JUMBO BLOCK CERT

Your own certificate authority and TLS certificates



Manual · Version 1.0.0

October 2026

ZANNI GROUP · Andreas Zanni e.K. · Brauhof 12 · 44866 Bochum · jumboblock.org

Contents

1. Overview
2. Basic terms
3. Workflow
4. Creating a certificate authority
5. CA file, unlocking and backup
6. Setting up the CA certificate on the devices
7. Issuing a certificate
8. Files and formats
9. FRITZ!Box
10. Signing a CSR
11. Register and revocation list
12. Checking certificates
13. Security and technology
14. Troubleshooting

1. Overview

CERT is the certificate tool of the JUMBO BLOCK Engineering Suite. It creates your own certificate authority (CA) and uses it to issue TLS certificates for routers, FRITZ!Box, NAS, servers, cameras and web applications. Devices on which the CA certificate has been set up as trusted once accept all certificates of this CA without warnings.

Many routers and computers work with self-generated or expired certificates. Browsers then show warnings and the authenticity of the remote station can no longer be established reliably. With CERT, valid certificates are ready within minutes – even in a crisis, without internet and without external services.

- **Fully offline:** CERT runs in the browser, also directly from a USB stick (`file://`). There is no server connection, no account and no cloud.
- **No third-party code:** certificates, requests, revocation lists and PKCS#12 are original program code. Keys, signatures and encryption use the browser's built-in Web Crypto API.
- **One file per CA:** certificate, encrypted key, issuance register and revocation list state are kept in one CA file (`.dbm`).
- **Checking included:** existing certificates can be examined for validity, algorithms and the requirements of current browsers.



Fig. 1: The six areas of CERT

Area	Purpose
Certificate authority	Create a CA, open, save, unlock and back up the CA file, download the CA certificate for the devices
Issue certificate	Generate a key pair and certificate for a device and download it in all required formats
Sign CSR	Sign certificate signing requests from devices that generate their own key
Register & revocation list	Overview of all issued certificates, revocation and creation of the revocation list (CRL)
Check certificate	Examine certificates, chains, requests, revocation lists and keys
Set up	Instructions for importing the CA certificate on all common systems

2. Basic terms

Term	Meaning
Certificate authority (CA)	Entity that issues certificates and signs them with its private key. Its own certificate (root certificate) is set up as trusted on the devices.
Certificate	Electronic ID of a device: name, addresses, public key, validity and the CA's signature. A certificate is public and may be shared.
Private key	Secret counterpart of the certificate. Only the device may know it. Whoever holds it can impersonate the device.
SAN (Subject Alternative Name)	List of host names and IP addresses the certificate is valid for. Browsers check this list only.
Common name (CN)	Display name in the certificate. Browsers no longer use it for address checks.
CSR	Certificate signing request. A device generates its key itself and only sends the public part with its name for signing.
Revocation list (CRL)	List of prematurely revoked certificates, signed by the CA.
PEM / DER	File formats for certificates and keys: PEM is text with `-----BEGIN …-----`, DER is the same information in binary form.
PKCS#12 (.p12 / .pfx)	Password-protected container with private key, certificate and certificate chain – common for Windows, IIS, NAS and many routers.

The two-tier principle

CERT works with your own root certification authority. Its certificate is set up **once** on every computer, phone and browser. After that, all device certificates of this CA are accepted automatically. New or replaced device certificates require no change on the end devices.

> Individual self-signed certificates per device would have to be set up on every end device individually and imported again after each replacement. Your own CA avoids this effort.

3. Workflow

1. **Create the certificate authority** – define name, key, validity, optional name constraints and password (chapter 4).
2. **Save and back up the CA file** – on at least two separate media, with the password kept separately (chapter 5).
3. **Download the CA certificate** and set it up on all end devices as a trusted root certification authority (chapter 6).
4. **Issue device certificates** – for devices that can import certificate and key (chapters 7 to 9), or **sign a CSR** for devices that generate their own key (chapter 10).
5. **Install the files on the devices** and check the connection in the browser.
6. **Save the CA file** – the register then contains all issued certificates.
7. **Monitor validity in the register**, renew certificates in time and revoke them if necessary (chapter 11).

Private keys of issued device certificates are not stored. The required files must be downloaded right after issuing.

4. Creating a certificate authority

A new CA is created in the *Certificate authority* area. The CA name and the password are mandatory.

