JUMBO BLOCK CRYPT

Encrypting and signing files and texts



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. Keyring
5. Identities
6. Contacts and trust
7. Encrypting
8. Decrypting
9. Signing
10. Verifying signatures
11. Interaction with other programs
12. Files and formats
13. Security and technology
14. Troubleshooting

1. Overview

CRYPT is the encryption tool in the KRITIS area of JUMBO BLOCK. It encrypts files and texts so that nobody can read or alter them unnoticed in transit, and uses signatures to prove where content comes from. In a crisis in particular, situation reports, plans, access data and approvals are exchanged via unusual routes – by USB stick, private messengers or a substitute mailbox. CRYPT makes these routes secure.

Fully offline: CRYPT 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: file format, certificate parsing and PKCS#12 are original program code. Key generation, ECDH, ECDSA, RSA, HKDF, PBKDF2 and AES use the browser's built-in Web Crypto API.

Also for large files: files are processed in segments of 4 MiB. Several gigabytes are possible; progress and cancellation are shown.

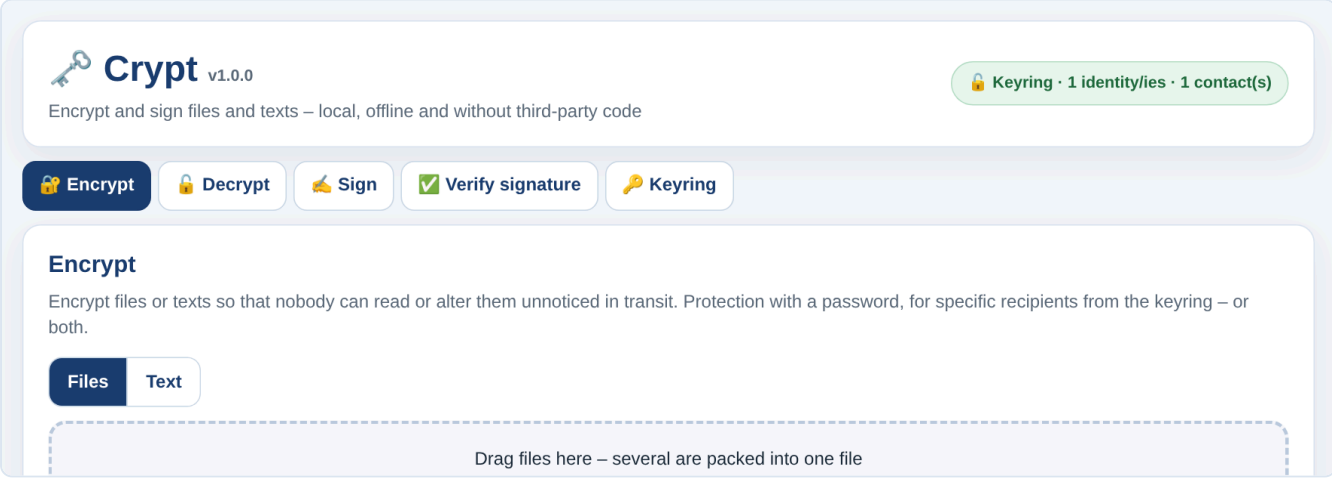


Fig. 1: The five areas of CRYPT

Area	Purpose
Encrypt	Encrypt files or text with a password, for recipients from the keyring or both, and optionally sign them as well
Decrypt	Open encrypted files (.jbc) or text blocks; completeness and sender are verified
Sign	Provide files with a signature file (.jbs) or turn texts into a signed block
Verify signature	Prove that a file or text is unchanged and who it comes from
Keyring	Manage own identities, contacts and trusted certificate authorities

2. Basic terms

Term	Meaning
Password encryption	Anyone who knows the password can open it. Simple, but the password must be transmitted via a second, secure route.
Recipient encryption	Only selected people can open it – with the private key from their keyring. No shared password is needed.
Keyring	Password-encrypted file (.dbm) with your identities, your contacts and the certificate authorities you trust.
Identity	Your own key pair: for signing and for receiving encrypted content. Either created in CRYPT or imported as a certificate from JUMBO BLOCK CERT.
Public key (.jbid)	The shareable part of an identity with name, unit and contact details. Others need it to verify your signatures and to send you encrypted content.
Fingerprint	Short string (e.g. `6C4A-5666-3B11-1D42-89E0-5838`) that uniquely identifies a public key. Compared by phone or in person during the exchange.
Contact	Another person's public key in your keyring – “verified” once you have compared the fingerprint with the person.
Certificate authority (CA)	E.g. your own CA from JUMBO BLOCK CERT. Signatures with certificates of a trusted CA count as confirmed without every contact having to be verified individually.
Signature	Cryptographic proof that content comes from a particular identity and has not been changed since.

3. Workflow

One-time setup

1. Create a keyring and protect it with a strong password (chapter 4).
2. Create your own identity or import a certificate from CERT (chapter 5).
3. Give your public key (.jbid) to the people you exchange content with; load their .jbid files as contacts.
4. Compare fingerprints by phone or in person and mark contacts as verified – or store your own certificate authority as trusted (chapter 6).
5. Save the keyring and keep a backup copy in the emergency folder.

