CryptSecure
Encryption Client

POST-QUANTUM BY DEFAULT · ENCRYPTED BEFORE IT LEAVES
NEW IN 3.0.5: YOUR KEYS LOCKED TO YOU

User Guide

Desktop application and command line for Windows

Version 3.0.5

November 2026 · Applies to CryptSecure 3.0.x (application releases 3.0.0 to 3.0.5)

Contents

About this guide 5

1 Welcome to CryptSecure 8
- 1.1 What CryptSecure does 8
- 1.2 Why post-quantum, and why now 8
- 1.3 What's new in 3.0.5 9
- 1.4 Four things worth knowing 9

2 Getting started 11
- 2.1 What you need 11
- 2.2 First start 11
- 2.3 Create your key 13
- 2.4 Connect to a broker 13
- 2.5 Publish your public key 14
- 2.6 Your first encrypted message 14

3 The window at a glance 15
- 3.1 How the sidebar is organized 16
- 3.2 The broker status light 16
- 3.3 Themes 17

4 Encrypting and sending 18
- 4.1 Choosing recipients 18
- 4.2 Recipient cards 18
- 4.3 Message and attachments 19
- 4.4 How the protection is chosen 19
- 4.5 Signing 20
- 4.6 Passphrase layer (optional) 20
- 4.7 Delivery options (broker only) 21
- 4.8 Three ways to finish 21
- 4.9 Monthly deposit allowance 22
- 4.10 Sending while offline 23
- 4.11 Several recipients and key types 23

5 Receiving and decrypting 25
- 5.1 Your Inbox 25
- 5.2 Decrypting a file or pasted text 26
- 5.3 Reading the result 26
- 5.4 Saving, acknowledging and clearing 27
- 5.5 When decryption fails 27
- 5.6 If a message cannot be opened right away 28

6 Sent items and chain of custody 29
- 6.1 Checking custody of messages you received 30
- 6.2 What each custody result means 30

7 Keys and contacts 31
- 7.1 Kinds of keys 31
- 7.2 Your keys 31
- 7.3 Contacts 32

7.4 What each kind of key can do 32

8 Safety numbers 34

8.1 Why they matter 34
8.2 What a safety number is 34
8.3 How to compare 34
8.4 When a key changes 35
8.5 The broker's own key 35
8.6 Safety numbers on received messages 35

9 Settings 36

9.1 Brokers 36
9.2 Appearance 36
9.3 Security 36
9.4 About 37

10 The command line 38

10.1 Running commands from CryptSecure.exe 38
10.2 Global options 39
10.3 Command reference 39
10.4 Worked examples 41
10.5 Scripting and automation 52
10.6 Exit codes 53

11 Security model and good practice 55

11.1 What CryptSecure protects against 55
11.2 What it cannot protect against 55
11.3 Good practice 55

12 Your keys: security, backup and recovery 56

12.1 How your keys are protected 56
12.2 What protects you from what 57
12.3 Unlocking your keys 58
12.4 The Protection window 59
12.5 Back up your keys 60
12.6 Moving to a new computer and restoring 61
12.7 Importing and exporting keys 61
12.8 When you lose access for good 62
12.9 Your part in keeping your keys safe 62
12.10 Updating CryptSecure and your keys 63
12.11 Security standards 63
12.12 Questions and answers 64

13 Standards and algorithms 65

13.1 Standards implemented 65
13.2 CNSA 2.0 66
13.3 NIST IR 8547 and the transition timeline 66
13.4 FIPS 140-3 66
13.5 About Internet-Drafts 66

14 Compatibility with CryptSecure 2.6 67

15 Troubleshooting 68

Appendix A Glossary 70

Appendix B Files and locations 71

Appendix C For administrators 72

About this guide

This guide covers CryptSecure Encryption Client 3.0.5 for Windows, the desktop application and the command line built into the same program, CryptSecure.exe. It also applies to the earlier 3.0.x releases (3.0.0 to 3.0.4); the revision history below says what changed in each release. Settings › About shows your application release.

Who it is for

- **Everyone** who sends or receives encrypted files: Chapters 1 to 9 explain the application screen by screen.
- **Power users and administrators** who script or automate: Chapter 10 is a complete command-line reference with real examples and their output.
- **Security, compliance and procurement reviewers**: Chapters 11 to 14 set out the security model, how your keys are protected, the standards CryptSecure implements and compatibility with CryptSecure 2.6.

Conventions

Style	Meaning
Encrypt & send	A button, menu item or label exactly as it appears in the application.
Settings › Brokers	A path through the interface: open Settings, then go to the Brokers section.
`cs keys list`	Something you type, or a file name.
Dark panels	A command-line session. Lines that start with C:\Work> are what you type; the rest is the program's real output.

The screenshots and command output in this guide were produced from the 3.0.x release builds (the key protection screens in Chapters 2 and 12 from 3.0.5; some earlier title bars and status lines show 3.0.0 to 3.0.4, everything else is the same; the broker examples in Chapter 10.4 were produced with 3.0.3) using example people (Alice Martin, Bob Chen, Carol Diaz) and an example broker at broker.example.com. Names, addresses, safety numbers, dates and file sizes on your screen will differ.

Revision history

Guide version	Date	Application	Changes
3.0.0	November 2026	3.0.0	First edition for CryptSecure 3.
3.0.0 rev. 2	November 2026	3.0.0	Microsoft Store edition: starting the app, moving keys from the 2.6 Store app, Store data folder, cs helper and command alias. Deposit allowance and size limits described as broker settings. NIST IR 8547 final and EO 14412 dates. Publisher, privacy and support details.
3.0.0 rev. 3	November 2026	3.0.0	Removed build instructions (the application is supplied ready to install). Command reference completed: --mode, --sender, passphrase options, --caveats and --dist-statement. Proof of delivery summary in Chapter 6. Labels aligned with the application.

Guide version	Date	Application	Changes
3.0.0 rev. 4	November 2026	3.0.0 (Store package 3.0.1)	Import older key… window (name, email, RSA .pem pair checked against each other, ML-KEM .bin pair, passphrase-protected .pem). One email address can have several keys; new messages use the strongest. Email column and Set email…. Commands keys import-old and keys set-email. Several-recipient and several-key behaviour documented with real output.
3.0.0 rev. 5	November 2026	3.0.2	A key picked with Choose key is the key used by every button, also when the address has another key. When a key counts as changed (Chapter 8.4); confirming a new key keeps keys you imported yourself. A named key wins over the same typed address; a chosen key for someone with no broker account is refused with an explanation.
3.0.0 rev. 6	November 2026	3.0.2	Version aligned across the products: the application reports 3.0.2 (Microsoft Store package 3.0.2.0, program file cryptsecure_v3.0.2.exe); the guide keeps its 3.0.0 title and applies to 3.0.x. CryptSecure for Outlook & Word 3.0.3 needs this application at 3.0.3 or newer. All CryptSecure products are released with the same version number.
3.0.0 rev. 7	November 2026	3.0.3	Checked against the broker's own code. Verify custody works with the real broker (3.0.2 showed "Could not verify the chain: no confirmation") and lists the deposit's events. Every recipient of a deposit to several people can collect it (3.0.2 let the broker erase it after the first). Pickup is checked against the broker's custody hash; acknowledge-before-release is handled; Inbox and Sent show the broker's real status (Picked up 1/2, Delivered, file erased, Available from …). Classification uses the broker's markings. New: Verify a document code… (cs verify-code). Microsoft Store package 3.0.3.0, program file cryptsecure_v3.0.3.exe. CryptSecure for Outlook & Word 3.0.3 (released together; every CryptSecure product now carries the same version) needs this release or newer.
3.0.0 rev. 8	November 2026	3.0.4	Audited the way people use it. Choose key ▾ adds to To (keys can be strung together; nothing typed is removed). CryptSecure 2.6 keys can be mixed with anyone: one part per 2.6 key (4.11, with the tested combination table). A typed address and another key of the same person are both used. Inbox › Verify custody for messages you received; custody results state the deposit's state (6.1, 6.2). A download that cannot be opened yet is kept until it opens (5.6). New 7.4: what each kind of key can do. Microsoft Store package 3.0.4.0, program file cryptsecure_v3.0.4.exe. Sign as keeps the key you pick, and no click is lost to the automatic once-a-minute broker check. CryptSecure for Outlook & Word 3.0.4 (released together) needs this release or newer.

Guide version	Date	Application	Changes
3.0.5 (rev. 9)	November 2026	3.0.5	Private keys protected. New Chapter 12, Your keys: security, backup and recovery: protection levels, unlocking, the Protection window, recovery packages and codes, restoring on another computer, importing and exporting, when access is lost for good, the owner's part, updating and going back, security standards. First start asks how to protect your keys. Back up my keys replaces the unprotected key backup. Command line: backup make, check and restore; keys protection, lock, activity and related commands; exit codes 16 to 18. Chapters 12 to 14 are now 13 to 15.

Legal notices

Copyright. © 2026 Opey's IT. All rights reserved.

Privacy. The application contains no telemetry, analytics or advertising and contacts no server except the brokers you add. What a broker can see is listed in Chapter 11.2.

Support. steveoppenheim@opey.org · www.opey.org/opeyit/cryptsecure/. Include the output of cs doctor (Chapter 15).

Trademarks. CryptSecure and the CryptSecure padlock mark are trademarks of Opey's IT. Microsoft, Windows, PowerShell and Windows Credential Manager are trademarks of the Microsoft group of companies. WordPress is a trademark of the WordPress Foundation. Let's Encrypt is a trademark of the Internet Security Research Group. Other names may be trademarks of their respective owners. Their use here is for identification only and does not imply endorsement.

Standards references. References to NIST, NSA and IETF publications describe the algorithms CryptSecure implements. They are not a statement that CryptSecure has been certified, validated, approved or endorsed by NIST, NSA, the IETF or any other body. CryptSecure 3.0.5 has not undergone FIPS 140-3 validation (see Chapter 13).

Export and import. CryptSecure contains strong encryption. Its export, re-export, import and use may be regulated in your country and in the destination country, including under the U.S. Export Administration Regulations. You are responsible for complying with the laws that apply to you.

No warranty. This guide is provided for information. It is not legal, regulatory or compliance advice. The software and this guide are provided "as is", to the extent permitted by law and subject to the license terms that accompany your copy of CryptSecure.

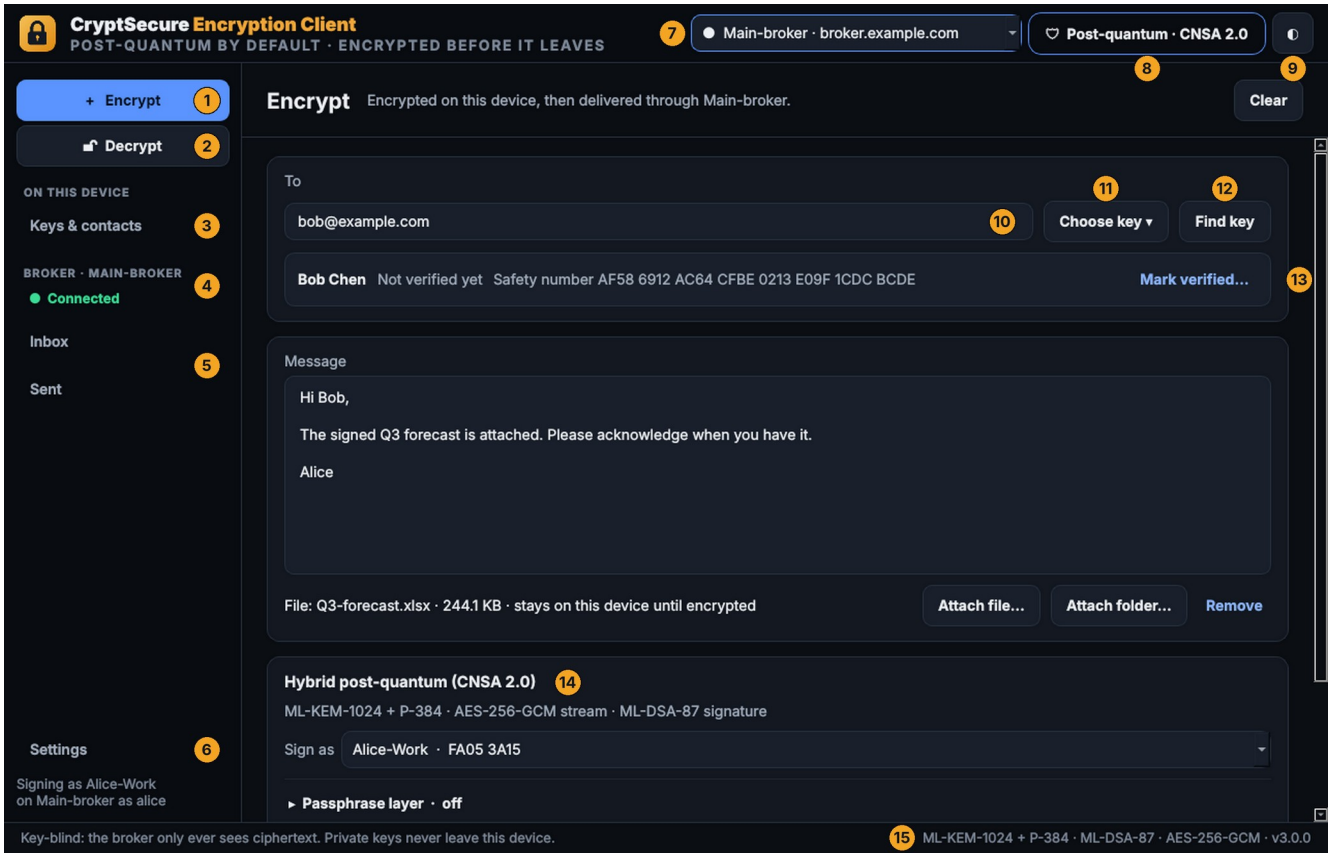
1 Welcome to CryptSecure

CryptSecure locks your files and messages on your own computer, before anything leaves it, using the newest post-quantum encryption standards. Only the people you choose can open them. Not the network, not the server, not us.

1.1 What CryptSecure does

CryptSecure Encryption Client turns a message, a file or a whole folder into an **envelope** that only your chosen recipients can open. It also signs the envelope, so recipients can confirm it came from you and was not changed on the way.

You can deliver an envelope however you like: save it and send it by email, put it on a USB drive, or paste it as text. You can also hand it to a **CryptSecure Broker**, a secure drop-box server that delivers it, records a tamper-evident chain of custody and tells you when it was picked up. The broker only ever stores ciphertext. It never has your keys and cannot read what you send.



The CryptSecure window. The numbered parts are explained in Chapter 3.

1.2 Why post-quantum, and why now

Most encryption in use today relies on RSA or elliptic curves. A large enough quantum computer could break both. Anything intercepted and stored today could then be read later, an approach known as “harvest now, decrypt later”. In August 2024 NIST published the first post-quantum encryption and signature standards, FIPS 203 (ML-KEM) and FIPS 204 (ML-DSA). In October 2026 its final transition report, NIST IR 8547, scheduled quantum-vulnerable public-key algorithms for deprecation after 2030 and disallowance after 2035.

CryptSecure 3 uses ML-KEM and ML-DSA by default. It pairs ML-KEM with a proven classical algorithm in a **hybrid** design, so a message stays protected unless both are broken. You do not need to configure anything to get this protection.

1.3 What's new in 3.0.5

A stolen laptop or a copied key folder no longer gives anyone your keys.

CryptSecure 3.0.5 locks your private keys to your computer and to you:

- **Your private keys are locked.** Every private key is kept encrypted on disk. It opens only with your unlock method: a security key, this computer's TPM (with or without a PIN) or a passphrase. A copy of your key folder is useless on its own (Chapter 12).
- **Unlocking ends by itself.** In the CryptSecure window one unlock lasts 15 minutes (you choose 0 to 60). Your keys also lock when the screen locks, when the computer sleeps, and when you choose Lock my keys now.
- **You see who asks.** The unlock window names the program that wants your key (CryptSecure, Outlook, Word or the command line), and Activity lists every use of your keys.
- **Back up my keys.** One recovery package holds every key and your broker passwords. It opens only with a recovery code shown to you once, and restores on Windows, Linux and Raspberry Pi.
- **Updating takes care of itself.** At the first start, everything is moved under protection and checked. You can still go back to 3.0.4 (Chapter 12.10).
- **Everything from before still opens.** Keys, backups and messages from CryptSecure 2.6 and from 3.0.0 to 3.0.4.

Introduced in 3.0.0 (3.0.2 to 3.0.4 added the improvements listed in the revision history):

- **Post-quantum by default.** Hybrid ML-KEM key encapsulation on every envelope, and ML-DSA-87 signatures on everything you send with a CryptSecure 3 key.
- **Several recipients, one envelope.** Encrypt a message and its files once for a whole group, in a file or through a broker. Each person opens it with their own key; nobody else can.
- **One email, several keys.** Keep your new post-quantum key next to your older keys on the same email address. Old messages keep opening; new ones always use the strongest key.
- **Signed and tamper-evident.** Recipients see who signed, whether you are a verified contact and whether anything was altered.
- **Built-in broker delivery.** Look up recipients' keys, send in one step, pick up, acknowledge and verify custody from inside the application.
- **Large files.** Folders and very large files are encrypted and decrypted in the background without freezing the window.
- **Command line.** Every function is scriptable, with JSON output and meaningful exit codes.
- **Fully compatible with CryptSecure 2.6.** Every 2.6 file opens and your 2.6 keys keep working. Keys kept elsewhere, including passphrase-protected .pem files, come in with Import older key… (Chapter 2.2).

1.4 Four things worth knowing

	In practice
Encrypted before it leaves	Encryption and decryption happen only on your computer. Plaintext never goes to the broker.

	In practice
Private keys stay home	Your private key never leaves this computer except in a recovery package you make yourself, which opens only with your recovery code (Chapter 12). Only your public key is shared.
Check who you are talking to	A 32-character **safety number** lets you confirm, by phone or in person, that a key really belongs to the person you think it does (Chapter 8).
Works without a broker	Encrypting, decrypting and managing keys need no server and no internet connection.

2 Getting started

2.1 What you need

Item	Details
Computer	Windows 10 or Windows 11, 64-bit.
The program	CryptSecure 3.0.5 (or newer), either from the Microsoft Store (listed as CryptSecurev3.0) or as the program file CryptSecure.exe supplied by your organization. The desktop application and the command line are the same program. CryptSecure for Outlook & Word 3.0.5 needs this application at 3.0.5 or newer. All CryptSecure products are released with the same version number.
Memory	Large files are processed in memory. Allow roughly three times the size of the largest file you will encrypt or decrypt.
A broker account (optional)	Only needed to send and receive through a broker. Your broker operator gives you the address, your username and instructions for creating an application password.

2.2 First start

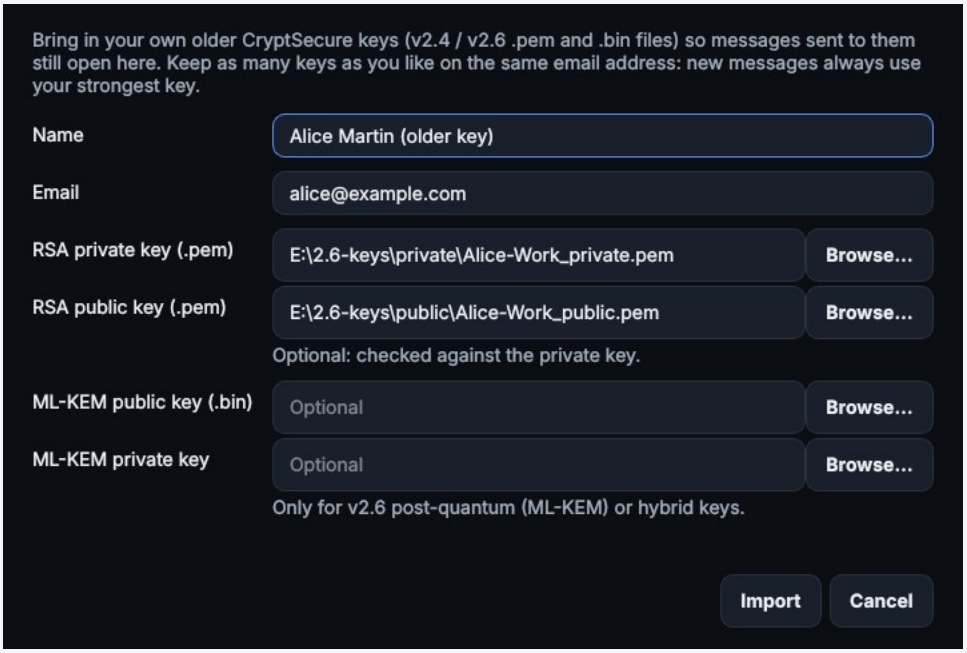
1. Start CryptSecure: choose **CryptSecurev3.0** in the Start menu if you installed it from the Microsoft Store, or double-click **CryptSecure.exe** if your organization supplied the program file. If Windows shows a security prompt for a newly downloaded program, confirm it comes from your organization or a source you trust before you allow it.
2. If you used the CryptSecure 2.6 program file on this computer, your keys are found automatically and you are told that they still work. If you used the Microsoft Store edition of 2.6, bring your keys over once, as described in the box below.
3. You are asked whether to create a CryptSecure 3 key. Choose **Yes**. You can also do this later under **Keys & contacts › New key…**.

> NOTE
>
> Run as a program file, CryptSecure 3 keeps its data in the same folder as version 2.6: `%APPDATA%\CryptSecure`. Nothing is moved or deleted. Before writing anything new, CryptSecure 3 makes one backup copy of your 2.6 key list (`key_registry.v2.6-backup.json`).

