Auctio

USER MANUAL

The contracting authority

From the tender key to the opening of the deposited technical bids: every step, with the screens.

Auctio — encrypted deposit of technical bids beyond Sintel's limits · https://auctio.it · manual version: September 2026 · ThinkDifferent S.r.l.

Contents

1. What Auctio is and what it does for you
2. Who does what
3. Account activation and first sign-in
4. The tender: what the operator of the service does, what you do
5. The tender key
6. What to write in the notice and the tender rules
7. During the tender
8. The closure
9. Opening the bids
10. Verifying a receipt
11. After the opening: retention and destruction
12. When something goes wrong
13. Glossary

1. What Auctio is and what it does for you

Auctio is an encrypted deposit that sits *beside* the tender platform. It serves when a technical bid — drawings, models, reports — exceeds Sintel's size limits (600 MB in total per operator, 100 MB per single file). The economic operator deposits the file here, encrypted in their browser with your tender's key; on Sintel they attach the **deposit receipt**, a PDF declaring the digest of what they deposited, sealed by the operator of the service and timestamped.

Three things are guaranteed by construction, not by promise:

- **Secrecy until the deadline.** The tender's private key is born in your browser and never leaves it. The operator of the service keeps envelopes it cannot open.
- **Integrity and certain date.** Every deposit yields a receipt with the file's digest, sealed and timestamped; the closure record lists every deposit and enters a public, unchangeable log anchored on Bitcoin.
- **Verifiable by anyone.** Receipts and records are checked with public tools, asking the platform nothing: years later too, in litigation too.

> **What does not change.** The procedure stays on Sintel: notice, administrative and economic envelopes, offer document, signature, ranking. Auctio only keeps the technical file that is too large, and proves its deposit.

2. Who does what

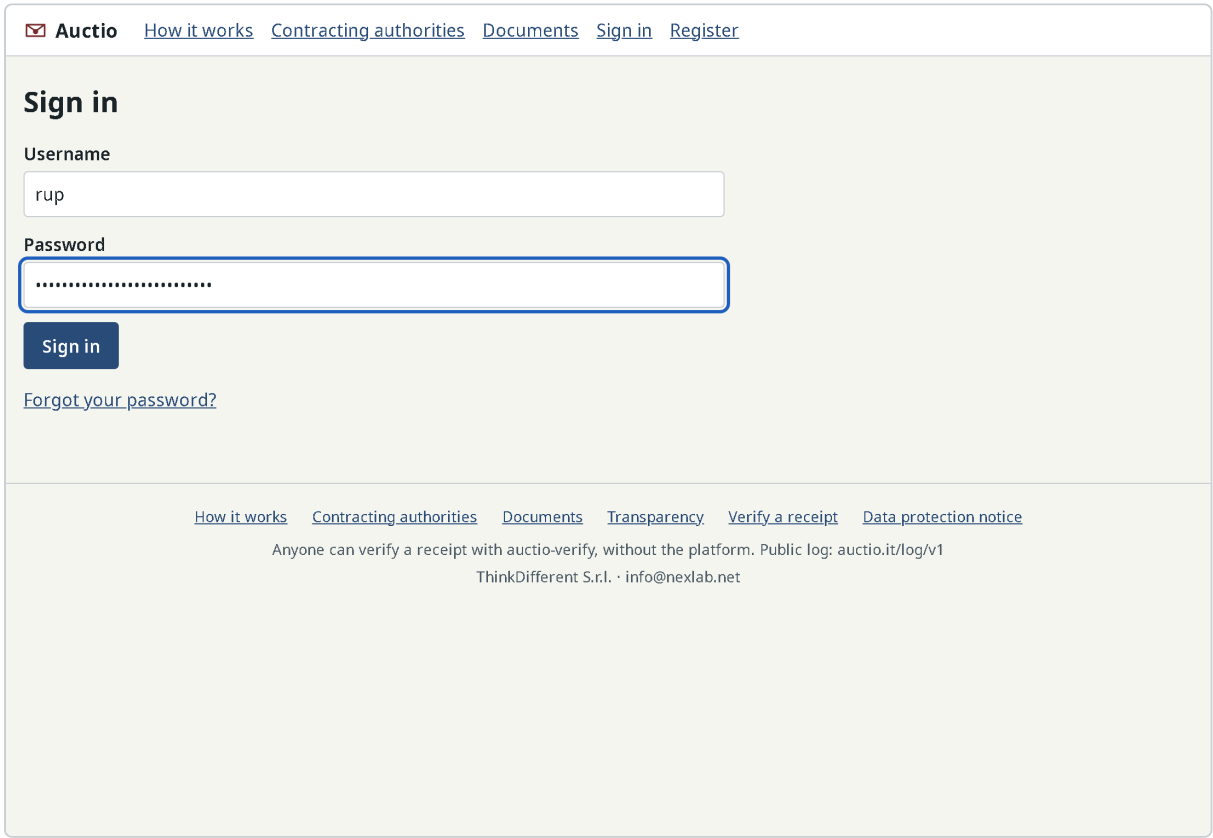
WHO	DOES	WHERE
The **operator of the service**	creates the tender (CIG, subject, deadline, limits), assigns you as contracting authority, opens it to deposits	Auctio, manager's area
The **contracting authority** (you: the RUP or a delegate)	generates the tender key, keeps the key file and the recovery kit, publishes the fingerprint in the notice; after the closure opens the envelopes	Auctio, "My tenders"
The **economic operator**	registers, deposits the encrypted file, gets the receipt and attaches it to the bid	Auctio and Sintel

The operator of the service never sees the content of a deposit: it sees sizes, instants and accounts. You see nothing before the deadline: the tender closes by itself and only then is the opening possible.

3. Account activation and first sign-in

Contracting authorities' accounts are opened by the operator of the service after checking the body and its contact person: there is no self-registration. Write to the address on auctio.it/stazioni-appaltanti naming the body, the contact person and, if already known, the tender. You will receive a username and an initial password, to change at first sign-in.

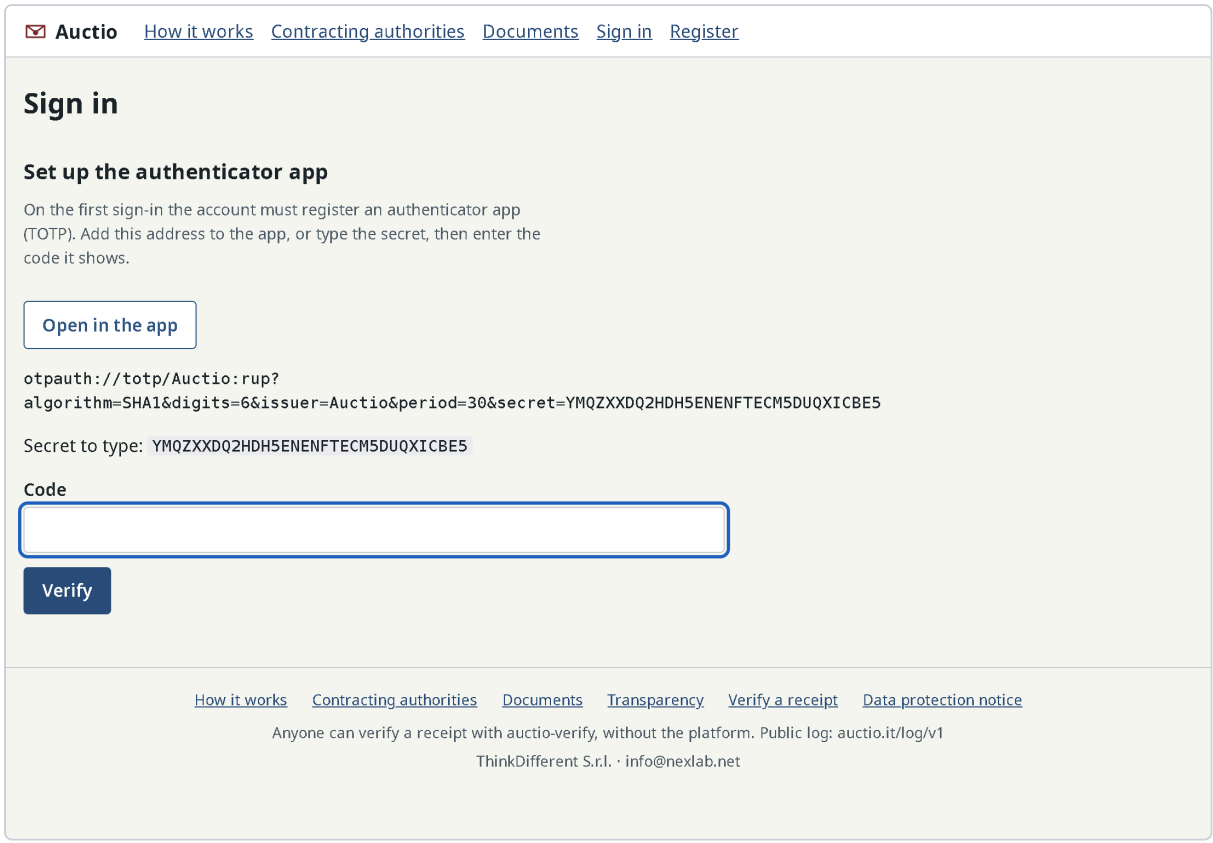
1 **Sign in with username and password.**



auctio.it/login: username and password.