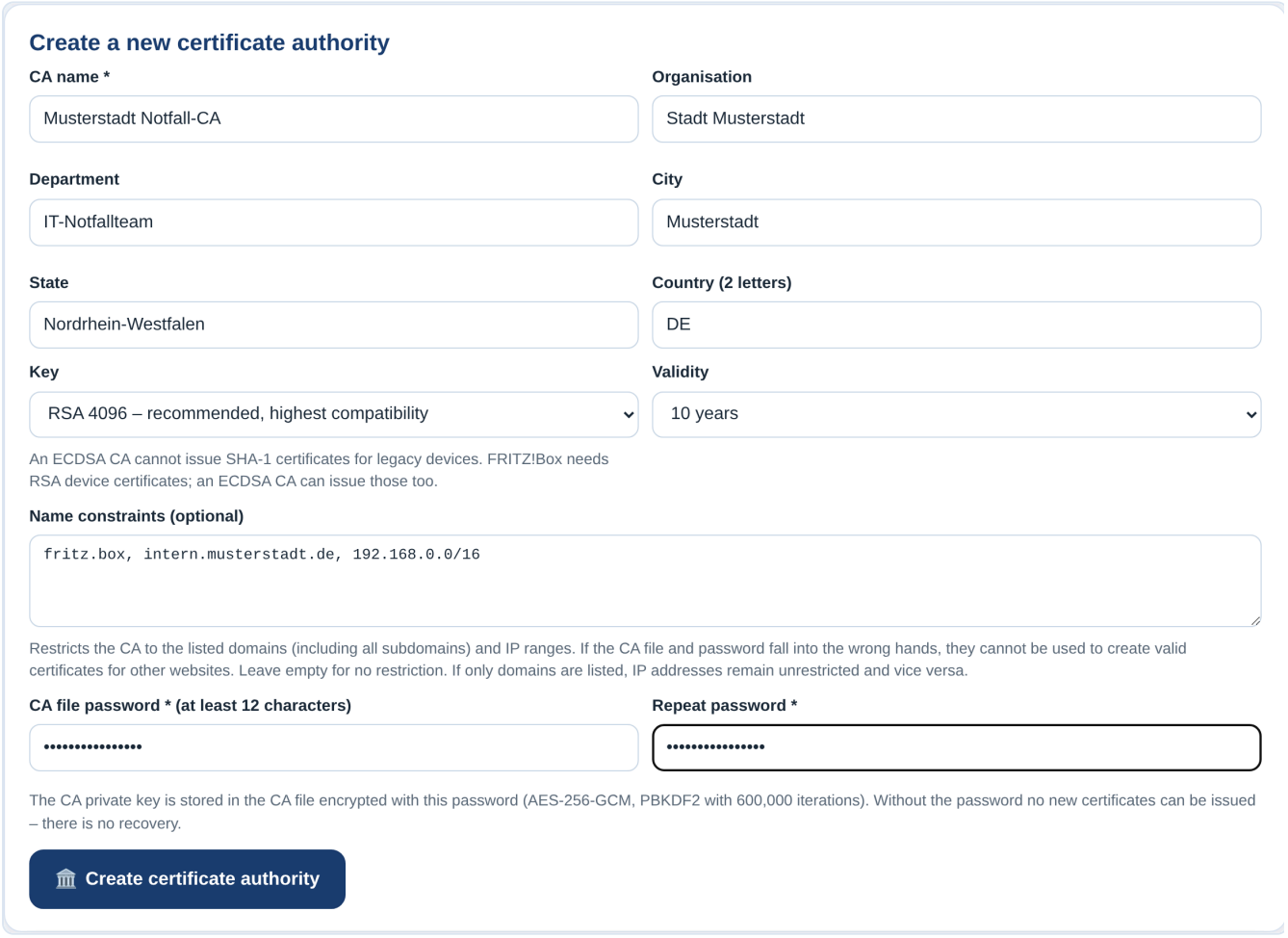


Fig. 2: Creating a new certificate authority

Field	Meaning
CA name *	Appears on all devices as the name of the root certification authority, e.g. “Sampletown Emergency CA”.
Organisation, department, city, state	Optional information on the origin of the CA.
Country	Two-letter country code, default DE.
Key	RSA 4096 (recommended, highest compatibility), RSA 3072, RSA 2048, ECDSA P-384 or ECDSA P-256.
Validity	5, 10, 15 or 20 years, default 10 years. Device certificates end no later than the CA.
Name constraints	Optional: domains and IP ranges for which the CA may exclusively issue certificates.
Password *	At least 12 characters. Encrypts the CA's private key in the CA file.

Choice of key

RSA 4096 is understood by practically all systems, including older devices. ECDSA keys are shorter and faster but are not supported by very old devices. An ECDSA CA cannot issue SHA-1 certificates for legacy devices. An ECDSA CA can still issue RSA device certificates for FRITZ!Box.

Name constraints

Name constraints restrict the CA to the listed domains (including all subdomains) and IP ranges, e.g. `fritz.box, internal.example.org, 192.168.0.0/16`. If the CA file and password fall into the wrong hands, they cannot be used to create certificates that are valid for other websites.

- Domains include all subdomains: `internal.example.org` also covers `nas.internal.example.org`.
- IP ranges are entered in CIDR notation (`192.168.0.0/16`, `fd00::/8`). A single address without a range only covers itself.
- If only domains are listed, IP addresses remain unrestricted – and vice versa.

> Name constraints are a fixed part of the CA certificate and cannot be changed later. Certificates for names outside the constraints are rejected by browsers.

After *Create certificate authority*, CERT generates the key pair (a few seconds for RSA 4096) and encrypts the key. The CA file should then be saved immediately.

5. CA file, unlocking and backup

After creating or opening a CA, the *Certificate authority* area shows its details.

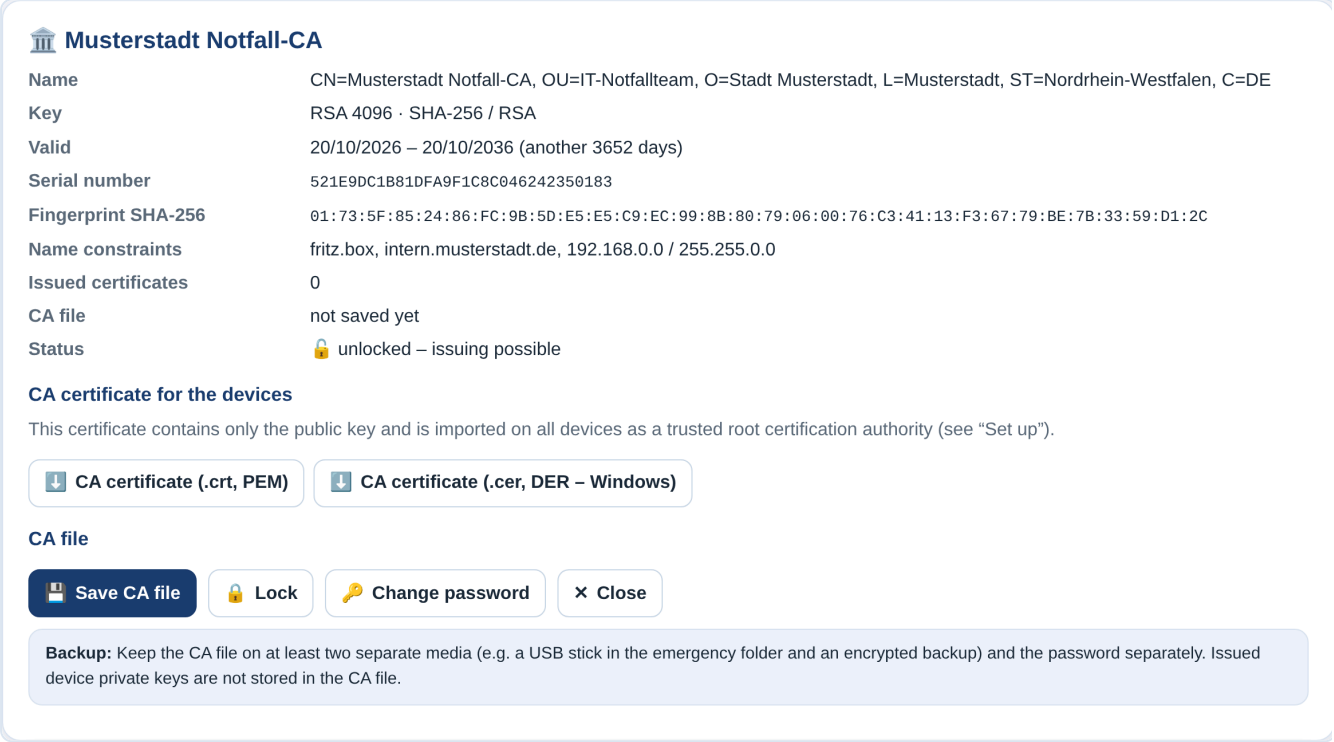


Fig. 3: Open certificate authority

Button	Function
CA certificate (.crt, PEM)	Public CA certificate for macOS, Linux, Firefox, Android and iOS.
CA certificate (.cer, DER – Windows)	The same certificate in binary format, installable on Windows by double-clicking.
Save CA file	Saves certificate, encrypted key, register and settings as `.dbm`.
Unlock / Lock	Decrypts the CA key with the password, or removes it from memory again.
Change password	Encrypts the CA key with a new password. Save the CA file afterwards.
Close	Closes the CA. CERT asks for confirmation if there are unsaved changes.

Opening and unlocking

A CA is loaded via *Open existing CA file* or by dragging the `.dbm` file into the window. CERT checks the file's SHA-256 checksum. If it does not match, the file was changed outside CERT; CERT still opens it and shows a notice.

CERT then asks for the password. After decryption it additionally checks that the key matches the CA certificate. The register can also be viewed while locked; issuing, signing and creating revocation lists require the CA to be unlocked.

Saving

Every issuance, revocation and settings change modifies the register. CERT shows unsaved changes above the CA details; closing the browser window triggers a warning. The default file name is `CA_<name>.dbm`.

Structure of the CA file

The CA file follows the suite's common DBM envelope with `_hashAlgorithm` and `_hash`. The program version is included in the checksum. CERT's own data is stored under the keys `cert_ca`, `cert_register` and `cert_crl`; unknown keys of other programs are kept unchanged when saving.

