IIC TSA — Timestamp Policy v1

Publication notice. This document is published by HTTPS CARD — INTERNET IDENTITY CARD LIMITED (England and Wales, company no. 09168431) for the IIC TSA time-stamping service at `https://tsa.internetidentitycard.com/tsa`. IIC TSA is designed to RFC 3161 / RFC 5816. It is **not** a qualified trust service under eIDAS or UK eIDAS, and it is not certified, accredited or recognised by any authority. IIC TSA and VerifBox (verifbox.com) belong to the same company: for VerifBox users, IIC TSA is not an independent third party. Use of the service is subject to the IIC TSA terms of use.

Version 1.0 — approved 10 October 2026; erratum 1 of 11 October 2026 (section 8, retention lock deferred). Owner approval was given by launching the production service on 10 October 2026; the service's policy.json reports status "approved". Production ceremony performed, production services deployed and accepted (40 of 40 checks), VerifBox integration live; production opened while the 14-day accuracy campaign runs in parallel. Design as approved by the owner on 9 October 2026 — two-level hierarchy, Cloud KMS SOFTWARE key, single operator, on-demand time guardian, daily third-party anchoring. Supersedes draft 9.

Status: version 1.0, approved 10 October 2026. The policy OID is 1.3.6.1.4.1.67100.1.1.1.1, assigned in the OID registry of PEN 67100; the acceptance criteria of the specification passed on the test project and in production on 10 October 2026 (section 10). IIC TSA is a trade name of the operator, which runs it within its Internet Identity Card business line. IIC TSA is not a qualified trust service under eIDAS, not certified, accredited or FIPS-certified, and not recognised by any government. IIC TSA and VerifBox (verifbox.com), a timestamping website that is a client of IIC TSA since 10 October 2026, belong to the same company: IIC TSA is not an independent third party for VerifBox users. This policy implements IIC TSA — Technical Specification v1.0; IIC TSA — Risk Register v1.1 assesses its risks. This policy is published in English and French; the English version is authoritative for the technical interpretation of the TSA protocol and policy.

1. Identification

Item	Value
Operator	HTTPS CARD - INTERNET IDENTITY CARD LIMITED, company no. 09168431, 124 City Road, London EC1V 2NX, United Kingdom
Service	Internet Identity Card Time Stamp Authority (IIC TSA), address `https://tsa.internetidentitycard.com/tsa`

Item	Value
Protocol	RFC 3161, as updated by RFC 5816 (HTTP transport only)
Policy OID	`1.3.6.1.4.1.67100.1.1.1.1`, assigned in the OID registry of PEN 67100 (allocated by IANA to the operator). Arc `.1.1.1` holds the IIC TSA timestamp policies; each new version takes the next number (`.1.1.1.2`, ...). Test tokens carry the distinct test policy `1.3.6.1.4.1.67100.1.1.9.1`, never accepted in production.
Policy version	1.0 (approved 10 October 2026)
Contact	an address on internetidentitycard.com, to be created

2. Scope and meaning of a token

An IIC TSA timestamp token proves that the TSA signed, at the time stated in its `genTime`, a given hash value. The TSA receives only that hash, never the document, and does not examine, store or interpret it.

A token does not prove who created or owns the document, its content, its accuracy or its lawfulness. It is not a signature of the document and it is not a qualified electronic timestamp.

Until acceptance criteria A1 to A10 of the specification (except A7, HSM attestation, superseded by the simplified design) have passed on the test project and again after the production ceremony, tokens have been accepted by a second verifier written independently of the token encoder (OpenSSL and the operator's own verifier are not enough on their own), and the endpoint has opened, the only accurate public statement is: *IIC TSA is designed to RFC 3161 / RFC 5816; local tests pass; acceptance on the test project passed (43 of 43 checks, 10 October 2026); production acceptance passed (40 of 40 checks, 10 October 2026 18:22 UTC); a second verifier written independently of the token encoder is still pending.* Once those conditions are met, and only then, the authorised statement becomes: *Internet Identity Card operates its own Time Stamp Authority, designed to RFC 3161 / RFC 5816, for independently verifiable cryptographic timestamping.* No statement may describe IIC TSA as qualified, certified, accredited, official or independent of VerifBox.

3. Requests

Requests are DER-encoded `TimeStampReq` messages sent by HTTP POST, content type `application/timestamp-query`, at most 8,192 bytes.