2 **At first sign-in, set up the authenticator app.**

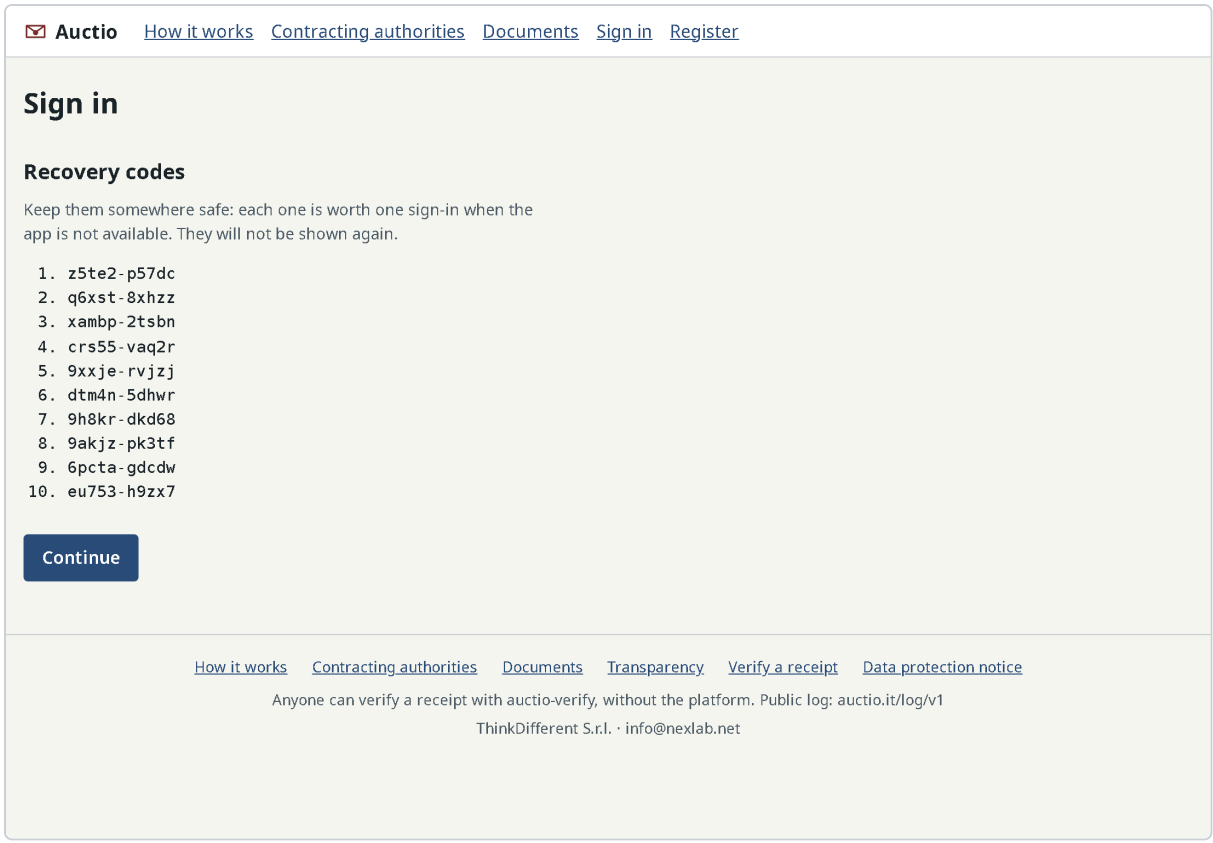
Your account uses a second factor: a six-digit code from an app on your phone (Google Authenticator, Microsoft Authenticator, FreeOTP, Aegis or similar). On the phone, "Open in the app" registers the account directly; from a computer, copy the "Secret to type" into the app. Then type the six-digit code the app shows.



Setting up the authenticator: once.

3 **Keep the recovery codes.**

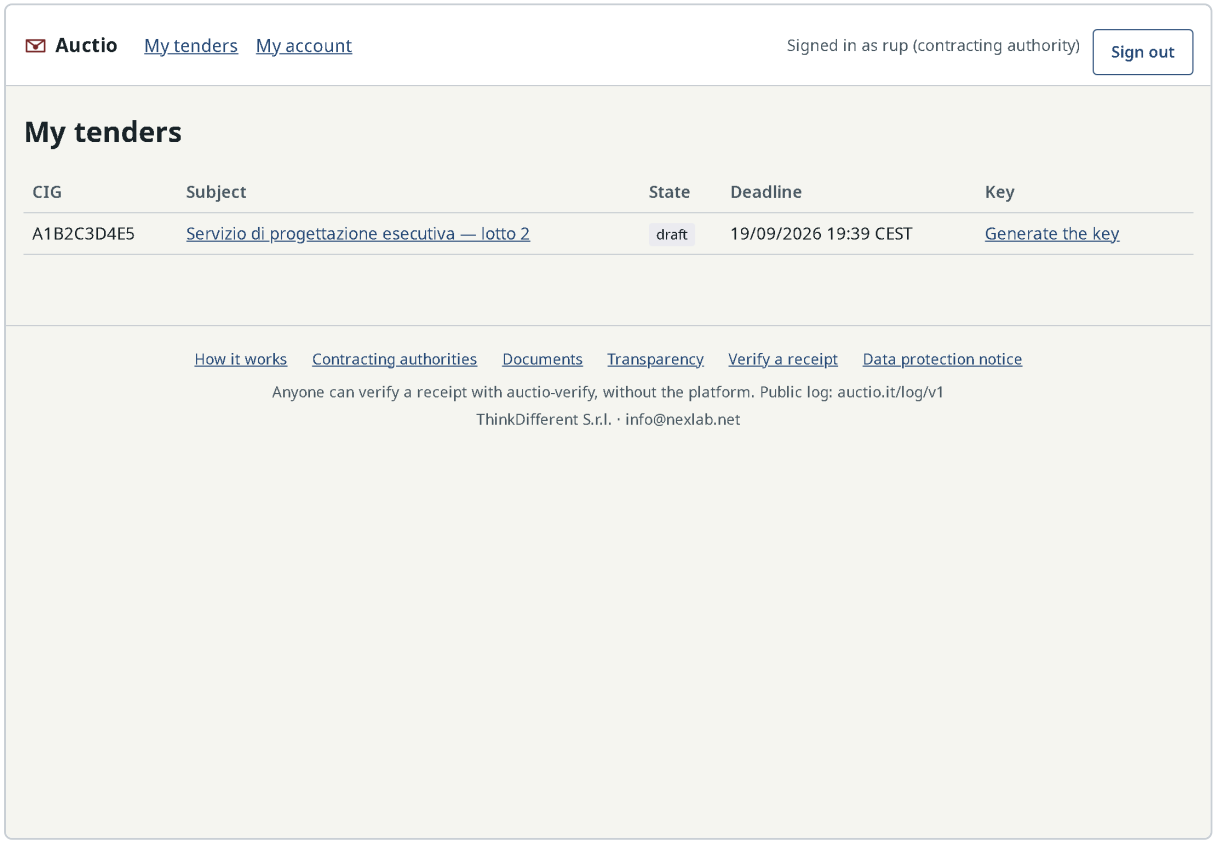
They are shown once. They serve if you lose the phone: each is good for one sign-in. Print them or put them in the body's password manager.



The recovery codes: with no phone and no codes, access is restored by the operator of the service.

4 Change the password

from "My account". Then "My tenders" lists the tenders assigned to you.



"My tenders": state and possible actions for each.

> Later sign-ins ask for the password and the app's code. A session lasts at most twelve hours and ends after thirty minutes of inactivity.

4. The tender: what the operator of the service does, what you do

When you agree a tender with the operator of the service, give them: the **CIG**, the **subject**, the **deadline** for bids (day, hour and minute, Italian time: the same as Sintel's), the **maximum size** per file and the **maximum volume** per operator if you want to limit them, and the **retention period** after the opening (in days), after which the deposited files are destroyed.

The operator of the service creates the tender as a *draft* and assigns you. Then it is your turn: the key (chapter 5). Once the key is published, the operator of the service *opens* the tender to deposits. It stays open until the deadline, then closes by itself.