MOVING FROM THE MICROSOFT STORE EDITION OF 2.6

Each Microsoft Store app keeps its own private copy of its data, so CryptSecure 3 from the Store cannot see the keys saved by CryptSecure 2.6 from the Store. Import them once:

1. In CryptSecure 3, open **Keys & contacts** and choose **Import older key…**. The **Import an older key** window opens with your name and email address already filled in.
2. Next to **RSA private key (.pem)**, choose **Browse…**. Click in the address bar, paste the folder below and press **Enter**:
`%LOCALAPPDATA%\Packages\0peyIT.CryptSecurev2.6_8pezjp7a9jcbr\LocalCache\Roaming\CryptSecure\keys`
3. Open the **private** folder, select your key (its name ends in **_private.pem**) and choose **Open**. Optional: under **RSA public key (.pem)**, choose the matching file in the **public** folder (ends in **_public.pem**); CryptSecure checks that the two belong together.
4. For a hybrid or ML-KEM key, also choose its two ML-KEM files: **_mlkem_public.bin** (public folder) and **_mlkem_private.bin** (private folder). Leave them empty for an RSA key.
5. Choose **Import**. Repeat for each key you want to keep. If a key is protected by a passphrase, CryptSecure asks for it.

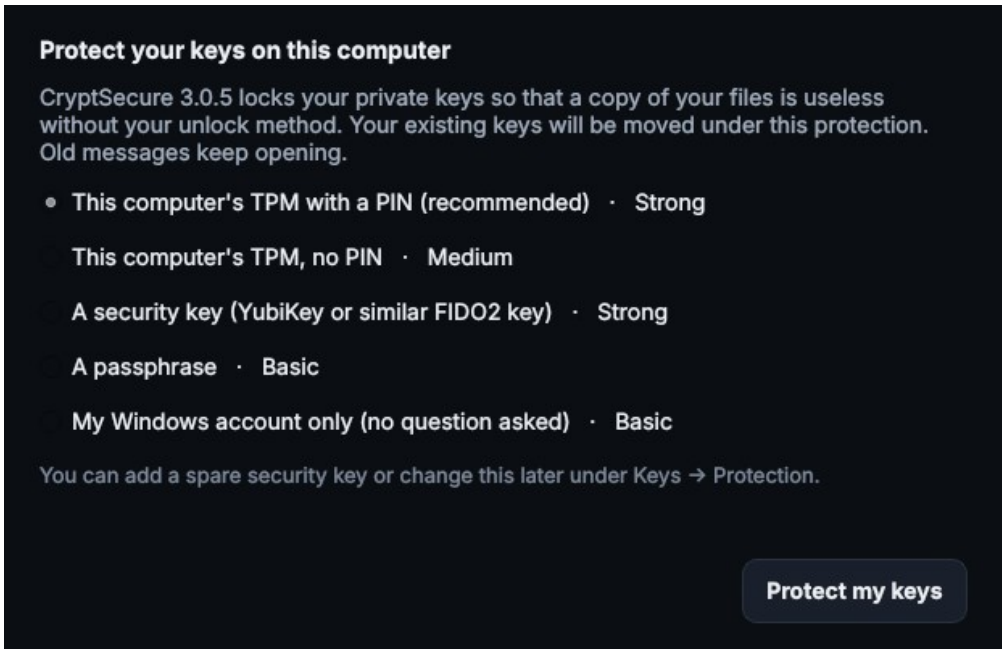


The Import an older key window, with an RSA key pair chosen.

Your 2.6 app and its keys are left untouched. Once a key is imported, everything encrypted to it opens in CryptSecure 3. **One email, several keys:** you can keep your new CryptSecure 3 key and any number of older keys on the same email address. Old messages open with whichever key they were sent to; new messages to that address always use the strongest key, CryptSecure 3 first.

Protect your keys

At the first start of CryptSecure 3.0.5, before anything else, CryptSecure asks how your keys are protected on this computer. The recommended choice is selected; if you update from an older version, your keys are moved under this protection and old messages keep opening. Chapter 12 explains each choice.



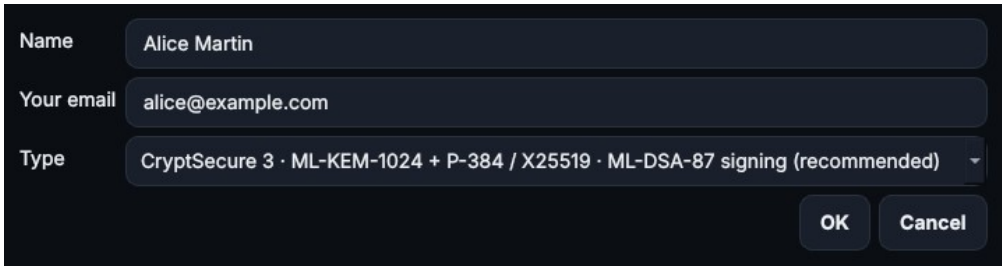
First start on a Windows computer with a TPM.

If you choose a passphrase, type it twice. It needs at least 12 characters. If you forget it, only your recovery package can bring your keys back (Chapter 12).

2.3 Create your key

Your key has two halves. The **private key** stays on this computer and is what opens messages sent to you and signs what you send. The **public key** is what you share so people can encrypt to you.

1. Open **Keys & contacts** and choose **New key…**.
2. Enter your name and your email address. Use the address your broker account uses, so people can find you.
3. Leave **Type** on **CryptSecure 3 … (recommended)** and choose **OK**.



The New key dialog.

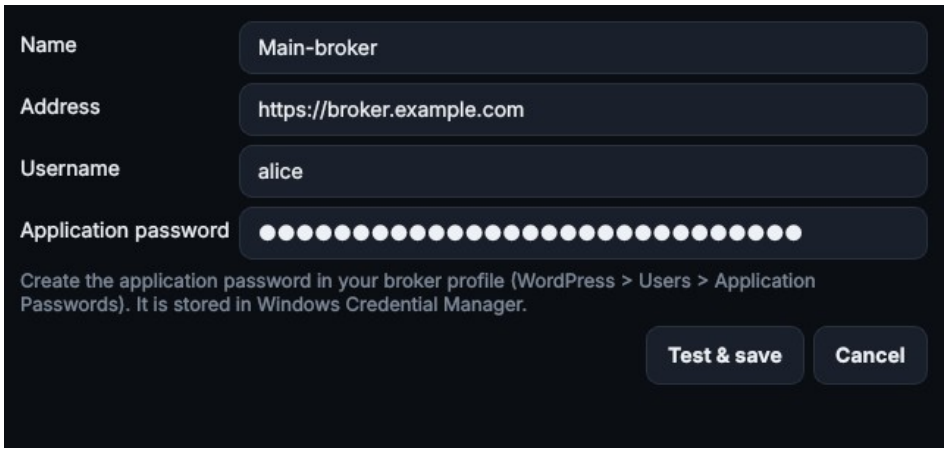
Key generation takes a second or two. The first CryptSecure 3 key you create becomes your **signing key**, the one used to sign everything you send.

2.4 Connect to a broker

Skip this step if you will only save, copy or email encrypted files yourself.

1. In your broker's web portal, create an **application password**: in WordPress, go to Users › Profile › Application Passwords. An application password is a separate password just for CryptSecure. You can revoke it at any time without changing your main password.

2. In CryptSecure, choose **+ Add a broker** in the sidebar, or go to **Settings › Brokers › Add broker…**.
3. Enter a name you will recognize, the broker address (it must start with https://), your username and the application password, then choose **Test & save**.



The Add broker dialog.

CryptSecure checks the connection and your login, reads the broker's security policy and **pins** the broker's signing key, so it can warn you if that key ever changes. Your application password is kept with your protected keys (Chapter 12), not in a plain file.

> **SECURITY**
>
> Never enter your main website password in CryptSecure. Use an application password, so the CryptSecure login can be revoked on its own if your computer is lost.
>
> CryptSecure refuses connections without HTTPS and refuses certificates that are expired or untrusted. There is no option to bypass this, by design.

2.5 Publish your public key

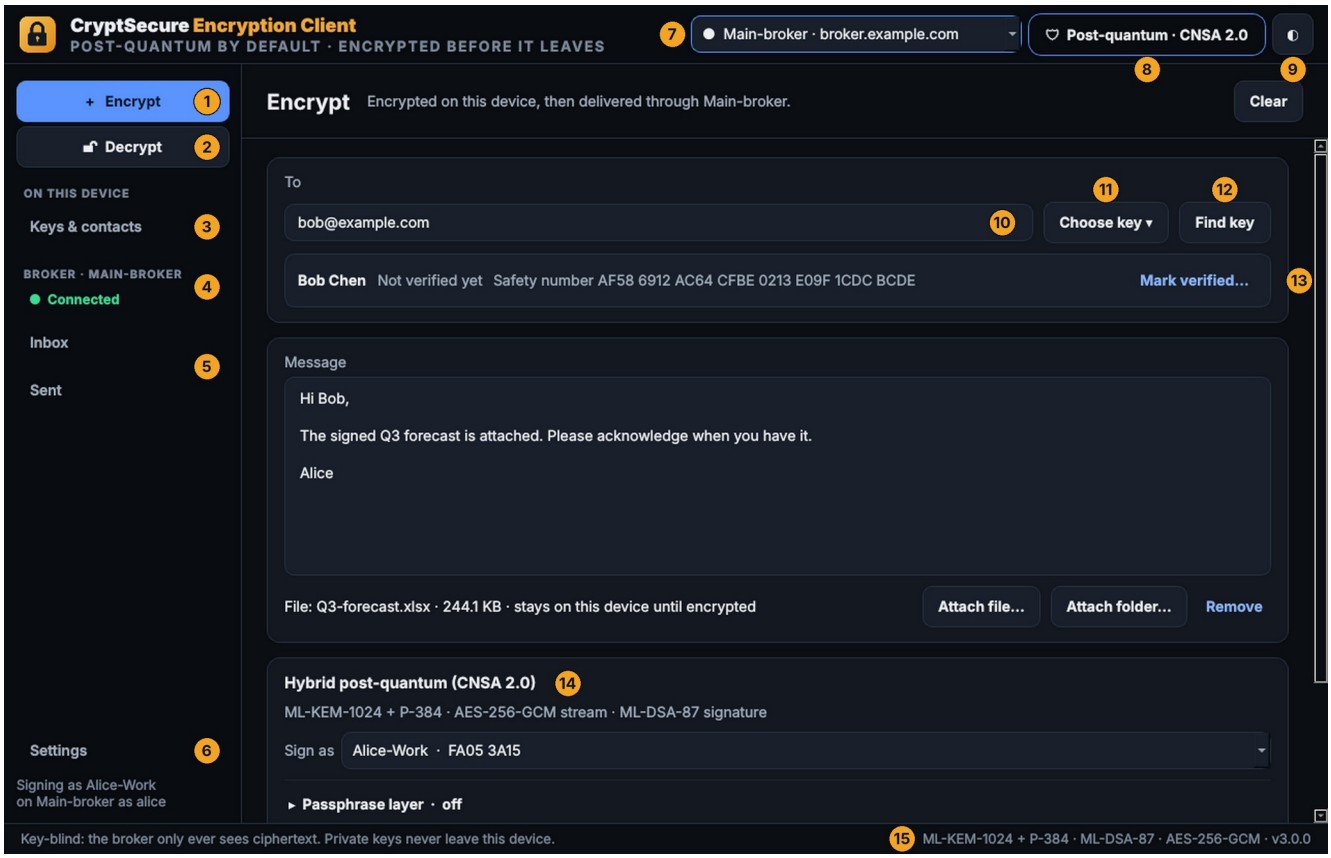
So that others can find you, go to **Keys & contacts**, select your key under **My keys** and choose **Publish to broker**. The Published column then shows **On** followed by the broker's name. When you create a key while a broker is connected, CryptSecure offers to publish it for you.

2.6 Your first encrypted message

1. Choose **+ Encrypt** in the sidebar.
2. In **To**, type the recipient's email address and choose **Find key**. Or choose **Choose key ▾** to pick someone already in your contacts.
3. Check the card that appears under the address (Chapter 4.2 explains each kind). The first time, compare safety numbers with the person (Chapter 8).
4. Type your message, or choose **Attach file…** or **Attach folder…**.
5. Choose **Encrypt & send** to deliver it through the broker, or **Encrypt only (save file)…** to save it.

That's it. The recipient opens it with **Decrypt**, or from their **Inbox** if you sent it through a broker.

3 The window at a glance



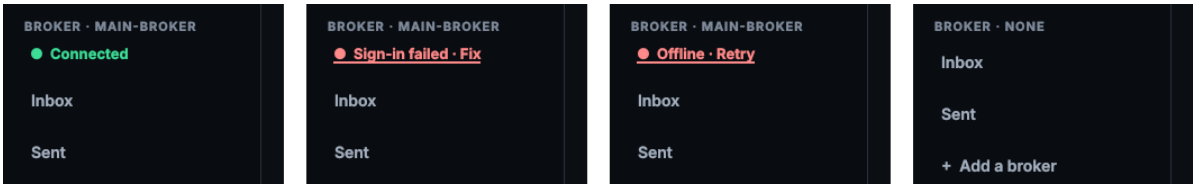
#	Part	What it does
1	**+ Encrypt**	Opens the Encrypt page: write, attach, choose recipients, then send, save or copy.
2	**Decrypt**	Opens an encrypted file or pasted text. Works without a broker.
3	**Keys & contacts**	Your keys, your contacts and their safety numbers. Listed under **On this device** because it all lives on your computer.
4	**Broker section and status light**	The name of the active broker and whether CryptSecure can reach it (see below).
5	**Inbox** and **Sent**	What is waiting for you on the active broker, and what you have sent. The blue badge counts deposits ready to pick up.
6	**Settings**	Brokers, appearance, security options, key folder and version information.
7	Broker switcher	Switches the active broker when you use more than one.
8	Protection badge	Shows **Post-quantum** and the broker's security profile, or a warning when a message cannot be post-quantum.
9	Theme button	Switches between dark and light appearance.
10	**To**	Recipients: email addresses, contact names or key names, separated by commas.
11	**Choose key ▾**	Picks a recipient from your own keys and contacts. Choosing again replaces the recipient.
12	**Find key**	Looks the address up in the broker's directory.

#	Part	What it does
13	Recipient card	Who the key belongs to, its safety number and whether you have verified it.
14	Protection	The encryption that will be used. It is chosen automatically for your recipients and broker.
15	Footer	The algorithms in use and the version number.

3.1 How the sidebar is organized

The sidebar is grouped by **where things happen**, the same way an email program lists folders under each account. Everything under **On this device** works offline and never leaves your computer. Everything under **Broker** lives on the server named in the heading and changes when you switch brokers. Encrypt and Decrypt sit at the top because they work in both places.

3.2 The broker status light

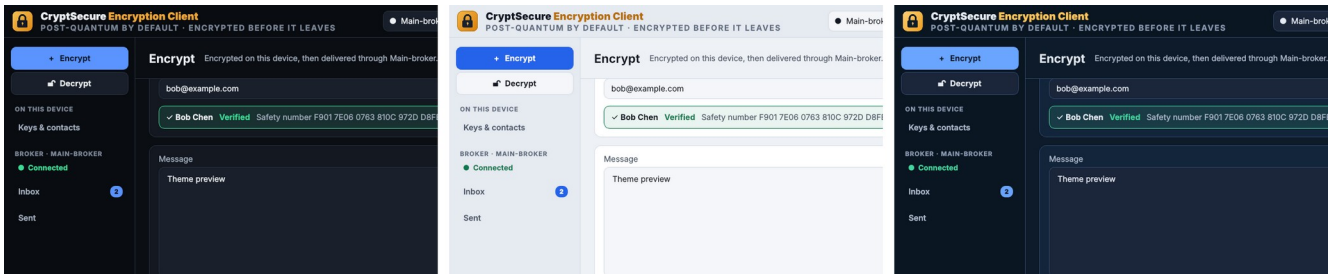


Connected, sign-in failed, offline, and no broker set up.

You see	Meaning	Click it to
● **Connected** (green)	The broker answered and your login works.	Check again.
○ **Connecting…** (amber)	A check is in progress.	
● **Offline · Retry** (red)	The broker cannot be reached: no internet connection, or the server is down.	Try again.
● **Sign-in failed · Fix** (red)	The broker rejected your application password.	Open Settings to update it.
● **Certificate problem · Details** (red)	The broker's security certificate is expired, untrusted or for a different address.	Open Settings. Hover for the full explanation.
● **Broker key changed · Details** (red)	The broker's pinned signing key is different from the one you first connected to.	Open Settings. Contact the broker operator before continuing.
No light; **BROKER · NONE**	No broker is set up. This is normal if you only encrypt locally.	Use **+ Add a broker**.

The light re-checks every minute and also turns red the moment a broker action fails. Hover over it to see the full reason and when it was last checked. Every state has a word next to the color, so it can be read without relying on color alone.

3.3 Themes



Dark, Light and Blue themes. System follows your Windows setting.

Change the theme in **Settings › Appearance**, or cycle dark and light with the ◐ button at the top right. The CryptSecure logo is drawn to suit each theme.

4 Encrypting and sending

Everything you encrypt is processed on your computer. The Encrypt page tells you, in its subtitle, where the result will go: “Encrypted on this device, then delivered through Main-broker”, or, with no broker, “Save it or copy it”.

4.1 Choosing recipients

Type recipients in **To**, separated by commas. Each can be an email address, a contact’s name or a key name. Then:

- **Find key** looks each address up in the active broker’s directory and shows a card for each recipient.
- **Choose key ▾** lists your own keys and your contacts, including keys carried over from CryptSecure 2.6. Picking an entry ADDS it to the end of To and keeps everything already there, so you can string several keys together, one pick after another. Picking a key that is already in To does not add it twice; Esc closes the list without changing anything. The key you pick is the key used.



Choose key lists your own keys first, then your contacts.

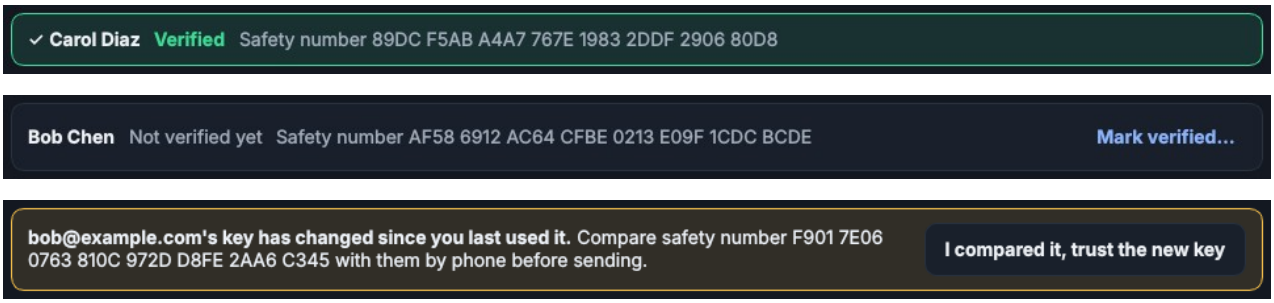
To encrypt for yourself (for example, to store a file safely), choose one of your own keys.

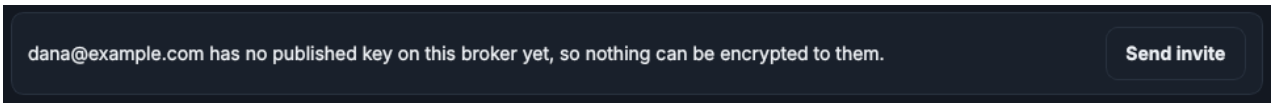
Several people at once. Type several recipients in **To**, separated by commas, pick several keys with **Choose key ▾**, or mix the two. Everyone with a CryptSecure 3 key shares one envelope, and each person opens it with their own key. Anyone not on the list cannot. Add yourself to keep a copy you can open. A recipient with a CryptSecure 2.6 key gets a part of their own (see 4.11, which lists every combination). Through a broker, every person on the list can collect it: the broker counts it as delivered, and may erase the file, only after all of them have (the command line’s --quorum N changes this). To remove someone, delete them from To.

When an address has several keys. Someone may have a new CryptSecure 3 key and an older 2.6 key on the same address. When you type the address, CryptSecure encrypts to the strongest: CryptSecure 3 first, then hybrid, ML-KEM and RSA; a verified key before an unverified one; then the newest. To use one particular key instead, pick it with **Choose key ▾**: the key you pick is the key used by **Encrypt & send**, **Encrypt only** and **Encrypt & copy as text**, and the message is delivered to that key’s email address. If you type an address AND pick another key of the same person, both are used (each gets the message). The same key listed twice counts once.

4.2 Recipient cards

A card appears under To for each recipient. Read it before you send.





From top: verified, not verified yet, key changed, and no published key.

Card	Meaning	What to do
✓ **Verified** (green)	You compared safety numbers with this person and marked the key verified.	Nothing. Send.
Not verified yet	The key came from the broker directory or a file. It encrypts correctly, but you have not confirmed whose it is.	For anything sensitive, compare safety numbers first, then choose **Mark verified…**.
"…'s key has changed since you last used it" (amber)	The broker no longer offers the key it gave you before for this address. They may have a new computer, or someone may be impersonating them. A key you picked with Choose key ▾ is not affected.	Sending stops. Call them and compare the new safety number. Only then choose **I compared it, trust the new key**.
No published key	This address has no key on the broker yet.	Choose **Send invite** to ask them to join.

4.3 Message and attachments

- Type or paste your message in **Message**. You can drop a file onto the message box to attach it.
- **Attach file…** attaches one file. **Attach folder…** attaches a whole folder. It is packed into a single compressed archive and unpacked again for the recipient.
- The message box holds up to **5 MB** of text, which keeps the application responsive. For anything larger, put it in a file and attach it. Attachments have no size limit in CryptSecure itself, but a broker sets its own maximum deposit size: 128 MB by default, and lower on some plans. `cs broker test` shows the limit for your account.
- Attachments stay on your computer until they are encrypted, as the line under the box reminds you.