Hash algorithm	OID	Hash length	Accepted
SHA-256	2.16.840.1.101.3.4.2.1	32 bytes	yes
SHA-384	2.16.840.1.101.3.4.2.2	48 bytes	yes
SHA-512	2.16.840.1.101.3.4.2.3	64 bytes	yes
SHA-1, MD5, others	—	—	no

Condition	Response
Algorithm not accepted	rejection, `badAlg`
Malformed request, version other than 1, wrong hash length, BER or trailing data	rejection, `badDataFormat`
`reqPolicy` present and different from this policy	rejection, `unacceptedPolicy`
Any extension: none is recognised in v1	rejection, `unacceptedExtension`
Clock not in SYNC state	rejection, `timeNotAvailable`
Cloud KMS key unavailable, or a check before or after signing failed	rejection, `systemFailure`
More requests than the per-address rate limit (section 10)	HTTP 429 with Retry-After: 60; no token

A nonce, when present, is copied unchanged into the token. `certReq` is honoured: when TRUE, the response carries the TSA certificate only, never the root; when absent, no certificate. A `certReq` written out as FALSE is not DER, which never encodes a default value, and is rejected with badDataFormat. A request carrying any extension is rejected, as RFC 3161 requires for extensions the service does not recognise. No client authentication is required: the service timestamps a hash, not an identity.

4. Tokens

Each token is a CMS `SignedData` carrying a `TSTInfo`, with a single signature: the TSA's (RFC 5816).

TSTInfo field	Value
`version`	1
`policy`	the policy OID of section 1
`messageImprint`	exact copy of the request
`serialNumber`	positive integer generated from 128 bits of a cryptographically secure random generator, never derived from the request. The TSA keeps an append-only serial journal (section 8) and rejects any detected reuse: on collision, no token is issued with that number and a new one is drawn

TSTInfo field	Value
`genTime`	UTC, rounded to the nearest whole second, `Z` suffix, no fraction
`accuracy`	the declared accuracy of `genTime` with respect to UTC: 1 second; every token is gated by a live measurement against four time sources, and a 14-day measurement campaign runs in parallel (section 5)
`ordering`	FALSE: tokens are not ordered among themselves
`nonce`	present if and only if the request carried one, bit-for-bit identical
`tsa`	always present: a GeneralName (directoryName) equal to the subject name of the TSA certificate

The signed attributes include `SigningCertificateV2` with the `ESSCertIDv2` (SHA-256) of the TSA certificate; the SHA-1 `ESSCertID` is not issued. The signature algorithm is fixed for each key generation and published with it. **First generation: ECDSA on curve P-256 with SHA-256.** The signature is computed once, over the SHA-256 digest of the DER-encoded signed attributes, with no second hashing. SHA-1 is never used. A later generation may use ECDSA P-384 with SHA-384, under a new policy version.

5. Time source

Tokens are issued only in the SYNC state; there is no fallback to an unchecked system clock (fail-closed). The service does not set the system clock: an on-demand time guardian inside the Cloud Run service (minimum instances 0) measures it against four time sources reached over TLS, run by four different operators. A test in the test project on 10 October 2026 showed that Cloud Run blocks outbound UDP: PTB, NIST, Netnod, Cloudflare, Google NTP and the metadata server gave no answer; the owner chose TLS sources the same day. NTP support stays in the code for environments that allow UDP; NTS (RFC 8915) is no longer a target on Cloud Run, as it needs UDP.

At least one retained time source SHALL provide a documented traceability path to UTC, directly or through a source whose traceability is documented. Only the PTB web clock is traceable: if PTB is unreachable, no source is traceable and no token is issued (`timeNotAvailable`). A source is usable when it answers within 2.5 seconds with a measured uncertainty of at most 100 ms.

Source	Endpoint	Measurement	Traceable to UTC
PTB web clock	`wss://uhr.ptb.de/time` (secure WebSocket)	server time in milliseconds, with the server's own uncertainty	yes, to UTC(PTB)

Source	Endpoint	Measurement	Traceable to UTC
Akamai	`https://time.akamai.com/?iso&ms`	ISO time in milliseconds	no
Google	`https://www.google.com/generate_204`	HTTP Date header: the instant it moves to the next second is bracketed between two requests; uncertainty = half the bracket	no
Cloudflare	`https://www.cloudflare.com/cdn-cgi/trace`	HTTP Date header, as for Google	no

NIST's web time service is not used: NIST forbids outside use of it. PTB's web clock publishes no terms for automated use; the load is low, four messages per measurement and measurements only on demand.

