Zero-Downtime SAML Certificate Rotation

A runbook for rotating your identity provider's SAML signing certificate without locking users out.

Gremlin API · Company Security Administration · Applies to SSO/SAML-enabled organizations

---

> **What this covers.** Your SAML identity provider (IdP) periodically rotates the X.509 certificate it uses to sign SAML assertions. Historically, swapping that certificate in Gremlin was a hard cutover: for the moment between "IdP switched" and "Gremlin updated," logins failed. Gremlin now supports a **secondary certificate** so both the old and new certificates are accepted at the same time, letting you rotate with zero downtime.

Contents

1. When to use this · 2. How dual-certificate validation works · 3. Before you start · 4. The rotation runbook (step by step) · 5. API reference · 6. Rolling back · 7. Troubleshooting · 8. Opening a support case

1. When to use this

Run this procedure whenever your IdP's SAML signing certificate is going to change, including:

- A scheduled certificate expiry / renewal in Okta, Azure AD (Entra ID), OneLogin, Ping, ADFS, etc.
- An emergency certificate reissue (for example after a suspected key compromise).
- Migrating to a new IdP tenant or application that signs with a different key.

If you rotate the certificate *without* this procedure (by overwriting the primary certificate in one shot), any assertion signed with the other certificate is rejected until both sides match, which can lock out your users. The dual-certificate flow removes that gap.

2. How dual-certificate validation works

Your Gremlin company can hold two IdP signing certificates at once:

Slot	Role
Primary certificate	The certificate configured on your SAML settings. Tried first on every login.
Secondary certificate	An optional fallback. Used only when the primary does not validate the incoming assertion.

On each SAML login, Gremlin validates the signed assertion in this order:

1. **Primary first.** If the assertion's signature validates against the primary certificate (and the certificate itself is currently valid), the login succeeds.
2. **Fall back to secondary.** If the primary certificate is expired, not yet valid, or simply does not match the assertion's signature, Gremlin then tries the secondary certificate. If that validates, the login succeeds.
3. **Both fail.** Only if *neither* certificate validates the assertion is the login rejected (HTTP 401).

> **The key property:** while both certificates are configured, it does not matter which one your IdP signs with. Old key or new key, the login works. That overlap window is what makes the rotation seamless.

3. Before you start

Requirement	Detail
Permission	A Gremlin user with **Company Security Write** access (Company/Org Security Administrator). You can only modify your own company.
API credentials	A team API key. Sent as `Authorization: Key <YOUR_API_KEY>`.
Your Company ID	The `{identifier}` in the URLs below. It must match your own company, or the API returns `403 Forbidden`.
The new certificate	The new IdP signing certificate as an X.509 / PEM certificate. The `BEGIN CERTIFICATE` / `END CERTIFICATE` headers and line breaks are optional, Gremlin normalizes them. This is the same format you already use for the primary certificate.

Throughout this runbook, replace `{identifier}` with your Company ID and `$GREMLIN_API_KEY` with your API key. The API base URL is `https://api.gremlin.com/v1`.

4. The rotation runbook

Perform these steps in order. Your users stay logged in and able to authenticate the entire time.

STEP 1

Add the new certificate as the secondary

Do this **before** your IdP starts signing with the new key. This opens the overlap window.

```
curl -X PUT \
  https://api.gremlin.com/v1/companies/{identifier}/saml/cert-secondary \
```

```
  -H "Authorization: Key $GREMLIN_API_KEY" \
  --data-urlencode "certificate=$(cat new-idp-cert.pem)"
```

A `200 OK` means the secondary is stored. Gremlin now accepts assertions signed by *either* the current (primary) or the new (secondary) certificate.

STEP 2

Switch your IdP to the new certificate

In your identity provider, activate the new signing certificate for the Gremlin application. From this point your IdP signs assertions with the new key.

Because the new key is already configured as your secondary, logins keep working: the primary (old) no longer matches, so Gremlin automatically falls back to the secondary (new) and succeeds.

> **Verify before continuing.** Have a test user log in (or use an incognito window). Confirm SSO succeeds. Do not proceed to Step 3 until you have confirmed logins work with the new certificate.