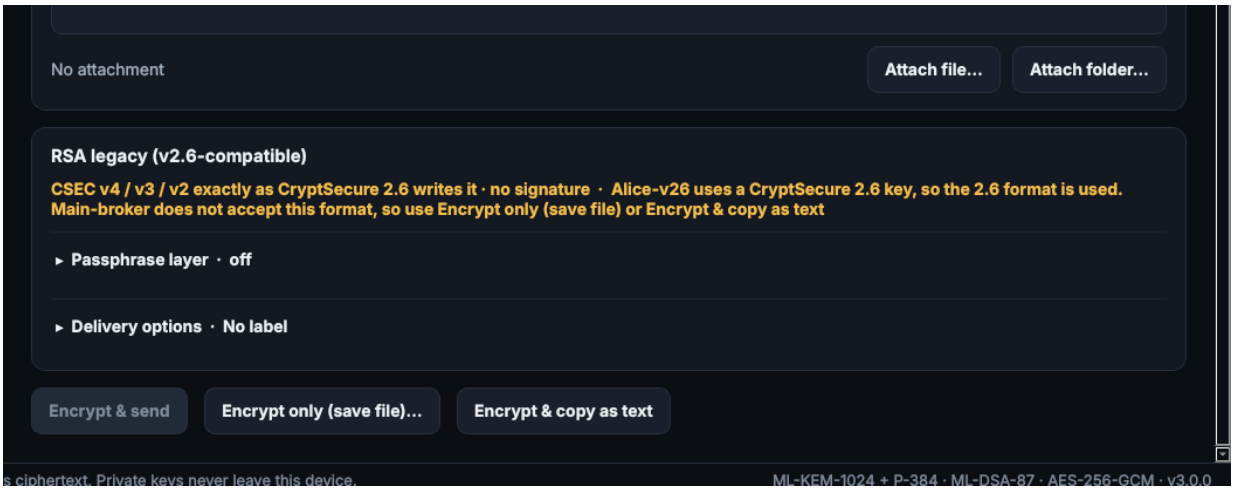
4.4 How the protection is chosen

You do not pick an encryption method. CryptSecure chooses the strongest one that every recipient's key supports and the active broker accepts, and shows it in the Protection panel. This avoids combinations that cannot work.

Protection shown	Algorithms	When it is used
Hybrid post-quantum (CNSA 2.0)	ML-KEM-1024 + ECDH P-384, HKDF-SHA-384, AES-256-GCM, ML-DSA-87 signature	The default with CryptSecure 3 keys. Uses the CNSA 2.0 algorithm set (Chapter 13).
Hybrid post-quantum (X-Wing)	ML-KEM-768 + X25519, HKDF-SHA-256, AES-256-GCM, ML-DSA-87 signature	When a broker's policy prefers it, for interoperability with commercial systems.
Pure ML-KEM-1024	ML-KEM-1024 only, HKDF-SHA-384, AES-256-GCM, ML-DSA-87 signature	Where a broker's policy does not allow hybrid key exchange.
RSA legacy (v2.6-compatible)	The CryptSecure 2.6 formats: RSA-OAEP and/or ML-KEM-1024 with AES-256-GCM. Not signed.	Only for a single recipient whose key is a 2.6 key, so they can open it in 2.6.

If a recipient only has an **RSA** key from CryptSecure 2.6, the badge at the top of the window changes to ⚠️ **Not post-quantum** and the panel says why. Ask that person to create a CryptSecure 3 key.

Some brokers do not accept deposits in the 2.6 format. For a 2.6 recipient on such a broker, **Encrypt & send** is switched off and the panel explains why. You can still use **Encrypt only (save file)...** or **Encrypt & copy as text** and deliver the file yourself.



A CryptSecure 2.6 recipient on a broker that refuses the 2.6 format: save or copy instead.

4.5 Signing

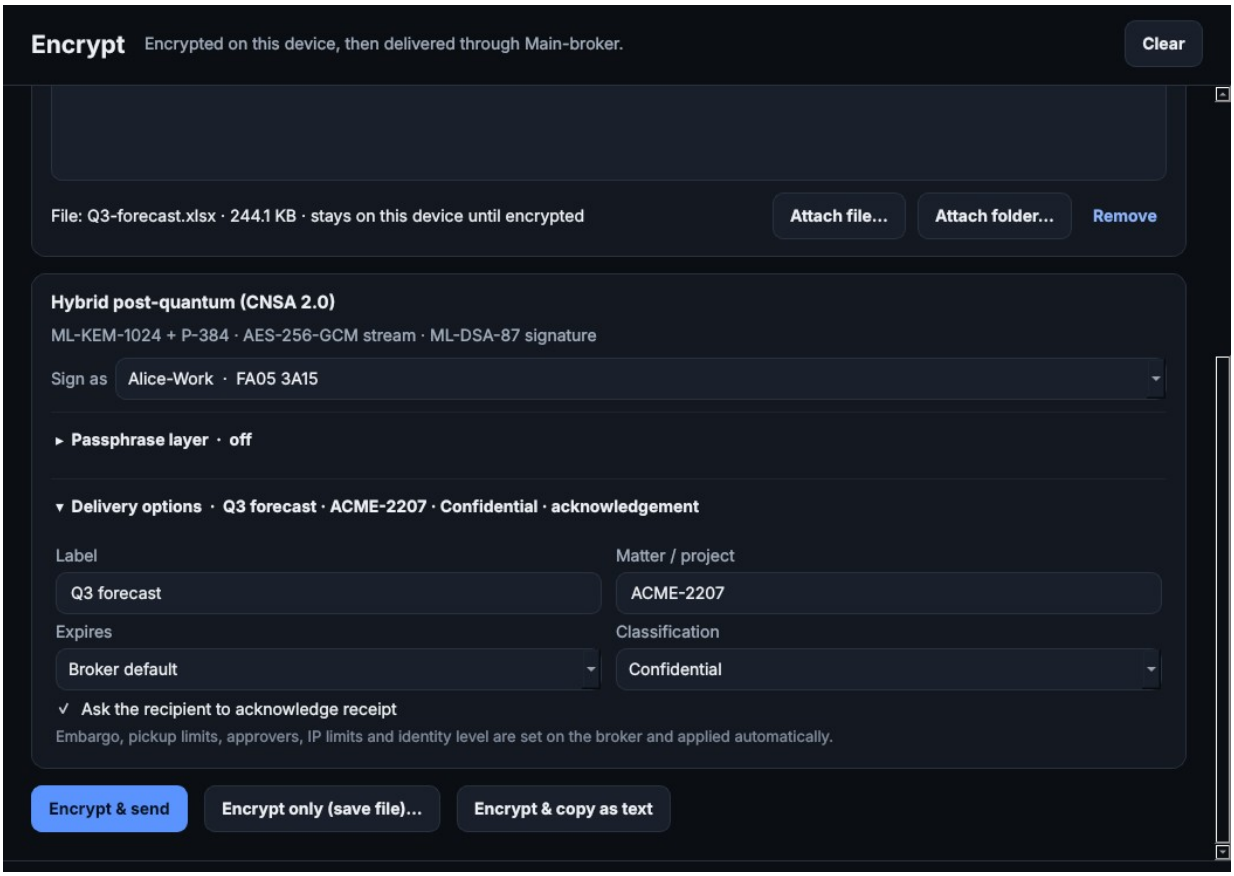
Sign as shows the key that signs the message and the first characters of its safety number. Everything you send with a CryptSecure 3 key is signed with ML-DSA-87, so recipients can confirm it came from you. If you have no CryptSecure 3 key yet, Sign as says so and envelopes are sent unsigned. To change your default signing key, select another of your keys in Keys & contacts and choose **Use for signing**. To sign with another of your keys just this time, pick it in Sign as: it stays picked until you close CryptSecure. Messages in the 2.6 format cannot be signed, so Sign as is hidden for them.

4.6 Passphrase layer (optional)

Open **Passphrase layer** to add a password on top of the encryption. The recipient then needs both their key and the passphrase. CryptSecure derives a key from the passphrase with Argon2id, deliberately slow to resist guessing, and shows a strength meter as you type.

> IMPORTANT
>
> Share the passphrase through a different channel than the file, for example by phone. There is no way to recover a forgotten passphrase.
>
> Some brokers do not accept passphrase-protected deposits. The panel tells you before you send. You can still save such a file and deliver it yourself.

4.7 Delivery options (broker only)



Delivery options open, with the three ways to finish below.

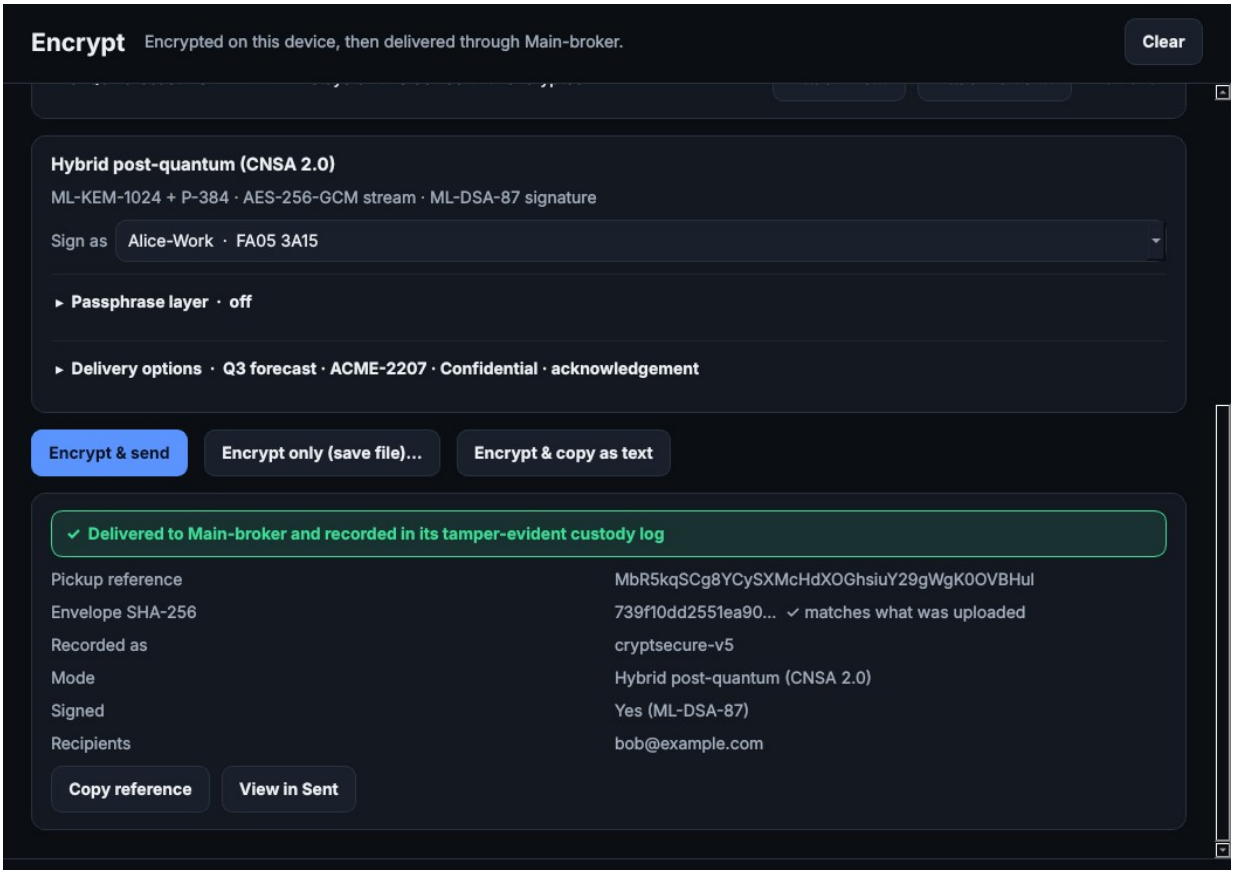
Option	Purpose
Label	A short title the recipient sees in their Inbox, for example “Q3 forecast”. Labels are not encrypted, so keep them non-sensitive.
Matter / project	A reference such as a case or project number, used for filtering and custody records. Not encrypted.
Expires	Broker default, 24 hours, 7 days or 30 days. After this the broker deletes the deposit.
Classification	None, Public, Internal, Confidential, CUI or CUI//SP (Specified): the broker’s own markings. Recorded with the deposit and printed on its custody records.
Ask the recipient to acknowledge receipt	The recipient is asked to confirm receipt, and you see when they did. Brokers from version 1.11 release the file only after the recipient acknowledges.

Embargoes, pickup limits, approvers, IP restrictions and identity requirements are set by the broker operator and applied automatically.

4.8 Three ways to finish

Button	Result
Encrypt & send	Encrypts, uploads to the active broker and confirms that the broker stored exactly what you sent, by comparing SHA-256 fingerprints.

Button	Result
Encrypt only (save file)...	Saves the envelope as a file (.csec, or .txt for the 2.6 format). Nothing is sent.
Encrypt & copy as text	Copies the envelope as text you can paste into an email or chat. Results over 5 MB are saved to a file instead.

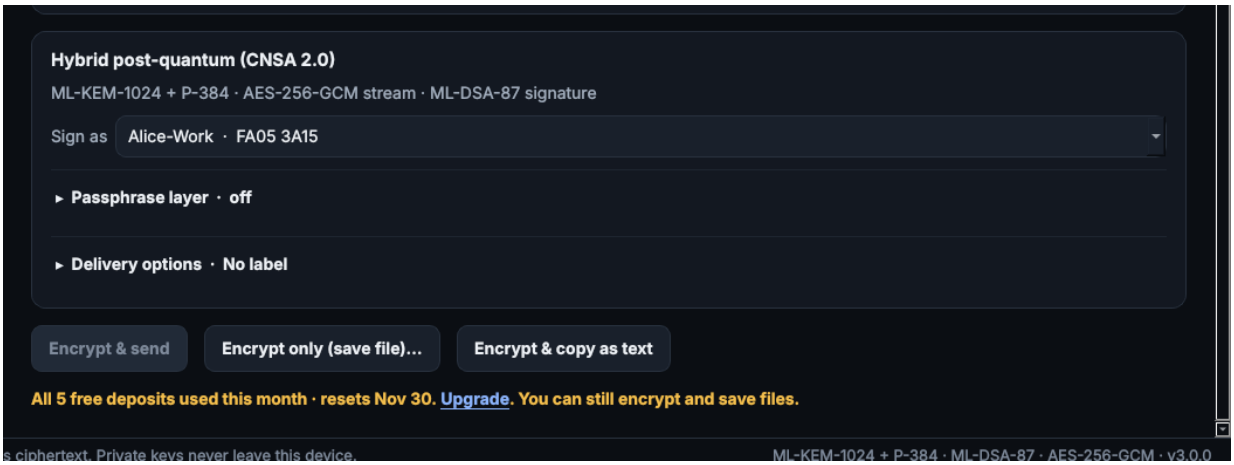


The receipt after Encrypt & send.

The receipt shows the **pickup reference** (the deposit's unique ID), the envelope's SHA-256 fingerprint and whether it matches what the broker stored, the protection used, whether it was signed and the recipients. **Copy reference** copies the ID; **View in Sent** follows it.

4.9 Monthly deposit allowance

Some brokers limit free accounts to a set number of **deposits per calendar month**. The broker decides the number and shows it on its plans page. The allowance only limits sending through the broker. Encrypting, saving, copying, decrypting and picking up are never limited. When the broker reports an allowance, the Encrypt page shows how many free deposits you have left. When they are used up, **Encrypt & send** is switched off until the reset date, and an **Upgrade** link appears if the broker provides one.



All free deposits used: sending is paused, saving still works.

4.10 Sending while offline

If **Settings › Security › Queue deposits while the broker is unreachable** is on, a deposit that cannot be delivered is encrypted and kept in the **Outbox** on your computer. Send it later with **Sent › Send queued now**. Queued envelopes are already encrypted. A deposit is never queued if it would exceed your monthly allowance.

4.11 Several recipients and key types

Any number of recipients can share one message, with any mix of key types. What CryptSecure makes depends on their keys:

- **Only CryptSecure 3 keys:** one envelope for everyone, post-quantum and signed. One deposit, one file or one block of text.
- **One or more CryptSecure 2.6 keys in To:** the 2.6 format can hold only one recipient, so CryptSecure makes separate encryptions: one envelope for all the CryptSecure 3 keys, plus one part for each CryptSecure 2.6 key. The protection line under To says so before you send, for example “2 separate encryptions, because a CryptSecure 2.6 key can only receive a file of its own”.
- **Encrypt & send:** one deposit per part, each with its own pickup reference; the receipt and Sent list each one. If anything would stop one part (a changed key, someone with no account on the broker, a broker that refuses the 2.6 format, not enough monthly allowance for every part), nothing is sent at all and the reason is shown.
- **Encrypt only (save file):** you choose a folder; one file per part, named after who it is for (...-for-Bob.csec; 2.6 parts are .txt files, the format CryptSecure 2.6 reads).
- **Encrypt & copy as text:** one text with a block per part, each headed “===== CryptSecure: for <name> =====”. CryptSecure 3 and the Outlook & Word panel open the whole text and pick the part meant for you. A CryptSecure 2.6 user copies only the lines under their name.
- **Files and folders** go into every part. **Signing:** the CryptSecure 3 part is signed; 2.6 parts cannot carry a signature. **Passphrase layer:** added to every part; each recipient needs the passphrase.
- **Post-quantum:** CryptSecure 3, 2.6 hybrid and 2.6 ML-KEM keys are post-quantum; a 2.6 RSA-only key is not, and the header badge says Not post-quantum while one is in To.
- **Broker policy:** a CNSA 2.0 (post-quantum-only) broker does not accept 2.6-format deposits. With a 2.6 key in To, Encrypt & send is then unavailable and says why; Encrypt only and Encrypt & copy as text still work.
- **Limits:** a key without an email address cannot be delivered through a broker (use Encrypt only or Encrypt & copy, or add an address with Set email...); someone without an account on the broker cannot collect a deposit (Send

invite); copied text is limited to 5 MB; each deposit is limited to the broker's maximum size; each part counts as one deposit against a monthly allowance.

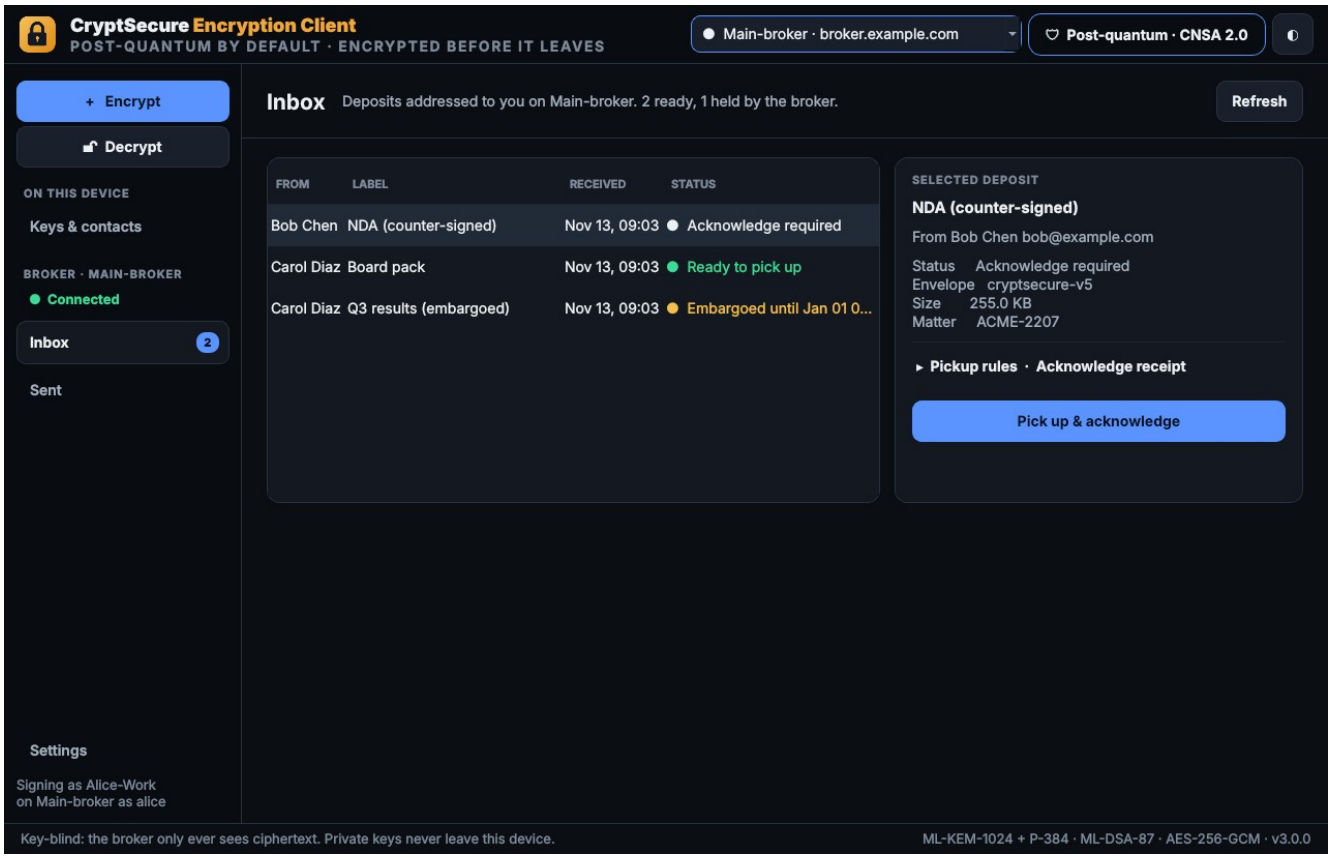
Every combination below was carried out for real in the 3.0.4 audit and again in the 3.0.5 audit, with a file attached and signing on, and every recipient opened the result:

Who is in To	What is made	Encrypt & send: broker accepts 2.6 formats	Encrypt & send: post-quantum-only broker	Encrypt only, Encrypt & copy, command line, Outlook & Word
One CryptSecure 3 recipient	1 envelope	1 envelope	1 envelope	Yes (1 envelope)
Several CryptSecure 3 recipients	1 envelope	1 envelope	1 envelope	Yes (1 envelope)
CryptSecure 3 recipients + yourself	1 envelope	1 envelope	1 envelope	Yes (1 envelope)
One CryptSecure 2.6 hybrid key	1 envelope	1 envelope	Not sent: this broker refuses the 2.6 format	Yes (1 envelope)
One CryptSecure 2.6 ML-KEM key	1 envelope	1 envelope	Not sent: this broker refuses the 2.6 format	Yes (1 envelope)
One CryptSecure 2.6 RSA key (not post-quantum)	1 envelope	1 envelope	Not sent: this broker refuses the 2.6 format	Yes (1 envelope)
CryptSecure 3 + one 2.6 key	2 parts	2 parts	Not sent: this broker refuses the 2.6 format	Yes (2 parts)
CryptSecure 3 (several) + 2.6 keys of every kind	4 parts	4 parts	Not sent: this broker refuses the 2.6 format	Yes (4 parts)
Only 2.6 keys (several)	3 parts	3 parts	Not sent: this broker refuses the 2.6 format	Yes (3 parts)
Your CryptSecure 3 key + your own older 2.6 key	2 parts	2 parts	Not sent: this broker refuses the 2.6 format	Yes (2 parts)
A key with no email address + others	1 envelope	Not sent: a key has no email address	Not sent: a key has no email address	Yes (1 envelope)
Someone with no account on the broker + others	1 envelope	Not sent: someone has no account on the broker	Not sent: someone has no account on the broker	Yes (1 envelope)

Recipient combinations. "Not sent" means nothing was deposited; the reason is shown on screen.