State	Condition	Issuance
SYNC	at least 3 usable sources, at least one of them traceable to UTC (PTB); spread at most 250 ms; system clock within ±100 ms of their median; no clock step of more than 1 second pending	allowed
HOLDOVER	partial loss after a recent SYNC	not allowed in v1
FAIL	any condition of SYNC unmet (spread or offset above threshold, fewer than 3 usable sources, none traceable to UTC), a clock step awaiting readmission, or an expired lease	not allowed

A SYNC result gives a time lease of 30 seconds, and the signer acts only under a valid lease; the guardian withdraws the lease as soon as it enters FAIL. Cloud Run gives no CPU between requests, so there is no background measurement (TSA_BACKGROUND_GUARDIAN=0). When a request arrives and no valid lease exists, the guardian queries the four sources before issuing (one measurement cycle; each source has at most 2.5 seconds to answer); without a SYNC result, no token is issued and the request is rejected with `timeNotAvailable`. The signer reads `genTime` under the lease, before signing; if the lease expires or is withdrawn during signing, no token is issued. A clock step of more than one second forces FAIL until manual readmission; any other FAIL ends when the conditions of SYNC are met again. The hosting infrastructure smears leap seconds over 24 hours; on those days the clock leaves the thresholds, the guardian moves to FAIL and issuance pauses. The declared accuracy is 1 second and covers the worst case that SYNC allows: 0.5 s of rounding, plus the offset threshold, plus a bound on the error of the

sources' median. The thresholds above are initial values. Staging measurements on 10 October 2026 (offset, with uncertainty): PTB about 0 to 3 ms (±8–12 ms), Akamai about −1 to −2 ms (±5 ms), Google ±33 ms, Cloudflare ±43 ms; spread 24 ms. Production acceptance on 10 October 2026 at 18:22 UTC: SYNC with four sources (PTB +0.9 ms ±8 ms, Akamai 0 ms ±4 ms, Google +3.5 ms ±33 ms, Cloudflare +2.6 ms ±44 ms; spread 3.5 ms). By owner decision of 10 October 2026, production opens without waiting for the 14-day measurement campaign, because every token is already gated by a live measurement against the four sources. The campaign runs in parallel on the test service (one call every 10 minutes, started 10 October 2026 at 12:30 UTC) and reports the share of SYNC cycles, the p99 of the median offset and the p99 of the spread; its report is reviewed at day 14, when the thresholds may be adjusted.

6. Keys and certificates

The token signing key is a Google Cloud KMS key, algorithm EC_SIGN_P256_SHA256, protection level SOFTWARE: it is **not** protected by a hardware security module (HSM). It is used for nothing but timestamp tokens: its certificate is built from the public key read from Cloud KMS, and no certificate request or health check is ever signed with it. The application never holds the private key; the signing service reaches Cloud KMS under its own least-privilege service account (signer and public key viewer on that key only), with no key file, PIN or credential in code, configuration or environment variables; any other secret lives in a separate secret store, with rotation and an access log. Before signing, the service checks that the Cloud KMS public key equals the certificate's and that the key's protection level is the expected one; it checks the CRC32C integrity of the digest and of the signature, normalises each ECDSA signature to low-S and verifies it locally before releasing the token. Cloud KMS Data Access audit logs are enabled. The scheduled destruction delay of a key version is 30 days in production (1 day in test). A move to HSM protection remains possible if a client requires it; it means a new key and a new certificate.

The hierarchy is private, dedicated and has two levels: an offline root, IIC TSA Root R1 (test: IIC TSA TEST Root R1), and the TSA certificate, which the root issues directly. There is no issuing CA. The root never signs tokens; the TSA key signs nothing else.