In operation

- **Passing on confidential content:** encrypt for recipients and sign (chapter 7). The recipient opens it with their keyring and sees that the content comes from you.
- **Without a keyring:** encrypt with a password and communicate the password via another route, e.g. by phone.
- **Approvals and instructions:** sign files or texts (chapter 9). Anyone can verify the signature – even without a keyring of their own.

4. Keyring

The keyring is created or opened in the *Keyring* tab. It is not needed for password encryption or for verifying signatures.

1. Under *Create a new keyring* enter a password of at least 12 characters twice.
2. Click *Create keyring*.
3. Create an identity (chapter 5) and save in the *Keyring file* area.

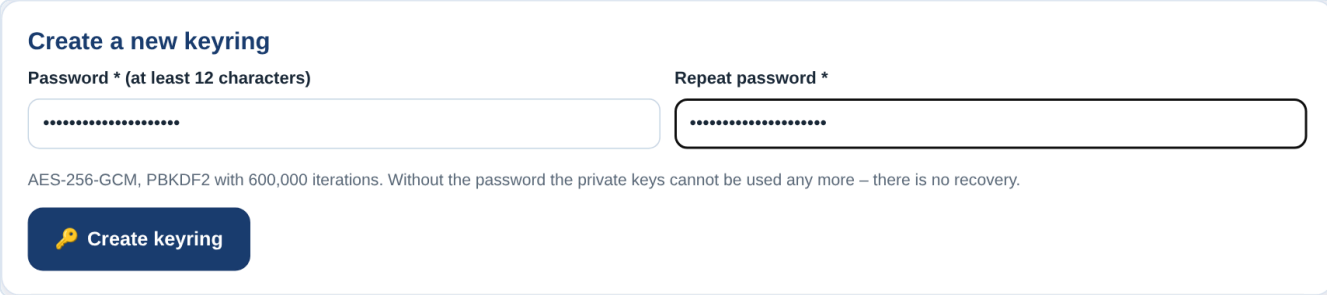


Fig. 2: Creating a new keyring

> Without the password the private keys cannot be used any more – there is no recovery. Keep the password separate from the keyring.

Opening, saving, locking

Open keyring reads an existing file; the password is requested afterwards. The *Keyring file* area shows the file, the encryption and unsaved changes.

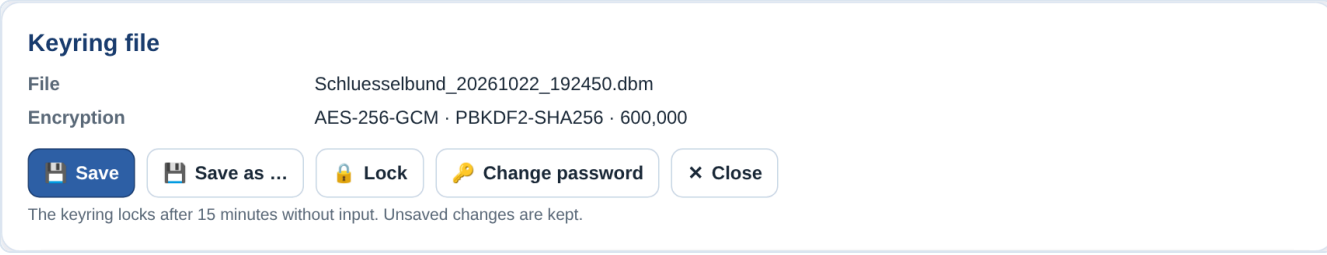


Fig. 3: Keyring file with save, lock and password change

Button	Effect
Save / Save as …	Writes the keyring encrypted to a file (.dbm). In Chrome and Edge the same file is overwritten on the next save; in other browsers a download is created.
Lock	Removes the keys from memory. The password is required to continue; unsaved changes are kept.
Change password	Set a new password, then save
Close	Close the keyring; CRYPT asks if there are unsaved changes

After 15 minutes without input the keyring locks itself.



Fig. 4: Locked keyring

5. Identities

An identity is your own key pair. You use it to sign and to receive content encrypted for you. A keyring can contain several identities, e.g. for different functions; one of them is the default.

Creating your own key pair

Create own key pair asks for name, unit or function and contact details. Name and unit appear in your signatures and in the public key; the contact details help others compare the fingerprint.

Importing a certificate from CERT

If you operate your own certificate authority with JUMBO BLOCK CERT, personal certificates can be used as identities. *Import certificate from Cert* reads the PKCS#12 file (.p12/.pfx, with its password) or key (.key) and certificate (.crt) together. RSA and ECDSA (P-256, P-384, P-521) are supported. The certificate must be intended for signing and valid. If the file also contains the CA certificate, CRYPT offers to add the certificate authority as trusted.



Fig. 5: Own identity with fingerprint and export of the public key

Passing on the public key

The *.jbid* button next to the identity saves the public key. This file contains no private key and may be passed on freely – by e-mail, messenger or USB stick. CRYPT displays the fingerprint, which you compare with the recipient.

A removed identity cannot be restored. Content encrypted only for this identity can no longer be read afterwards.