> **Backup:** keep the CA file on at least two separate media (e.g. a USB stick in the emergency folder and an encrypted backup), with the password kept separately. Without the password no new certificates can be issued – recovery is not possible.

6. Setting up the CA certificate on the devices

The CA certificate is imported once per end device as a trusted root certification authority. It contains only the public key. The CA's private key and the CA file are never transferred to end devices. The instructions are also available in the program's *Set up* area.

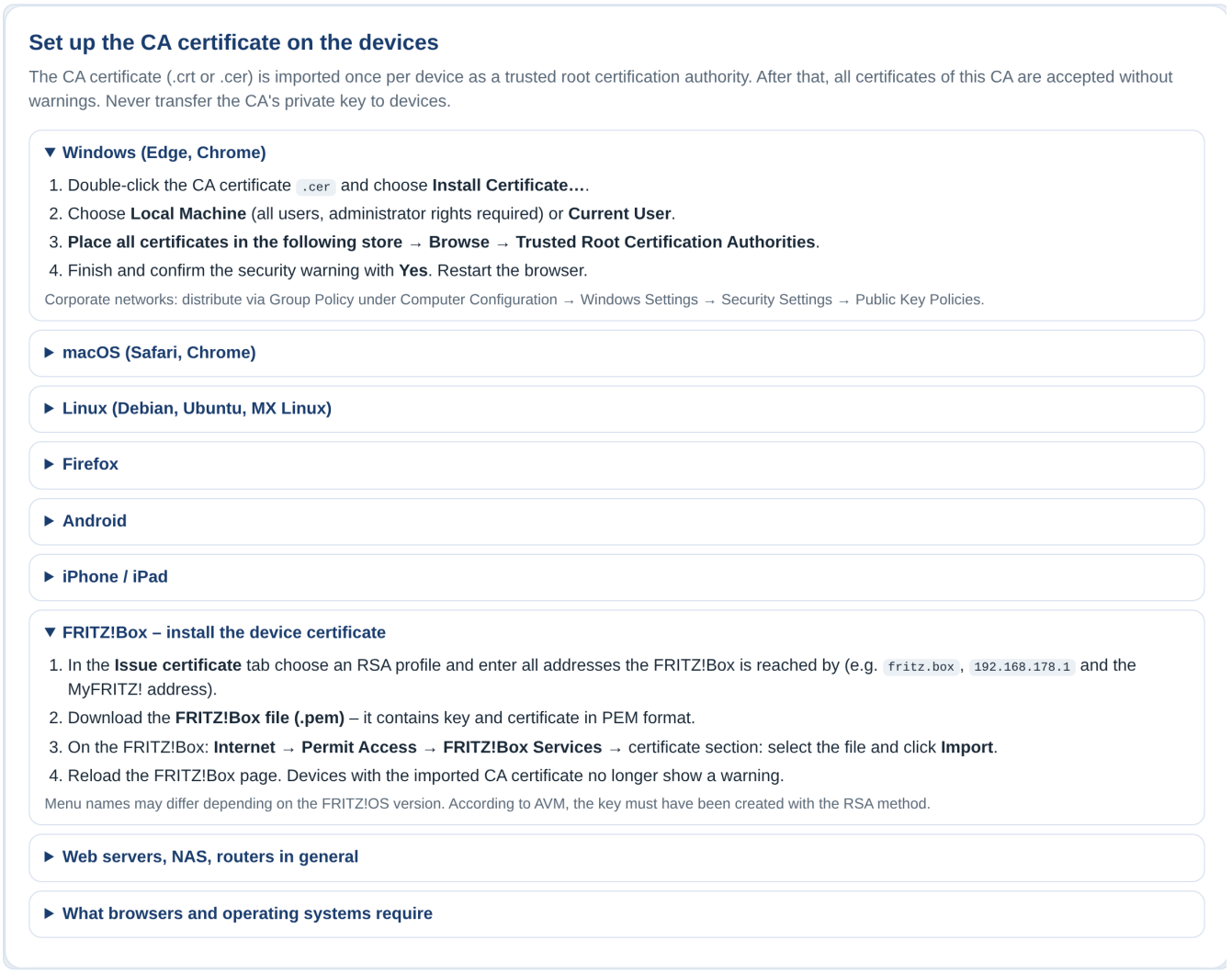


Fig. 4: Set up area with instructions

Windows (Edge, Chrome)

1. Double-click the CA certificate `.cer` and choose *Install Certificate…*.
2. Choose *Local Machine* (all users, administrator rights required) or *Current User*.
3. *Place all certificates in the following store → Browse → Trusted Root Certification Authorities*.
4. Finish, confirm the security warning with *Yes* and restart the browser.

In corporate networks the certificate can be distributed via Group Policy: Computer Configuration → Windows Settings → Security Settings → Public Key Policies.

macOS (Safari, Chrome)

1. Double-click the CA certificate `.crt`; choose the *System* keychain in Keychain Access.
2. Double-click the certificate and expand *Trust*.
3. Set *When using this certificate* to *Always Trust* and confirm with your password.

Linux (Debian, Ubuntu, MX Linux)

```
sudo cp ca.crt /usr/local/share/ca-certificates/
sudo update-ca-certificates
```

The `.crt` extension is mandatory. This applies to programs that use the system store.

Chrome and Chromium on Linux use their own certificate store (`~/.pki/nssdb`) and do not adopt the system-wide certificates. Importing via terminal is reliable; restart Chrome afterwards:

```
sudo apt install libnss3-tools
certutil -d sql:$HOME/.pki/nssdb -A -t "C,," -n "My CA" -i ca.crt
```

Alternatively open the certificate manager via `chrome://certificate-manager` and import the CA certificate under the trusted certificates installed by you – not under your own certificates, which expect PKCS#12 files.

Firefox

1. Settings → *Privacy & Security* → *View Certificates…*
2. Tab *Authorities* → *Import…* → choose the CA certificate (not in the *Your Certificates* tab).
3. Tick *Trust this CA to identify websites* and confirm.

Firefox uses its own certificate store. On Windows and macOS, Firefox adopts the system certificates if `security.enterprise_roots.enabled` is set to `true` in `about:config`.

Android

1. Transfer the CA certificate `.crt` to the phone.
2. Settings → Security (depending on manufacturer: Security and privacy → More security settings) → *Encryption & credentials* → *Install a certificate* → *CA certificate*.
3. Confirm the warning and choose the file.

Chrome and Firefox accept user-installed CAs. Some apps only trust the preinstalled certification authorities.

iPhone and iPad

1. Open the CA certificate `.crt` via Safari, Mail or AirDrop – it is downloaded as a profile.
2. Settings → *Profile Downloaded* → *Install*.

3. Settings → General → About → *Certificate Trust Settings* → enable the certificate under *Enable full trust for root certificates*.

7. Issuing a certificate

In the *Issue certificate* area, the key pair and certificate for a device are created directly in the browser. This is suitable for all devices that can import their own certificate including the key: FRITZ!Box, NAS, routers, firewalls, web servers and web applications.

