

MOCK HEARING – SIMULATION ONLY – TECHNICAL EXHIBIT

EXHIBIT 16 – CERTIDOX INDEPENDENT VERIFICATION BUNDLE FOR EXHIBIT 05

Exhibit 16 consists of this explanatory document together with the native Certidox verification bundle associated with Exhibit 05.

The bundle allows a reviewing party to reconstruct and examine the certified reference document and to verify its cryptographic integrity and associated RFC 3161 timestamp **without using the Certidox mobile application**. The bundle is provided as a native ZIP archive so that the verification procedures can be performed directly by the reviewing party using standard technical tools.

WHAT THE BUNDLE INDEPENDENTLY ESTABLISHES

Using the native verification bundle, a reviewing party can independently:

- **reconstruct the certified reference document as document.pdf;**
- calculate the SHA-256 of the reconstructed certified payload and **verify that it matches the payload_sha256 value recorded in manifest.json;**
- calculate the SHA-256 of the exact `manifest.json` file and **verify that it matches the message imprint contained in the DigiStamp RFC 3161 timestamp response;**
- verify the associated RFC 3161 timestamp using DigiStamp trust material obtained independently of Certidox; and
- independently confirm the linkage between the QR code, the reading key, `document.enc`, the reconstructed `document.pdf` and the corresponding `manifest.json`.

Except for current live certification status, these procedures do not require access to the Certidox live verification interface or reliance on a Certidox-generated verification display.

VERIFICATION CHAIN

The independent verification path can be summarized as follows:

1. Independent decoding of the QR code appearing in Exhibit 05 yields a Certidox verification URL containing a verification identifier and reading key. The reading key is identical to the value contained in `reading_key.txt` in the native verification bundle.
2. Using that reading key, AES-256-GCM decryption of `document.enc` reconstructs the certified reference document as `document.pdf`.
3. The reconstructed `document.pdf` has SHA-256:
e90e117a63524859e834c4038c587de907c5e1b0c29367e213727cf93c59cd9e
That value matches `payload_sha256` in `manifest.json`.

4. The exact `manifest.json` has SHA-256:
3fe8ae11c52a862eb3d65b41988fb2a06ff24424051dfef2e64a0a0851cc3b40
That value matches the SHA-256 message imprint contained in `response.tsr`.
5. The RFC 3161 response records a DigiStamp `genTime` of September 24, 2026 at 14:58:26.26 GMT, corresponding to 9:58:26.26 AM CDT.

Independent verification reconstructs the exact certified reference document linked to the QR code appearing on Exhibit 05. That reference shows Service A and Option B selected and Options C and D unselected. Its SHA-256 matches `payload_sha256` in the exact `manifest.json`, whose own SHA-256 matches the message imprint contained in the DigiStamp RFC 3161 timestamp response.

This chain permits the reviewing party to move from the QR code appearing on Exhibit 05 to the exact reconstructed certified reference document and its associated independently verifiable timestamp evidence.

INDEPENDENT RECONSTRUCTION OF THE CERTIFIED DOCUMENT

Using the encrypted document data and the reading key contained in the bundle, the reviewing party can reconstruct the certified reference document as `document.pdf`.

The reconstructed document has the following SHA-256 hash:
e90e117a63524859e834c4038c587de907c5e1b0c29367e213727cf93c59cd9e

Independently calculating the SHA-256 of Exhibit 05 produces the same value. Accordingly, the document reconstructed from the verification bundle is byte-for-byte identical to Exhibit 05.

The reconstructed document shows:

- **Service A selected;**
- **Option B selected;**
- **Option C not selected; and**
- **Option D not selected.**

The reconstructed certified document shows the same service selections as Exhibit 05: Service A and Option B selected, and Options C and D unselected.

The reconstructed document also displays Laura Bennett and ApexGrid Infrastructure, Inc., consistent with the information displayed through the Certidox verification interface in Exhibit 15.

The reconstruction therefore provides an independently reproducible copy of the exact certified reference document associated with the QR code appearing on Exhibit 05.

QR CODE LINKAGE

The QR code appearing on Exhibit 05 can be decoded independently.

The QR code contains information linking the document to its Certidox verification material, including:

- a Certidox verification identifier whose bundle-specific portion corresponds to the `certification_id` recorded in `manifest.json`; and
- the reading key contained in `reading_key.txt` and used to decrypt `document.enc` and reconstruct `document.pdf`.