6. Contacts and trust

Contacts are other people's public keys. *Load contact (.jbid)* imports one or more files. CRYPT checks whether the file is unchanged and whether it is one of your own identities.

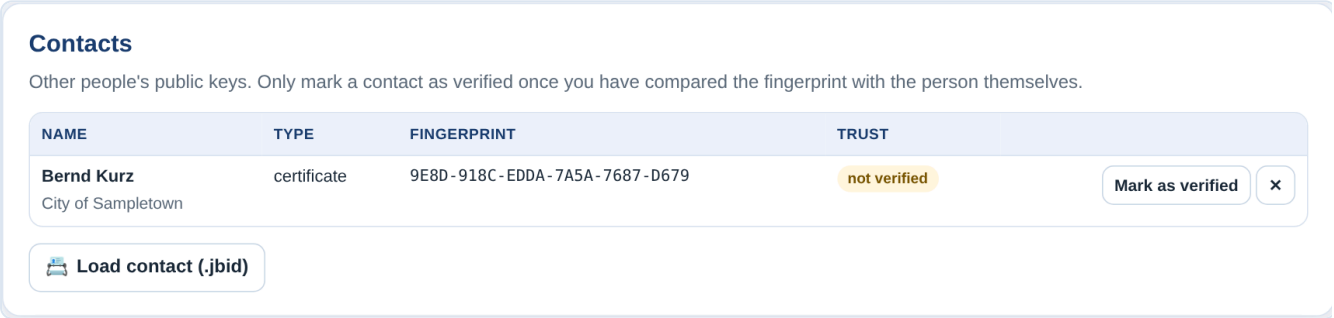


Fig. 6: Contact, not yet verified

Only once you have compared the fingerprint with the person themselves – by phone via a known number or in person – do you mark the contact with *Mark as verified*. This rules out that someone has slipped you a foreign key under a false name.

Trusted certificate authorities

With many participants, individual verification is laborious. If your organisation operates its own certificate authority with JUMBO BLOCK CERT, load its CA certificate with *Load CA certificate (.crt, .cer)*. Signatures with certificates of this authority then count as confirmed.

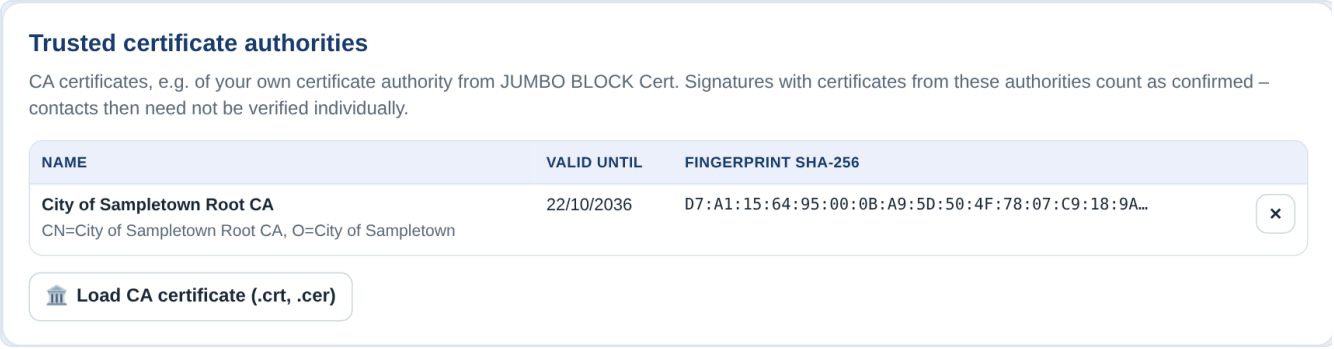


Fig. 7: Own certificate authority stored as trusted

> Only add your own or a reliably known certificate authority. All certificates it issues then count as confirmed.

Trust levels

Level	Meaning
Your own identity	The content was signed with one of your identities.
Contact, fingerprint verified	Known sender, confirmed in person
Certificate of a trusted CA	The certificate verifiably comes from a stored certificate authority and was valid at the time of signing.
Contact, fingerprint not yet verified	Key known, but origin not confirmed
unknown	Key not in the keyring – compare the fingerprint with the person; the sender can be added directly as a contact.

7. Encrypting

In the *Encrypt* area first choose whether *Files* or a *Text* should be encrypted.

1. **Content:** drag files into the field or add them via *Choose files*. Several files are packed into one encrypted file; names and modification dates are kept. For *Text* enter the text.
2. **Protection:** *With password* (at least 12 characters) and/or *For recipients*. Recipients are selected from the identities and contacts of the keyring; individual public keys can also be added without a keyring via *Load a recipient's key (.jbid)*.
3. **Sender:** with *Also sign with my identity* the recipient sees when opening that the content comes from you.
4. Click *Encrypt*.