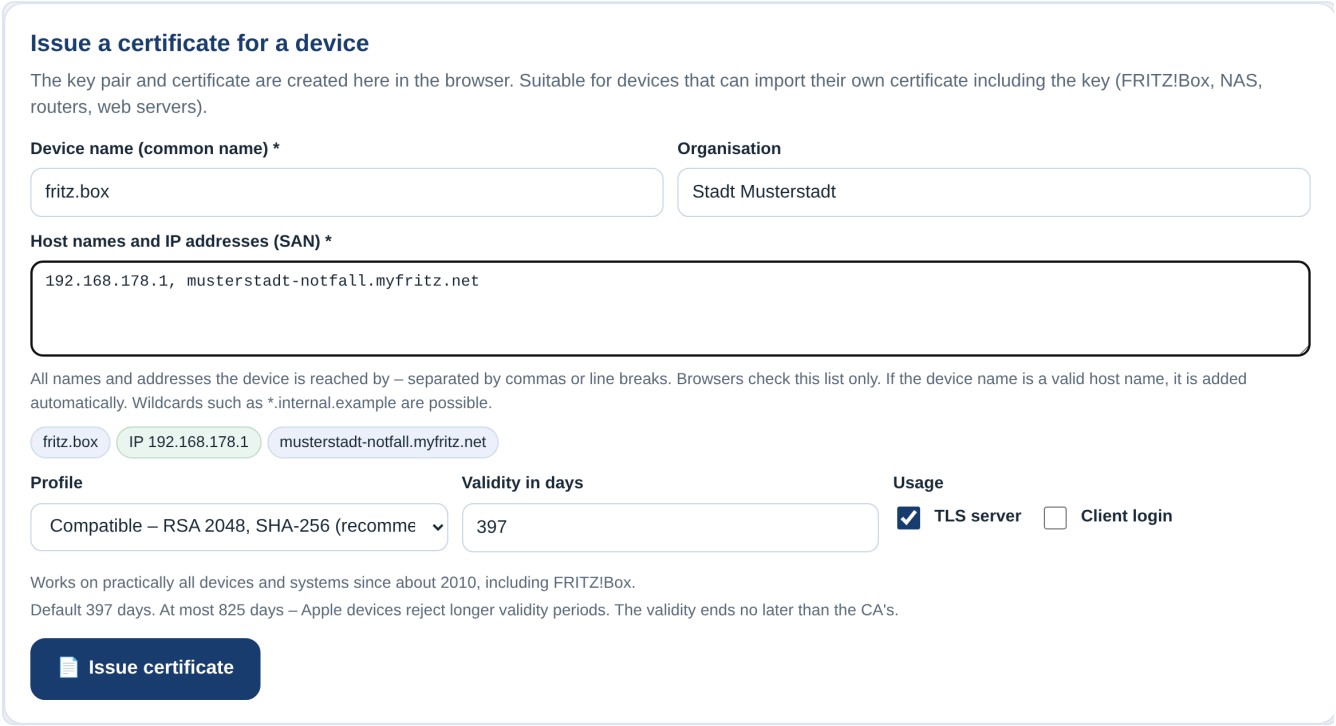


Fig. 5: Issuing a certificate for a device

Field	Meaning
Device name (common name) *	Display name of the certificate, e.g. `fritz.box`. If it is a valid host name, it is added to the address list automatically.
Organisation	Optional entry in the certificate.
Host names and IP addresses (SAN) *	All names and addresses the device is reached by, separated by commas or line breaks.
Profile	Key type and signature algorithm (see table below).
Validity in days	1 to 825 days, default 397 days.
Usage	*TLS server* for web and device interfaces (default), *Client login* for certificates a device or user logs in with (e.g. VPN).

Addresses (SAN)

Below the input field CERT shows every recognised address as a chip: host names in blue, IP addresses in green, invalid entries in red. Possible entries:

- host names such as `fritz.box` or `nas.internal.example.org`, also with umlauts (converted automatically to the international notation),
- wildcards for exactly one level, e.g. `*.internal.example.org`,
- IPv4 addresses such as `192.168.178.1` and IPv6 addresses such as `fd00::1`.

> The certificate is only valid for the listed addresses. If a device is also reached by IP address or another name (e.g. the MyFRITZ! address), these must be listed as well.

Profiles

Profile	Key / signature	Use
Compatible (default)	RSA 2048 / SHA-256	Practically all devices and systems since about 2010, including FRITZ!Box
Strong	RSA 3072 / SHA-256	Longer key, slightly slower handshake, suitable for FRITZ!Box
Modern	ECDSA P-256 / SHA-256	Current systems; not for FRITZ!Box and very old devices
Legacy devices	RSA 2048 / SHA-1	Only for devices that cannot handle SHA-256; only available with an RSA CA

> Certificates with SHA-1 are rejected by current browsers and Apple systems. The *Legacy devices* profile is intended exclusively for old devices that understand no other algorithm.

Validity

The default of 397 days matches the maximum validity of public certificates. More than 825 days is not possible because Apple devices reject longer server certificates. If the validity extends beyond the CA's expiry date, CERT shortens it to that date and shows a notice. The start of validity is one hour before the time of issue so that small clock deviations of devices do not interfere.

After *Issue certificate*, the certificate is entered in the register and the result is shown with all download options (chapter 8).

8. Files and formats

After issuing, CERT shows the certificate details and all download options.