The root certificate has subject C = GB, O = HTTPS CARD - INTERNET IDENTITY CARD LIMITED; its key is ECDSA P-384, it is signed with SHA-384 and valid for 30 years minus 1 second; `basicConstraints` CA:TRUE, pathlen:0, critical; `keyUsage` keyCertSign and cRLSign, critical; subject key identifier per RFC 7093 method 1 (leftmost 160 bits of SHA-256). The root private key is a software key, stored as PKCS#8 encrypted with scrypt and AES-256-CBC under a passphrase of at least 20 characters, written on paper and kept apart. It is generated and used offline on the owner's own computer, in a Docker container built from `ceremony/Dockerfile` (Debian 13, OpenSSL 3.5) and run with `--network none`, with network cables and Wi-Fi off. It is used about once a year, to issue the CRL, and to certify a new TSA key. Backups: the whole ceremony folder, encrypted key included, on two USB drives stored in two different places. The root key is never placed in Cloud Shell, Git, a container image or the cloud.

Production ceremony, 10 October 2026. Performed offline on the owner's iMac (Docker, Debian 13, OpenSSL 3.5, `--network none`) by the single operator. Root certificate IIC TSA Root R1 (P-384), valid 10 October 2026 to 9 October 2056, SHA-256 `86:52:E3:D1:3E:72:5F:7C:75:7C:FA:05:3D:64:13:60:AC:A4:43:CF:6C:09:D7:D6:DD:D4:4B:EA:85:1E:D3:B9`. TSA certificate IIC TSA tsa-2026-a, SHA-256 `E4:F9:8E:7A:ED:CD:1A:07:D8:1C:A5:9D:1C:19:92:AB:E7:85:7A:2C:A1:CC:8D:F6:9F:72:05:FF:AD:34:FB:E8`. First root CRL valid until 10 October 2027 17:29 UTC. The encrypted root key is on two USB keys of different brands, verified identical and each decryptable; the passphrase is on paper, kept apart; the working copy was deleted from the computer. Only root.pem, root.crl and the TSA certificate went to Cloud Shell.

Operators. IIC TSA has a single operator, the owner, who holds every role. There is no separation of duties. Compensating controls: Cloud KMS Data Access audit logs, least-privilege service accounts, the retention-locked issuance log (section 8) and its daily third-party anchoring.

TSA certificate field	Value
`basicConstraints`	CA:FALSE, critical
`keyUsage`	digitalSignature only, critical
`extendedKeyUsage`	id-kp-timeStamping (1.3.6.1.5.5.7.3.8) only, critical; SHALL NOT contain anyExtendedKeyUsage
`subjectKeyIdentifier` / `authorityKeyIdentifier`	present; subject key identifier per RFC 7093 method 1
Subject	CN = IIC TSA {generation} (first: IIC TSA tsa-2026-a; test: IIC TSA TEST tsa-test-2026-a), O = HTTPS CARD - INTERNET IDENTITY CARD LIMITED, C = GB; issuer: IIC TSA Root R1; public key: ECDSA P-256, the Cloud KMS key
Validity	135 months minus 1 second; the private key signs tokens for at most 15 months after the certificate's notBefore date and is destroyed within 18 months of the certificate's issuance
CRL distribution point	https://tsa.internetidentitycard.com/tsa/crl (test: the private run.app address of the test service)

The TSA signing certificate SHALL NOT be used for TLS, document signing, code signing, email signing or any purpose other than timestamp generation. The TLS certificate of tsa.internetidentitycard.com and the VerifBox document signing keys are distinct keys and are never accepted as token signers. Each key generation has a monotonic identifier (for example tsa-2026-a). The next certificate is published before the switch; the previous key stops signing at a documented instant and stays available for verification.

7. Expiry, revocation and compromise

Expiry of the TSA certificate alone does not invalidate a token issued while the certificate was valid; tokens are verified at their `genTime`. Validation remains subject to the certificate status, the evidence on key compromise and the long-term validation rules below.

The root issues the only CRL of the hierarchy: DER-encoded, valid for 365 days and re-issued every year, and an emergency CRL at once on a compromise. It is published at `https://tsa.internetidentitycard.com/tsa/crl` and `https://tsa.internetidentitycard.com/tsa/root-crl`. The current root CRL is archived every day with the daily batch of the issuance log (section 8), and VerifBox publishes a copy at `https://verifbox.com/.well-known/iic-tsa-racine.crl`, a second location outside the service, so that revocation evidence survives an outage or the loss of the domain. Every revocation carries a reason code; a revocation for keyCompromise carries the trust-cutoff instant as its invalidity date.