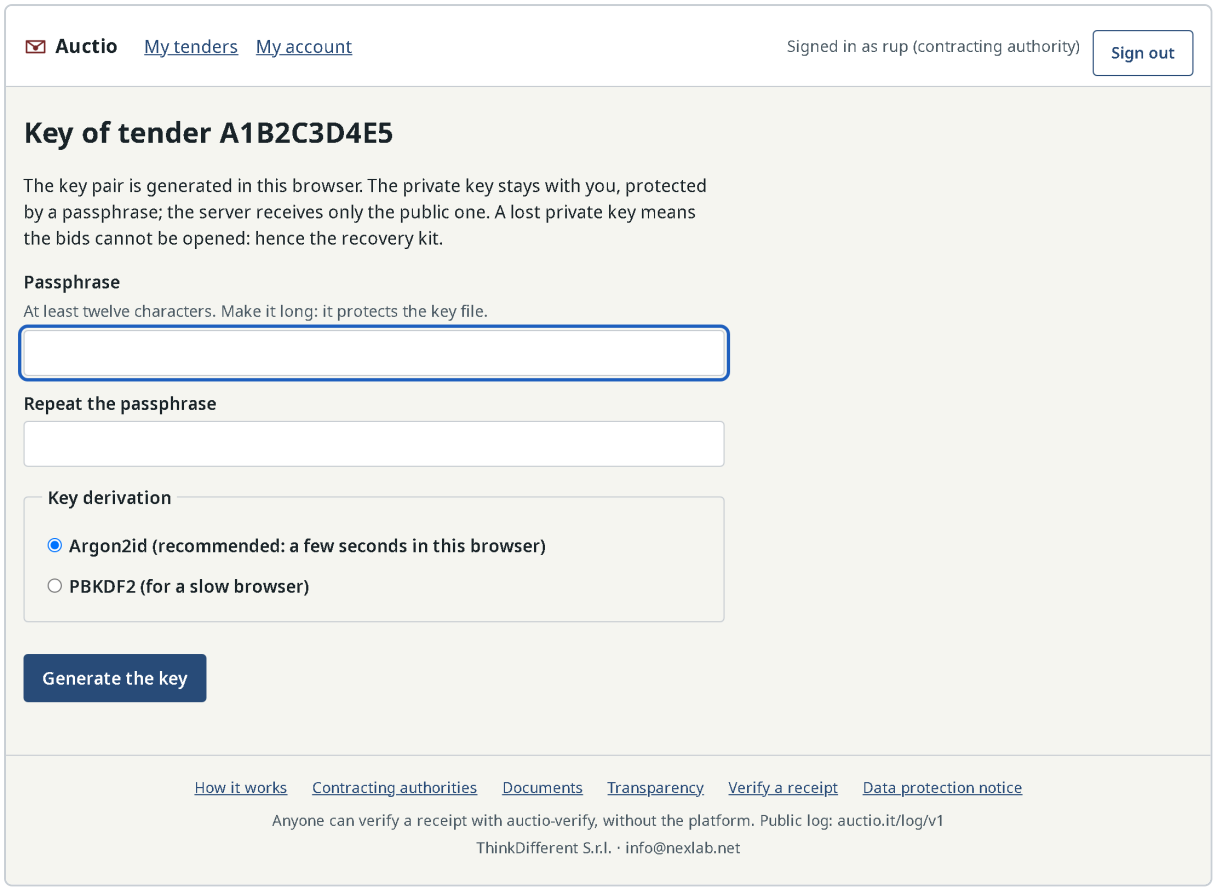
draft	created, not yet visible to operators; here you generate the key
open	operators deposit; the public page shows the key's fingerprint
closed	the deadline has passed; no deposit enters or leaves; the record is issued
opening	you are downloading and decrypting the envelopes
retention	opening completed; the files remain for the period set
destroyed	the files have been deleted; receipts and record remain

5. The tender key

The key is a pair: a **public** part, which the service receives and operators encrypt with, and a **private** part, which stays with you alone and is the only thing in the world that opens the envelopes. Everything happens in your browser; the service never sees the private part.

> **If the private key is lost, the bids cannot be opened.** Nobody — not the operator of the service, not whoever wrote the software — can recover them. That is why there is a key file *and* a recovery kit, and why this is the longest chapter of the manual.

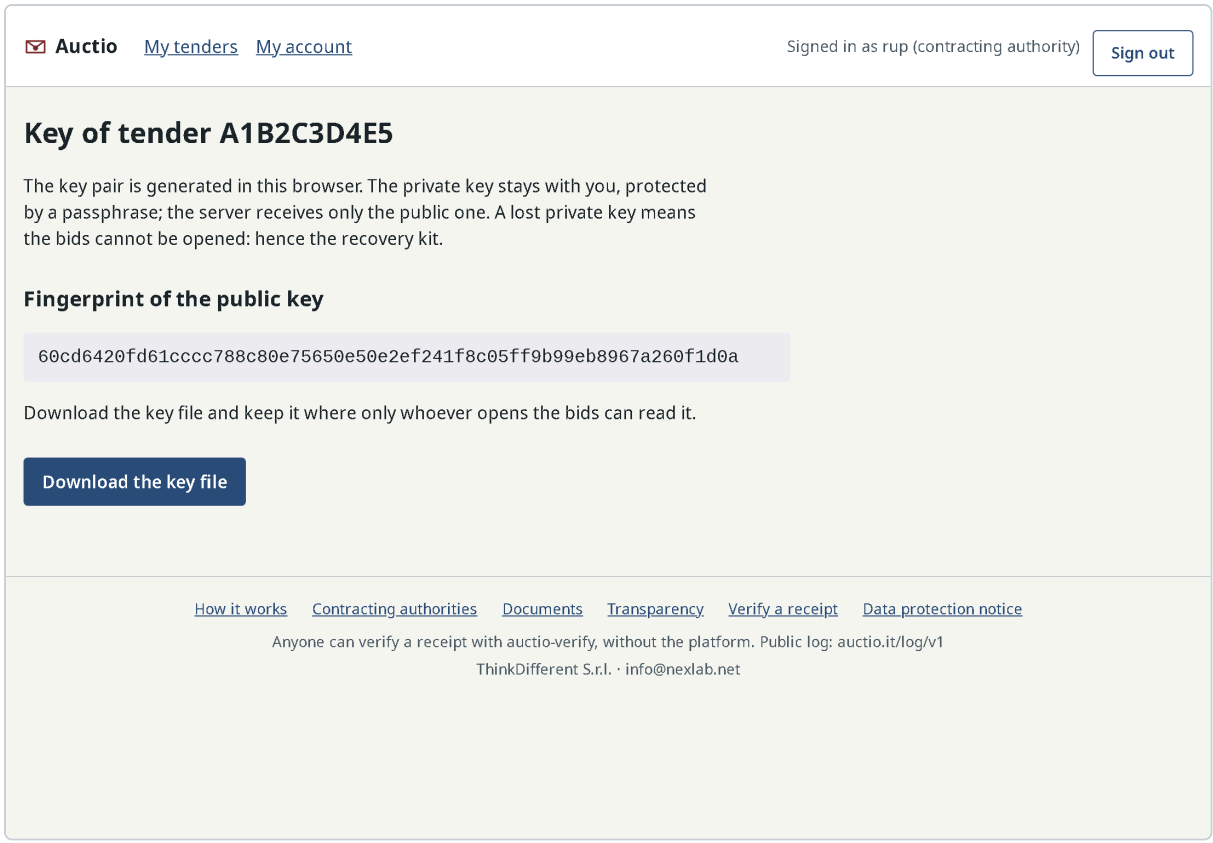
1. **Choose a passphrase.**
   From "My tenders", on the draft tender, "Generate the key". Type a passphrase of at least twelve characters, twice. A phrase of four or five uncommon words beats a short, complicated password: you will type it again at the opening, weeks later.



The passphrase protects the key file; generation takes a few seconds.

2 **Download the key file.**

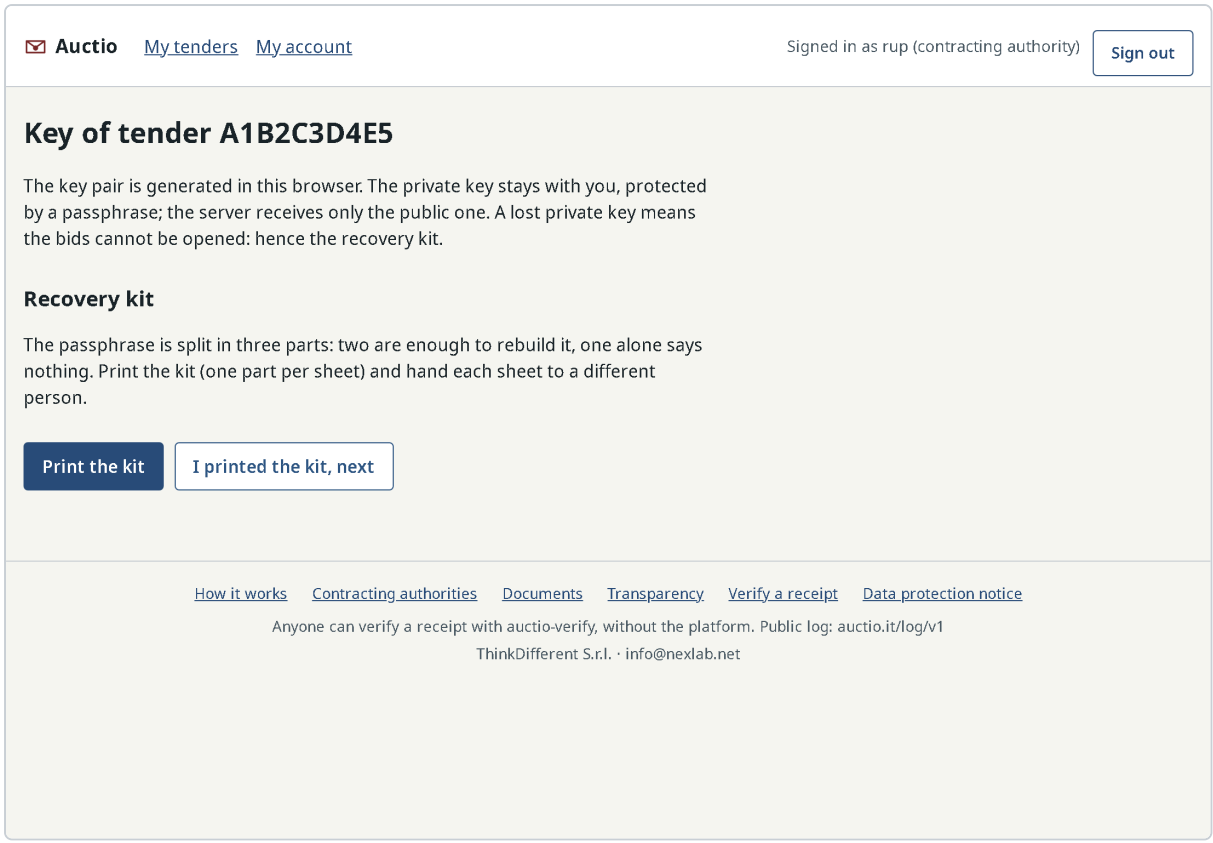
It is called `gara-<CIG>.key`. Copy it to two different places: the tender folder on the body's server and a USB key kept under lock. This screen also shows the **public key's fingerprint**: the one for the notice (chapter 6).