5 Receiving and decrypting

5.1 Your Inbox



The Inbox, with a deposit selected.

The Inbox lists deposits addressed to you on the active broker: who sent each one, its label, when it arrived and its status. Select a deposit to see its details and pickup rules on the right. **Refresh** reloads the list.

Status	Meaning
Ready to pick up	You can download and decrypt it now.
Acknowledge required	The sender asked you to confirm receipt. Choose Pick up & acknowledge. Brokers from version 1.11 release the file only after you acknowledge: Pick up alone is then refused, and CryptSecure offers to acknowledge and download in one step.
Embargoed	Held by the broker until the time shown (Embargoed, or Available from …). Nothing to do yet. Some brokers list an embargoed deposit only once its embargo ends.
Awaiting approver	An approver must release it first.
Picked up	Already collected. When everyone has collected it the broker may erase the file: it then shows as Delivered, file erased and can no longer be downloaded; its custody record remains.
Expired / Withdrawn by sender	No longer available.

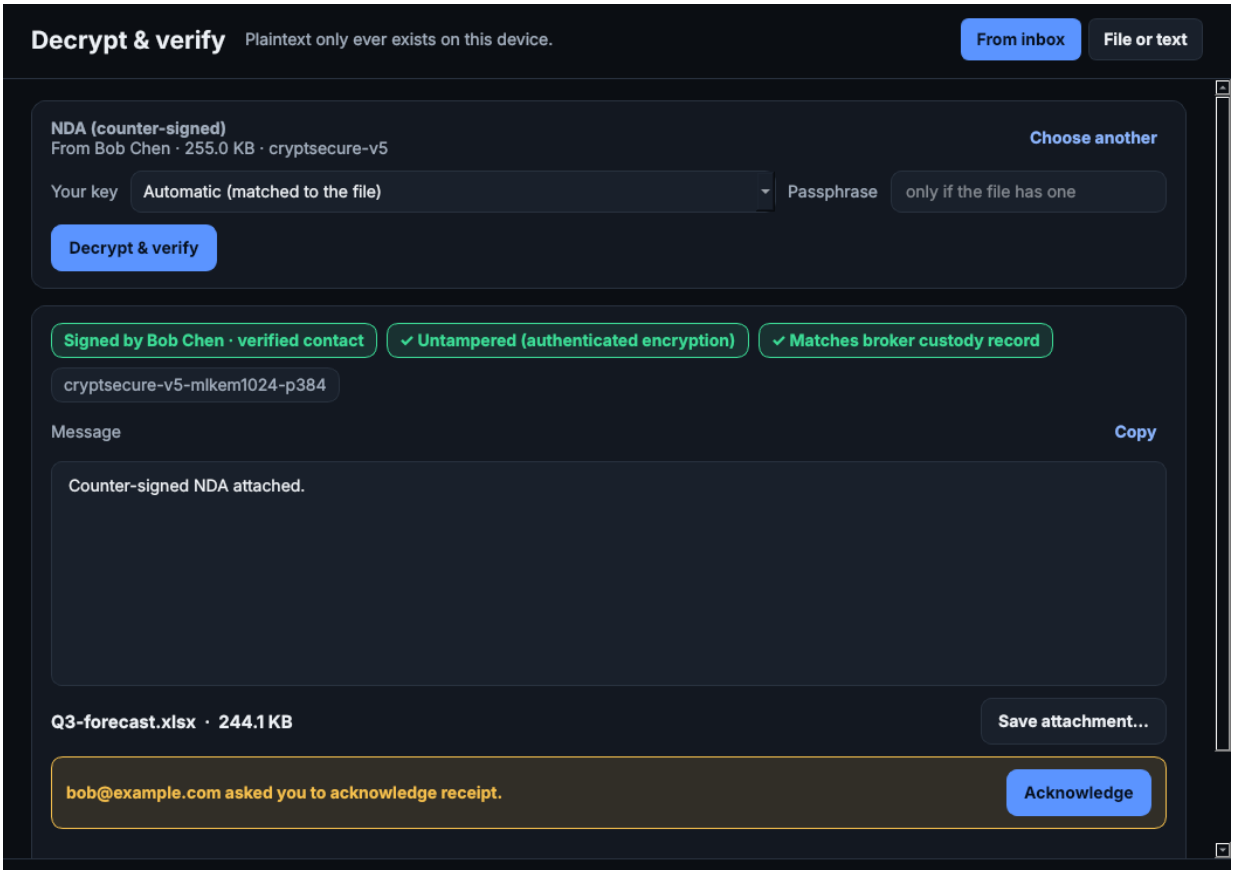
Select a deposit and choose **Pick up & decrypt** (or **Pick up & acknowledge**). CryptSecure downloads the ciphertext, checks it against the broker's custody record, decrypts it on your computer and opens the result on the Decrypt page.

5.2 Decrypting a file or pasted text

1. Choose **Decrypt** in the sidebar. It opens on **File or text**.
2. Choose **Open file...**, drop a file onto the box, or paste ciphertext.
3. Leave **Your key** on **Automatic**, which finds the right key. Enter a **Passphrase** only if the file has one.
4. Choose **Decrypt & verify**.

CryptSecure opens every format from CryptSecure 2.4 to 3.0. There is no size limit when you open a file: it is read and decrypted in the background. Pasted text is limited to 5 MB, so use **Open file...** for anything larger.

5.3 Reading the result



A decrypted, signed deposit with an acknowledgement request.

The row of badges above the message is the most important part of the screen:

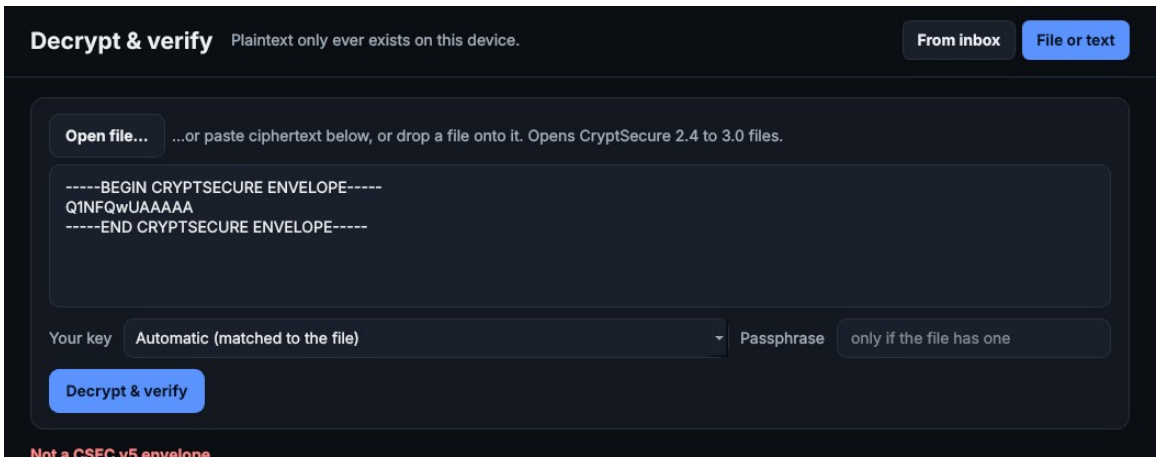
Badge	Meaning
Signed by Bob Chen · verified contact (green)	Signed with a key you have verified. You know who sent it.
Signed by ... · contact not yet verified (amber)	The signature is valid and the key is in your contacts, but you have not compared safety numbers.
Signed by ... · NOT in your contacts (amber)	The signature is mathematically valid, but the signer is unknown to you. Treat the name as unconfirmed.

Badge	Meaning
Signed by you	You sent it, for example to yourself.
Not signed (amber)	No signature, typical of CryptSecure 2.6 files. The sender cannot be confirmed.
✓ **Untampered (authenticated encryption)**	Every byte decrypted and authenticated. Nothing was altered, reordered or cut short.
✓ **Matches broker custody record**	The file you received is exactly the one the broker recorded at deposit.
Format badge, for example cryptsecure-v5-mlkem1024-p384	The envelope format and protection used.

5.4 Saving, acknowledging and clearing

- **Copy** copies the message. Copied plaintext is cleared from the clipboard after 30 seconds by default (Settings › Security).
- **Save attachment…** saves an attached file. For an attached folder the button reads **Extract folder…** and unpacks it where you choose. CryptSecure refuses archive entries that would write outside that folder.
- Messages larger than 5 MB are not displayed. Use **Save message…** to write them to a file.
- **Acknowledge** confirms receipt to the sender when they asked for it.
- **Clear**, or the **Esc** key anywhere in the window, wipes the decrypted text from the screen and clears the clipboard.

5.5 When decryption fails



Message	Cause and fix
None of your keys can open this envelope. It was encrypted for B402 E0FA …	It was encrypted for someone else's key. The characters are the start of the intended recipient's safety number. Compare them with your own in Keys & contacts.
This file has a passphrase layer.	Enter the passphrase the sender gave you.
Envelope content failed authentication (tampered, reordered or truncated).	The file was damaged or altered. Ask for it again. Never trust a partly decrypted file; CryptSecure does not show one.
This is not CryptSecure ciphertext.	The file or pasted text is not a CryptSecure envelope, or part of it is missing.

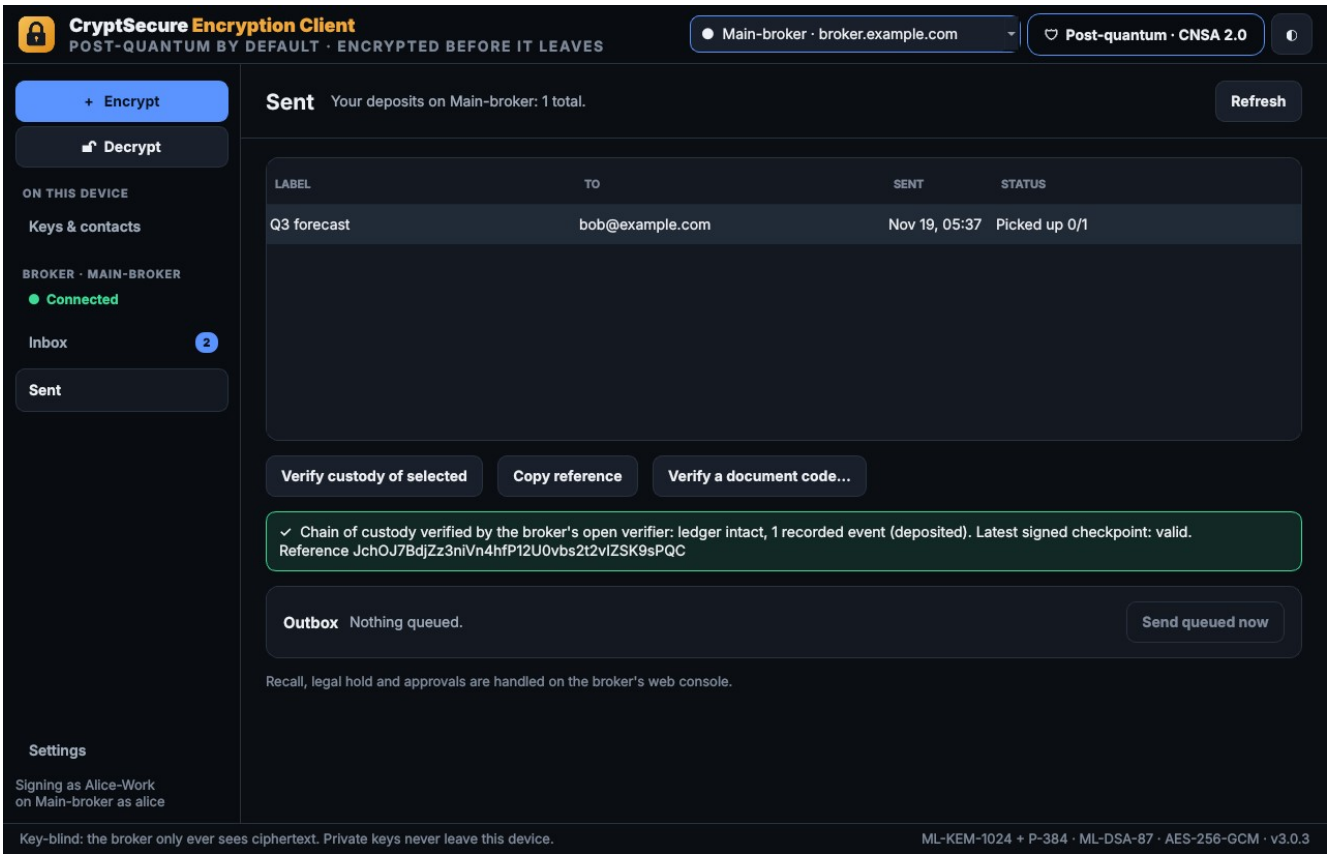
Message	Cause and fix
Not a CSEC v5 envelope.	The text looks like an envelope but is incomplete. Copy the whole block, including the BEGIN and END lines.

5.6 If a message cannot be opened right away

When you pick up a deposit, CryptSecure first keeps the downloaded ciphertext on your computer and then decrypts it. Brokers usually erase their copy as soon as you collect it, so if it cannot be opened yet (the passphrase is missing or wrong, or the key is on another computer), the encrypted copy stays on your computer and the Inbox button reads **Try again (downloaded, not opened yet)**. Enter the passphrase in **Decrypt** and choose **Decrypt & verify**.

Once it opens, the kept copy is deleted, so the sender's rules still hold: file erased after pickup, readable for a limited time after opening, pickup limits. Use **Save message...** or **Save attachment...** if you need to keep it.

6 Sent items and chain of custody



Sent, with a custody check completed.

Sent lists your deposits on the active broker and what has happened to each: how many recipients have collected it (Picked up 1/2), Delivered, file erased once everyone has, Withdrawn by sender or Expired. Select one and choose:

- **Verify custody of selected** to ask the broker's open verifier whether its tamper-evident ledger is intact and to read this deposit's recorded events: deposited, each pickup, acknowledgements and file erased. The result also reports the broker's latest signed checkpoint. If the ledger has been altered, it says BROKEN.
- **Copy reference** to copy the deposit ID for your records or for the recipient.

The **Outbox** panel shows deposits queued while the broker was unreachable. **Send queued now** delivers them. Verify a document code… checks the verification code printed on a broker certificate or History page, for example EH7G-MPKV-49MQ-4G4D-TNR7-CVBG (cs verify-code on the command line). Recalls, legal holds and approvals are handled in the broker's web console.

WHAT YOU CAN PROVE, END TO END

Every deposit through a broker leaves a trail you can check, from your computer to the recipient's:

1. **Deposited intact.** The broker's SHA-256 receipt matches the fingerprint of the envelope your computer produced (`sha256_match: True` on the command line).
2. **Delivered intact.** Before anything is decrypted, the recipient's copy is checked against the broker's custody record (✓ **Matches broker custody record**).
3. **From you, unchanged.** Your ML-DSA-87 signature shows the recipient who sent it and that nothing was altered.
4. **Received.** If you asked for an acknowledgement, the recipient's confirmation is recorded with the deposit.
5. **Record untouched. Verify custody of selected** (`cs custody` on the command line) has the broker's open verifier confirm that its tamper-evident ledger is intact, and lists this deposit's recorded events.

Chapter 10.4 follows one deposit through all five steps, from `cs submit` to `cs custody`, with real output.

6.1 Checking custody of messages you received

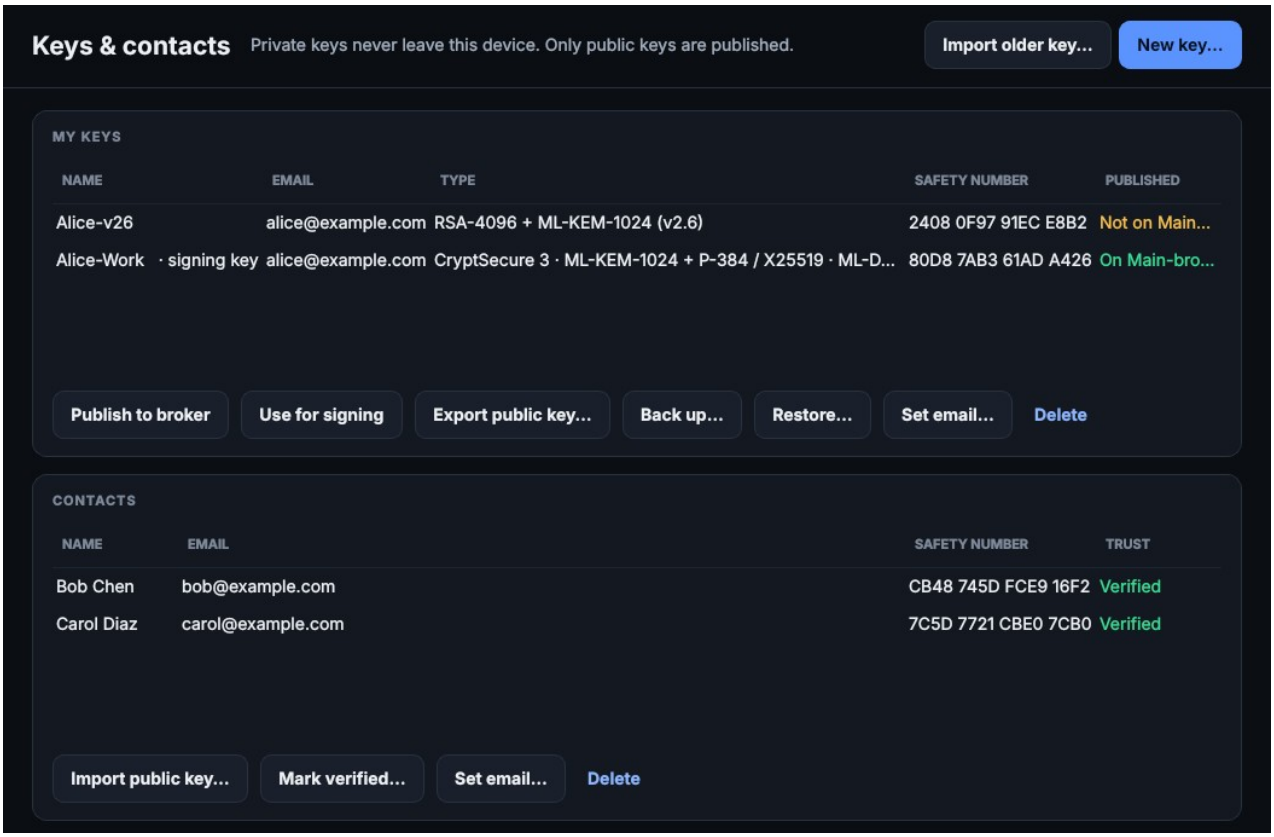
In the **Inbox**, select a deposit and choose **Verify custody**. It asks the broker's open verifier the same questions as Sent does, so the sender and every recipient see the same record: whether the broker's ledger is intact, every event recorded for this deposit, and its state. You can check before you pick it up, right after, or long after the broker has erased the file. On the command line: cs custody <reference>.

When you pick a deposit up, the Decrypt page also shows **Matches broker custody record**: the SHA-256 of what you downloaded equals what the broker recorded at deposit.