Failure to publish or retrieve the current CRL SHALL NOT cause previously issued tokens to be silently treated as valid or invalid. A verifier SHALL apply the revocation evidence available for the relevant validation time; when revocation checking is requested and no evidence is available, the result is reported as undetermined, not as valid.

On suspected compromise of the signing key, issuance stops immediately (FAIL), the root issues an emergency CRL revoking the TSA certificate with reason keyCompromise, and a trust-cutoff instant is published as its invalidity date. Tokens issued before that instant are not automatically invalidated: they remain within the generation's declared validation scope, subject to the evidence available on the state of the signing key when each was issued and to the validation rules below. The cutoff is a published bound, fixed from the issuance log, the Cloud KMS Data Access audit logs and the last verified state of the key; it is not a proof that every earlier token was legitimately issued. That proof comes from the issuance log (section 8): a token whose signature digest appears in a daily batch timestamped by a third-party TSA, or anchored in Bitcoin, before the cutoff was issued by the service. The root is not revoked for an incident limited to the TSA key. The operator's notification duties are in section 9.

Validation of tokens, including in the long term

A relying party checks the following points separately, and a verifier reports each result rather than a single verdict:

1. The CMS signature of the token is cryptographically valid.
2. The TSA certificate chains to the published root, carries the critical `id-kp-timeStamping` extended key usage, and was within its validity period at `genTime`.
3. The TSA certificate was not revoked at `genTime`, according to a CRL issued after genTime: the first archived CRL whose thisUpdate is later than genTime, or a later one.
4. If the key was later revoked for keyCompromise, `genTime` is before the published trust-cutoff instant, and the token's signature digest appears in a batch of the issuance log timestamped or anchored before that instant; otherwise the token rests

on other evidence, such as a second timestamp from an independent source; without such evidence, the result is undetermined.
5. The `policy` field is an OID the relying party accepts: `1.3.6.1.4.1.67100.1.1.1.1` for this policy, never a test OID.
6. The `messageImprint` matches the data being verified and, where a nonce was sent, the nonce matches.

For validation long after issuance, the token is kept together with the certificate chain and a CRL issued after `genTime`, which the operator archives (above). Before the hash or signature algorithm of a token is considered weak, the token and its validation data should be protected by a new timestamp, as long-term validation formats such as CAdES-A do; the operator announces any such weakening of its algorithms in advance. A new timestamp issued by the IIC TSA itself adds no independence. Where independent evidence is needed, it comes from a source that does not depend on the IIC TSA: a TSA operated by a third party, or another proof of existence the relying party accepts, such as an anchor in the Bitcoin blockchain through OpenTimestamps. Such an anchor shows that the token existed before a given block; it does not certify the time stated by the TSA and is not an RFC 3161 timestamp.

8. Issuance log

The TSA keeps an issuance log **without any hash value from requests**: the fingerprints submitted are never stored, in line with the operator's privacy commitments. Each token itself carries its hash and signature, which is what a verifier needs.

Kept for each token	Not kept
serial number, `genTime`, key generation, clock state, clock generation and lease identifiers, presence of a nonce (yes/no), status (issued, or not issued with the reason lease_expired, sign_failed or check_failed), and the SHA-256 of the token's signature	hash value submitted, nonce value, IP address, request body, document, file name

The log is a Cloud Storage bucket holding one object per serial number (`entries/<32 hex digits>`), created only if absent, so no entry is ever overwritten. A token is returned only after its entry is written. It also serves as the serial journal used to detect reuse (section 4). Retention: 12 years, longer than the 135-month validity of a TSA certificate, enforced by a 12-year bucket retention policy. **Erratum of 11 October 2026:** the irreversible *lock* of that retention policy has not yet been applied; by owner decision of 11 October 2026 it is deferred until after the first weeks of operation. Until it is applied, an account holding administrative rights on the project could in theory shorten the retention policy; the daily third-party time-stamping and Bitcoin anchoring of the journal (below) remain the evidence that the past has not been rewritten. The date of the lock will be published in this section and on `/tsa/status` (test: 1 day, not locked). Every day at 00:20 UTC, a Cloud Run job started by Cloud Scheduler writes all entries and marks of the previous UTC day to `batches/journal-YYYY-MM-DD.jsonl` (sorted, canonical JSON lines). The SHA-256 of that file is