The fingerprint (64 hexadecimal characters) and the key file download.

3 **Print the recovery kit.**

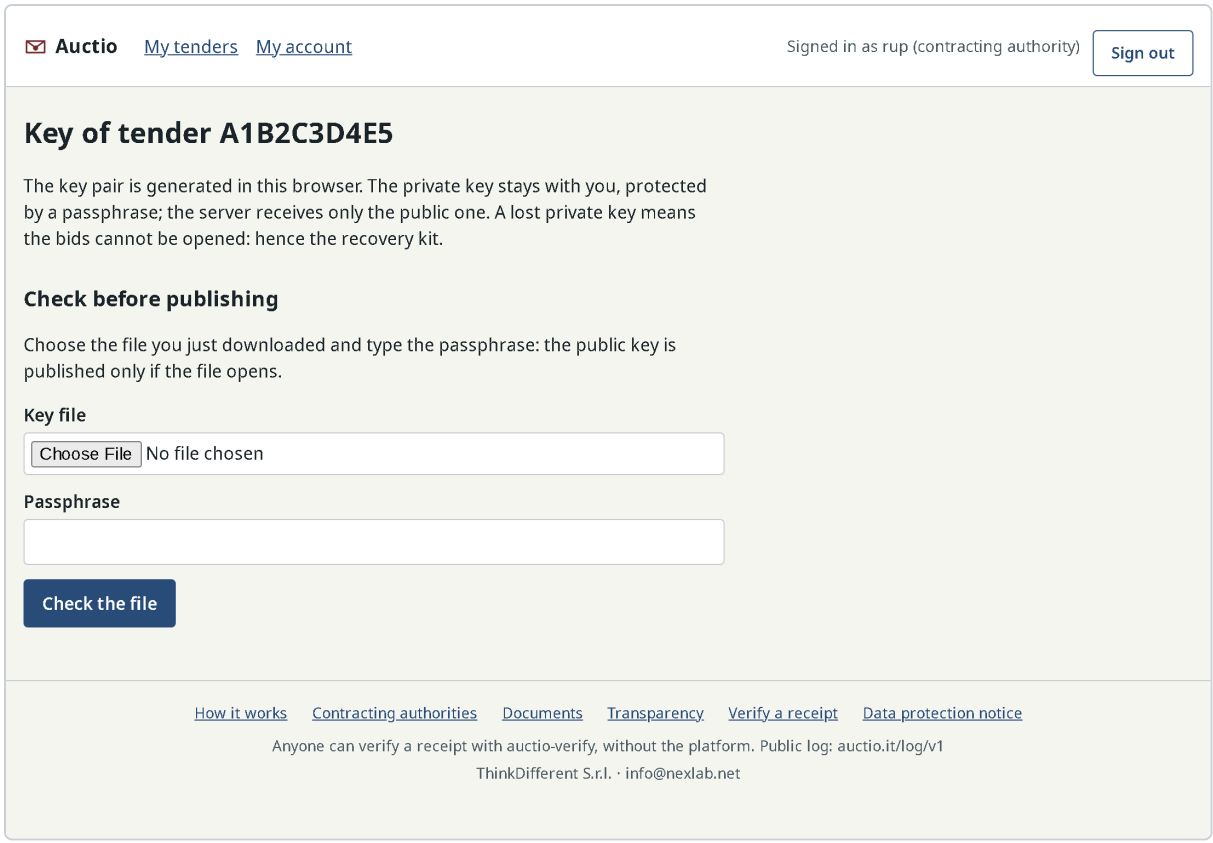
The passphrase is split into three parts: two rebuild it, one alone says nothing. "Print the kit" gives one sheet per part; then "I printed the kit, next". Hand each sheet to a different person (RUP, secretary, management) or seal it in three different places. Whoever forgets the passphrase rebuilds it with two sheets.



The kit: three sheets, two are enough. Print it before going on.

4 Verify the file.

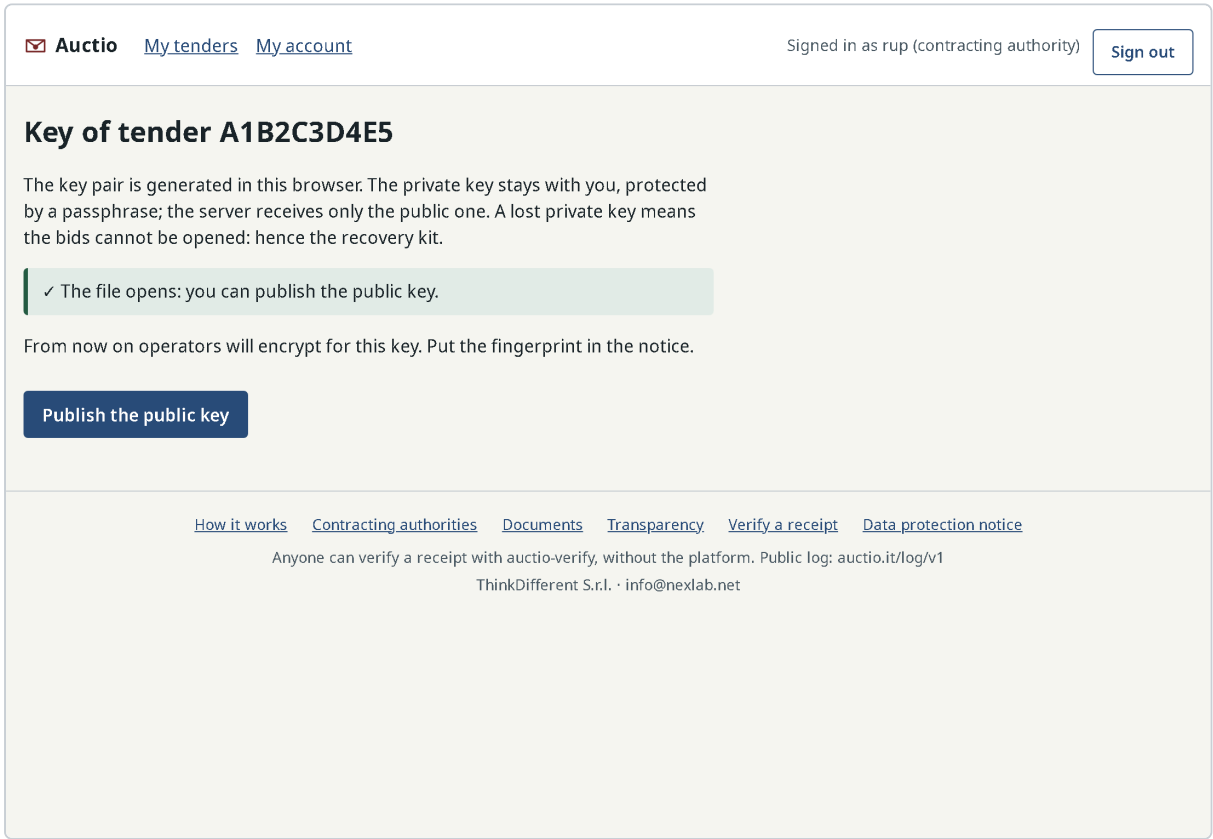
"Verify before publishing": the page asks you to load the file you just downloaded and to type the passphrase again, then "Verify the file": it proves the file opens and is the right one. If it does not open, start again from step 1: the key is not published yet, nothing is lost.