6.2 What each custody result means

Result shown	What it means	Who sees it
✓ … Not picked up yet: it is still waiting at the broker.	Deposited and recorded; nobody has collected it yet.	Sender and every recipient
✓ … Picked up 1 time.	Some recipients have collected it; the others still can.	Sender and every recipient
✓ … Picked up 2 times, acknowledged.	Collected by everyone (and acknowledged, when the sender asked for it).	Sender and every recipient
✓ … (deposited → picked up → file erased)	The broker erased the ciphertext after the last pickup. The record and the SHA-256 fingerprint remain, so custody can still be proven.	Sender and every recipient
This deposit was recalled by the sender	Withdrawn before delivery; it can no longer be picked up.	Sender; recipients if it is still listed
This deposit expired before it was picked up	Nobody collected it in time; the broker removed the ciphertext.	Sender
This deposit was erased by the broker operator	Crypto-erased (for example on request); the record and SHA-256 remain.	Sender and recipients
This deposit was denied by the approver	An approver did not release it; it was never delivered.	Sender
CHAIN BROKEN	The broker's own verifier reports its ledger has been altered. Do not rely on the record; tell the broker operator.	Sender and every recipient
Could not verify the chain	The broker's verifier did not answer (offline, or a broker without the open verifier). Try again later.	Sender and every recipient

7 Keys and contacts



Keys & contacts: your keys above, other people's public keys below. Each key shows the email address it belongs to.

7.1 Kinds of keys

Type shown	What it is
CryptSecure 3 · ML-KEM-1024 + P-384 / X25519 · ML-DSA-87	The recommended key. Supports all three post-quantum protections and signs with ML-DSA-87.
RSA-4096 + ML-KEM-1024 (v2.6)	A CryptSecure 2.6 hybrid key. Post-quantum, but it cannot sign and only works with the 2.6 format.
ML-KEM-1024 (v2.6)	A CryptSecure 2.6 post-quantum-only key.
RSA-4096 (v2.6)	A CryptSecure 2.6 RSA key. Not post-quantum.

7.2 Your keys

Your keys are listed with their **Name**, **Email**, **Type**, **Safety number** and whether they are **Published**. One email address can have several keys, for example your CryptSecure 3 key and an older 2.6 key: old messages open with whichever key they were sent to, and new messages to you use the strongest one. **New key...** and **Import older key...** are at the top of the page.

Button	What it does
New key...	Creates a key. Choose CryptSecure 3 unless you need to talk to someone still on 2.6.

Button	What it does
Publish to broker	Puts your public key in the active broker's directory so people can find you.
Use for signing	Makes the selected CryptSecure 3 key your default signing key.
Export public key...	Saves your public key to a .cspub file you can email or hand to someone.
Back up my keys...	Makes a recovery package of all your keys, protected by a recovery code shown once (Chapter 12).
Restore...	Restores your keys from a recovery package, or a key from a CryptSecure 3.0.0 to 3.0.4 backup file, for example on a new computer.
Protection	How your keys are protected: level, ways to unlock, unlock time, backup and recovery (Chapter 12). Shown at the top of the page.
For CryptSecure 2.6...	Writes one of your 2.6-format keys as the plain files CryptSecure 2.6 reads. These files are not protected.
Import older key...	Brings in your own older keys (v2.4 / v2.6 .pem and .bin files) with your email address, so old messages still open. You can keep several keys on one email address.
Set email...	Attaches an email address to the selected key, or changes it. The same button is under Contacts.
Delete	Removes the key from this computer. Messages encrypted to it can no longer be opened here.

IMPORTANT

Your private keys are protected on this computer (Chapter 12). A copied key folder does not open on another computer or user account. To move your keys or to be safe if this computer is lost, use **Back up my keys...**: it makes a recovery package that opens only with a recovery code shown once.

Without a recovery package and its code, lost keys cannot be recovered by anyone.

7.3 Contacts

Contacts are other people's public keys. They are added automatically when you use **Find key**, or manually with **Import public key...** from a .cspub, .pem or .json public key file. Set email... attaches or changes the address of a contact. Select a contact and choose **Mark verified...** after you have compared safety numbers. The **Trust** column then shows **Verified**.

7.4 What each kind of key can do

Every cell below was proven in the 3.0.5 audit by doing it: real keys, real messages, a test broker, the command line the Outlook & Word add-in uses and, for the 2.6 columns, the real CryptSecure 2.6.0 program. In short: a CryptSecure 3 key can do everything; 2.6 keys can receive (and open) everything sent to them, but cannot sign, cannot share one envelope with other recipients (they get a part of their own), cannot go through a post-quantum-only broker, and an RSA-only key is not post-quantum.

What you can do	CryptSecure 3	2.6 hybrid (RSA + ML-KEM)	2.6 ML-KEM only	2.6 RSA only
Create it in the window (Keys & contacts > New key...)	Yes	Yes	No	Yes
Create it from the command line (cs keys new / new-legacy)	Yes	Yes	Yes	Yes

What you can do	CryptSecure 3	2.6 hybrid (RSA + ML-KEM)	2.6 ML-KEM only	2.6 RSA only
Bring in your own older key (Import older key… / cs keys import-old)	No	Yes	Yes	Yes
Post-quantum protection	Yes	Yes	Yes	No
Others can encrypt to it	Yes	Yes	Yes	Yes
Opens messages sent to it	Yes	Yes	Yes	Yes
Sign messages with it (ML-DSA-87)	Yes	No	No	No
Messages sent to it carry the sender's signature	Yes	No	No	No
Publish it to a broker	Yes	Yes	Yes	Yes
Others find it with Find key	Yes	Yes	Yes	Yes
Encrypt & send to it via a broker that accepts 2.6 formats	Yes	Yes	Yes	Yes
Encrypt & send to it via a CNSA 2.0 / post-quantum-only broker	Yes	No	No	No
Pick it up from the Inbox and open it	Yes	Yes	Yes	Yes
Share ONE envelope with other recipients	Yes	No	No	No
Be one of several recipients (its own part when needed)	Yes	Yes	Yes	Yes
Passphrase layer on top	Yes	Yes	Yes	Yes
Files	Yes	Yes	Yes	Yes
Folders	Yes	Yes	Yes	Yes
Encrypt & copy as text	Yes	Yes	Yes	Yes
Message readable in CryptSecure 2.6.0	No	Yes	Yes	Yes
Outlook & Word add-in: encrypt to it	Yes	Yes	Yes	Yes
Outlook & Word add-in: sign with it	Yes	No	No	No
Export public key	Yes	Yes	Yes	Yes
Import as a contact	Yes	Yes	Yes	Yes
Back up my keys… / Restore…	Yes	Yes	Yes	Yes
Set email…	Yes	Yes	Yes	Yes
Mark verified…	Yes	Yes	Yes	Yes
Delete	Yes	Yes	Yes	Yes

Key capability matrix. Back up my keys… makes one recovery package with every key, CryptSecure 3 and 2.6; Restore… brings them all back on another computer.

8 Safety numbers

A safety number is how you make sure a key really belongs to the person you think it does. Checking it once, with each person, closes the one gap encryption alone cannot.

8.1 Why they matter

Encryption guarantees that only the holder of a key can read a message. It cannot tell you **who holds the key**. If an attacker tricked you into using their key instead of Bob's, for example by planting it in a directory, your message would be perfectly encrypted, for the attacker. The safety number lets you and Bob confirm, through a channel the attacker does not control, that you are using Bob's real key.

8.2 What a safety number is

> **A SAFETY NUMBER**
> `2B3E 250C E30A FF51 AAF1 3057 954D DC3C`

- It is **32 characters**, shown as eight groups of four, using 0–9 and A–F.
- It is calculated from the person's **whole public key**: the SHA-384 fingerprint of every key part (ML-KEM, elliptic curve and ML-DSA), of which the safety number is the first 128 bits.
- It is the **same on both computers**. Your own number in My keys matches what everyone else sees in their contacts for you.
- It **changes if the key changes**. Any different key gives a completely different number. Producing a different key with the same 32 characters is not practically possible.
- It is **not secret**. You can read it aloud, print it on a business card or show it on a video call.

Shorter forms appear where space is tight: the first four groups in tables, the first two next to **Sign as**. Always compare **all eight groups** when you verify.

8.3 How to compare

1. Contact the person through a channel you already trust: a phone number you know, a video call, or in person. Do not use an address or number that came with the key.
2. You read out the safety number shown on your screen for **them** (on their recipient card, or in Contacts). They read out the number shown for **their own key** in My keys. Or the other way round.
3. Compare all eight groups. If every group matches, choose **Mark verified…**, or `cs keys verify` on the command line.
4. If anything differs, do not send. Ask them to check they published the right key, then look the key up again.

> **TIP**
> Verify once per person and key. CryptSecure remembers, shows ✓ Verified on every message to or from them, and warns you if their key ever changes.

8.4 When a key changes

People get new computers and create new keys, so a changed key is often innocent. But it is also exactly what an impersonation looks like. A key counts as changed only when the broker stops offering the key it gave you before. A key you imported yourself (a 2.6 .pem file or a contact file) never counts as changed, and the broker offering an additional key for someone is not a change: the new key is added next to the ones you have. When a key the broker gave you is replaced, CryptSecure:

- shows the **"…'s key has changed since you last used it"** card and stops **Encrypt & send** for that recipient;
- on the command line, stops `submit` with exit code 7 and shows the new safety number;
- keeps the old key until you confirm the new one (confirming never deletes a key you imported yourself) with **I compared it, trust the new key**, or `cs lookup <email> --accept-changed`.

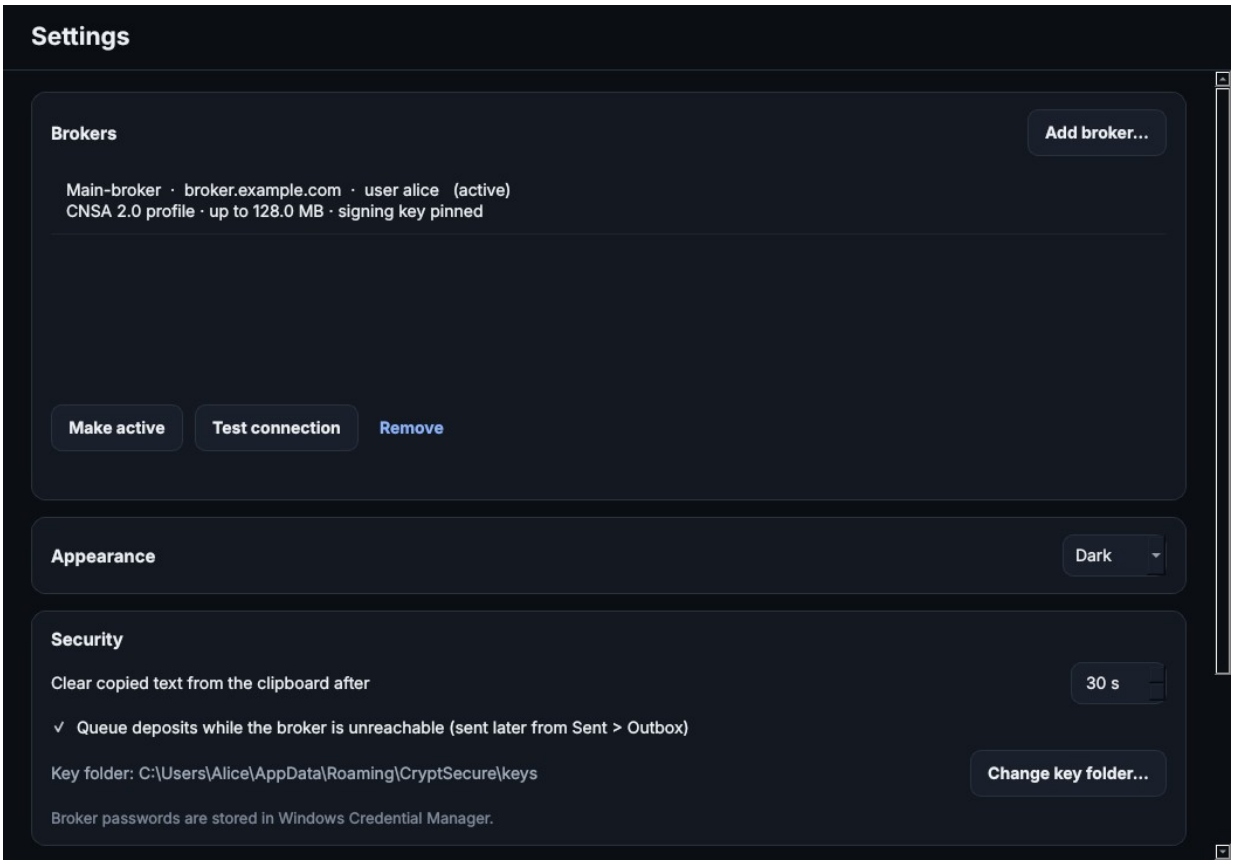
8.5 The broker's own key

Brokers sign their responses. When you add a broker, CryptSecure pins the broker's signing key, shown as **signing key pinned** in Settings, and as `pinned_signing_key` on the command line. If it later changes, CryptSecure refuses to work with that broker and the status light shows **Broker key changed**. Confirm with the broker operator before you remove and re-add the broker.

8.6 Safety numbers on received messages

When you decrypt a signed message, CryptSecure finds the signer's key by its fingerprint. The signature badge tells you whether that key is yours, a verified contact, an unverified contact or unknown (see Chapter 5.3). An unknown signer's name is only what the sender typed, so verify before you rely on it.

9 Settings



Settings.

9.1 Brokers

Lists your brokers with their address, your username, the broker's security profile, maximum deposit size and pinned signing key. **Add broker...**, **Make active**, **Test connection** and **Remove** do what they say. Remove also deletes the saved application password. You can use several brokers; one is active at a time and the switcher at the top of the window changes it. Inbox, Sent and Find key always use the active broker.

9.2 Appearance

Dark, Light, Blue, or System to follow Windows.

9.3 Security

Setting	Default	Effect
Clear copied text from the clipboard after	30 s	Copied plaintext (decrypted messages, references) is removed from the clipboard after this many seconds. 0 means never. Copied ciphertext is not cleared, since it is safe to share.
Queue deposits while the broker is unreachable	Off	Keeps deposits in the Outbox when the broker cannot be reached (Chapter 4.10).

Setting	Default	Effect
Key folder / Change key folder…	%APPDATA%\ CryptSecure\keys	Where keys are stored. Change it only if your organization requires another location.
Credential storage	Protected with your keys	Where broker application passwords are kept. Before 3.0.5: Windows Credential Manager.

9.4 About

Shows the version number, the cryptographic libraries in use and your data folder. Quote this information when you ask for support.

10 The command line

Every function of CryptSecure is also available as a command, for scripting, automation and batch work. The commands are part of CryptSecure.exe: there is no separate program to install. The command line uses the same keys, contacts, brokers and settings as the desktop application.

10.1 Running commands from CryptSecure.exe

CryptSecure.exe is a Windows desktop program. When started with the word `cli` it runs a command instead of opening the window, and writes the result to the console it was started from:

```
CryptSecure.exe cli <command> [options]
```

Because it is a desktop program, the Windows console does not wait for it to finish unless you ask it to. The simplest way to handle this is a small helper called `cs`, which you set up once.

Command Prompt: create cs.cmd

Save the following as **cs.cmd** in the same folder as CryptSecure.exe, then add that folder to your PATH. (Using the Microsoft Store edition? See the second version below.)

CS.CMD

```
@echo off
rem cs.cmd - run CryptSecure commands and wait for the result
start "" /b /wait "%~dp0CryptSecure.exe" cli %*
exit /b %ERRORLEVEL%
```

The Microsoft Store edition installs a **CryptSecure.exe** command that works in any Command Prompt or PowerShell window, but its program folder cannot be changed. Save this version of **cs.cmd** in any folder that is on your PATH instead:

CS.CMD (MICROSOFT STORE EDITION)

```
@echo off
rem cs.cmd - run CryptSecure commands and wait for the result
start "" /b /wait CryptSecure.exe cli %*
exit /b %ERRORLEVEL%
```

Now `cs keys list` runs `CryptSecure.exe cli keys list`, waits, shows the output in the same window and sets %ERRORLEVEL% to the result. The rest of this chapter uses `cs`.

PowerShell: add a cs function

Add this line to your PowerShell profile (`notepad $PROFILE`), adjusting the path to where CryptSecure.exe is. With the Microsoft Store edition, use just CryptSecure.exe, without a path:

POWERSHELL PROFILE

```
function cs { & "C:\Program Files\CryptSecure\CryptSecure.exe" cli @args | Write-Output }
```

Piping through Write-Output makes PowerShell wait for the result. It also lets you capture output, for example `cs --json keys list | ConvertFrom-Json`. The exit code is in `$LASTEXITCODE`.

> **TIP**
> Automating at scale? Your administrator can also build a console edition of the same program (Appendix C) that needs no helper: `CryptSecure-cli.exe keys list`.

10.2 Global options

Global options go **before** the command name, for example `cs --json inbox`.

Option	Effect
`--json`	Prints a single JSON object instead of text. Every result includes "ok": true or false; errors add "error", "code" and "exit_code". Use this in scripts.
`--quiet`	Prints nothing. Use the exit code.
`--caller NAME`	Used by the Outlook & Word add-in: the unlock window names Outlook or Word.
`--home <folder>`	Uses another data folder instead of %APPDATA%\CryptSecure, for example for a service account or a test.
`-h`, `--help`	Help for the program, or for any command: `cs submit --help`.

Progress lines that start with `...` are written to standard error, so they never mix with JSON output. Long operations can be cancelled with Ctrl+C.

10.3 Command reference

Command	What it does
`doctor`	Shows the version, cryptographic libraries, data folder and credential store.
`keys list`	Lists your keys and contacts with their safety numbers.
`keys new <label> [--name N] [--email E]`	Creates a CryptSecure 3 key.
`keys new-legacy <label> [--type hybrid|rsa|mlkem] [--bits 2048|3072|4096] [--email E]`	Creates a CryptSecure 2.6-format key, for people still on 2.6. Defaults: hybrid, 4096.
`keys export-public <label> [-o FILE]`	Prints your public key, or writes it to FILE.
`keys import-public <file> [--email E] [--name N] [--label L]`	Adds a contact from a public key file.
`keys import-old --rsa-private F [--rsa-public F] [--mlkem-public F --mlkem-private F] [--email E] [--name N] [--passphrase-env VAR]`	Brings in your own older key files (v2.4 / v2.6). The public .pem is checked against the private key. Any number of keys can share one email address.
`keys set-email <label> <email>`	Attaches an email address to a key or contact, or changes it.
`keys verify <label>`	Marks a contact verified, after you have compared safety numbers.
`keys default <label>`	Sets your default signing key (a CryptSecure 3 key of your own).

Command	What it does
`backup make -o FILE`	Makes a recovery package of all your keys and shows the recovery code once.
`keys restore <file> [--label L]`	Restores a key from a CryptSecure 3.0.0 to 3.0.4 backup file (.csid-backup).
`backup check <file> --code-env V`	Checks that a recovery package opens with its code. Changes nothing.
`backup restore <file> --code-env V`	Restores every key in a recovery package that is not already here.
`keys protection`	Shows the level and the ways to unlock your keys.
`keys setup [--method M]`	Sets up key protection now: auto, passphrase, security-key, tpm, tpm-pin or windows.
`keys add-method <method>`	Adds a way to unlock: passphrase, security-key, tpm, tpm-pin or windows.
`keys remove-method <id>`	Removes a way to unlock (not the last one).
`keys lock`	Locks your keys in this command line (each command-line run locks them when it ends; the window has Lock my keys now).
`keys unlock-minutes <0-60>`	How long one unlock lasts; 0 asks every time.
`keys activity [--limit N]`	When your keys were used, for what, and who asked.
`keys move-2.6 --yes`	Protects your CryptSecure 2.6 key files (2.6 can then no longer use them).
`keys export-2.6 <label> --out DIR --yes`	Writes a 2.6-format key as plain files for CryptSecure 2.6 (not protected).
`keys prepare-downgrade --yes`	Before going back to 3.0.4: writes keys and broker passwords where 3.0.4 reads them.
`keys stay`	After keys prepare-downgrade, if you stay on (or come back to) 3.0.5: protects those copies again.
`keys remove-old-copies --yes`	Wipes the 3.0.4 key copies kept for 30 days after updating.
`keys delete <label> --yes`	Deletes a key or contact. `--yes` is required.
`broker add --name N --url URL --user U [--password-env VAR] [--no-test]`	Adds a broker, tests it and pins its key. The application password is read from the environment variable VAR, or prompted for.
`broker list`	Lists brokers; the active one is marked.
`broker use <id>`	Makes a broker active.
`broker test [<id>]`	Checks connection, login and policy, and reports the deposit allowance.
`broker remove <id>`	Removes a broker and its saved password.
`lookup <email> [--accept-changed]`	Finds a recipient's key in the active broker's directory.
`publish <label>`	Publishes your public key to the active broker.
`invite <email>`	Invites someone who has no key yet.
`encrypt --to R… -o FILE [-m TEXT | --message-file F] [-a PATH] [--text] [--mode M] [--sender L] [--passphrase-env VAR | --passphrase-prompt]`	Encrypts for local keys or contacts and writes a file. `--text` writes text instead of binary. `-a` takes a file or a folder. `--mode` forces a protection: auto (the default, Chapter 4.4), MLKEM1024-P384, MLKEM768-X25519, MLKEM1024 or LEGACY-V26.