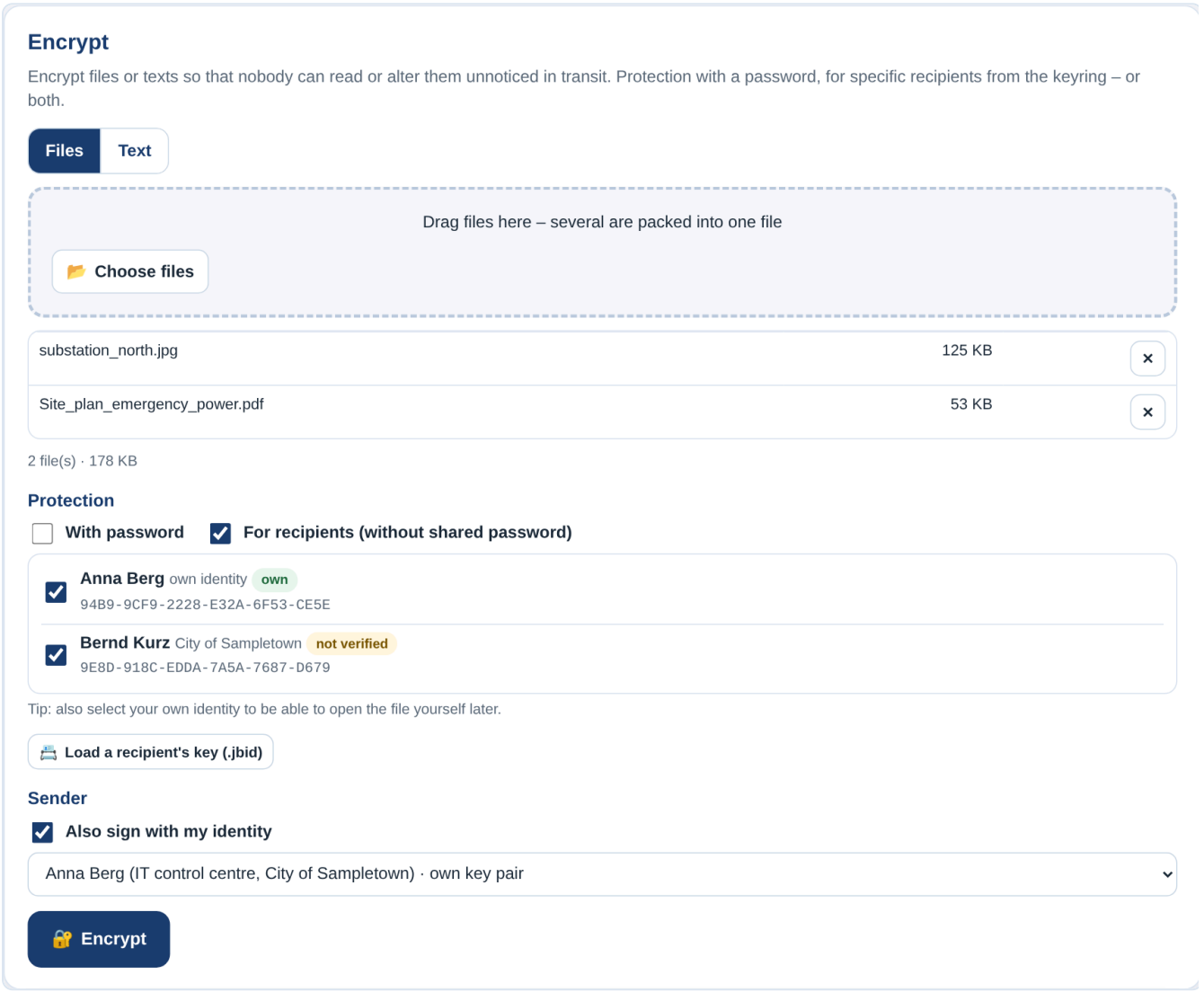


Fig. 8: Encrypting and signing two files for two recipients

Tip: also select your own identity to be able to open the file yourself later. CRYPT preselects it by default.

Files are saved as `.jbc`. For texts a text block appears, which is pasted completely, including the header and footer lines, into e-mail or messenger.