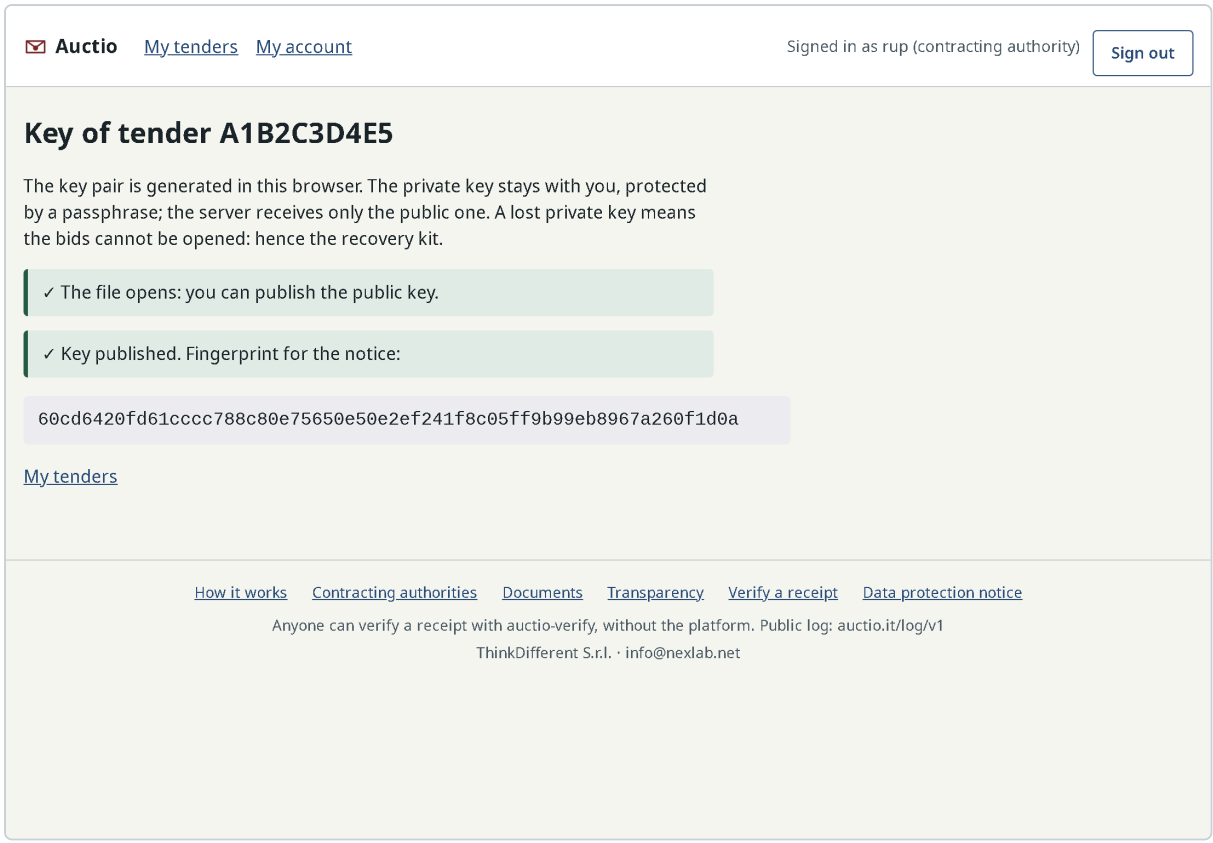
The opening rehearsal: the file and the passphrase you will use at the opening.

5 Publish the public key.

From now on the fingerprint is on the tender's public page and the key can no longer change: bids will be encrypted for it.



Publication is final.



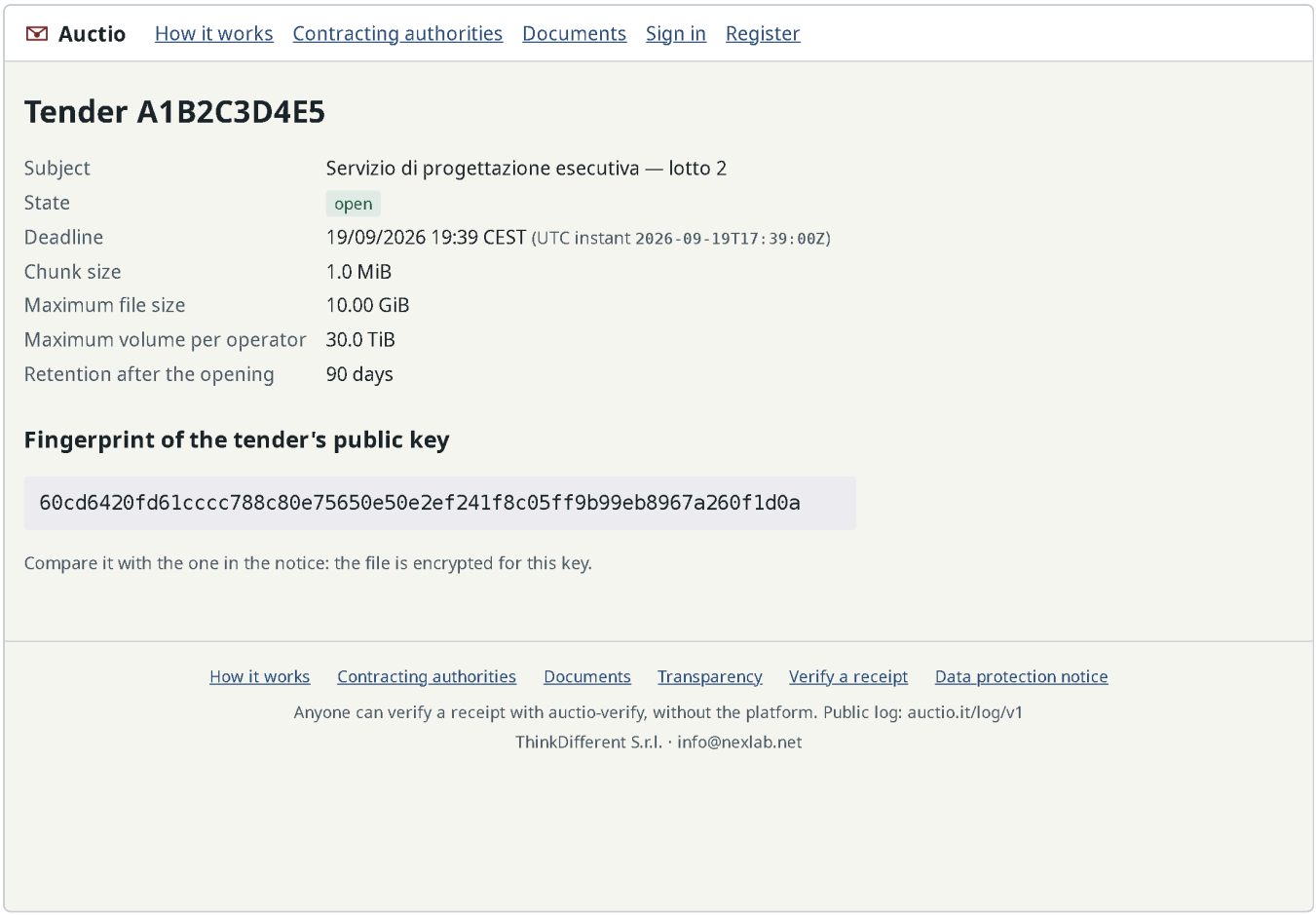
Key published: the fingerprint is the notice's.

> **Checklist before telling the operator of the service "open the tender":** the `.key` file in two places · the three kit sheets with three people or in three places · the file verification passed · the fingerprint copied into the notice.

6. What to write in the notice and the tender rules

Three things, all on the tender's public page (`auctio.it/t/<identifier>`, which the operator of the service gives you):

1. the **address of the tender's page** on Auctio, where the operator deposits;
2. the **fingerprint of the tender's public key**, so the operator checks they encrypt for you and nobody else;
3. the **obligation to attach the deposit receipt**, as PDF, in the technical envelope on Sintel, signed like the other documents of the bid.



The tender's public page, visible to all: CIG, subject, deadline (Italian time and UTC instant), fingerprint, limits.

A clause for the tender rules

```
Technical documentation exceeding the size limits of the Sintel platform.
The economic operator deposits the file, encrypted for the public key of this procedure
(SHA-256 fingerprint: <fingerprint>), on the Auctio service at <https://auctio.it/t/…>,
within the deadline for the submission of bids. The deposit receipt issued by the
service, as PDF, is attached to the technical envelope on Sintel and digitally signed
like the other documents of the bid. The deposited file is an integral part of the
technical bid; it is opened after the deadline by the contracting authority, and the
outcome of the opening is recorded in the minutes.
```