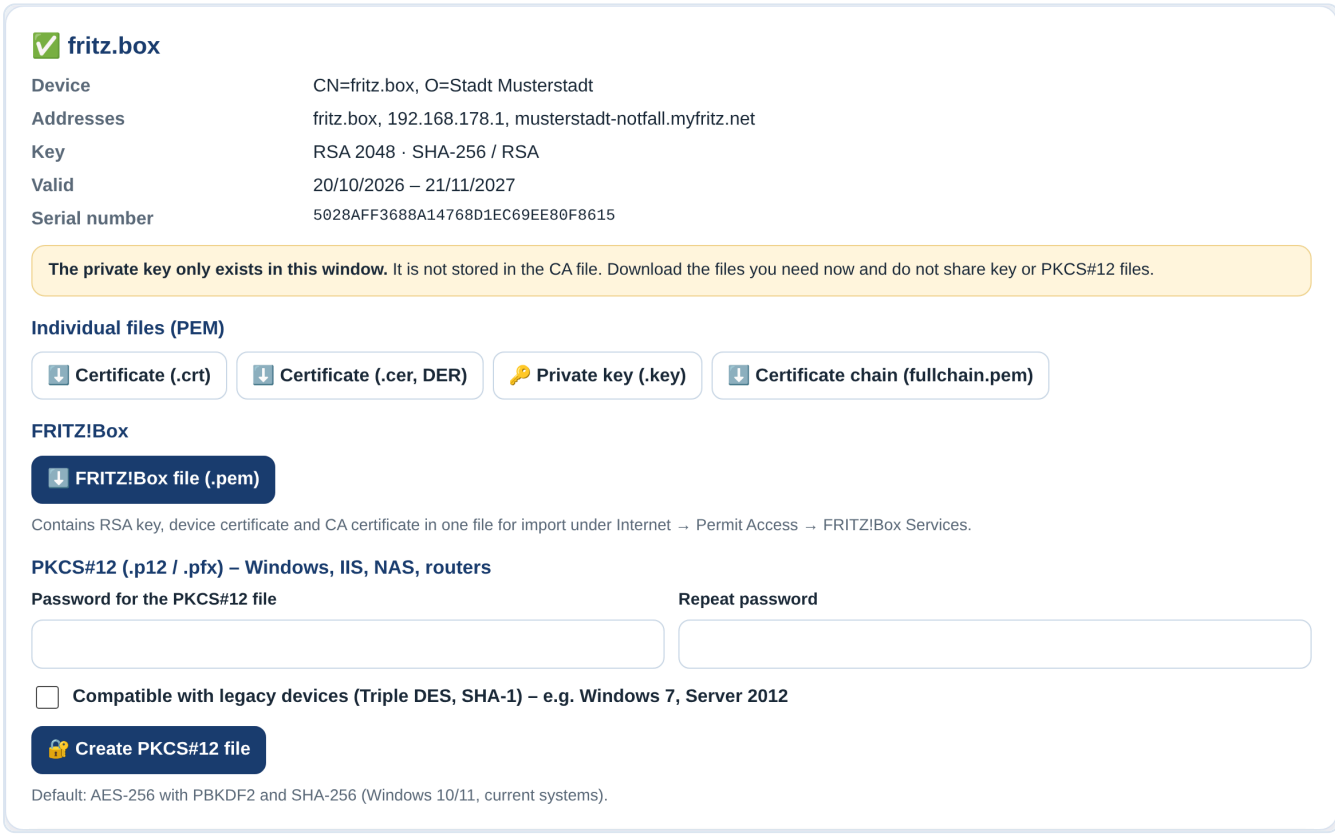


Fig. 6: Result with all download options

File	Content	Use
Certificate (.crt)	Device certificate, PEM	Servers and devices that expect certificate and key separately
Certificate (.cer, DER)	Device certificate, binary	Windows and devices with DER import
Private key (.key)	Key, PEM (PKCS#8), unencrypted	nginx, Apache, devices with separate key import
Certificate chain (fullchain.pem)	Device certificate and CA certificate	Web servers, so that clients receive the complete chain
FRITZ!Box file (.pem)	RSA key, device certificate and CA certificate in one file	Import into FRITZ!Box (RSA profiles only)
PKCS#12 (.p12)	Key, certificate and CA certificate, password protected	Windows, IIS, many NAS and routers

PKCS#12

A separate password of at least 8 characters is set for the PKCS#12 file. It is requested when importing on the target device.

- **Default:** AES-256 with PBKDF2 and SHA-256 – for Windows 10/11 and current systems.
- **Compatible with legacy devices:** Triple DES with SHA-1 – for Windows 7, Windows Server 2012 and older devices that cannot read the modern algorithm.

> **The private key only exists in this window.** It is not stored in the CA file. Keep key, FRITZ!Box and PKCS#12 files secret, do not send them by email and delete them from the working computer after installation. If the key is lost, simply issue a new certificate.

Web server configuration

System	Certificate	Key
nginx	`ssl_certificate` → fullchain.pem	`ssl_certificate_key` → .key
Apache	`SSLCertificateFile` → fullchain.pem	`SSLCertificateKeyFile` → .key
Windows / IIS	Import the PKCS#12 file	contained in the PKCS#12 file

9. FRITZ!Box

The FRITZ!Box generates its own self-signed certificate by default. With a certificate from your own CA, browser warnings disappear on all devices on which the CA certificate is set up.

1. In the *Issue certificate* area, enter `fritz.box` as the device name.
2. Under *Host names and IP addresses*, enter all addresses of the FRITZ!Box, e.g. `192.168.178.1` and the MyFRITZ! address.
3. Choose an RSA profile (*Compatible* or *Strong*) and issue the certificate.
4. Download the **FRITZ!Box file (.pem)**.
5. On the FRITZ!Box: *Internet* → *Permit Access* → *FRITZ!Box Services* → certificate section: select the file and click *Import*.
6. Reload the FRITZ!Box page.

> According to AVM, certificate and private key must be in PEM format and the key must have been created with the RSA method. The CERT FRITZ!Box file meets both requirements. Menu names may differ depending on the FRITZ!OS version.

For ECDSA profiles CERT does not offer a FRITZ!Box file because the FRITZ!Box does not accept such keys.

10. Signing a CSR

Many devices – servers, NAS, cameras, firewalls – generate their key themselves and only provide a certificate signing request (CSR, usually a `.csr` or `.req` file). CERT signs it with your CA. The private key never leaves the device.

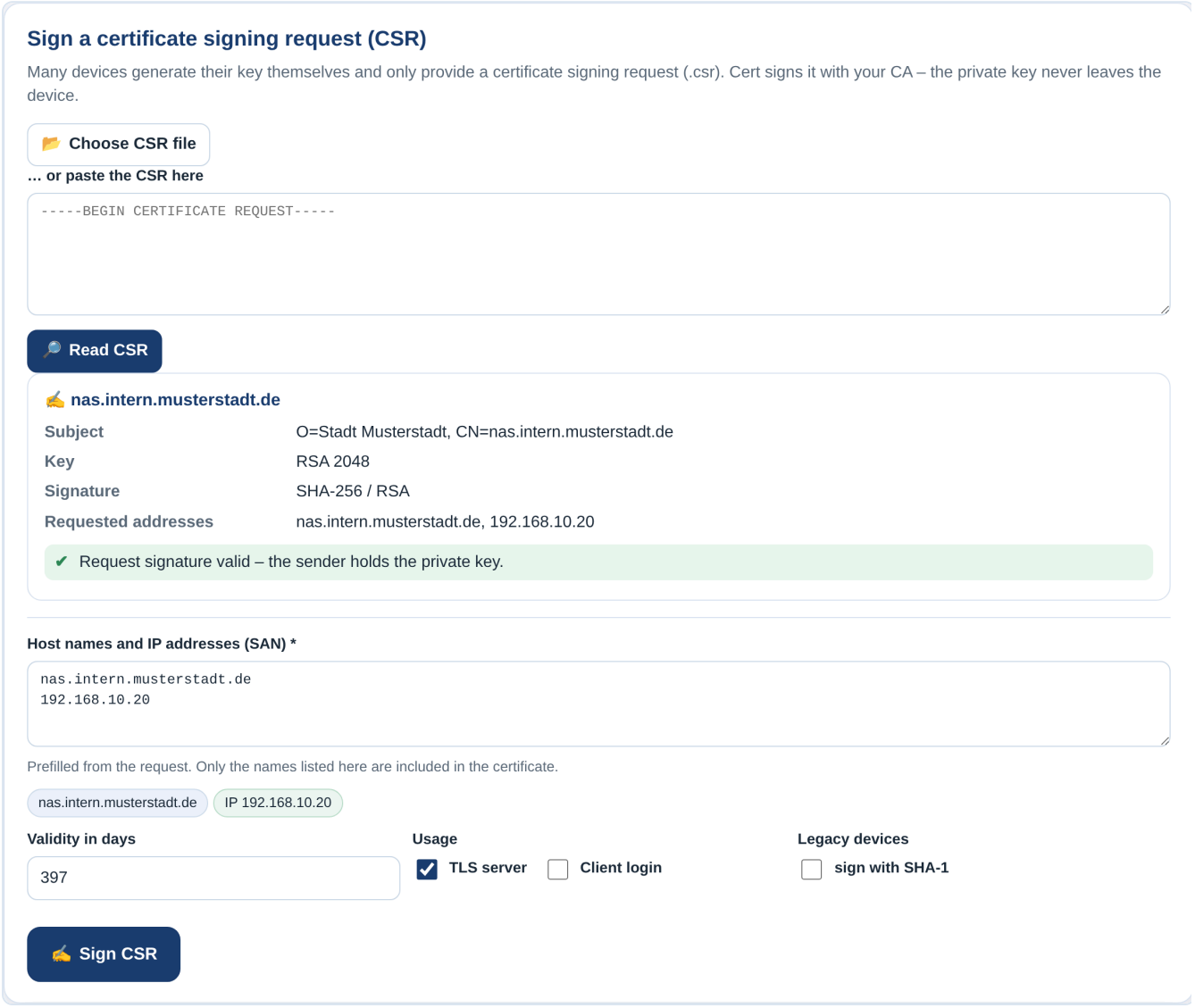


Fig. 7: Read signing request with options

1. Generate and export the CSR on the device.
2. Load it in CERT via *Choose CSR file*, by pasting the PEM text or by dragging the file into the window, then click *Read CSR*.
3. CERT shows subject, key, signature algorithm and the requested addresses and verifies the request signature.
4. Check the addresses and add more if needed, set validity and usage.
5. Click *Sign CSR* and download the certificate or the certificate chain.
6. Install the certificate on the device that created the request.