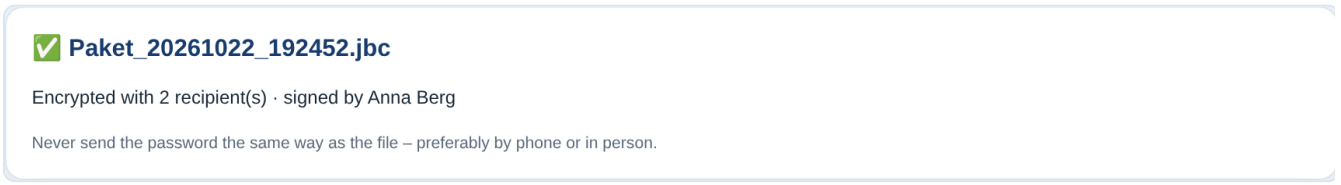


Fig. 9: Result: encrypted file for two recipients, signed

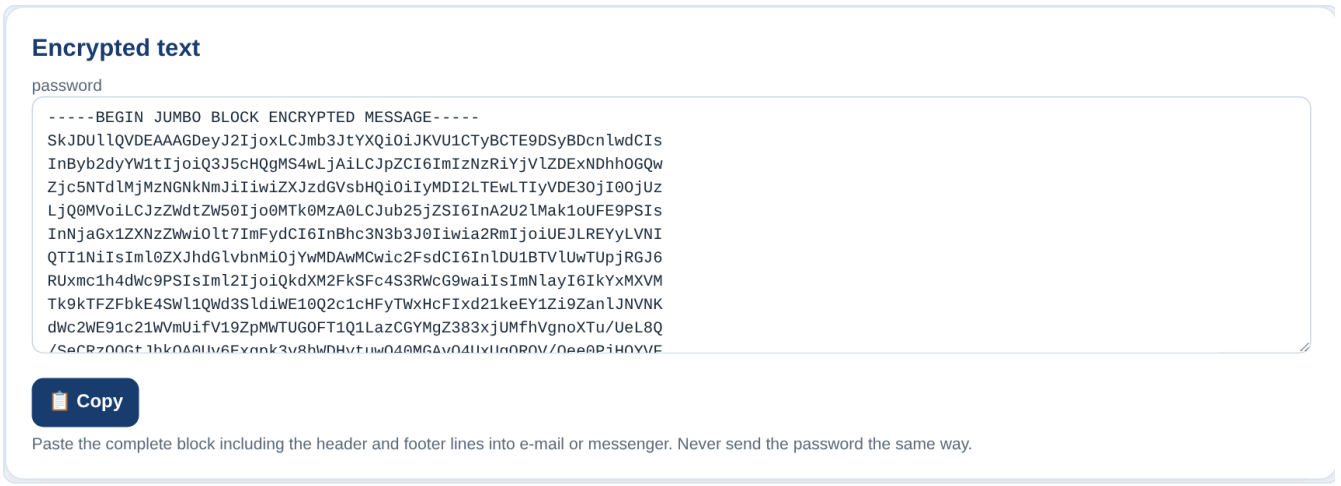


Fig. 10: Encrypted text block with password

> Never send the password the same way as the encrypted file – preferably by phone or in person.

8. Decrypting

1. In the *Decrypt* area choose the file (.jbc) or drag it into the field – or paste an encrypted text block and click *Read*. Surrounding text such as a greeting in the e-mail does not interfere.
2. CRYPT shows how the content is protected: password and/or recipients. For recipient encryption, CRYPT recognises with the keyring open whether one of your identities is included.
3. Enter the password if required and click *Decrypt*.

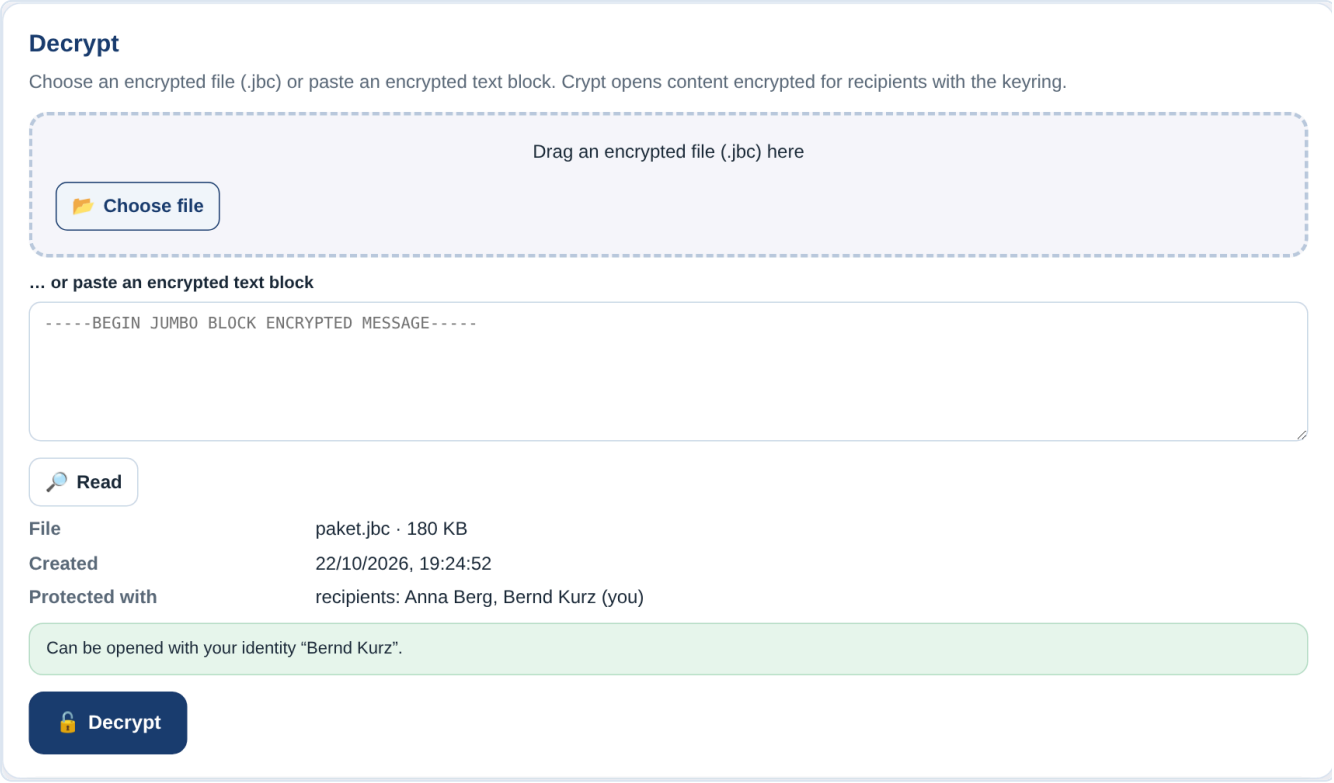


Fig. 11: File for recipients – can be opened with your own identity

When decrypting, CRYPT checks every segment and finally the checksum of the entire content. Only if everything is correct does *Decrypted and unchanged* appear. A contained signature is shown with its trust level.