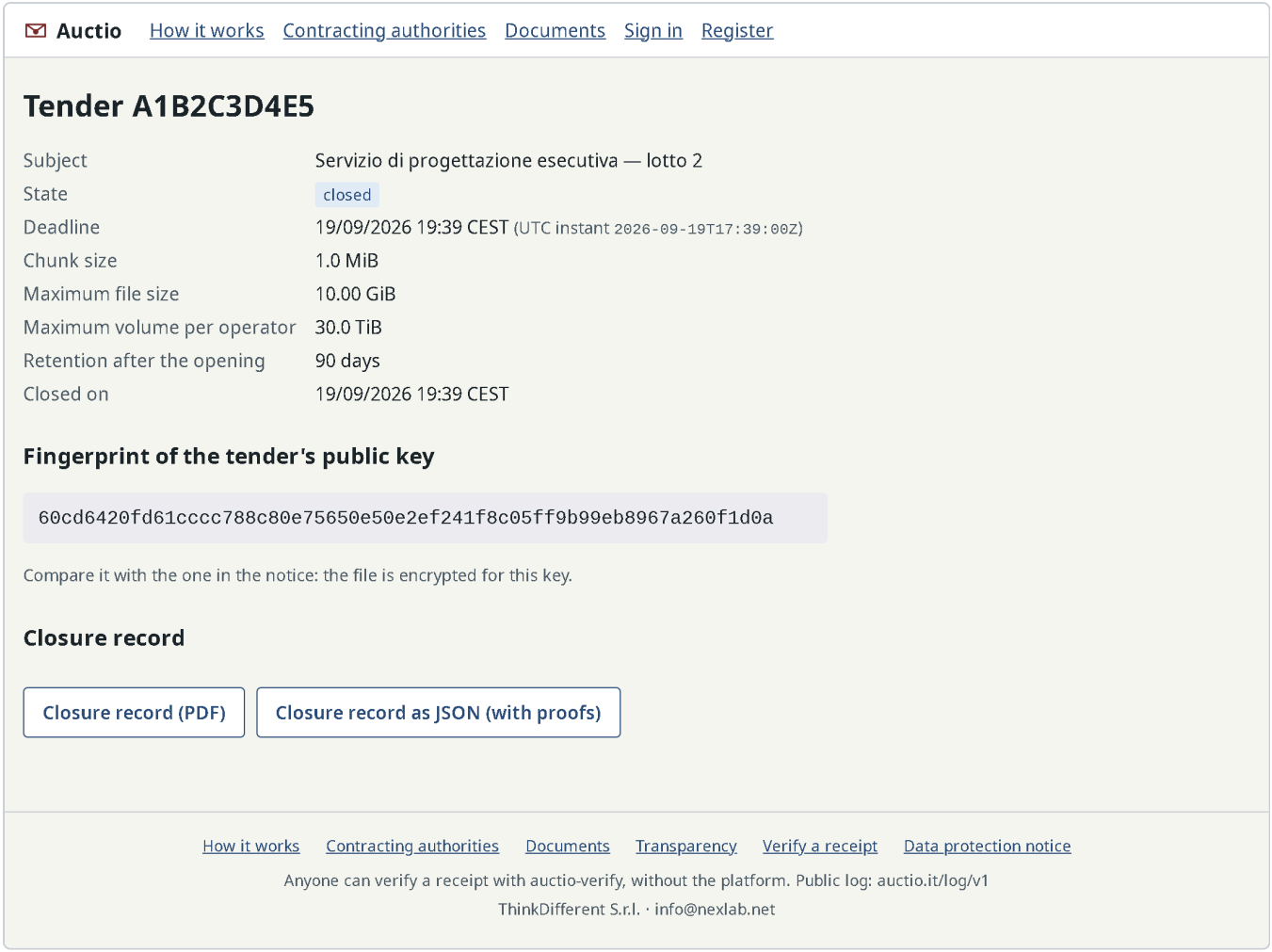
The deadline on Auctio must be Sintel's: give the operator of the service day, hour and minute. An operator depositing after the deadline is refused; one who deposited before but did not attach the receipt on Sintel is in the position of a bid missing an attachment — how you judge it is for the tender rules, not for Auctio.

7. During the tender

Nothing to do, and nothing to see: before the deadline the deposits are visible to nobody, you included. "My tenders" shows the tender as *open*. If the deadline must move (extension), ask the operator of the service: the new deadline appears on the public page, operators who already deposited need do nothing. If the tender is cancelled, the operator of the service cancels it and the deposits are destroyed.

8. The closure

At the deadline, to the minute, the tender closes by itself. From then on no deposit enters or can be deleted. The service issues the **closure record**: the list of completed deposits with their digests, sealed, timestamped and entered in the public log. You get an e-mail notice.



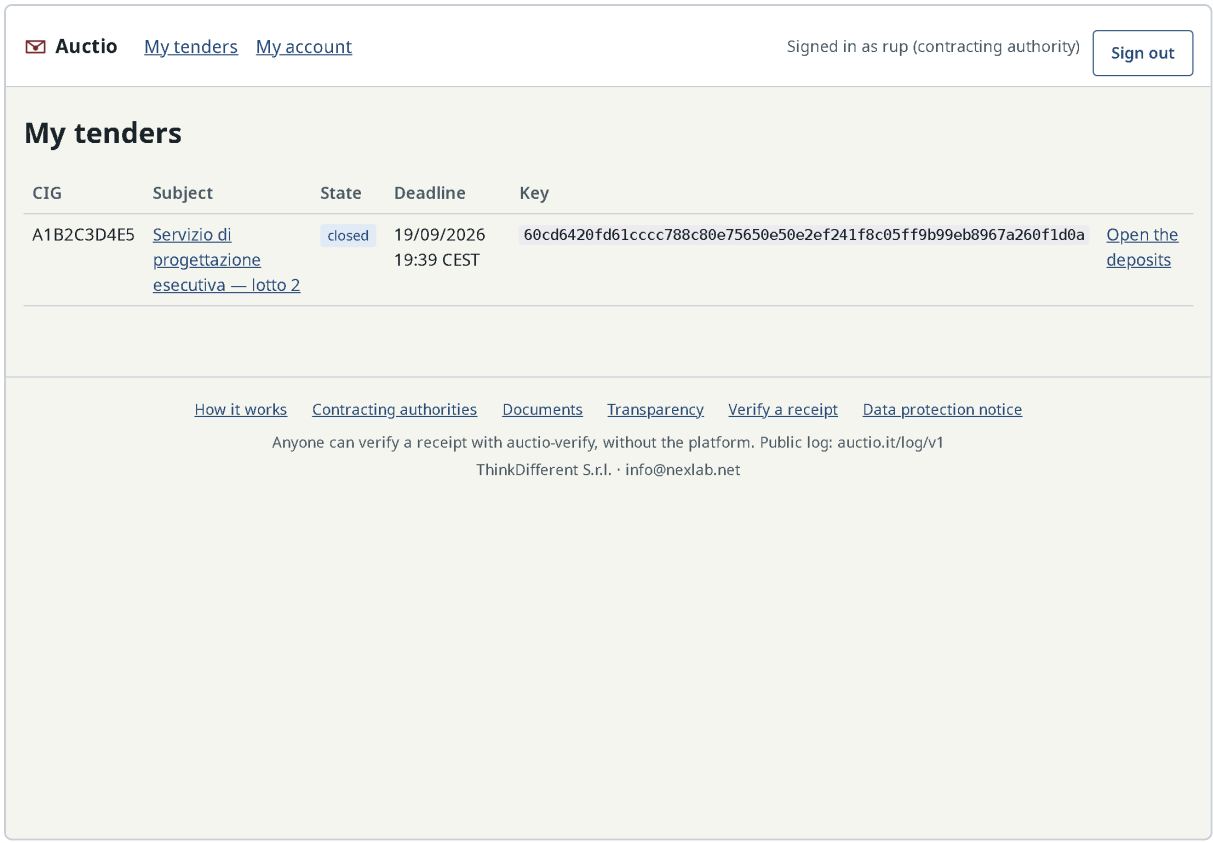
The public page after the closure: the record can be downloaded by anyone, as PDF and as JSON with the proofs.

The record is public and names no operator: it holds identifiers and digests. It is the document with which, in session, you state which deposits arrived within the deadline.

9. Opening the bids

The opening happens on your computer: the envelopes are downloaded and decrypted *in the browser*, with your private key; the files in clear never pass through the service. Do it in the session the tender rules provide for, with the key file (or two kit sheets) at hand.

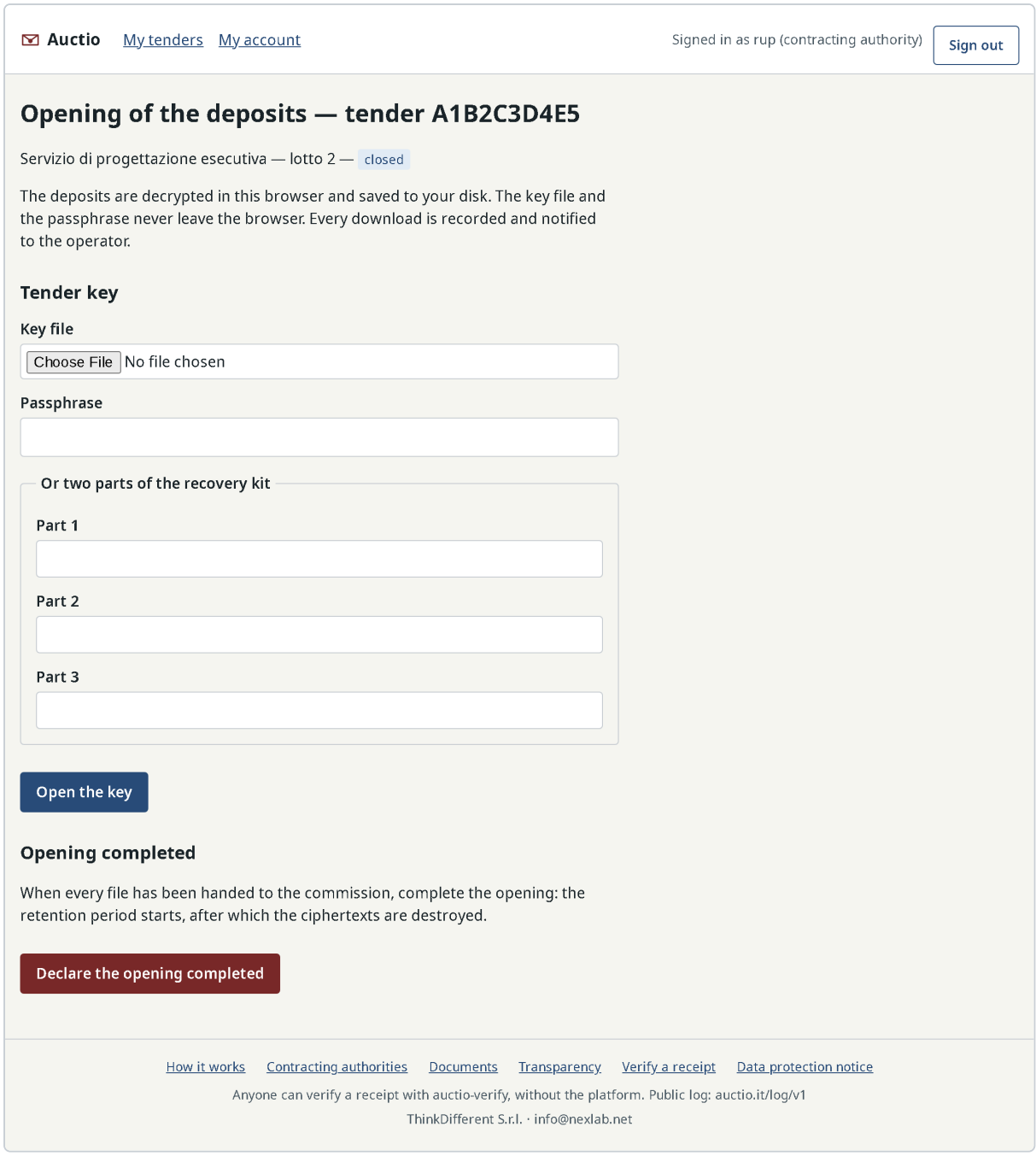
1. **"My tenders" → the closed tender → "Open the deposits".**



The closed tender, ready for the opening.