timestamped by DigiCert's public RFC 3161 server (`http://timestamp.digicert.com`, policy 2.16.840.1.114412.7.1, used without contract; owner decision of 9 October 2026), or by FreeTSA (`https://freetsa.org/tsr`) as a fallback only, never by the IIC TSA itself, and anchored in Bitcoin through OpenTimestamps; neither ever becomes a source of `genTime`. The file and its proofs are copied, with their SHA-256, to an archive bucket in another region (europe-west4, Archive storage class). A failure of these third parties never blocks token issuance. The signature digest is not the hash value submitted: it cannot be computed without the token itself, so it does not let anyone test whether a given document was timestamped. The entries of a given day are provided to a relying party on request.

Request-body logging and client-address logging are disabled, and Cloud Run request logs are excluded because they contain client IP addresses. Aggregate security and availability telemetry (request counts, rejection codes, latency) may be kept, provided it contains no request body, hash value, nonce value or client IP address. The client IP address is held in memory for at most one minute, solely for rate limiting, and is not written to any log the operator keeps; that the request and audit logs stored in the TSA projects, the platform's default log buckets included, retain no client address, request body or signed-attributes digest is checked before opening; what the cloud provider keeps outside those projects is not under the operator's control. The issuance log and the records of client requests therefore contain no personal data. Administrative and security audit records, such as the Cloud KMS Data Access audit logs, may contain the identities of authorised personnel; they are protected and retained under the operator's privacy and retention rules.

9. What is not promised

- No qualification: the service is not a qualified trust service under eIDAS or UK eIDAS, and holds no ETSI audit or listing in any trust list.
- No certification: neither the service nor its key is claimed to be certified, accredited or FIPS-certified; the TSA key has Cloud KMS SOFTWARE protection, not HSM protection.
- No availability commitment: during a FAIL state, a key rotation or an incident, requests are rejected rather than timestamped.
- No review of content: the TSA never receives the document and makes no statement about it.
- No guarantee of legal effect: the evidential value of a token is assessed by the competent court or authority. As a non-qualified electronic time stamp, a token is not denied legal effect or admissibility as evidence solely because it is electronic or not qualified (Article 41 of eIDAS).
- No independence from VerifBox: IIC TSA and VerifBox belong to the same company; for VerifBox users, IIC TSA is not an independent third party.
- No separation of duties: the service has a single operator (section 6).

Legal obligations. The operator provides a non-qualified trust service in the United Kingdom. Under UK eIDAS Article 19, it takes appropriate technical and organisational security measures, notifies the ICO within 24 hours of becoming aware of a breach with a

significant impact on the service, and informs the affected parties. It processes personal data under UK GDPR (ICO registration ZA457585): the issuance log holds no personal data, and a client IP address is held in memory for at most one minute (section 8).

10. Publication and changes

Endpoint	Content
`GET /tsa/policy.json`	short machine-readable summary of this policy (JSON)
`GET /tsa/certificate`	current TSA certificate (PEM)
`GET /tsa/chain`	root certificate (PEM)
`GET /tsa/crl`	current root CRL (DER)
`GET /tsa/root-crl`	current root CRL (DER), same as /tsa/crl
`GET /tsa/status`	clock state, key generation, time of the last token, dated notices (rotation, revocation, incident, termination); no client data
`GET /tsa/health`	200 only when the clock is in SYNC and the key answers
`POST /tsa`	timestamp requests (section 3)

The service runs on Google Cloud, region europe-west1 (Belgium), with separate test and production projects. The public service `iic-tsa` answers at `https://tsa.internetidentitycard.com/tsa` (Cloud Run domain mapping), at most 60 requests a minute per client address. **Confirmed by the owner on 10 October 2026:** production also runs a private service, `iic-tsa-verifbox`, from the same image, with the same key and the same issuance log, reachable only by VerifBox's service account (IAM), at most 600 requests a minute per address.