Command	What it does
`decrypt <file> [-o DIR] [--key L] [--passphrase-env VAR | --passphrase-prompt]`	Decrypts a file and prints the message; `-o` saves the attachment.
`submit --to EMAIL… [-m | --message-file] [-a PATH] [--mode M] [--sender L] [--passphrase-env VAR | --passphrase-prompt] [--label] [--matter] [--classification] [--caveats] [--dist-statement] [--expires-at] [--available-from] [--embargo-until] [--pickup-limit N] [--quorum N] [--ttl-after-open-mins N] [--approver-email] [--ack] [--approval] [--require-verified]`	Encrypts and deposits to the active broker in one step. `--require-verified` refuses recipients you have not verified. `--caveats` and `--dist-statement` record a handling caveat and a distribution statement (a letter A to F) with the deposit, like `--classification`; they are visible to the broker, not encrypted.
`inbox`	Lists deposits waiting for you.
`sent`	Lists your deposits and their status.
`pickup <uid> [-o DIR] [--ack] [--passphrase-env VAR | --passphrase-prompt]`	Downloads, checks and decrypts a deposit; `--ack` also acknowledges it. Add `--passphrase-env` or `--passphrase-prompt` for deposits with a passphrase layer.
`ack <uid>`	Acknowledges receipt of a deposit.
`custody <uid>`	Verifies the chain of custody for a deposit.
`verify-code <code>`	Checks a custody document's verification code (as printed on a broker certificate or History page) with the broker's open verifier: authentic and unaltered, not valid, or unknown.
`outbox [--flush]`	Shows queued deposits, or sends them.

SECURITY

Passwords and passphrases are never accepted as command-line arguments, which other programs on the computer could see. Pass them in an environment variable (`--password-env`, `--passphrase-env`) or type them at the prompt (`--passphrase-prompt`).

In Command Prompt: `set CS_APP_PW=…`. In PowerShell: `$env:CS_APP_PW = "…"`. Clear the variable when you are done.

10.4 Worked examples

The examples follow Alice, Bob and Carol from first setup to delivery. All output is reproduced exactly as the release build produced it; a line shown as … marks output shortened for this guide. Only computer-specific values (dates, IDs, fingerprints) will differ for you.

Check the installation

```
C:\Work> cs doctor
version: 3.0.5
  … (built-in component versions)
suites:
  - MLKEM1024-P384
  - MLKEM768-X25519
  - MLKEM1024
data_folder: C:\Users\Alice\AppData\Roaming\CryptSecure
credential_store: protected together with your keys
key_protection: Strong
```

The full output also lists the versions of the cryptographic components built into your copy of CryptSecure. Include all of it when you contact support.

Create a key and list it

```
C:\Work> cs keys new Alice-Work --name "Alice Martin" --email alice@example.com
label: Alice-Work
kind: cs3
private: True
email: alice@example.com
name: Alice Martin
verified: True
safety_number: BDC9 00FD 9F35 F149 863F 1A37 CAF9 FCC0
source: local
published_on: (none)
description: CryptSecure 3 · ML-KEM-1024 + P-384 / X25519 · ML-DSA-87

C:\Work> cs keys list
keys:
  Alice-Work  (your key)
     Alice Martin <alice@example.com>
     CryptSecure 3 · ML-KEM-1024 + P-384 / X25519 · ML-DSA-87
     safety number BDC9 00FD 9F35 F149 863F 1A37 CAF9 FCC0

default_signing_key: Alice-Work
```

Your own keys always show `verified: True`. The safety number is what other people compare with you.

Add the broker, test it, publish your key

```
C:\Work> set CS_APP_PW=abcd EFGH ijkl MNOP qrst UVWX
C:\Work> cs broker add --name Main --url https://broker.example.com --user alice --password-
env
    CS_APP_PW
added: main
active: True
pinned_signing_key: 93c61eb414d239c9

C:\Work> set CS_APP_PW=

C:\Work> cs broker test
broker: main
ok: True
info_endpoint: True
profile: CNSA 2.0
modes:
  - MLKEM1024-P384
  - MLKEM1024
default_mode: MLKEM1024-P384
max_bytes: 134217728
allow_passphrase: False
deposit_allowance: not reported by broker

C:\Work> cs publish Alice-Work
published: Alice-Work
broker: main
response: {'ok': True, 'fingerprint':
'cd7a5a663feb45b3c2fda643d49285d0ed7805dd4ac943efc7d714096
    ee43dc8'}
```

`broker test` reports the broker's policy: its profile, the protections it accepts (`modes`), the default, the maximum deposit size in bytes, whether passphrase-protected deposits are allowed, and your deposit allowance if the broker reports one.

Find a recipient and verify them

```
C:\Work> cs lookup bob@example.com
email: bob@example.com
state: unverified
safety_number: ED1F 92BB 2FC6 D2BA 10A0 E8CE 342F 58AC
algo: cryptsecure-v3
label: bob@example.com
modes:
  - MLKEM1024-P384
  - MLKEM768-X25519
  - MLKEM1024

C:\Work> cs keys verify bob@example.com
label: bob@example.com
kind: cs3
private: False
email: bob@example.com
name: Bob Chen
verified: True
safety_number: ED1F 92BB 2FC6 D2BA 10A0 E8CE 342F 58AC
source: broker:main
published_on: (none)
description: CryptSecure 3 · ML-KEM-1024 + P-384 / X25519 · ML-DSA-87
```

Alice ran `keys verify` only after Bob read his safety number to her over the phone and all eight groups matched.

Encrypt a file to save or email

```
C:\Work> cs encrypt --to bob@example.com -m "Draft is final. Numbers attached." -a
    Q3-forecast.xlsx -o forecast.csec
  ... Reading attachment
  ... Wrapping content key (MLKEM1024-P384) for Bob Chen
  ... Encrypting (AES-256-GCM stream)
  ... Signing (ML-DSA-87)
written_to: forecast.csec
kind: cryptsecure-v5-mlkem1024-p384
mode: MLKEM1024-P384
signed: True
bytes: 59031
sha256: 76e4fd9b306a6a9bfabb87ff1bfa4efcf37bf0deec5dff7f4714b566df920465
recipients:
  - bob@example.com
```

Several recipients, a message read from a file, a whole folder, and text output for pasting into an email:

```
C:\Work> cs lookup carol@example.com
email: carol@example.com
state: unverified
safety_number: 216B D4B6 DB22 BC99 E84D 462D 6AB7 DFE2
algo: cryptsecure-v3
label: carol@example.com
...

C:\Work> cs encrypt --to bob@example.com carol@example.com --message-file note.txt -a
Contracts
    --text -o minutes.txt
  ... Compressing folder
  ... Wrapping content key (MLKEM1024-P384) for Bob Chen
  ... Wrapping content key (MLKEM1024-P384) for Carol Diaz
  ... Encrypting (AES-256-GCM stream)
  ... Signing (ML-DSA-87)
written_to: minutes.txt
kind: cryptsecure-v5-mlkem1024-p384
mode: MLKEM1024-P384
signed: True
bytes: 40784
sha256: 29c25654cb7320f84208d93e85c3c9be553445c4845c31a5f788be20ad12f134
recipients:
  - bob@example.com
  - carol@example.com
```

With a passphrase layer (the passphrase comes from an environment variable):

```
C:\Work> set CS_PASS=orbit-maple-42-lantern
C:\Work> cs encrypt --to bob@example.com -m "Wire details" --passphrase-env CS_PASS -o
wire.csec
  ... Wrapping content key (MLKEM1024-P384) for Bob Chen
  ... Encrypting (AES-256-GCM stream)
  ... Signing (ML-DSA-87)
  ... Adding passphrase layer (Argon2id)
written_to: wire.csec
kind: cryptsecure-passphrase
mode: MLKEM1024-P384
signed: True
bytes: 11047
sha256: 0417edf51d7d082b847370d085bbf5b49e10006e53914453b4ce1727161792ea
recipients:
  - bob@example.com
```

Decrypt

```
C:\Bob> cs decrypt forecast.csec -o Received
  ... Reading envelope
  ... Unwrapping content key (MLKEM1024-P384)
  ... Decrypting (AES-256-GCM stream)
  ... Verifying sender signature (ML-DSA-87)
Signed by Alice Martin · contact not yet verified
------------------------------------------------------------
Draft is final. Numbers attached.
------------------------------------------------------------
Attachment: Q3-forecast.xlsx (46.9 KB) saved to Received\Q3-forecast.xlsx
```

The first line is the signature result (see Chapter 5.3). Bob has not verified Alice yet, so it says so. Carol has never looked Alice up, so for her the signer is not in her contacts. Her attached folder is unpacked into Received:

```
C:\Carol> cs decrypt minutes.txt -o Received
  ... Reading envelope
  ... Unwrapping content key (MLKEM1024-P384)
  ... Decrypting (AES-256-GCM stream)
  ... Verifying sender signature (ML-DSA-87)
Signed by Alice Martin · NOT in your contacts (signature valid, identity unconfirmed)
------------------------------------------------------------
Board minutes attached. Please review before Friday.

------------------------------------------------------------
Attachment: Contracts_1794578103.zip (26.7 KB) saved to Received\Contracts_1794578103
```

Send through the broker, pick up, acknowledge

```
C:\Work> cs submit --to bob@example.com -m "Q3 forecast attached." -a Q3-forecast.xlsx
    --label "Q3 forecast" --matter ACME-2207 --classification Confidential --ack
  ... Fetching bob@example.com's public key
  ... Reading attachment
  ... Wrapping content key (MLKEM1024-P384) for Bob Chen
  ... Encrypting (AES-256-GCM stream)
  ... Signing (ML-DSA-87)
  ... Uploading to Main and checking SHA-256
ok: True
uid: s7H3GIXlFqFWNbQOExMkzWr4nYFyJJRhyTP2hr8f
deposit_id: 1
sha256: 93c00e324b2df931dd1313088d54d061a0bfbff65c9ae84587fa3c31e22570bb
kind: cryptsecure-v5
sha256_local: 93c00e324b2df931dd1313088d54d061a0bfbff65c9ae84587fa3c31e22570bb
sha256_match: True
mode: MLKEM1024-P384
signed: True
broker: Main
recipients:
  - bob@example.com
kind_local: cryptsecure-v5-mlkem1024-p384
queued: False
```

`sha256_match: True` confirms the broker stored exactly the envelope Alice's computer produced. Bob sees it in his inbox and picks it up:

```
C:\Bob> cs inbox
inbox:
  Q3 forecast  ·  from alice@example.com
     2026-11-19 10:34 UTC  ·  Ready to pick up
     uid s7H3GIXlFqFWNbQOExMkzWr4nYFyJJRhyTP2hr8f

C:\Bob> cs pickup s7H3GIXlFqFWNbQOExMkzWr4nYFyJJRhyTP2hr8f -o Downloads --ack
... Downloading ciphertext
... Acknowledging receipt (the sender requires it before release)
... Downloading ciphertext
... Reading envelope
... Unwrapping content key (MLKEM1024-P384)
... Decrypting (AES-256-GCM stream)
... Verifying sender signature (ML-DSA-87)
Signed by Alice Martin · contact not yet verified
------------------------------------------------------------
Q3 forecast attached.
------------------------------------------------------------
Attachment: Q3-forecast.xlsx (46.9 KB) saved to Downloads\Q3-forecast.xlsx
✓ Matches the broker's custody record (SHA-256)
Receipt acknowledged.
```

Alice checks the status and the chain of custody:

```
C:\Work> cs sent
sent:
  Q3 forecast
     2026-11-19 10:34 UTC  ·  Picked up 1/1 · file erased
     to bob@example.com
     uid s7H3GIXlFqFWNbQOExMkzWr4nYFyJJRhyTP2hr8f

C:\Work> cs custody s7H3GIXlFqFWNbQOExMkzWr4nYFyJJRhyTP2hr8f
✓ Chain of custody verified by the broker's open verifier: ledger intact, 5 recorded events
(deposited → pickup refused → acknowledged → picked up → file erased). Latest signed
checkpoint: valid.
  reference : s7H3GIXlFqFWNbQOExMkzWr4nYFyJJRhyTP2hr8f
  ledger    : INTACT (checked by the broker's open verifier)
  checkpoint: valid signature, covers ledger up to entry 5 (2026-11-19 10:34 UTC)
  events    :
    2026-11-19 10:34 UTC  deposited       alice@example.com
    2026-11-19 10:34 UTC  pickup refused  bob@example.com
    2026-11-19 10:34 UTC  acknowledged    bob@example.com
    2026-11-19 10:34 UTC  picked up       bob@example.com
    2026-11-19 10:34 UTC  file erased     bob@example.com
```

Share your public key without a broker

```
C:\Work> cs keys export-public Alice-Work -o alice.cspub
label: Alice-Work
written_to: alice.cspub
```

```
C:\Carol> cs keys import-public alice.cspub --label Alice-Card
label: Alice-Card
kind: cs3
private: False
email: alice@example.com
name: Alice Martin
verified: False
safety_number: BDC9 00FD 9F35 F149 863F 1A37 CAF9 FCC0
source: import
published_on: (none)
description: CryptSecure 3 · ML-KEM-1024 + P-384 / X25519 · ML-DSA-87
```

Carol now compares the safety number with Alice before running `cs keys verify Alice-Card`.

Protect, back up and restore keys

```
KEY PROTECTION AND BACKUP
C:\Work> cs keys protection
level: Strong
set_up: True
methods:
  - id=s3b1d0f, name=Security key, level=Strong, use=daily, added=2026-11-21, kinds=fido2
  - id=s9e44a2, name=Spare key, level=Strong, use=daily, added=2026-11-21, kinds=fido2
unlock_minutes: 15

C:\Work> cs backup make -o E:\Alice.csrecovery
written_to: E:\Alice.csrecovery
recovery_code: A4HJ-V8E6-AR2D-QKW7-2SS5-J1CW-EAPP-2D8P-03KA-MXD4-E5JP-EKGE-XNW0-K1WM
important: Write the recovery code on paper now. It is shown only once. ...

C:\Work> cs keys backup Alice-Work
Error (usage): Unprotected key backups were replaced by recovery packages. Use: backup make -o
    FILE

ON THE NEW COMPUTER
C:\Users\Alice> set RC=A4HJ-V8E6-AR2D-QKW7-2SS5-J1CW-EAPP-2D8P-03KA-MXD4-E5JP-EKGE-XNW0-K1WM
C:\Users\Alice> cs backup restore E:\Alice.csrecovery --code-env RC
restored:
  - Alice-v26
  - Alice-Work
```

Bring in an older key on the same email address

Alice already has her CryptSecure 3 key, Alice-Work. She brings in her old 2.6 RSA key pair so that files sent to it years ago still open:

```
C:\Work> cs keys import-old --rsa-private E:\2.6-keys\private\Alice-Work_private.pem --rsa-public E:\2.6-keys\public\Alice-Work_public.pem --email alice@example.com --name "Alice Martin (2.6 key)"
label: Alice Martin (2.6 key)
kind: rsa
private: True
email: alice@example.com
name: Alice Martin (2.6 key)
verified: True
safety_number: B458 23ED C442 1526 1971 C2C8 67E0 7D83
source: import
published_on: (none)
description: RSA-4096 (v2.6)
```

Both keys now carry the same email address:

```
C:\Work> cs keys list
keys:
  Alice Martin (2.6 key)  (your key)
     Alice Martin (2.6 key) <alice@example.com>
     RSA-4096 (v2.6)
     safety number B458 23ED C442 1526 1971 C2C8 67E0 7D83

  Alice-Work  (your key)
     Alice Martin <alice@example.com>
     CryptSecure 3 · ML-KEM-1024 + P-384 / X25519 · ML-DSA-87
     safety number 4B1C B614 26E9 3A1A 0145 3435 3176 7710

default_signing_key: Alice-Work
```

New messages to alice@example.com use her CryptSecure 3 key automatically, never the older RSA key:

```
C:\Work> cs encrypt --to alice@example.com -m "Note to self" -o note.csec
  ... Wrapping content key (MLKEM1024-P384) for Alice Martin
  ... Encrypting (AES-256-GCM stream)
  ... Signing (ML-DSA-87)
written_to: note.csec
kind: cryptsecure-v5-mlkem1024-p384
mode: MLKEM1024-P384
signed: True
bytes: 10994
sha256: 3882fd372040ecd8762d3c2bbdb47d4dd1fd7e343fee7407717ac87ee7be4175
recipients:
  - Alice-Work
```

Choosing the public key by mistake is caught with a clear message (exit code 4):

```
C:\Work> cs keys import-old --rsa-private E:\2.6-keys\public\Alice-Work_public.pem
Error (format): Alice-Work_public.pem is a PUBLIC key. Choose your private key file
(usually ..._private.pem) here; the public key is taken from it automatically. To add only
someone's public key, use Import public key under Contacts.
```

When someone's key changes

```
C:\Work> cs lookup bob@example.com
email: bob@example.com
state: changed
safety_number: A45C 77B0 AD37 B3C6 3DF8 07AE 6657 A9FC
algo: cryptsecure-v3
label: bob@example.com
modes: (none)
warning: Key changed. Compare the safety number with the person, then rerun with
    --accept-changed.

C:\Work> cs submit --to bob@example.com -m "hi"
  ... Fetching bob@example.com's public key
Error (key_changed): bob@example.com's key has CHANGED since you last used it. Compare the new
    safety number (A45C 77B0 AD37 B3C6 3DF8 07AE 6657 A9FC) with them before sending.
(exit code 7)

C:\Work> cs lookup bob@example.com --accept-changed
email: bob@example.com
state: verified
safety_number: A45C 77B0 AD37 B3C6 3DF8 07AE 6657 A9FC
algo: cryptsecure-v3
label: bob@example.com
modes:
  - MLKEM1024-P384
  - MLKEM768-X25519
  - MLKEM1024
```

Alice only accepted the new key after Bob confirmed the new number on the phone.

Offline queue

```
C:\Work> cs submit --to bob@example.com -m "Sent from the train"
  ... Fetching bob@example.com's public key
  ... Wrapping content key (MLKEM1024-P384) for Bob Chen
  ... Encrypting (AES-256-GCM stream)
  ... Signing (ML-DSA-87)
  ... Uploading to Main and checking SHA-256
queued: True
queue_id: 20261113-085518-757b84
kind: cryptsecure-v5-mlkem1024-p384
sha256_local: b01ac8f87155ebbd48129c08b9d34526ed7abac1863c545ec07a7c09839c30bc
mode: MLKEM1024-P384
signed: True
broker: Main

C:\Work> cs outbox
queued:
  20261113-085518-757b84  ·  to bob@example.com  ·  queued 2026-11-13 13:55 UTC

C:\Work> cs outbox --flush
sent:
  delivered 20261113-085518-757b84  ·  uid wwYq3zyf8t637tYp0CYDqfb3iGWvVA95yR8vnOkI  ·  SHA-
256
      receipt matches
```

The deposit was queued because the broker could not be reached and queuing is on in Settings. `outbox --flush` sent it once the connection returned.

Errors you may see

```
C:\Work> cs decrypt forecast.csec
  ... Reading envelope
Error (key_missing): None of your keys can open this envelope. It was encrypted for ED1F 92BB
    2FC6 D2BA.
(exit code 6)

C:\Work> cs decrypt wire.csec
  ... Reading envelope
Error (passphrase_required): This file has a passphrase layer. Enter the passphrase and try
    again.
(exit code 6)

C:\Work> cs decrypt tampered.csec
  ... Reading envelope
  ... Unwrapping content key (MLKEM1024-P384)
  ... Decrypting (AES-256-GCM stream)
Error (integrity): Envelope content failed authentication (tampered, reordered or truncated).
(exit code 5)

C:\Work> cs submit --to bob@example.com -m "one more"
Error (csb_quota): You've used all 5 free deposits on Main this month. Your allowance resets
on
    Nov 30. You can still encrypt and save files, and pick up anything sent to you. To
upgrade:
    https://broker.example/upgrade
(exit code 15)

C:\Work> cs broker test
Error (network): Could not reach broker.example.com: [WinError 10061] No connection could be
    made because the target machine actively refused it
(exit code 14)

C:\Work> cs keys delete Alice-v26
Error (usage): Add --yes to confirm deleting the key files.
(exit code 2)
```

The “(exit code …)” lines are added here for clarity; CryptSecure does not print them. Check %ERRORLEVEL% or $LASTEXITCODE.

10.5 Scripting and automation

Use `--json` and the exit code. Every JSON result has "ok"; on failure it also has a stable machine-readable "code" and the "exit_code".

```
C:\Work> cs --json submit --to bob@example.com -m "nightly export" --label Nightly
{
  "ok": true,
  "uid": "bYFIOIaHg0Lklhd84ua6W95ZxsbZAzmdNdROvmr8",
  "deposit_id": 7,
  "sha256": "d8d2d40b9081ebc481788301cfdc87bf9be03a3f5dc4902e97eb3a2210387199",
  "kind": "cryptsecure-v5",
  "sha256_local": "d8d2d40b9081ebc481788301cfdc87bf9be03a3f5dc4902e97eb3a2210387199",
  "sha256_match": true,
  "mode": "MLKEM1024-P384",
  "signed": true,
  "broker": "Main",
  "recipients": [
    "bob@example.com"
  ],
  "kind_local": "cryptsecure-v5-mlkem1024-p384",
  "queued": false,
  "quota": null
}
```

Example: nightly deposit from a batch file

NIGHTLY.CMD

```
@echo off
rem Encrypt and deposit tonight's export. Stop on any error.
cs --quiet submit --to finance@example.com -a "D:\Exports\nightly.csv" --label "Nightly
export"
    --require-verified
if errorlevel 15 (echo Monthly allowance used up & exit /b 15)
if errorlevel 1 (echo Deposit failed with code %ERRORLEVEL% & exit /b %ERRORLEVEL%)
echo Delivered.
```

`if errorlevel N` is true for N and above, so test the highest code first. Run the script under an account whose CryptSecure data folder already holds the broker and a verified contact for the recipient.

Example: pick up everything waiting (PowerShell)

PICKUP-ALL.PS1

```
$items = (cs --json inbox | ConvertFrom-Json).inbox | Where-Object can_pickup
foreach ($i in $items) {
    cs --quiet pickup $i.uid -o "D:\Incoming" --ack
    if ($LASTEXITCODE -ne 0) { Write-Warning "Pickup of $($i.label) failed:
$LASTEXITCODE" }
}
```

10.6 Exit codes

Code	Meaning	Typical cause
0	Success	
1	General error	File not found, disk error.
2	Usage	Unknown command or option, missing `--yes`, a key that already exists.
4	Format	The input is not a CryptSecure envelope, or is damaged.