Check	Meaning
Request signature valid	The sender holds the corresponding private key. Only then can the CSR be signed.
Signature invalid	The request was altered or is damaged and will not be signed.
RSA below 2048 bits	CERT asks for confirmation before signing, as browsers and Apple systems reject such keys.
Algorithm not supported	RSA and ECDSA (P-256, P-384, P-521) with SHA-1 or SHA-2 are supported.

CERT takes the subject from the request and only the addresses listed in the *Host names and IP addresses* field into the certificate. With an RSA CA, *sign with SHA-1* can additionally be chosen for legacy devices.

11. Register and revocation list

Issuance register

The register contains all certificates issued with the CA – including signed CSRs – with device name, serial number, key, addresses, validity and status. It is stored in the CA file.

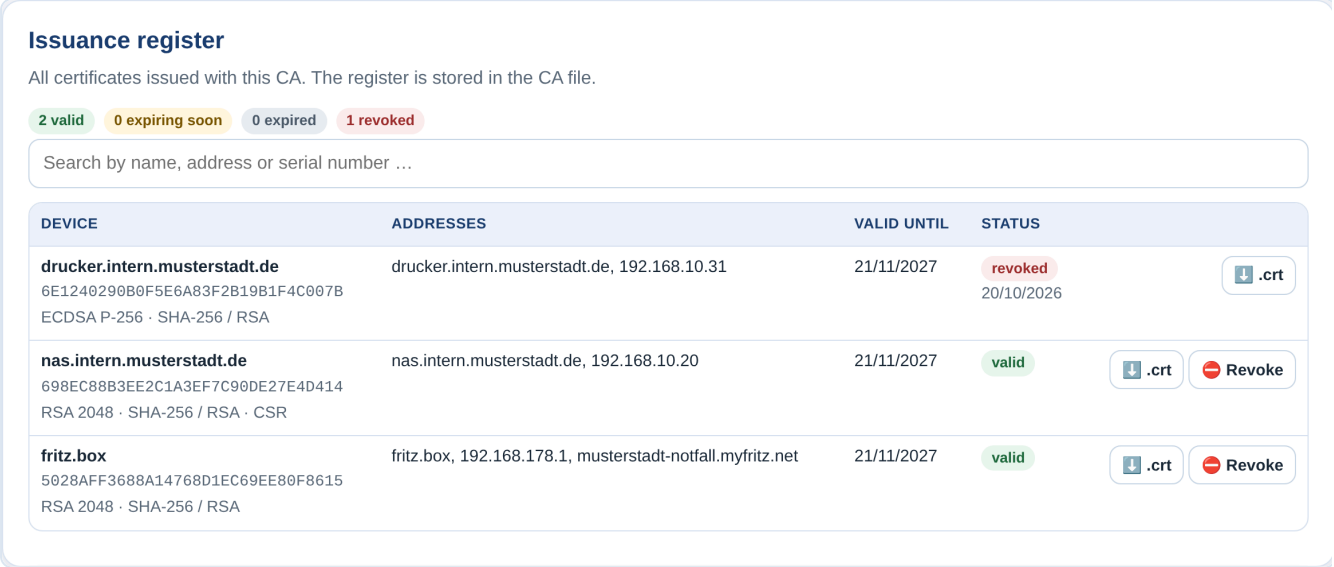


Fig. 8: Issuance register

Status	Meaning
valid	Certificate is valid for at least another 30 days
expires in … days	Less than 30 days left – reissue in time
expired	Validity exceeded
revoked	Revoked prematurely, with date

The search field filters by name, address or serial number. Any certificate can be downloaded again via *.crt*; private keys are not contained in the register.

Revoking a certificate

Revoke declares a certificate invalid prematurely, e.g. when a device was lost or decommissioned or a key became known. Available reasons are *Unspecified*, *Key compromise*, *Affiliation changed*, *Superseded* and *Cessation of operation*. Revocation is final. Revoking is also possible while the CA is locked; the revocation list requires an unlocked CA.

Revocation list (CRL)

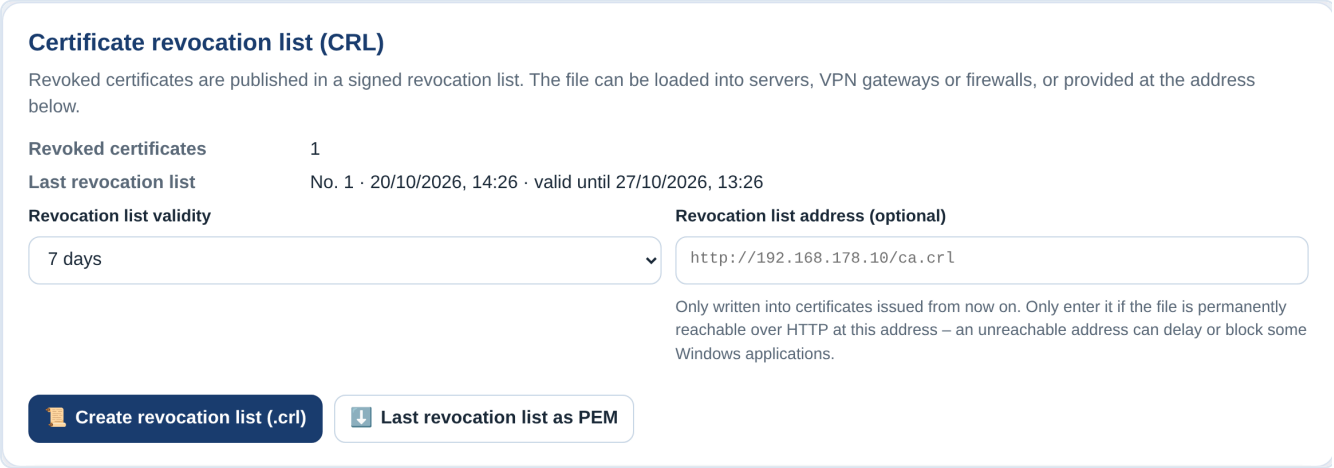


Fig. 9: Creating a revocation list

Create revocation list (.crl) generates a list of all revoked certificates signed by the CA with a sequential number and downloads it in DER format; a PEM version is also available. The revocation list is valid for 7, 30, 90, 180 or 365 days. A new list must be created before it expires.