Status on 10 October 2026 (go-live). Test project `iic-tsa-test-67100` (test policy `1.3.6.1.4.1.67100.1.1.9.1`): test ceremony done, test root SHA-256 `BA:7C:3F:11:CC:23:8D:EC:D7:E1:9E:60:DE:88:6E:10:FA:CA:78:5E:BF:F2:CD:7A:53:2A:12:34:3B:29:DC:76`, test TSA certificate SHA-256 `E8:DC:53:E3:36:24:7E:EB:6A:C3:0E:6E:78:6A:1E:D2:16:19:05:F3:7A:E2:E2:C7:63:35:C8:6A:DF:F9:E4:A6`; acceptance script 43 checks, 43 passed. Production project `iic-tsa-prod-67100` (Cloud Run, europe-west1): Cloud KMS key `tsa-2026-a`, public key SHA-256 `3d11b2af1f9a52d960e79c5aae3553c087a8627e32628a0ab51c0cb83a8a0d02`; production ceremony performed on 10 October 2026 (section 6); public service `iic-tsa` at `https://iic-tsa-cnnddoxp6q-ew.a.run.app/tsa` (60 requests a minute per address), private service `iic-tsa-verifbox` (IAM, 600 a minute) called by VerifBox's service account; daily batch job at 00:20 UTC (DigiCert, OpenTimestamps, archive copy). Domain mapping `tsa.internetidentitycard.com` created (CNAME to ghs.googlehosted.com in Cloud DNS); the Google-managed certificate was being provisioned at the time of writing. Production acceptance on 10 October 2026 at 18:22 UTC: 40 checks passed, 0 failed (`tests/staging-acceptance.sh`, TARGET=prod). First production token: serial `0xD22448FD5AD33BD7A393BB6024B28126` at

18:02:46 UTC, verified with OpenSSL. VerifBox integration live at about 18:52 UTC: the site publishes the root and CRL at `/.well-known/iic-tsa.json` and `/.well-known/iic-tsa-racine.crl`, its engine pins the root, each proof carries the token over the file fingerprint, verified independently with OpenSSL (Verification: OK) and the public Python verifier. Buckets with 12-year retention created; the retention lock (irreversible) is pending owner approval.

Any change to this policy produces a new version with a new OID; tokens always name the policy under which they were issued. A separate Practice Statement describes operations: Cloud KMS key, ceremonies, the single operator, backups, logging and incident handling.

Open items at approval (version 1.0, 10 October 2026):

- ☑ Key provider: Google Cloud KMS, SOFTWARE protection level (decided 9 October 2026)
- ☑ Outbound UDP/123 from Cloud Run: settled on 10 October 2026, blocked; replaced by four sources reached over TLS (PTB, Akamai, Google, Cloudflare; section 5)
- ☑ 14-day measurement campaign: no longer blocking, by owner decision of 10 October 2026; running in parallel on the test service since 10 October 2026 12:30 UTC, report reviewed at day 14 (section 5)
- ☐ A second verifier written independently of the token encoder
- ☐ French version of this policy
- ☑ TSA Practice Statement v1.0, approved 10 October 2026
- ☑ Acceptance of the RFC 3161 / RFC 5816 profile on the test project: 43 of 43 checks passed on 10 October 2026
- ☑ Two production services confirmed by the owner on 10 October 2026: public iic-tsa and private iic-tsa-verifbox for VerifBox (section 10)
- ☐ To be confirmed: whether the daily CRL archive is also published in the operator's public GitHub repository, as draft 7 stated
- ☐ Service domain (`tsa.internetidentitycard.com`: domain mapping and CNAME created on 10 October 2026, Google-managed HTTPS certificate still being provisioned) and a contact address on internetidentitycard.com
- ☐ Privacy notice of the IIC TSA, published on the Internet Identity Card website
- ☐ IIC TSA — Risk Register v1.1 approved with this policy on 10 October 2026; IIC TSA — Technical Specification v1.0 approval to be confirmed
- ☐ Request and audit logs stored in the TSA projects checked: no client address, request body or signed-attributes digest retained
- ☐ Retention lock on the production issuance-log and archive buckets (irreversible): deferred by owner decision of 11 October 2026 until after the first weeks of operation (section 8, erratum)
- ☐ Production budget alert
- ☐ Day-14 review of the measurement campaign (section 5)

Revision history

Draft revisions before approval do not consume a new policy OID. Once this policy is approved, any change produces a new version with the next OID (`.1.1.1.2`, ...). Only the latest draft is current; earlier copies are withdrawn.