Code	Meaning	Typical cause
5	Integrity	Wrong key or passphrase, or the data was tampered with.
6	Key missing	No key that can open the file; unknown recipient; passphrase required.
7	Policy	A recipient's key changed; the combination of recipients is not allowed; a 2.6 recipient with a broker that refuses the 2.6 format.
8	Dependency missing	A cryptographic component is unavailable (a damaged installation).
9	Cancelled	Stopped with Ctrl+C.
10	Broker error	The broker returned an error not listed below.
11	Broker authentication failed	Wrong or revoked application password.
12	Broker license inactive	The broker's own license has lapsed.
13	Rejected by broker policy	Too large, too many recipients, plaintext refused, or not permitted.
14	Network unreachable	No connection, or a refused or expired certificate.
15	Monthly deposit allowance used up	Free account; resets at the start of the next month.
16	Keys locked	Your keys were not unlocked: no answer, wrong passphrase or PIN, or the security key was not there.
17	Approval refused	The approval was refused, so nothing was done.
18	Made by a newer CryptSecure	Update CryptSecure to open this file. Nothing was changed.

11 Security model and good practice

11.1 What CryptSecure protects against

- **Interception in transit and at rest.** Envelopes are encrypted before they leave your computer. A broker, mail server, cloud drive or network observer sees only ciphertext.
- **Future quantum computers.** Hybrid post-quantum key encapsulation protects against “harvest now, decrypt later”.
- **Tampering.** Authenticated encryption detects any change, reordering or truncation. Nothing partial is ever shown.
- **Impersonation.** ML-DSA-87 signatures, safety numbers and key-change warnings show who sent a message and stop you encrypting to a substituted key.
- **A compromised or dishonest broker.** The broker never holds keys. Its signing key is pinned, uploads are checked by SHA-256 receipt and custody can be verified independently.

11.2 What it cannot protect against

No application can protect information on a computer that is already compromised. Be aware that:

- Malware, screen capture and keyloggers on your computer, or the recipient’s, can see messages before encryption or after decryption. While your keys are unlocked, a harmful program running as you can also use them (Chapter 12).
- Decrypted content is in memory while displayed, and Windows may write memory to disk. Use full-disk encryption such as BitLocker, and press Esc to clear when done.
- Labels, matter references, file sizes, sender and recipient addresses and timing are visible to the broker. Keep sensitive details in the encrypted message, not the label.
- Anyone who has both your recovery package and its recovery code can take your keys. Keep them apart (Chapter 12).
- A message is only as safe as the key you chose. Verify safety numbers for anything sensitive.

11.3 Good practice

1. Create a CryptSecure 3 key and ask your contacts to do the same. Retire RSA-only 2.6 keys.
2. Verify safety numbers with the people you exchange sensitive material with.
3. Use an application password for each broker, and revoke it if a computer is lost.
4. Make a recovery package, write the recovery code on paper and keep the two apart. Check your recovery once a year. Add a spare security key (Chapter 12).
5. Keep Windows up to date, use BitLocker and lock your screen when you step away.
6. Use `--require-verified` in scripts that send automatically.

12 Your keys: security, backup and recovery

Your private key is what opens every message sent to you. CryptSecure 3.0.5 locks it to this computer and to you, lets you back it up safely, and tells you exactly when access can be lost for good.

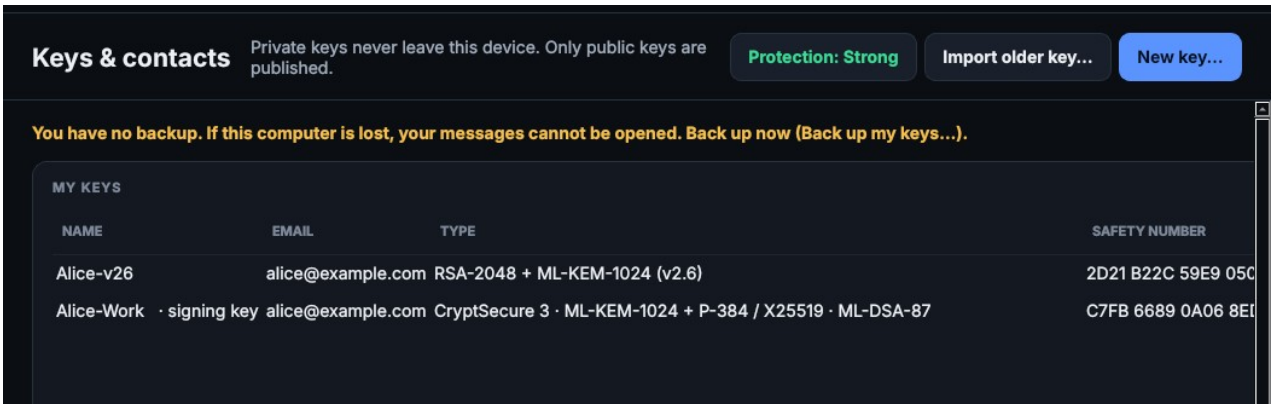
> **WARNING**
>
> **There is no back door.** If you lose your unlock method and you have no recovery package with its recovery code, nobody can recover your keys: not your IT department, not the broker, not CryptSecure. Messages sent to those keys can then never be opened.
>
> Make a recovery package on the first day (section 12.5) and keep the recovery code on paper, away from the computer.

12.1 How your keys are protected

Each private key is kept in its own protected key file (`.cskey`) in your key folder. The file is encrypted with AES-256-GCM; its key is derived with HKDF-SHA384 from a random keyring key that is itself locked by your **unlock method**. Without the unlock method a copy of the file is useless. Your public keys are not secret and are stored as before.

Level	Unlock method	What it means
Strong	A security key (YubiKey or another FIDO2 key with hmac-secret), or this computer's TPM with a PIN	Someone who copies your files gets nothing. While you are logged in, the keys open only with your PIN or a touch of the key.
Medium	This computer's TPM, without a PIN	Your keys open only on this computer. A copied key folder is useless elsewhere.
Basic	A passphrase, or on Windows without a TPM, your Windows account	With a passphrase, a copied file can only be attacked by guessing the passphrase, which is deliberately slow (about one second and 256 MB of memory per guess). With the Windows account, anyone who can sign in to your Windows account can use the keys.

The level shown is the **weakest** way in, because that is the one an attacker would use: if you add a passphrase next to a security key, the level shows Basic. You see the level on the **Protection** button at the top of Keys & contacts.



Keys & contacts with the Protection button and the reminder shown until you make a recovery package.

On Windows, Linux and Raspberry Pi

	Windows	Linux	Raspberry Pi
Recommended	TPM with a PIN, or a security key	A security key	A security key
Also offered	TPM without PIN; passphrase; Windows account only	TPM, only where your user can already use it; passphrase	Passphrase; an add-on TPM board if your user can use it
When nothing else exists	Windows account (Basic)	Passphrase, required (Basic)	Passphrase, required (Basic)

A recovery package made on any of these computers restores on any other (section 12.6).

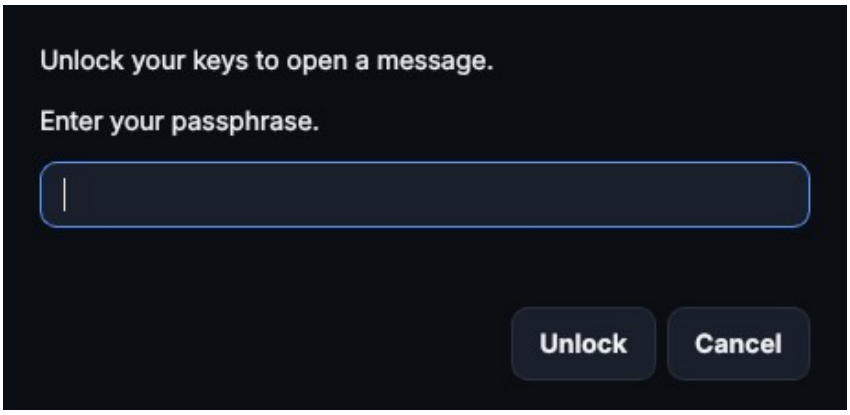
12.2 What protects you from what

This table gives the honest answer for CryptSecure 3.0.5. A “harmful program” means malware you ran by accident, running as you.

Situation	What happens
Laptop stolen while it is off or locked	Protected at Strong and Medium. At Basic, only as strong as your passphrase.
Someone copies your key files or backups (cloud sync, USB drive, IT backup)	Protected at Strong and Medium. At Basic, they can only try to guess the passphrase, slowly.
A harmful program copies your key files	Protected at Strong and Medium: the files are useless on another computer.
A harmful program uses your keys while you are logged in	**Possible** while your keys are unlocked, and at Medium or with the Windows account at any time. At Strong it cannot take the keys off the computer. This is the main limit of 3.0.5.
A harmful program reads CryptSecure’s memory while it opens a message	Possible during use. Keys are opened for each operation and not kept, but the computer itself must be trustworthy.
Someone at your unlocked computer	Can use your keys until the unlock ends (15 minutes unless you change it). Lock your screen: CryptSecure then locks your keys too.
Someone with administrator rights on your computer	Not protected. Keep administrator rights to yourself.
Someone steals your broker password	Cannot read your messages. They could publish a different key under your name: your contacts would see a key-change warning (Chapter 8).
The broker operator, your email provider, CryptSecure	Never have your private key and cannot open your messages.

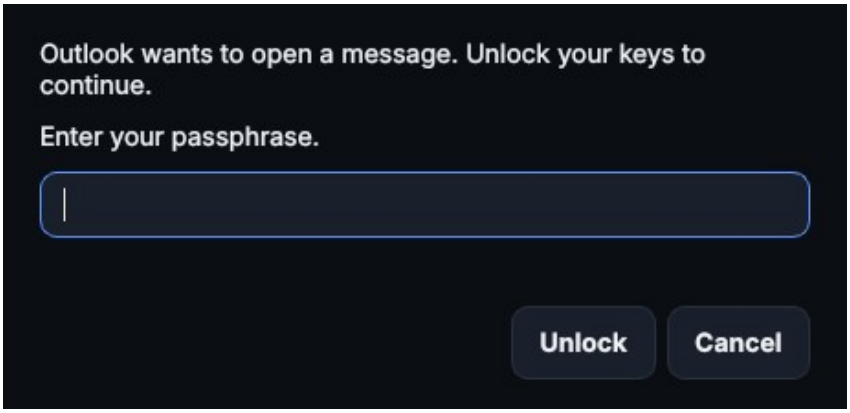
12.3 Unlocking your keys

When CryptSecure needs your private key, to open a message or to sign one, it asks you to unlock your keys. With a security key you touch it; with the TPM and a PIN, Windows asks for the PIN in its own window; with a passphrase, CryptSecure shows this window:



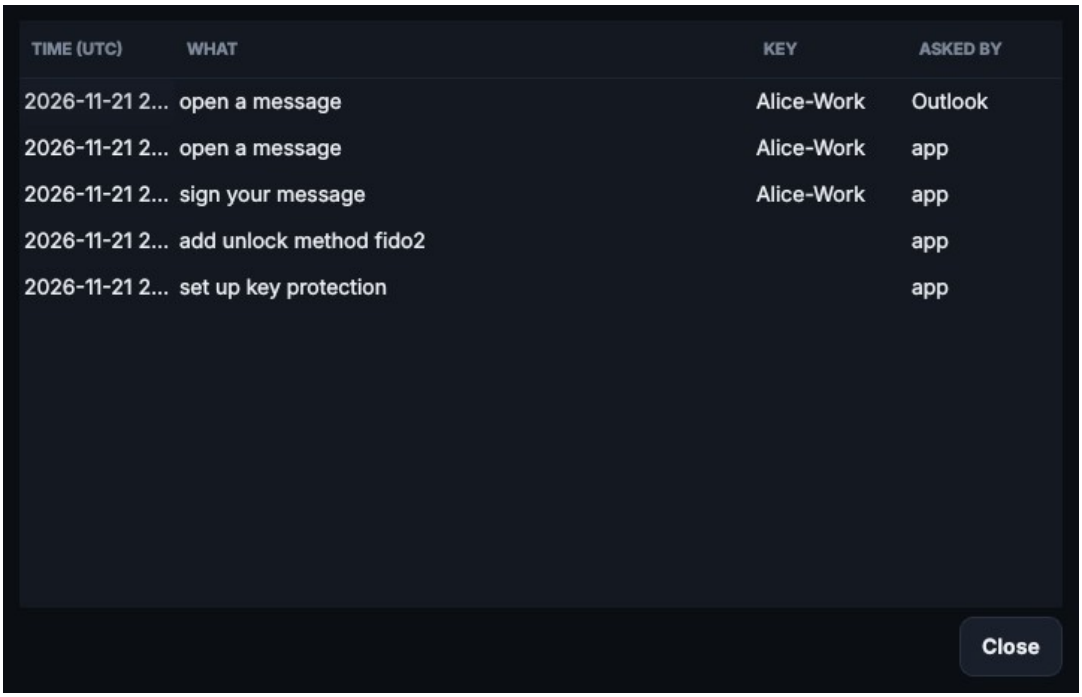
The unlock window says what the key is needed for.

- **In the CryptSecure window, one unlock lasts 15 minutes** by default. You can choose 0 to 60 minutes, or “ask every time”, in the Protection window. Outlook, Word and the command line are separate requests: each asks again when it needs your key (with the TPM without a PIN, the Windows default, nothing is asked).
- **Your keys lock themselves** when the time is up, when you lock your screen, when the computer sleeps and when you close CryptSecure. **Lock my keys now** locks them at once.
- **A wrong passphrase** keeps your keys locked and says so. Nothing is opened. Cancel does the same.
- **Outlook and Word** cannot see your passphrase or your keys. When they need your key, CryptSecure’s own window appears in front of Office and names the program that asks.



CryptSecure’s window when Outlook needs your key.

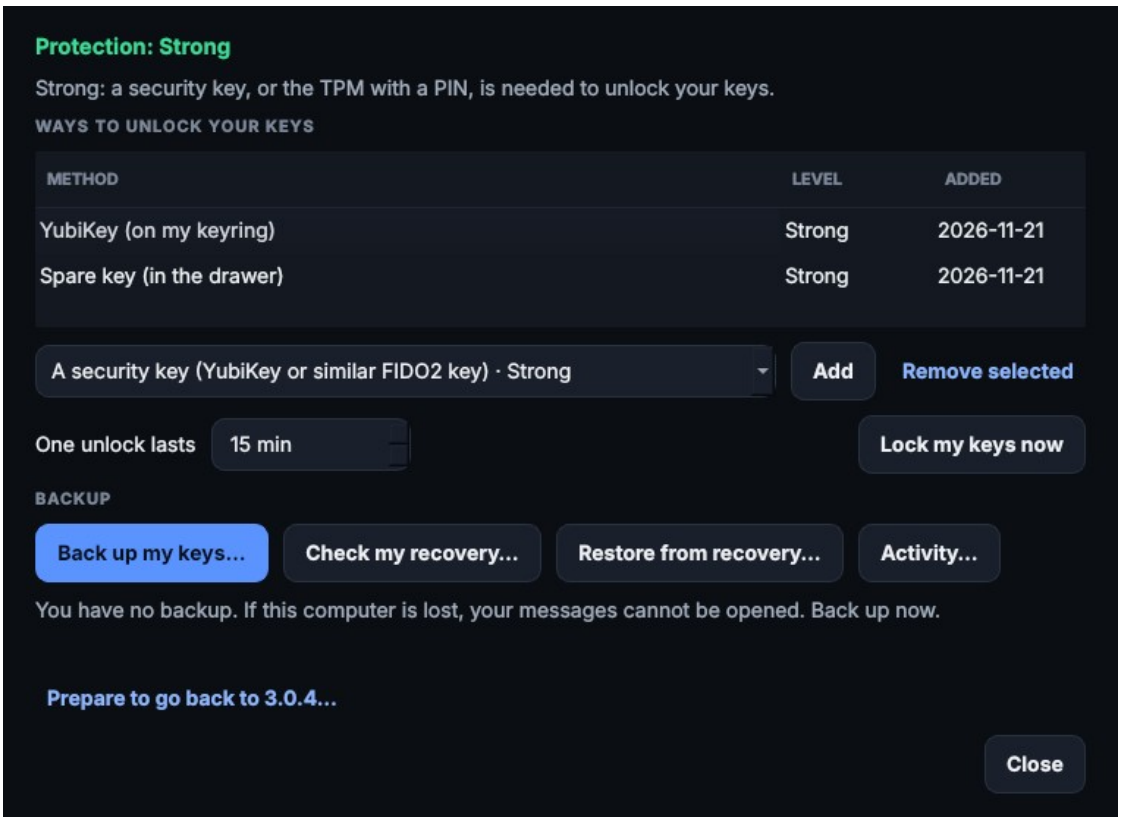
Activity in the Protection window lists when your keys were used, for what, and which program asked (the app, the command line, Outlook or Word). It never records message content. Something you did not do appearing here is a reason to change your passphrase and to check your computer.



Activity: when your keys were used and who asked.

12.4 The Protection window

Choose **Protection** at the top of Keys & contacts.

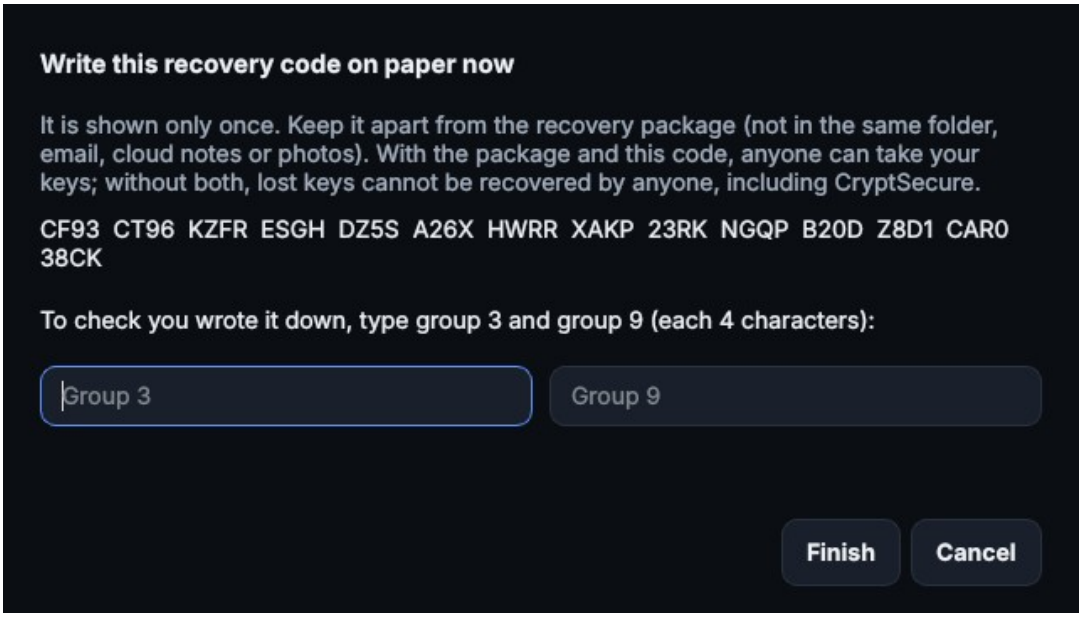


The Protection window: level, ways to unlock, unlock time and backup.

Control	What it does
Ways to unlock your keys	Every unlock method set up on this computer, with its level. Any one of them unlocks your keys.
Add	Adds the method chosen in the list: a spare security key (recommended), the TPM with a PIN, a passphrase. Adding asks you to unlock first.
Remove selected	Removes a method. The last one cannot be removed.
One unlock lasts	0 (ask every time) to 60 minutes.
Lock my keys now	Locks your keys immediately, or, if CryptSecure is using them at that moment, as soon as that operation ends.
Back up my keys… / Check my recovery… / Restore from recovery…	Sections 12.5 and 12.6.
I no longer use CryptSecure 2.6: protect these keys	Shown only if you have CryptSecure 2.6 key files (section 12.10).
Prepare to go back to 3.0.4…	Section 12.10.

12.5 Back up your keys

Back up my keys… (Keys & contacts, or the Protection window) makes a **recovery package**: one `.csrecovery` file with all your private keys, CryptSecure 3 and 2.6, and your broker passwords. It is encrypted with AES-256-GCM under a random **recovery code**, so any change to it is detected when it is opened, and it is signed with your signing key to show which of your keys made it.



The recovery code is shown once. Type groups 3 and 9 to finish.

- The **recovery code** has 14 groups of 4 letters and numbers. It is shown **once**. Write it on paper. To finish, type groups 3 and 9 back; if you cannot, the package is deleted and nothing is left half done.
- Keep the **package** and the **code** in **different places**: for example the package on a USB drive or in your cloud storage, the code on paper in a drawer or a safe. Never keep the code in email, chat, cloud notes or a photo.

- Anyone who has both the package and the code can take your keys. Anyone who has only one of them has nothing.
- **Check my recovery…** opens a package with its code and lists what it would restore, without changing anything. Do this once a year and before any repair of your computer.
- Make a new package after you create or import a key. The reminder on Keys & contacts stays until you have one.