STEP 3

Promote the new certificate to primary

Once the IdP is fully cut over and logins are confirmed, remove the old primary. Gremlin automatically promotes the secondary into the primary slot:

```
curl -X DELETE \
  https://api.gremlin.com/v1/companies/{identifier}/saml/cert \
  -H "Authorization: Key $GREMLIN_API_KEY"
```

After this call the new certificate is your **primary**, and there is no secondary. You are back to a clean single-certificate configuration. Rotation complete.

> **Guardrail:** if SAML is enabled and there is no secondary certificate configured, deleting the primary returns `400 Bad Request` rather than leaving you with no usable certificate. That is expected, it is the system protecting you from locking yourself out.

> **Alternative for Step 3.** If you prefer to set the primary explicitly instead of promoting, re-submit your full SAML settings with the new certificate as the primary (see `POST .../saml/props` in the reference), then call `DELETE .../saml/cert-secondary` to clear the fallback. The promote-via-delete flow above is simpler and is the recommended path.

5. API reference

All endpoints require the `COMPANY_SECURITY_WRITE` privilege and the `Authorization: Key` header. All are scoped to your own company.

`PUT` /companies/{identifier}/saml/cert-secondary

Sets (or replaces) the secondary IdP certificate.

Body (form-encoded)	`certificate` — the X.509 / PEM certificate. Required.
Returns	`200 OK` on success. `400` if the certificate cannot be parsed. `403` if the identifier is not your company.

`DELETE` /companies/{identifier}/saml/cert-secondary

Removes the secondary certificate. Use this to abandon a rotation or clean up. Always safe, it never affects the primary.

Returns	`200 OK`. `403` if the identifier is not your company.

`DELETE` /companies/{identifier}/saml/cert

Removes the primary certificate. If a secondary is configured, it is promoted to primary automatically.

Returns	`200 OK` on success. `400 Bad Request` if SAML is enabled and no secondary is configured (would leave you with no certificate). `403` if the identifier is not your company.

`POST` /companies/{identifier}/saml/props

Existing endpoint that sets the full SAML configuration, including the **primary** certificate. Referenced here for the alternative Step 3 flow.

Body (form-encoded)	`enabled`, `entityId`, `idpUrl`, `certificate` (primary), `forced`, `claimsForced`.

6. Rolling back

If something looks wrong during the overlap window (Step 1 or 2) and you have *not* yet promoted:

- **Revert the IdP** to sign with the original certificate. Since the original is still your primary, logins continue to work.
- **Remove the secondary** if you want to abandon the rotation: `DELETE /companies/{identifier}/saml/cert-secondary`.

Because the original certificate remains the primary until you explicitly delete it in Step 3, rollback before promotion carries no risk of lockout.

7. Troubleshooting

Symptom	Likely cause & action
Login fails right after switching the IdP (Step 2)	The new certificate was not stored as the secondary, or a different certificate was uploaded than the one the IdP now signs with. Re-run Step 1 with the exact certificate from your IdP's SAML metadata.
`400` "Cannot remove the primary IDP certificate…"	You tried to delete the primary while SAML is enabled and no secondary exists. Add a secondary first (Step 1), or this is expected if you were not mid-rotation.
`400` "Unable to parse certificate data"	The `certificate` value is not a valid X.509 certificate. Confirm you copied the certificate (not the private key or metadata XML). Headers and newlines are optional.
`403 Forbidden`	The `{identifier}` is not your own Company ID, or your user lacks Company Security Write. Use your own Company ID and an admin API key.
"Certificate is expired" / "not yet valid"	The primary certificate is outside its validity window and no valid secondary is configured. Add the current certificate as the secondary (Step 1), then promote (Step 3).
Login fails with both certificates set	Neither certificate matches the assertion signature. The IdP is signing with a third, different key. Upload that key.

8. Opening a support case

If logins are still failing after following this runbook, contact Gremlin Support. To get to a resolution fast, include:

- Your **Company ID** (`{identifier}`) and IdP vendor (Okta, Entra ID, etc.).
- Which **step** you were on when the problem appeared.
- The **exact HTTP status and error message** returned by the API calls above.

- The approximate **timestamp (with time zone)** of a failed login attempt.
- The **certificate serial number / thumbprint** of both the old and new certificates (never send private keys).
- Whether both primary and secondary were configured at the time of failure.

> **Never share private keys, API keys, or full session tokens** in a support case. Support only needs public certificates and the metadata above.
