a single intermediate or CA.
- **Monitor CT logs:** use a CT monitoring service to be alerted whenever a certificate is issued for your domains.
- **Track trust-store announcements:** follow root-program notices (Mozilla dev-security-policy, Chrome Root Program, CCADB) and CERT-In advisories.

Certificate authorities and organisations running their own PKI:

- **Test every key before certifying it:** run automated weak-key checks on each CSR, including a test for small or low-entropy private values. Comparing a public key against the first few thousand multiples of the curve generator take milliseconds and would have caught all eight keys.
- **Bind the CSR to the intended HSM key:** record each key's public-key fingerprint at generation, confirm under maker-checker control that the CSR and certificate match it, and use unique, unambiguous key labels.
- **Treat ceremony scripts as code:** peer-review, version and test every ceremony script, including approved deviations, on a non-production HSM, with independent witness or auditor sign-off.
- **Remove test and duplicate keys:** delete keys created for testing (such as wrap/unwrap tests) before the ceremony closes and reconcile the HSM key inventory against the ceremony record.
- **Treat anomalies as incidents:** root-cause CRL or OCSP signature-verification failures instead of re-signing; the invalid CRL signatures of May-June 2025 were an early warning.
- **Re-verify past ceremonies and cross-signs:** periodically re-validate all issued CA keys, as eMudhra has committed to do for every past ceremony. CAs that cross-sign should independently check subordinate keys first.

References

- Dennis Jackson (Mozilla), eMudhra Intermediate CA Compromise disclosure site (20 Oct 2026) <https://emudhra.we-have-a-problem.com/>
- Mozilla Bugzilla 2083449, Private key compromise of eMudhra Intermediate Certificates (accessed 21 Oct 2026) https://bugzilla.mozilla.org/show_bug.cgi?id=2083449
- Mozilla Bugzilla 1970728, eMudhra: Invalid CRL signatures https://bugzilla.mozilla.org/show_bug.cgi?id=1970728
- Mozilla Security Blog, Protecting Firefox users following the compromise of eMudhra intermediate certificates (20 Oct 2026) <https://blog.mozilla.org/security/2026/10/20/protecting-firefox-users-following-the-compromise-of-emudhra-intermediate-certificates/>
- eMudhra, Statement on Recent Certificate Security Incident <https://emudhra.com/en/post/certificate-security-incident>

Disclaimer

This document is prepared for informational purposes only and is intended to assist organisations and security professionals in understanding the eMudhra intermediate Certificate Authority (CA) security incident, its potential implications, and recommended mitigation measures. The information presented is based on publicly available sources as of 22 October 2026 and is subject to change as further information becomes available.

The analysis and recommendations provided herein are for general awareness and should not be construed as a definitive assessment of any organization's security posture. Readers are advised to verify relevant information with official sources and conduct appropriate assessments of their own systems and certificate infrastructure.

No warranties are made regarding the accuracy, completeness, or adequacy of the information contained herein. This document should not be relied upon as a substitute for professional cybersecurity advice or independent technical assessment.

ABOUT DSCI

Data Security Council of India (DSCI), set up by Nasscom, is a not-for-profit industry body and think tank on data protection, cyber security, and critical technologies in India. DSCI is committed to making cyberspace safe, secure, and privacy-assured, and to advancing critical technologies vital for national security through policy advocacy, thought leadership, capacity building, and outreach initiatives. DSCI engages with the government, regulators, defence, law enforcement agencies, embassies, industry members, and academia to position India as a leader in the global digital ecosystem.

Data Security Council of India

4th Floor, Nasscom Campus, Plot No. 7-10, Sector 126, Noida, UP-201303



[data-security-council-of-india/](#)



[DSCI_Connect](#)



[dsci.connect](#)



[dsci.connect](#)



[dscivideo](#)



[security chips](#)