The revocation list is loaded into servers, VPN gateways or firewalls that check a CRL, or provided at a fixed address.

Revocation list address

Optionally, an HTTP address can be entered at which the revocation list is permanently available, e.g. `http://192.168.178.10/ca.crl`. It is written into all certificates issued afterwards.

> Only enter the address if the file is permanently reachable there. An unreachable address can delay or block some Windows applications – especially in a crisis. Without an entry the field remains empty in the certificates.

12. Checking certificates

The *Check certificate* area examines existing files: certificates, certificate chains, signing requests, revocation lists, private keys and PKCS#12 containers, in PEM or DER format. Files are chosen, dragged into the window or pasted as PEM text. A file with several certificates is evaluated completely.

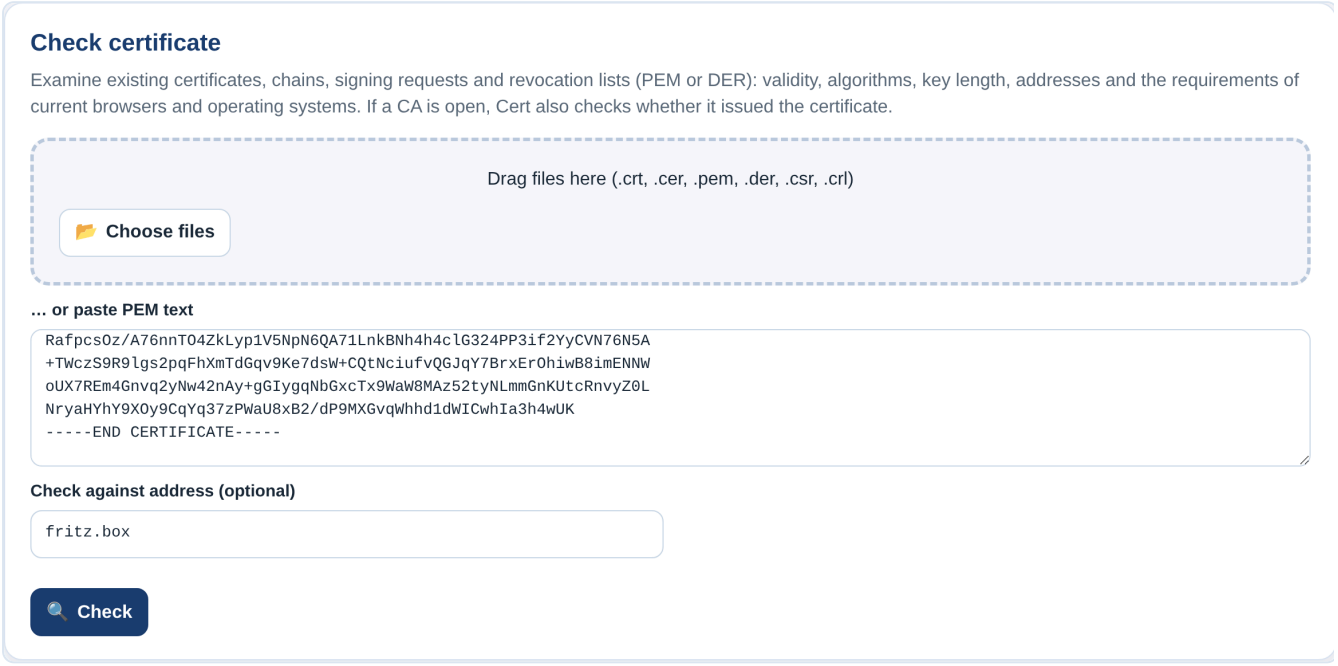


Fig. 10: Entering the data to check

For each certificate CERT shows subject, issuer, validity, serial number, key, signature algorithm, addresses, usages and the SHA-256 and SHA-1 fingerprints, followed by the assessment.

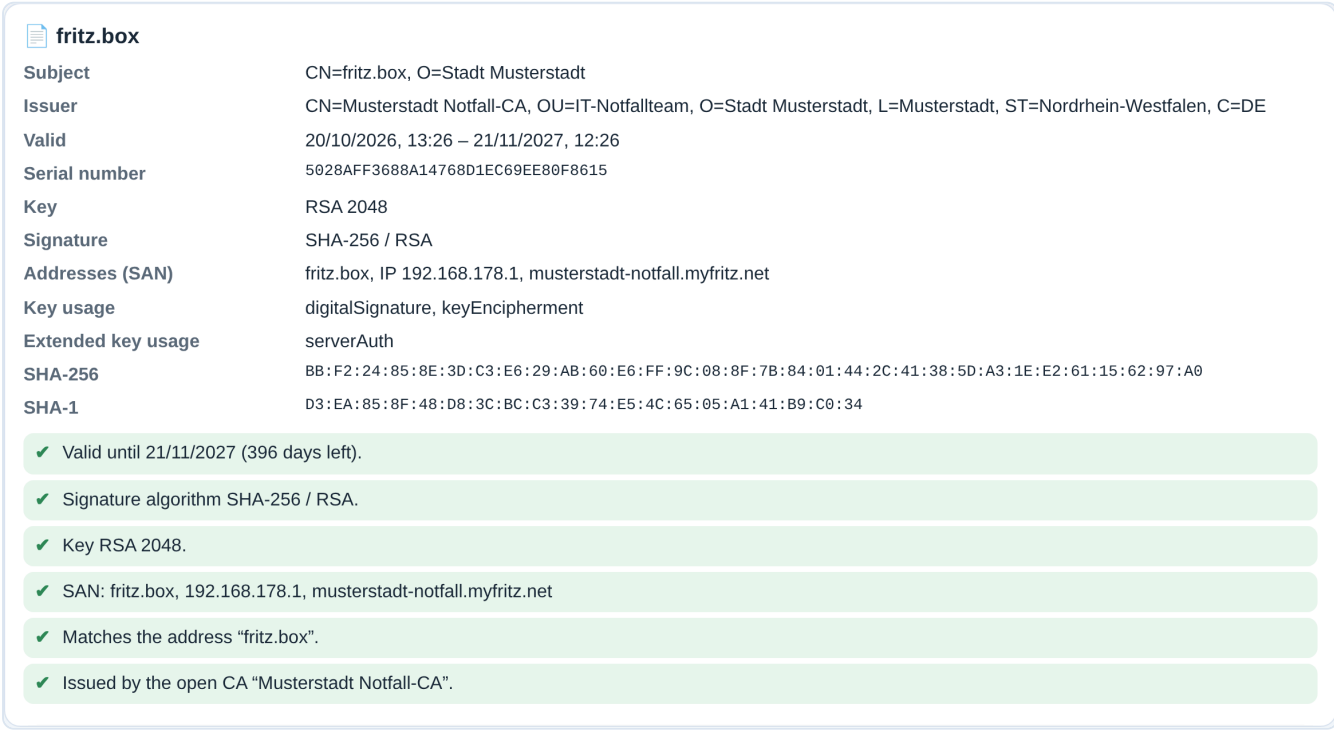


Fig. 11: Assessment of a device certificate

Marker	Meaning
✔ green	OK
! yellow	Notice that may cause problems (e.g. expiry in less than 30 days, missing serverAuth usage, self-signed)
✖ red	Rejected by browsers or operating systems (e.g. expired, SHA-1 or MD5, RSA below 2048 bits, no addresses in the SAN, validity of more than 825 days)
i blue	Information (e.g. CA with path length, name constraints)

Further checks

- **Check against address:** with an address entered (e.g. `fritz.box` or `192.168.178.1`), CERT checks whether the certificate is valid for it – including wildcards.
- **Chains:** if the issuing certificate is part of the same check, the signature is verified against it. Root certificates are checked for their self-signature.
- **Open CA:** if a CA is open, CERT checks whether it issued the certificate and whether it is revoked in the register. Revocation lists are verified against the open CA.
- **Not yet valid:** if the start of validity is in the future, the device date and time are usually set incorrectly.