BACK UP AND CHECK FROM THE COMMAND LINE

```
C:\Work> cs backup make -o E:\Alice.csrecovery
written_to: E:\Alice.csrecovery
recovery_code: A4HJ-V8E6-AR2D-QKW7-2SS5-J1CW-EAPP-2D8P-03KA-MXD4-E5JP-EKGE-XNW0-K1WM
important: Write the recovery code on paper now. It is shown only once. Keep it apart from the
  recovery package. Without both, lost keys cannot be recovered by anyone.

C:\Work> set RC=A4HJ-V8E6-AR2D-QKW7-2SS5-J1CW-EAPP-2D8P-03KA-MXD4-E5JP-EKGE-XNW0-K1WM
C:\Work> cs backup check E:\Alice.csrecovery --code-env RC
ok: True
made: 2026-11-21T21:00:45+00:00
by_version: 3.0.5
signed: True
keys:
  - label=Alice-v26, kind=legacy26, fp=c9fae285dca6...
  - label=Alice-Work, kind=identity, fp=bdac85879a2a...
broker_passwords: False
```

12.6 Moving to a new computer and restoring

On the new computer, install CryptSecure and protect it at first start (Chapter 2). Then choose **Restore from recovery…**, pick the `.csrecovery` file and type your recovery code. Every key in the package is added and protected with the new computer's unlock method; keys already there are left as they are. A package made on Windows restores on Linux or a Raspberry Pi, and the other way round.

Because a key file is locked to the computer and user that made it, **copying the key folder to another computer does not work**: CryptSecure says the key belongs to another computer and asks for a recovery package. This is what makes a stolen copy useless.

RESTORE ON THE NEW COMPUTER

```
C:\Users\Alice> cs backup restore E:\Alice.csrecovery --code-env RC
restored:
  - Alice-v26
  - Alice-Work
```

12.7 Importing and exporting keys

You want to	How
Share your public key	**Export public key…**, or publish it to your broker. Public keys are not secret.
Add someone's public key	**Import public key…** under Contacts.
Bring in your own keys from another computer	**Restore from recovery…** with a recovery package.
Bring in a backup made by CryptSecure 3.0.0 to 3.0.4 (`.csid-backup`)	**Restore…** and choose the file. The key is protected as it comes in. Then delete the old file: it is not protected.

You want to	How
Bring in your CryptSecure 2.6 (or 2.4) keys	**Import older key…** with the .pem and .bin files (Chapter 7). They are protected as they come in.
Use one of your 2.6-format keys in CryptSecure 2.6	Select it and choose **For CryptSecure 2.6…**. This writes plain, unprotected key files that 2.6 can read: delete them when you no longer use 2.6.
Export a private key	Only as a recovery package. Unprotected private key files are no longer offered, except the 2.6 files above.

12.8 When you lose access for good

Your keys can only be recovered with a recovery package **and** its recovery code. In every case below, without them, the keys and every message sent to them are lost for good.

What happens	Result without a recovery package and code	How to avoid it
The computer is lost, stolen, broken or reset	All keys lost	A recovery package and code, kept apart from the computer
Windows is reinstalled, the TPM is cleared in the BIOS, or the main board is replaced	Keys protected by the TPM are lost	The same; check your recovery before any repair
Your only security key is lost or broken	Keys locked to that key are lost	Add a spare security key, and keep a recovery package
You forget your passphrase	Keys lost	A recovery package; better, a security key
Your Windows user account is deleted	Keys protected by the Windows account are lost	A recovery package
You uninstall or reset the Microsoft Store app on Windows	Windows deletes the app's data with it, including your key folder: all keys lost	Make a recovery package before you uninstall or reset. (On Linux, removing the package keeps your data folder.)
You lose the recovery code, the computer still works	Nothing lost yet, but no safety net	Make a new recovery package at once
The package and the code are lost, then the computer fails	Keys lost	Check your recovery once a year
The recovery code is shared, photographed to the cloud or kept with the package	Someone else can take all your keys	Keep code and package apart; never send the code
You approve a request you did not start	Someone uses your key for that action	Unlock only when you just asked for something

12.9 Your part in keeping your keys safe

- Make a recovery package on the first day, and a new one after adding a key.
- Write the recovery code on paper and keep it away from the computer and from the package.
- Add a spare security key and keep it somewhere else.
- Check your recovery once a year and before any computer repair.

- Unlock only when you have just asked CryptSecure to do something.
- Lock the computer when you walk away.
- Keep Windows or Linux up to date, and do not run programs from unknown sources.
- Do not use the Basic level on a computer that other people use.
- Do not give anyone administrator rights to your computer.

12.10 Updating CryptSecure and your keys

- **Keys and old messages keep working after updates.** Every key, backup and message made by CryptSecure since 2.6 opens in this and every later version.
- **At the first start of 3.0.5** CryptSecure asks how to protect your keys, moves your 3.0.x keys under that protection and moves your broker passwords out of Windows Credential Manager (or the Linux keyring) into the same protection. A copy of each 3.0.x key file is kept for 30 days in `keys\private\pre-3.0.5` (still protected by Windows as before), then wiped.
- **CryptSecure 2.6 key files** are left where CryptSecure 2.6 reads them and shown as "Basic – also used by CryptSecure 2.6". When you no longer use 2.6, choose **I no longer use CryptSecure 2.6: protect these keys**.
- **Files from a newer version** are refused with "made by a newer CryptSecure, update to open it" and are never changed.
- **Going back to 3.0.4:** choose **Prepare to go back to 3.0.4...** first. It writes your keys and broker passwords again where 3.0.4 reads them (not protected by 3.0.5), then install 3.0.4. When you start CryptSecure 3.0.5 again, it asks whether you stay on 3.0.5; answer **Yes** (or choose **I stay on 3.0.5: protect these copies again** in the protection window) and the copies are protected again. Until then they stay for 3.0.4, at most 30 days.
- Make a recovery package before any major update.

12.11 Security standards

CryptSecure uses the NIST post-quantum standards ML-KEM-1024 and ML-DSA-87 together with AES-256, and is designed to align with NIST, NSA CNSA 2.0 and OWASP guidance. Your private keys never leave your control, and there is no back door. CryptSecure has not been certified under FIPS 140-3 or Common Criteria.

Standard	How CryptSecure 3.0.5 relates to it
NIST FIPS 203 (ML-KEM) and FIPS 204 (ML-DSA)	Uses ML-KEM-1024 and ML-DSA-87, with classical algorithms in a hybrid design.
NSA CNSA 2.0	Messages: ML-KEM-1024 with HKDF-SHA384 (default suite), ML-DSA-87, AES-256. Key protection: AES-256-GCM, HKDF-SHA384, SHA-384. The unlock inputs use other methods set by the hardware or standard concerned: Argon2id for passphrases, RSA-2048 OAEP-SHA256 inside the TPM, FIDO2 hmac-secret for security keys.
FIPS 140-3	Not validated. CryptSecure uses the open-source Python cryptography library.
NIST SP 800-57 Part 1	Follows its key lifecycle: creation, protection, use, backup, recovery and destruction are each defined and logged.
NIST SP 800-63B-4	Unlock uses authenticator types it describes: hardware security keys and a TPM with a PIN.
OWASP ASVS 5.0	Cryptography requirements self-assessed during the release audit; not independently certified.

Standard	How CryptSecure 3.0.5 relates to it
ISO/IEC 27001:2022, control 8.24	Helps organisations meet the control on use of cryptography. CryptSecure itself is not certified.
GDPR Article 32	Supports encryption as a security measure.

“Designed to align with” means the design follows these documents. It is not a certification, and no standards body has approved or endorsed CryptSecure.

12.12 Questions and answers

Question	Answer
Can CryptSecure get my keys back?	No. Only your recovery package and its code can. There is no back door.
Can anyone at CryptSecure, the broker or my email provider read my messages?	No. They never have your private key.
My computer was stolen. What now?	If your keys were Strong or Medium, the thief cannot use them. Restore your keys on a new computer from your recovery package, and revoke the broker application password of the stolen computer.
I think someone learned my passphrase.	Add a new way to unlock (a new passphrase or a security key), then remove the old passphrase in the Protection window. CryptSecure also wipes its own previous copy of the protection file. A copy of your key folder taken earlier (for example by a backup program) still opens with the old passphrase, so if someone may have such a copy, create a new key and tell your contacts.
I think I ran a harmful program.	Lock your keys (Lock my keys now), disconnect the computer from the network and have it checked. Look at Activity for uses you did not make. If in doubt, create a new key and tell your contacts.
Is a passphrase enough?	It protects copies of your files well if it is long and unique. A security key is stronger and easier: there is nothing to type that a keylogger could capture.
What does the broker know about my keys?	Only your public key and your email address.
Why does Outlook ask me to unlock in another window?	So that Outlook, Word and the add-in never see your passphrase or your keys.

13 Standards and algorithms

CryptSecure is built on published, peer-reviewed standards rather than proprietary cryptography. This chapter lists them, what each is used for, and its status as of this release.

13.1 Standards implemented

Standard	Title / subject	Used for in CryptSecure
NIST FIPS 203 (Aug 2024)	Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM)	Post-quantum key encapsulation: ML-KEM-1024 and ML-KEM-768.
NIST FIPS 204 (Aug 2024)	Module-Lattice-Based Digital Signature Standard (ML-DSA)	ML-DSA-87 signatures on envelopes sent with a CryptSecure 3 key.
IETF RFC 9180	Hybrid Public Key Encryption (HPKE)	The framework that wraps each recipient's content key.
IETF draft-ietf-hpke-pq	Post-quantum and PQ/traditional hybrid algorithms for HPKE (Internet-Draft, work in progress)	The MLKEM1024-P384, MLKEM768-X25519 and ML-KEM-1024 HPKE suites.
IRTF draft-connolly-cfrg-xwing-kem	X-Wing: general-purpose hybrid post-quantum KEM (Internet-Draft, work in progress)	The X-Wing protection (ML-KEM-768 + X25519).
NIST FIPS 197	Advanced Encryption Standard (AES)	AES-256 encryption of all content.
NIST SP 800-38D	Galois/Counter Mode (GCM)	Authenticated encryption, applied in 64 KiB segments with a final-segment marker so truncation is detected.
NIST FIPS 180-4	Secure Hash Standard (SHA-2)	SHA-384 fingerprints and safety numbers; SHA-256 deposit receipts.
IETF RFC 5869	HKDF	Deriving the content encryption key, bound to the envelope header.
NIST FIPS 186-5 / SP 800-186	Digital signatures; elliptic-curve domain parameters	The P-384 curve in the CNSA 2.0 hybrid.
IETF RFC 7748	Elliptic curves for security (X25519)	The classical half of X-Wing.
IETF RFC 9106	Argon2 memory-hard function	Argon2id for the optional passphrase layer.
IETF RFC 8017	PKCS #1 v2.2 (RSA-OAEP)	Reading and writing CryptSecure 2.6 RSA formats only.
NIST SP 800-132	Password-based key derivation (PBKDF2)	Opening older 2.x passphrase-protected files only.
IETF RFC 8446	TLS 1.3	Connections to brokers. Full certificate verification is always on.
IETF RFC 4648	Base64	Text (armored) envelopes.

13.2 CNSA 2.0

The NSA's Commercial National Security Algorithm Suite 2.0 specifies ML-KEM-1024 for key establishment, ML-DSA-87 for signatures, AES-256 for encryption and SHA-384 or SHA-512 for hashing. CryptSecure's default protection, labeled **Hybrid post-quantum (CNSA 2.0)**, uses exactly these, with ML-KEM-1024 combined with ECDH P-384, an algorithm from the earlier CNSA 1.0 suite, in a hybrid. CNSA 2.0 does not require a hybrid but permits one. The **Pure ML-KEM-1024** protection is available where a policy calls for ML-KEM alone.

> NOTE
> "CNSA 2.0" in CryptSecure names the algorithm set used. It does not mean that CryptSecure is approved by the NSA for National Security Systems, which requires separate evaluation and validated cryptographic modules.

13.3 NIST IR 8547 and the transition timeline

NIST IR 8547, Transition to Post-Quantum Cryptography Standards (final, October 2026), sets the U.S. federal timeline for moving off quantum-vulnerable public-key algorithms. They are deprecated after 2030 and disallowed after 2035. CryptSecure 3 envelopes already use FIPS 203 and FIPS 204. CryptSecure 2.6 **RSA-only** keys rely entirely on algorithms on that schedule and should be replaced with CryptSecure 3 keys.

The final report accommodates hybrid schemes that pair an approved post-quantum algorithm with a classical one, which is how CryptSecure's default protections are built. U.S. Executive Order 14412 (June 2026) and OMB Memorandum M-26-15 require federal high-value assets and high-impact systems to move to post-quantum key establishment by December 31, 2030, and to post-quantum signatures by December 31, 2031. CryptSecure 3 already does both.

13.4 FIPS 140-3

CryptSecure 3.0.x performs its cryptography with OpenSSL-based components and has **not** been validated under FIPS 140-3. Organizations that require validated cryptographic modules should treat CryptSecure 3.0.5 accordingly.

13.5 About Internet-Drafts

The post-quantum HPKE suites and X-Wing are defined in IETF and IRTF Internet-Drafts that may still change before becoming RFCs. CryptSecure records the suite used in every envelope header, so files made with this release remain readable by future releases even if the drafts change.

14 Compatibility with CryptSecure 2.6

	CryptSecure 3.0.x
Your 2.6 keys	Program-file installs: found and used in place. Nothing is moved, converted or deleted, and the 2.6 key list is backed up once before CryptSecure 3 first writes to it. Microsoft Store edition: import them once with Import older key… (Chapter 2.2).
Opening 2.6 files	Every format opens: CSEC v2 (RSA), v3 (ML-KEM-1024), v4 (RSA + ML-KEM-1024 hybrid), passphrase-protected files (Argon2id or PBKDF2) and pre-CSEC files from version 2.4.
Sending to 2.6 users	Choose their 2.6 key as the only recipient. CryptSecure writes the 2.6 format automatically, so they can open it in 2.6. These messages cannot be signed.
2.6 opening 3.0 files	Not possible. CryptSecure 3 envelopes (CSEC v5) need CryptSecure 3. Ask 2.6 users to upgrade, which keeps their existing keys and files.
Running both versions	Both can be installed side by side during a transition. Program-file installs share one data folder; the Microsoft Store editions each keep their own.

Compatibility in both directions was tested by running the real CryptSecure 2.6.0 program against this release: files made by each were opened by the other wherever the formats allow.

15 Troubleshooting

Problem	Solution
"...security certificate EXPIRED on ..." when adding or using a broker	The broker's certificate has expired. CryptSecure will not connect, because an expired certificate cannot prove the server is your broker. Ask the broker operator to renew it. If the message says this computer's clock is wrong, correct the date and time in Windows.
"...a certificate in its chain was rejected as expired"	This computer's list of trusted authorities is out of date. Run Windows Update, or install the latest CryptSecure release.
Sign-in failed · Fix / exit code 11	The application password is wrong or was revoked. Create a new one in the broker portal and update it in Settings › Brokers (remove and re-add the broker).
Offline · Retry / exit code 14	Check your internet connection and the broker address. Turn on the offline queue if you want deposits held until the connection returns.
I can see my 2.6 keys but cannot encrypt to them	Use **Choose key ▾** on the Encrypt page and pick the key. A 2.6 key can only be the sole recipient of a message.
Encrypt & send is greyed out	Hover over it for the reason: no broker, the monthly allowance is used up, or a 2.6 recipient on a broker that refuses the 2.6 format. Saving and copying still work.
"The message text is ... the limit is 5 MB"	Put large content in a file and attach it.
"None of your keys can open this envelope"	It was encrypted for another key. Compare the characters shown with your safety numbers; ask the sender to encrypt to your current key.
A contact's key has changed	Call them and compare the new safety number before accepting it (Chapter 8.4).
Encrypt & send asks me to trust a key I did not choose	Update CryptSecure to 3.0.4 or newer: a key picked with Choose key ▾ is then the key used. If you typed the address, the broker's key for it really changed: compare the new safety number (Chapter 8.4), or pick the key you want with Choose key ▾.
"... has no account on ... yet, so they could not pick it up"	The key you chose belongs to someone who has no account on this broker, so nothing could be delivered to them. Choose Send invite (type their address and choose Find key), or use Encrypt only (save file) or Encrypt & copy as text and send it another way.
Commands print nothing, or the prompt returns immediately	Use the `cs` helper from Chapter 10.1, which makes Windows wait for the result.
Windows reports missing libraries when starting	The program file is incomplete or damaged. Download or install CryptSecure.exe again.
I installed from the Microsoft Store and my 2.6 keys are missing	The Store editions of 2.6 and 3 keep separate data. Import your keys once with Keys & contacts › Import older key... (Chapter 2.2).
"CryptSecure.exe is not recognized" in a command window (Microsoft Store edition)	In Windows Settings, go to Apps › Advanced app settings › App execution aliases and turn CryptSecure on. Then open a new command window.
"...is a PUBLIC key. Choose your private key file..." when importing an older key	In Import older key..., the RSA private key box needs the file ending in _private.pem. Put the _public.pem file in the RSA public key box, or leave that box empty.
"These keys are already in your key store, as ..."	That key was imported before. Nothing more to do; to attach an email address to it, select it and choose Set email....

Problem	Solution
“v2.6 keys can only receive single-recipient files. Send to them separately.”	One recipient has only a CryptSecure 2.6 key. Send to that person on their own, or ask them to create a CryptSecure 3 key.

When you contact support, include the output of `cs doctor` (or Settings › About), the exact message you see, and the exit code if you used the command line. Never send your private key, your recovery package or your recovery code.

Appendix A Glossary

Term	Meaning
Application password	A separate password for one program, created in the broker portal and revocable on its own.
Armored text	An envelope written as text between BEGIN and END lines, so it can be pasted into email or chat.
Broker	A CryptSecure server that stores and delivers envelopes, keeps a custody log and hosts a key directory. It never holds private keys.
Chain of custody	A tamper-evident record of what happened to a deposit: deposited, picked up, acknowledged.
CNSA 2.0	The NSA's Commercial National Security Algorithm Suite 2.0 (Chapter 13).
Contact	Another person's public key stored on your computer.
Deposit	An envelope delivered to a broker for one or more recipients.
Envelope	The encrypted result: one file holding the encrypted content, a wrapped key for each recipient and the signature.
Hybrid	Combining a post-quantum algorithm with a classical one, so protection holds unless both are broken.
ML-KEM / ML-DSA	The NIST post-quantum standards for key encapsulation (FIPS 203) and signatures (FIPS 204).
Pinning	Remembering a key the first time and warning if it ever changes.
Post-quantum	Designed to resist attack by large quantum computers as well as conventional ones.
Private key	The half of your key that stays on your computer; it opens messages and signs.
Public key	The half of your key you share so others can encrypt to you and check your signatures.
Safety number	A 32-character summary of a public key used to confirm whose key it is (Chapter 8).
Signing key	Your CryptSecure 3 key used to sign what you send.

Appendix B Files and locations

Location or file	Contents
`%APPDATA%\CryptSecure\`	The data folder. Program-file installs share it with CryptSecure 2.6. With the Microsoft Store edition, Windows keeps it privately under %LOCALAPPDATA%\Packages\OpeyIT.CryptSecurev3.0_...\LocalCache\Roaming\CryptSecure.
`keys\private\<label>.cskey`	Your private keys (CryptSecure 3 and 2.6-format), each in a protected key file (Chapter 12).
`keys\private\_keyring.cskey`	Your ways to unlock (key protection). Never edit it.
`keys\private\_vault.cskey`	Broker application passwords, protected with your keys.
`keys\private\pre-3.0.5\`	Your 3.0.x key files, kept 30 days after updating so you can go back to 3.0.4, then wiped.
`logs\key-use.log`	Activity: when your keys were used and who asked. Never message content.
`keys\public\<label>.cspub`	Public keys: yours and your contacts'.
`keys\…\*.pem, *.bin`	CryptSecure 2.6 keys, left where 2.6 reads them until you protect them (Chapter 12).
`key_registry.json`	The 2.6 key list, still in 2.6 format.
`key_registry.v2.6-backup.json`	One-time backup of the 2.6 key list.
`cs3_registry.json`	CryptSecure 3 key details: verification, sources, publishing.
`settings.json`	Your settings and broker profiles (no passwords).
`outbox\`	Encrypted deposits waiting to be sent.
Windows Credential Manager	Held broker passwords before 3.0.5. Emptied at the first start of 3.0.5.

Extension	File type
.csec	A CryptSecure envelope (binary).
.txt	An armored envelope (CryptSecure 3) or base64 text (CryptSecure 2.6).
.cspub	A public key to share.
.csid-backup	An unprotected key backup made by 3.0.0 to 3.0.4. Restore it, then delete it.
.csrecovery	A recovery package: all your keys, protected by a recovery code.
.cskey	A protected private key file. It opens only on the computer and user that made it.

Appendix C For administrators

Microsoft Store package (MSIX)

The Microsoft Store edition is an MSIX package named **OpeyIT.CryptSecurev3.0** (display name CryptSecurev3.0). It contains the desktop application, named **cryptsecure_v3.0.5.exe** (cryptsecure_v3.0.0.exe in 3.0.0 and 3.0.1; cryptsecure_v3.0.2.exe to cryptsecure_v3.0.4.exe in those releases) and registers the command alias **CryptSecure.exe**, so `CryptSecure.exe cli …` works from any Command Prompt or PowerShell window.

Packaged apps keep a private copy of AppData for each package. The Store editions of 2.6 and 3 therefore do not share keys, and users move them once as described in Chapter 2.2.

Environment variable

Variable	Effect
`CRYPTSECURE_HOME`	Uses another data folder, for example for a service account. Equivalent to `--home` on the command line.

Broker requirements

CryptSecure 3 talks to brokers implementing the cryptsecure/v1 API over HTTPS with a certificate from a publicly trusted authority, for example Let’s Encrypt, including its intermediate certificate. Brokers that publish their capabilities let CryptSecure apply their policy automatically: profile, accepted protections, maximum size, passphrase policy and deposit allowance. For other brokers safe defaults are used.



CryptSecure Encryption Client

POST-QUANTUM BY DEFAULT · ENCRYPTED BEFORE IT LEAVES
NEW IN 3.0.5: YOUR KEYS LOCKED TO YOU

User Guide · Version 3.0.5 · November 2026