Declaration of a TLSNotary Notary Signing Key

Service: `pavoltravnik.witness.directcase.ai` · **Document version:** 1.0 · **Date:** 26 September 2026

> **TEST MODE.** The service is experimental and provided without warranty. Receipts are intended for testing and evaluation only and must not be relied upon as evidence in legal, financial or other critical matters. The key below may be retired at any time; its history is published at `/keys.json`.

1. DECLARANT

Ing. Mgr. Pavol Trávnik, born 22 July 1991, Zborovská 124/68, Malá Strana, 150 00 Praha 5, Czech Republic, acting as a **natural person** and sole operator of the service.

2. WHAT THE SERVICE DOES

The service is a *notary* (verifier) built on the open-source TLSNotary protocol. When a user (the *prover*) loads a web page over HTTPS, the notary takes part in the TLS session — by multi-party computation (MPC-TLS) or as a proxy — without being able to forge the server's data. The prover may redact private parts; the notary then signs a **receipt**: a commitment to the server's identity (checked against its TLS certificate) and to the disclosed content, with the time of verification. Anyone can later check the receipt without trusting the prover.

A valid receipt proves that the disclosed content was served over TLS by the named server at the stated time, and that this notary witnessed it. **It does not prove** that the content is true, nor anything about redacted parts. The notary is operated independently of the websites it observes.

3. DECLARED SIGNING KEY

Key id	`73a6bdba645a59e4` (first 8 bytes of SHA-256 of the public key)
Algorithm	ECDSA, curve secp256k1, SHA-256 (`secp256k1-ecdsa-sha256`)
Public key (SEC1, hex)	`029ba59091fb1d63de1b7db1e92bc2bd0ffadc82a8d6522c102cd3db5635b9b744`
SHA-256 of the public key	`73a6bdba645a59e4ce5641cfa5fbe8e7bf0aad33e90004cc803ecf3ff47f181f`
Active since	2026-09-25T21:52:43Z

I declare that this key was generated by the service under my sole control, that receipts signed with it are issued by the service described above, and that I will publish the retirement of this key at `/keys.json`.

4. HOW TO VERIFY A RECEIPT

1. Check that this PDF carries a valid eIDAS signature of the declarant (e.g. with the EU DSS validator).
2. Take the receipt `{alg, key_id, payload, signature}` and confirm that `key_id` and the public key match section 3.
3. Verify the ECDSA signature (64-byte `r‖s`, hex) over SHA-256 of the exact UTF-8 bytes of `payload`, or send the receipt to `POST https://pavoltravnik.witness.directcase.ai/verify`.
4. Check that `verified_at` in the payload lies within the key's active window.

Signed electronically by the declarant with an electronic signature under Regulation (EU) No 910/2014 (eIDAS).