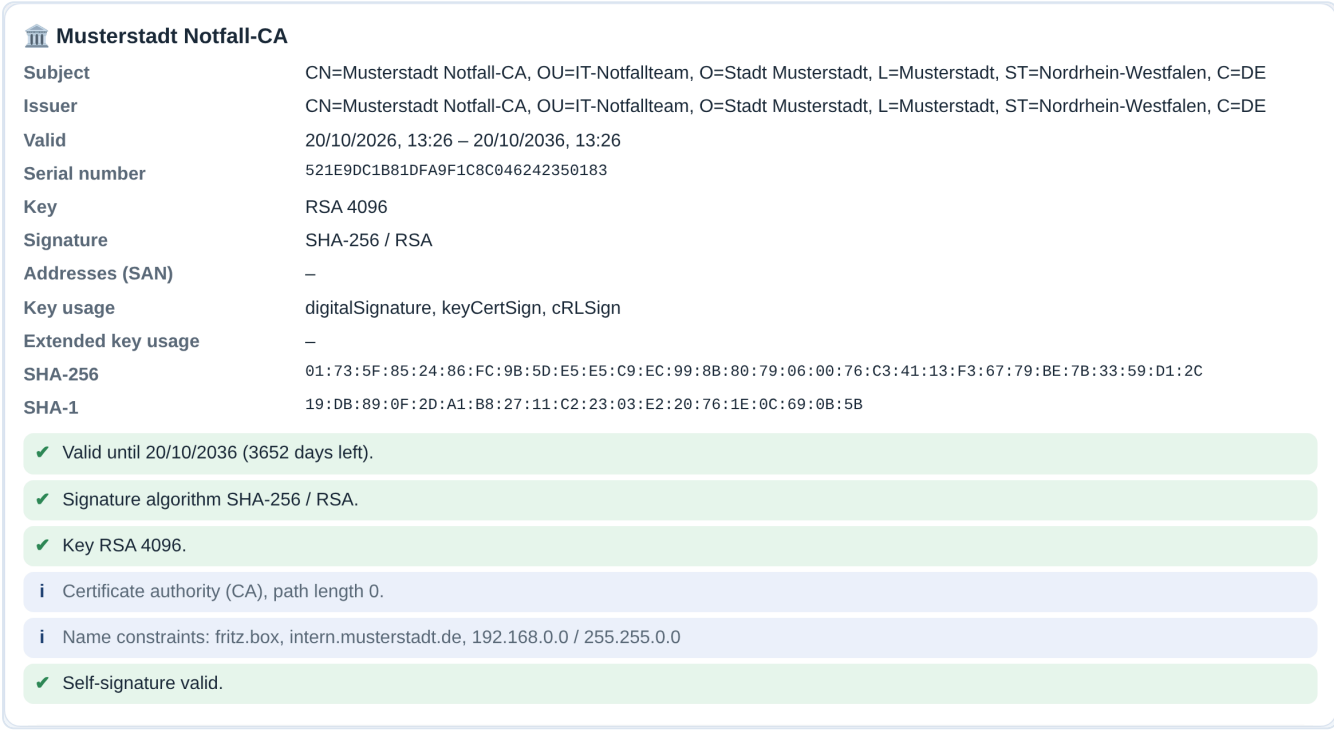


Fig. 12: Assessment of a CA certificate with name constraints

13. Security and technology

Area	Implementation
Processing	Fully local in the browser, no network connection (Content Security Policy with `connect-src 'none'`)
Key generation and signing	Browser Web Crypto API: RSA (PKCS#1 v1.5) and ECDSA
Protection of the CA key	AES-256-GCM, key derivation PBKDF2-SHA-256 with 600,000 iterations, random salt and IV
CA file integrity	SHA-256 checksum over the entire content including the program version
Serial numbers	128 bits, random
CA certificate	X.509 v3, BasicConstraints CA with path length 0, KeyUsage keyCertSign and cRLSign, key identifiers, optional NameConstraints
Device certificate	X.509 v3, SAN, KeyUsage, ExtendedKeyUsage serverAuth/clientAuth, key identifiers, optional revocation list address
PKCS#12	Default: PBES2 with AES-256-CBC and PBKDF2/HMAC-SHA-256, MAC SHA-256 · Legacy: Triple DES and MAC SHA-1
Revocation list	X.509 CRL v2 with CRL number, key identifier and revocation reason
Program code	ASN.1/DER, X.509, PKCS#8, PKCS#10, PKCS#12 and Triple DES as original ZANNI GROUP code, no third-party libraries

Limits

- CERT does not replace a public certificate authority: certificates of your own CA are only accepted on devices on which the CA certificate is set up.
- Security depends on protecting the CA file and password. The computer on which CERT is unlocked should be trustworthy.
- OCSP queries and intermediate certification authorities are not part of CERT. For use in your own network and in a crisis, direct issuance by the root CA is sufficient.
- When opened over the network, the browser only provides the Web Crypto API via HTTPS. From a USB stick (`file://`) and via jumboblock.org, CERT works without restriction.

14. Troubleshooting

Problem	Cause and solution
Browser warns despite certificate	The CA certificate is not set up on this device as a trusted root certification authority (chapter 6), or the browser was not restarted after the import. Firefox uses its own store.
Browser reports the CA certificate as “corrupted” on import	The CA certificate was imported in the *Your certificates* section, which expects PKCS#12 files with a private key. The CA certificate belongs with the authorities or trusted certificates; for Chrome on Linux see chapter 6.
Warning “name mismatch”	The address used is missing in the SAN. Reissue the certificate with all names and IP addresses and verify it in the *Check certificate* area with *Check against address*.
Certificate “not yet valid” or “expired”	Check the device date and time. Devices without a clock battery start with a wrong date after a power failure.
iPhone/iPad or Mac rejects it	Fully trust the CA certificate under *Certificate Trust Settings*. Validity at most 825 days, usage TLS server, no SHA-1 – met by the *Compatible*, *Strong* and *Modern* profiles.
FRITZ!Box does not accept the file	Use an RSA profile and import the *FRITZ!Box file (.pem)*, not the individual certificate.
Windows does not import the PKCS#12 file	For Windows 7, Server 2012 or older choose the option *Compatible with legacy devices*.
“The password is incorrect” when unlocking	Check upper/lower case and keyboard layout. Without the correct password the CA key cannot be recovered.
“Checksum mismatch” when opening	The CA file was changed outside CERT. Check the register and settings and use a backup if entries are missing.
Certificate rejected despite name constraints	The name or IP address is outside the CA's constraints. The constraints are shown in the *Certificate authority* area.
Private key lost	Issue a new certificate and revoke the old one in the register.
Web Crypto API not available	Use an up-to-date browser; use HTTPS when opened over the network.