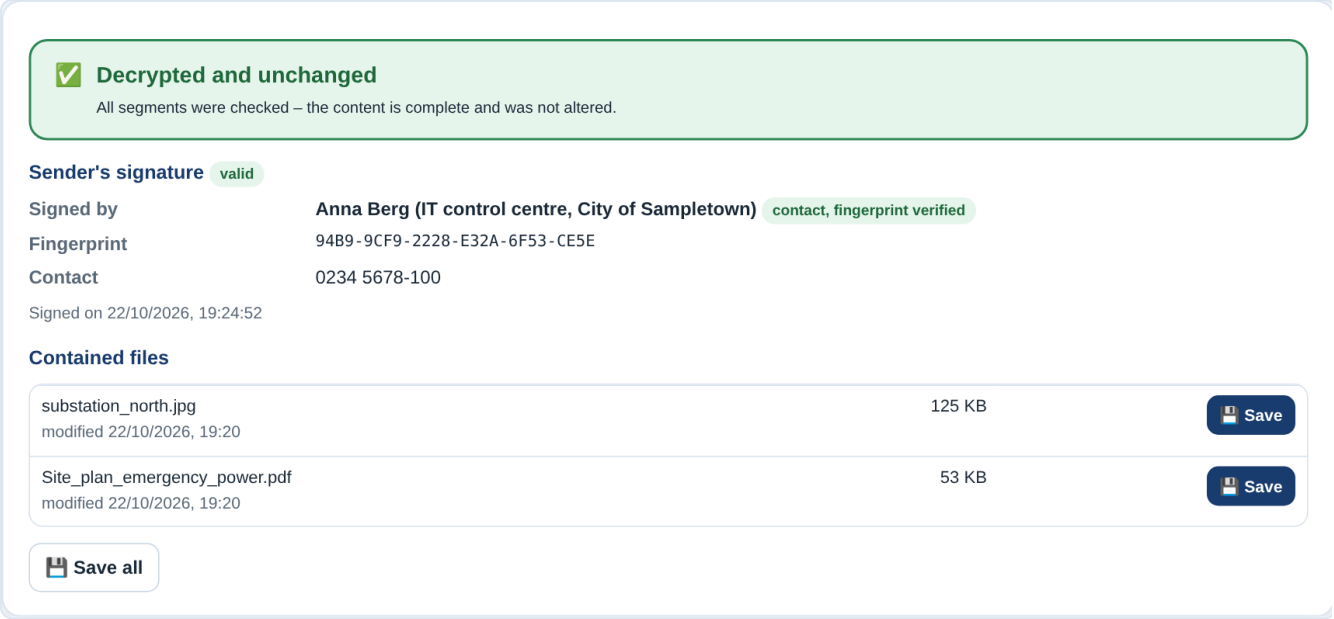


Fig. 12: Decrypted and unchanged, signed by a verified contact

The contained files can be saved individually or with *Save all*. Texts appear in a field for copying.

If CRYPT reports an error while decrypting, the file was damaged, truncated or altered. Partial results are not output.

9. Signing

A signature proves that a file or text comes from you and has been unchanged since. The keyring must be open to sign.

Signing files

1. Choose the identity under *Sign as*.
2. Add files. Optionally enter a *Note*, e.g. “approved for restart” – the note is part of the signature.
3. Click *Sign*. A signature file `file.ext.jbs` is created for each file.