2 Open the key.

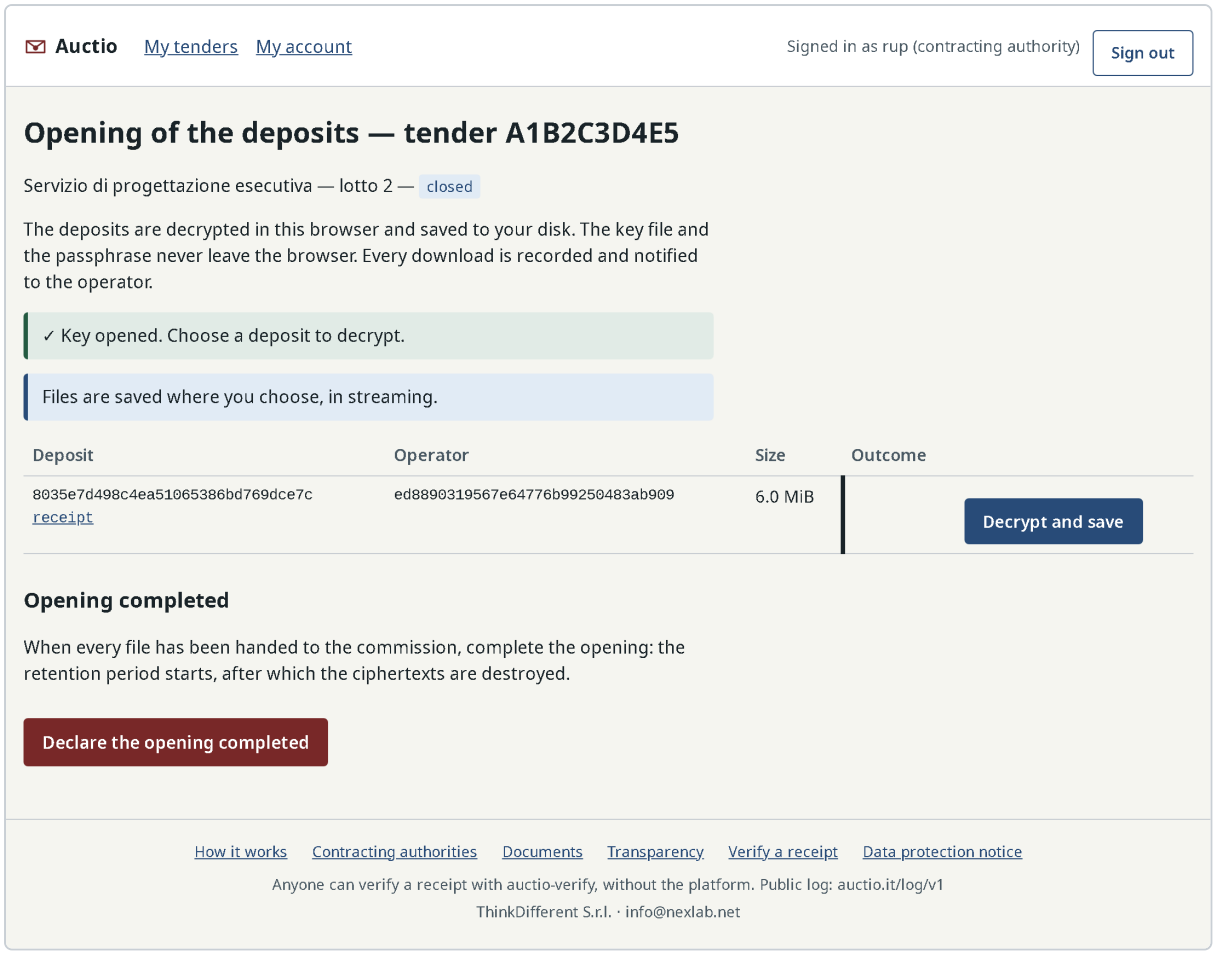
Load the file `gara-<CIG>.key` and type the passphrase. If the passphrase is lost, use two parts of the kit: the "Part 1/2/3" fields take the sheets' text. The key is decrypted in the browser: a few seconds.



The key file and the passphrase — or two parts of the kit.

3 Download every deposit.

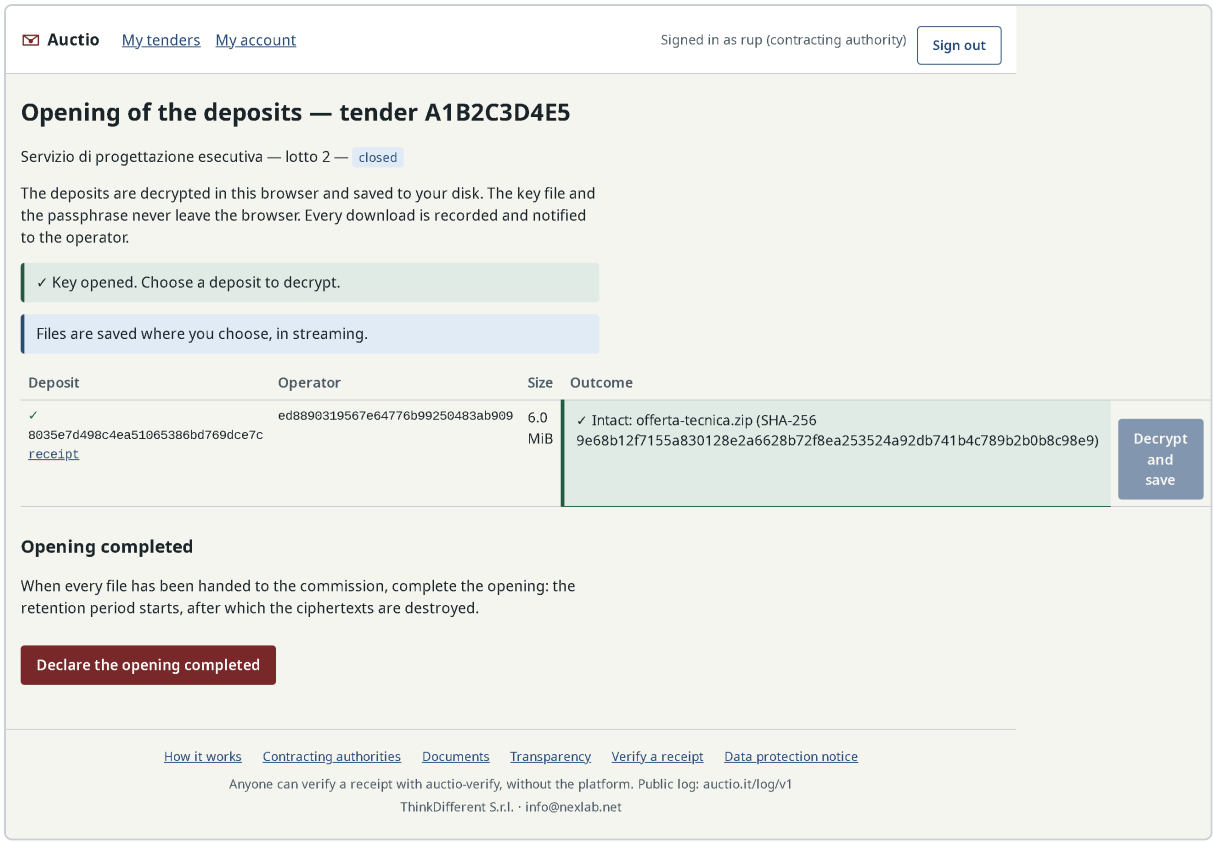
The table lists the record's deposits, with the operator, the size and the instant. "Decrypt and save" writes the file in clear to your computer, under the original name the operator gave it. For large files the browser asks where to save and shows progress.



The deposits listed in the record, one per row.

4 **Read the outcome.**

For each decrypted file the browser recomputes the digest and compares it with the one declared in the receipt: ✓ **intact** or ✗ **not intact**. A file that is not intact is a different file from the one whose digest the operator declared: record it as such. Every download is notified to the operator concerned.



Outcome of the opening: intact, recorded.