Draft	Date	Change
1.0, erratum 1	2026-10-11	Section 8: the retention lock is deferred by owner decision (not yet applied); no change to the tokens, their format or their meaning, so the policy version and OID are unchanged. Section 10: open item updated
1.0	2026-10-10	Approved: owner approval given by launching the production service (policy.json status "approved"). Production ceremony recorded (root R1 and tsa-2026-a fingerprints, first root CRL to 10 October 2027, USB backups); production services iic-tsa and iic-tsa-verifbox deployed in iic-tsa-prod-67100, domain mapping created; production acceptance 40 of 40 checks at 18:22 UTC, guardian SYNC with four sources; first production token at 18:02:46 UTC; VerifBox integration live; open items updated (retention lock, budget alert, HTTPS certificate, second verifier, day-14 review)

Draft	Date	Change
9	2026-10-10	Cloud Run found to block outbound UDP: the time guardian now measures against four sources over TLS (PTB web clock, traceable; Akamai; Google and Cloudflare through the HTTP Date header), with a 2.5-second timeout and at most 100 ms uncertainty per source; no token without PTB; NIST, Netnod and NTP sources dropped on Cloud Run, NTS no longer a target; measurement on demand only (no background guardian); staging evidence (test ceremony, 43 of 43 acceptance checks, measured offsets) and production infrastructure recorded; production opens without waiting for the 14-day campaign, which runs in parallel on the test service; open items updated
8	2026-10-09	Simplified design approved by the owner on 9 October 2026: two-level hierarchy (no issuing CA; certReq returns the TSA certificate only; root CRL only); TSA key in Cloud KMS at SOFTWARE protection level, with pre-signing checks; offline software root key; single operator stated, with compensating controls; on-demand time guardian with named NTP sources, 1-second accuracy, fail-closed, NTS a target; journal in a locked 12-year retention bucket, daily batch timestamped by DigiCert (FreeTSA fallback) and anchored through OpenTimestamps; same-company disclosure for VerifBox; UK eIDAS Article 19 and UK GDPR obligations; endpoints aligned with the service; public statement limited to "designed to RFC 3161 / RFC 5816"; A7 (HSM attestation) superseded

Draft	Date	Change
7	2026-10-08	A certReq written out as FALSE rejected as non-DER, and the certificates returned when it is TRUE stated; the check on logs limited to the logs stored in the TSA projects, what the cloud provider keeps elsewhere being outside the operator's control
6	2026-10-08	Any request extension rejected, as RFC 3161 requires; lease withdrawn on entering FAIL, and automatic return to SYNC except after a clock step; genTime read by the signer under the lease; leap-second sentence corrected; declared accuracy covering the worst case of SYNC, with a source traceable to UTC among the usable ones; revocation judged from a CRL issued after genTime, and an undetermined result without evidence after a compromise; daily batches timestamped by a third party and anchored in Bitcoin, no longer optional; retention raised to 12 years; the statement on client addresses limited to logs the operator keeps until the platform logs are checked; the public statement allowed only after the production run and the opening; no token issued to clients before the criteria pass; the leaf certificate built from the module's public key; the stale open item on D1 and D2 replaced; the FAIL state lists every unmet condition of SYNC; notices listed on /tsa/status

Draft	Date	Change
5	2026-10-07	Specification and risk register cited under their IIC TSA titles; genTime rounded to the nearest second; the module reached through a service identity; root CRL published; a reason code on every revocation, and the cutoff as invalidity date on keyCompromise; independent evidence for renewal may also be a Bitcoin anchor. Proposed for review: D1, the signature digest of each token kept in the issuance log as evidence after a compromise; D2, a leaf certificate valid for up to 135 months whose key signs for at most 15 months
4	2026-10-07	Certificate-expiry wording aligned with the validation rules; a renewal timestamp from the IIC TSA itself adds no independence; measured accuracy and time thresholds written into this policy before its approval; draft number and date shown in the identification table
3	2026-10-07	Compromise wording aligned with the Practice Statement; validation of tokens, including in the long term, added; personal-data statement made exact; third-party TSA for the journal cross-check
2	2026-10-07	The TSA is placed in the Internet Identity Card business line and renamed IIC TSA; policy OID assigned. Copies titled "VerifBox TSA" are withdrawn; no change of substance
1	2026-10-04	Initial draft, including the corrections of the first reviews

© 2026 Https Card — Internet Identity Card Ltd, 124 City Road, London EC1V 2NX, United Kingdom. UK Company No. 09168431. Governed by the laws of England and Wales.