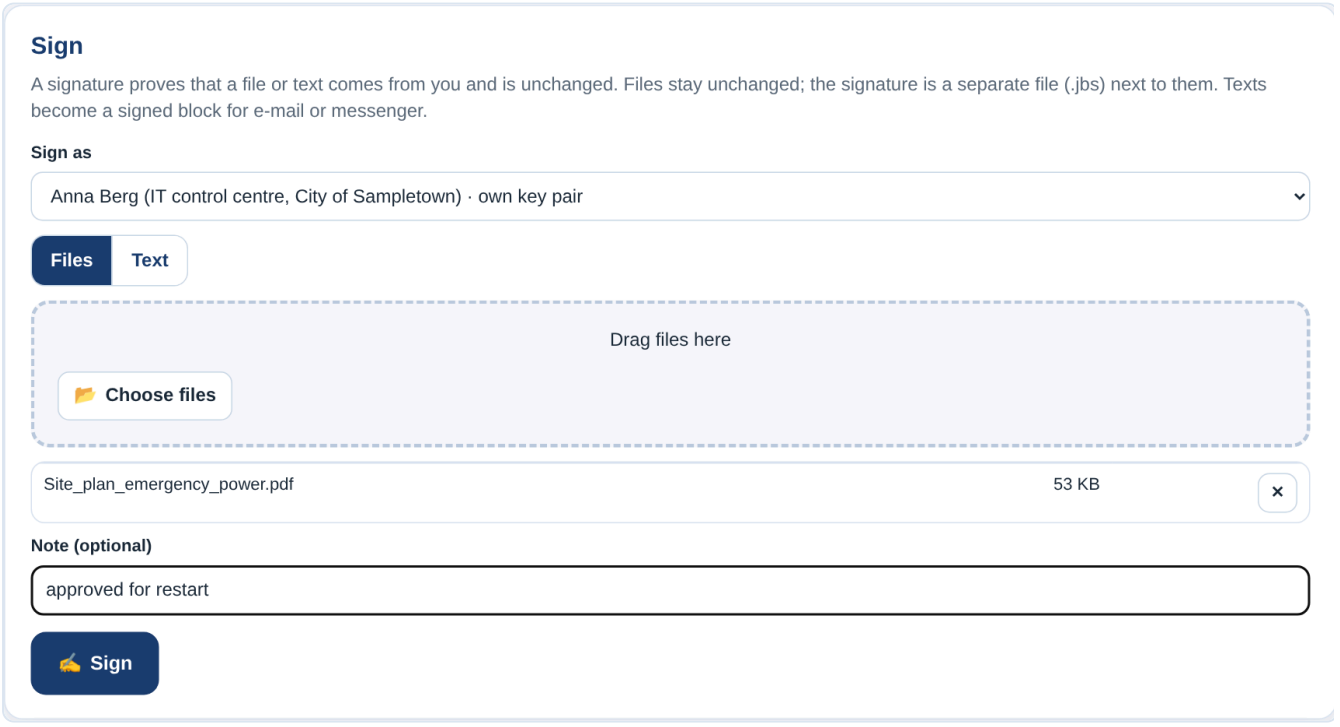


Fig. 13: Signing a file with a note



Fig. 14: Result: signature file

The file itself stays unchanged. Pass on the file and the signature file together.

Signing text

With *Text* a signed text block is created from the readable text and a signature part. The block is passed on completely. Line endings and trailing spaces are normalised so that the signature remains valid after passing through e-mail programs; any change to the content makes it invalid.

Signed text

```
-----BEGIN JUMBO BLOCK SIGNED MESSAGE-----
Approval: pumping station south may be reconnected.
Anna Berg, IT control centre
-----BEGIN JUMBO BLOCK SIGNATURE-----
eyJvYmoiOnsidHlwIjoidGV4dCIsInRleHRfc2hhMjU2IjoiMWQzYTk5NGU5MzBi
NGJjODg5Mjk4MzA1N2JjYmY3ZDRkZDBmYWE1NTJmZmFlZDQ1NTEzODU1ZWY2NTg2
N2UwZiIsInplaXQiOiIyMDI2LTEwLTIyVDE3OjI0OjU3LjAyNFoiLCJzaWduaWVy
ZXIiOiI5NEI5LTlDRjktMjIyOC1FMzJBLTZGNTMtQ0U1RSJ9LCJhbGciOiIxLjIu
ODQwLjEwMDQ1LjQuMy4yIiwid2VydCI6Ik1FVUNJRGlGbGZlZWtwd0tvNWFYZzNs
ZHhHOHJTVVhDeU1Ib01MSzNwNG1pZHhhNkFpRUFvVDNzYmlhTWpIZGVJTi9KeHBa
Zmk3RkFqMGx6bkhYVm5tREt0bzFBc2Y0PSIsImthcnRlIjp7ImthcnRlIjp7InYi
OjEsImlkIjoiMmI5NDU0Y2I5Mjg4NjY3YThmNjVlYjVhYWQyODBkYTgiLCJhcnQi
```

Copy

Pass on the complete block. Recipients verify it in Crypt under “Verify signature”. Any change to the text invalidates the signature.

Fig. 15: Signed text block

10. Verifying signatures

Signatures can be verified by anyone – even without a keyring of their own. With the keyring open, CRYPT additionally recognises contacts and certificates of your own certificate authority.

1. In the *Verify signature* area choose the file and its signature file (.jbs) together or drag them into the field – or, for *Text*, paste the signed block.
2. Click *Verify*.