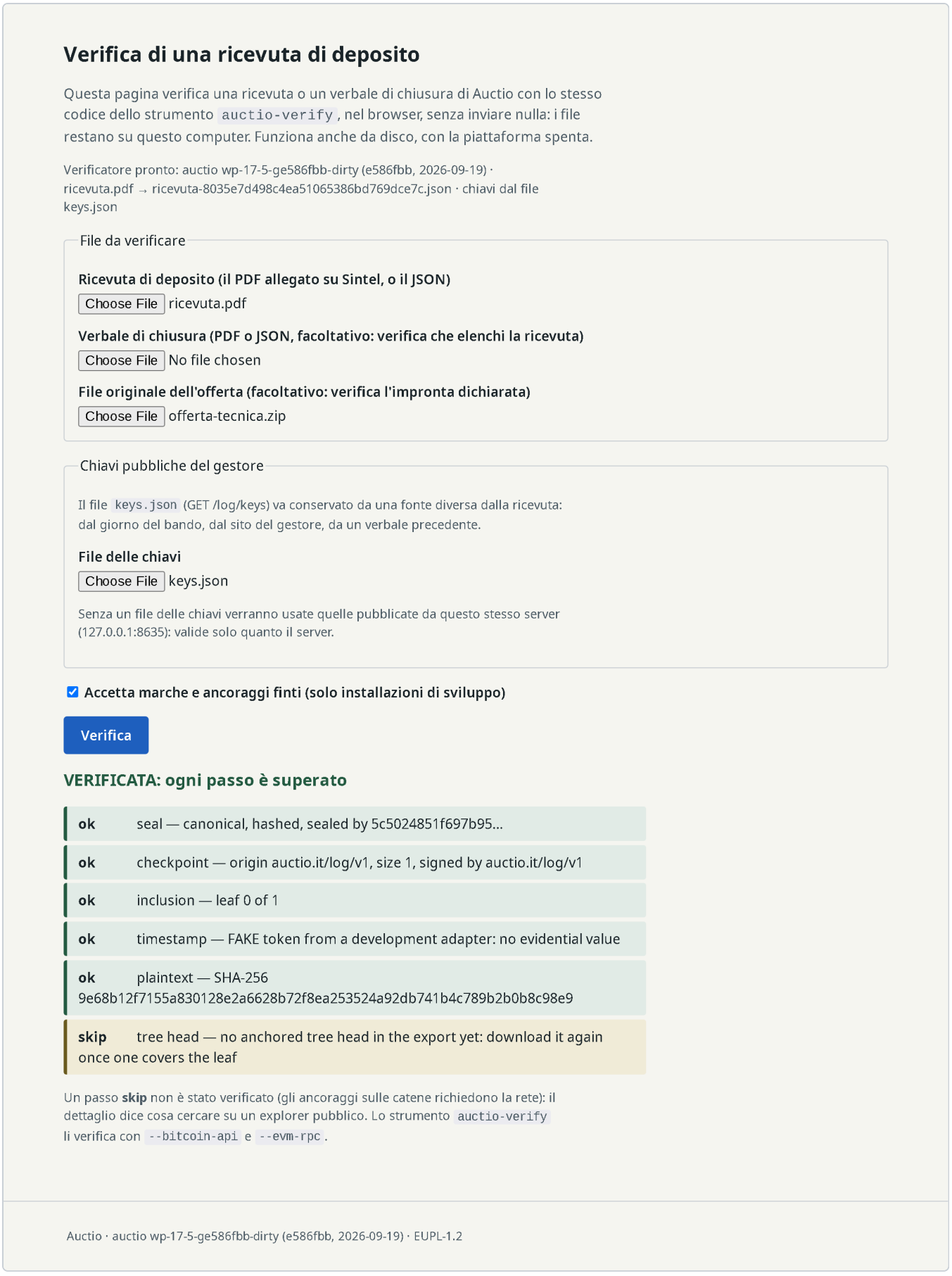
5 **Complete the opening.**
When every deposit has been downloaded, "Opening completed" records the end of the session and starts the retention period.

> **If a file was downloaded but not opened.** You may repeat the download as often as needed until the opening is completed, and after, during retention: every download is recorded and notified.

10. Verifying a receipt

The receipt the operator attached on Sintel is to be compared with the deposit: is it the same file? Was it issued by Auctio, before the deadline, for this tender? You verify it yourself, without the service, in two ways:

- **in the browser**, at auctio.it/verify/ (the verifier page is in Italian): load the receipt (Sintel's PDF), the published keys (`auctio.it/log/keys`, downloadable from the same page) and, to check the content too, the decrypted file. No data leaves the page;
- **with the `auctio-verify` tool** on your computer, shipped with every version of the software (the site's "Transparency" page), for a verification that can be filed.



Verifica di una ricevuta di deposito

Questa pagina verifica una ricevuta o un verbale di chiusura di Auctio con lo stesso codice dello strumento `auctio-verify`, nel browser, senza inviare nulla: i file restano su questo computer. Funziona anche da disco, con la piattaforma spenta.

Verificatore pronto: auctio wp-17-5-ge586fbb-dirty (e586fbb, 2026-09-19) · ricevuta.pdf → ricevuta-8035e7d498c4ea51065386bd769dce7c.json · chiavi dal file keys.json

File da verificare

Ricevuta di deposito (il PDF allegato su Sintel, o il JSON)
Choose File ricevuta.pdf

Verbale di chiusura (PDF o JSON, facoltativo: verifica che elenchi la ricevuta)
Choose File No file chosen

File originale dell'offerta (facoltativo: verifica l'impronta dichiarata)
Choose File offerta-tecnica.zip

Chiavi pubbliche del gestore

Il file `keys.json` (GET /log/keys) va conservato da una fonte diversa dalla ricevuta: dal giorno del bando, dal sito del gestore, da un verbale precedente.

File delle chiavi
Choose File keys.json

Senza un file delle chiavi verranno usate quelle pubblicate da questo stesso server (127.0.0.1:8635): valide solo quanto il server.

☑ **Accetta marche e ancoraggi finti (solo installazioni di sviluppo)**

Verifica

VERIFICATA: ogni passo è superato

ok seal — canonical, hashed, sealed by 5c5024851f697b95…

ok checkpoint — origin auctio.it/log/v1, size 1, signed by auctio.it/log/v1

ok inclusion — leaf 0 of 1

ok timestamp — FAKE token from a development adapter: no evidential value

ok plaintext — SHA-256 9e68b12f7155a830128e2a6628b72f8ea253524a92db741b4c789b2b0b8c98e9

skip tree head — no anchored tree head in the export yet: download it again once one covers the leaf

Un passo **skip** non è stato verificato (gli ancoraggi sulle catene richiedono la rete): il dettaglio dice cosa cercare su un explorer pubblico. Lo strumento `auctio-verify` li verifica con `--bitcoin-api` e `--evm-rpc`.

Auctio · auctio wp-17-5-ge586fbb-dirty (e586fbb, 2026-09-19) · EUPL-1.2

The verification: every step (seal, timestamp, log, content) with its outcome.

A **NOT VERIFIED** outcome names the step that failed: *seal* (the receipt was changed or is not Auctio's), *timestamp* (the timestamp does not match), *plaintext* (the file is not the one declared), *tree head* (the log does not contain it).

11. After the opening: retention and destruction

Once the opening is completed, the files remain in the service for the retention period set for the tender (on the public page). You may download them again in that period — for an access request, for an appeal. At the end of the period the encrypted files are **destroyed** and the destruction is recorded; receipts, record and log remain forever: they are the document of the procedure.

File: the closure record (PDF and JSON), the receipts attached on Sintel, the outcome of the opening, and — in the restricted file — the key file and the kit sheets until the end of the retention period.

12. When something goes wrong

SITUATION	WHAT TO DO
I lost the phone with the authenticator app	use a recovery code to sign in, then set the app up again from "My account"; with no codes, ask the operator of the service to reset the second factor
I forgot the passphrase	two kit sheets rebuild it, in the opening form
I lost the key file	look in the second copy; if it is truly lost *and* the tender is already open to deposits, the envelopes cannot be opened: tell the operator of the service at once and consider, with your legal office, reopening the deadline with a new tender
The file verification of step 5.4 fails	the key is not published yet: start again from the passphrase, no operator is involved
An operator says they cannot deposit	check the tender is *open* and the deadline not passed; the rest (sizes, connection) is in the operator's manual; the operator of the service sees the uploads' state
A file is "not intact"	a fact for the minutes: the decrypted file does not have the digest declared in the receipt; repeat the download once to rule out a transfer error
The deadline must be extended	the operator of the service does it; those who deposited need do nothing

For everything else: the operator of the service, at the contacts at the foot of every page of the site.

13. Glossary

Tender key
The tender's cryptographic key pair: the public one encrypts, the private one decrypts. Generated in your browser.

Fingerprint (hash, digest)

A 64-character string that uniquely identifies a file or a key (SHA-256). One different byte changes it.

Passphrase
The secret phrase that protects the key file.

Recovery kit
The passphrase split into three parts; two rebuild it.

Deposit receipt
The document Auctio issues to the operator: the file's digest, the instant, the tender, the seal of the operator of the service, the timestamp. As PDF/A for Sintel.

Closure record
The list of deposits arrived within the deadline, sealed, timestamped and entered in the public log.

Timestamp
A third-party authority's attestation that a document existed at a given instant (RFC 3161).

Public log
The transparency log, append-only, that receipts and records enter; anchored on Bitcoin. At auctio.it/log/.

Intact / not intact
The outcome of comparing the decrypted file's digest with the one declared in the receipt.