Fig. 16: Authentic and unchanged – sender is a verified contact

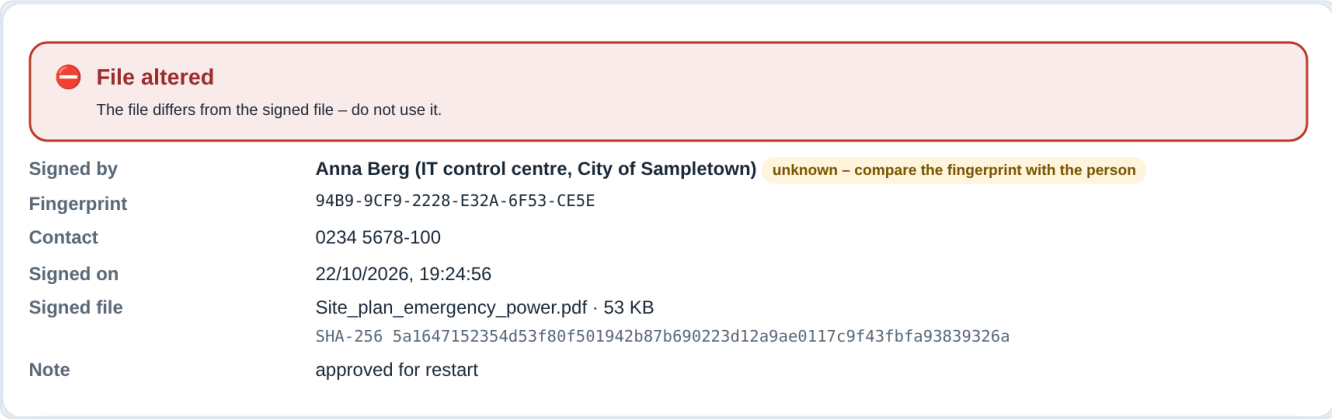


Fig. 17: File altered – do not use

Result	Meaning
Authentic and unchanged	Content unchanged, sender confirmed (own identity, verified contact or trusted CA)
Unchanged – sender not yet confirmed	Content unchanged, but the key is not confirmed. Compare the fingerprint with the person.
File altered / text altered	The content differs from the signed content – do not use it.
Signature invalid	The signature does not belong to this content or was altered.
Certificate forged	The certificate claims to come from a trusted CA but is not signed by it. Do not use the content.

CRYPT also shows the signing time, note, name and checksum of the signed file. For certificates it checks whether the certificate was valid and intended for signing at the time of signing. A certificate that has since expired is only shown as a note if it was valid at the time of signing.

11. Interaction with other programs

Program	Use of CRYPT
JUMBO BLOCK CERT	Personal certificates of your own CA as identities; CA certificate as trusted certificate authority
JUMBO BLOCK REPORT	Signs the state of an incident file with an identity from the keyring. The resulting signature file (.jbs) is verified in CRYPT under *Verify signature* together with the verification file.
JUMBO BLOCK INVENTORY	Encrypts and signs deliveries between units and central office with the keyring; units and central office exchange their public keys (.jbid) for this.

In all cases the programs open the keyring read-only and only for the duration of the session. The keyring file is not changed.

12. Files and formats

File	Content	Passing on
Keyring (.dbm)	Identities with private keys, contacts, certificate authorities – password-encrypted	Own backup copies only
Public key (.jbid)	Name, unit, contact details, public keys, possibly certificate	Pass on freely, compare fingerprint
Encrypted file (.jbc)	One or more files or a text, encrypted, optionally signed	Any route
Signature file (.jbs)	Signature, signing time, note, checksum and name of the file, sender's public key	Together with the file
Text blocks	`-----BEGIN JUMBO BLOCK ENCRYPTED MESSAGE-----` or `SIGNED MESSAGE` with `SIGNATURE`	E-mail, messenger, text file

13. Security and technology

Area	Implementation
Content encryption	AES-256-GCM with a random content key, segments of 4 MiB with individual authentication, final SHA-256 checksum of the entire content
Password	PBKDF2-SHA-256 with 600,000 iterations, at least 12 characters
Recipients	ECDH P-256 with an ephemeral key per recipient and HKDF-SHA-256; content key wrapped per recipient
Signature	ECDSA P-256 for own key pairs; RSA or ECDSA for certificates from CERT
Fingerprint	SHA-256 over the public signing key, the first 96 bits in groups of four
Keyring	AES-256-GCM, PBKDF2-SHA-256 with 600,000 iterations, automatic locking after 15 minutes
Privacy	No network connection, no telemetry, no storage of keys in the browser

Limits

- Protection depends on the password and the device: on a compromised computer, input can be intercepted.
- A signature proves the identity, not the person behind it. Therefore compare fingerprints or use your own CA.
- Revocation of keys is not distributed automatically. If a keyring is lost, inform your contacts and create a new identity.
- When opened over the network, the browser only provides the Web Crypto API via HTTPS. From a USB stick (`file://`) and via jumboblock.org, CRYPT works without restriction.

14. Troubleshooting

Problem	Cause and solution
“None of your identities is among the recipients”	The file was encrypted for other people or for an older identity. Ask the sender to encrypt again with your current public key.
“The password is incorrect”	Check upper/lower case and keyboard layout. Without the correct password the content cannot be opened.
“The encrypted data is incomplete”	File truncated while copying or downloading. Have it transferred again.
“No encrypted text block found”	Block not pasted completely. Copy the header and footer lines as well.
Signature “Unchanged – sender not yet confirmed”	Compare the fingerprint with the person and mark the contact as verified, or store the CA.
“File altered”	Wrong version or file edited after signing. Use the original file.
PKCS#12 file not supported	Older encryption methods in the .p12 file. Export again from CERT or use key (.key) and certificate (.crt).
“The certificate is not intended for signing”	Server certificate instead of personal certificate. Issue a certificate with client/signature usage in CERT.
Keyring checksum does not match	File changed outside CRYPT. The content is nevertheless confirmed by the encryption; save again.
Web Crypto API not available	Use an up-to-date browser; use HTTPS when opened over the network.