The certification identifier recorded in `manifest.json` is:
`c790931bc5a9e2b02f126c8507ee42ff1c222f69dfda9bd4f1beb9445f45921c`

Using the reading key obtained from the QR code appearing on Exhibit 05, the reviewing party can reconstruct `document.pdf` from `document.enc`. **The SHA-256 of that reconstructed document matches `payload_sha256` in `manifest.json`. The SHA-256 of the exact `manifest.json`, in turn, matches the message imprint contained in the DigiStamp RFC 3161 timestamp response.**

This provides a direct technical linkage between the QR code appearing on Exhibit 05, the verification bundle, the reading key, the reconstructed certified reference document and the independently verifiable timestamp evidence.

INDEPENDENT TIMESTAMP VERIFICATION

The bundle includes an RFC 3161 timestamp response containing a timestamp token digitally signed by DigiStamp Timestamp Authority.

The timestamp verification can be performed without contacting Certidox. The DigiStamp CA certificate material used to validate the timestamp response can be obtained directly from DigiStamp by the included verification procedure or independently supplied by the reviewing party. The reviewing party can therefore validate the timestamp evidence **using trust material obtained independently of Certidox.**

The DigiStamp RFC 3161 timestamp does not directly timestamp `document.pdf`.

The SHA-256 of the exact `manifest.json` contained in this bundle is:
`3fe8ae11c52a862eb3d65b41988fb2a06ff24424051dfef2e64a0a0851cc3b40`

The SHA-256 message imprint contained in `response.tsr` is the same value.

The manifest, in turn, records the SHA-256 of the reconstructed certified payload as `payload_sha256:`
`e90e117a63524859e834c4038c587de907c5e1b0c29367e213727cf93c59cd9e`

That value matches the independently calculated SHA-256 of `document.pdf`.

The RFC 3161 response records a `genTime` of September 24, 2026 at 14:58:26.26 GMT, corresponding to September 24, 2026 at 9:58:26.26 AM CDT.

A successful verification reports: **Verification: OK**

and concludes: `Timestamp verification passed.`

For a fully independent assessment of the Timestamp Authority trust chain, the reviewing party can independently validate the DigiStamp Root CA certificate fingerprint against DigiStamp's official repository.

When successfully verified, the RFC 3161 timestamp establishes that **the SHA-256 hash of the exact `manifest.json` matches the message imprint timestamped at the DigiStamp `genTime`**. Because that exact manifest records the `payload_sha256` matching the reconstructed `document.pdf`, **the certified reference document is cryptographically linked to the manifest** whose SHA-256 hash matches the message imprint timestamped at the DigiStamp `genTime`.

The DigiStamp Timestamp Authority does not independently originate or attest to the accuracy of the `certification_event_time_utc` value recorded by Certidox in that manifest.

Exhibit 05 can therefore be examined against a fixed certified reference whose exact digital content and historical timestamp evidence are independently verifiable.

RECORDED CERTIFIER INFORMATION

The exact `manifest.json` also records **Laura Bennett as the certifier, lbennett@apexgrid-infra.com as the certifier email address, and ApexGrid Infrastructure, Inc. as the certifying organization.**

Because those recorded values form part of the exact `manifest.json` whose SHA-256 matches the message imprint contained in the DigiStamp RFC 3161 timestamp response, successful verification establishes that those values were contained in that manifest no later than the DigiStamp `genTime` and have not subsequently been altered without invalidating the timestamp verification.

This does not, by itself, independently establish the underlying human identity, authorship, authority or legal capacity of Laura Bennett.

DISTINCT TIME VALUES

The certification record contains distinct time values that serve different purposes:

- Certidox certification-event time: 2026-09-24T14:58:22Z, recorded in `manifest.json`, corresponding to September 24, 2026 at 9:58:22 AM CDT; and
- DigiStamp RFC 3161 genTime: September 24, 2026 at 14:58:26.26 GMT, recorded in `response.tsr`, corresponding to September 24, 2026 at 9:58:26.26 AM CDT.

These time fields arise from distinct sources and should not be treated as interchangeable. The user-facing Certidox screen reproduced in Exhibit 15 contains an interface display value of 9:58:26 AM CDT. The independent verification bundle does not, by itself, establish which interface transformation produced that displayed value, and no independent technical conclusion in Exhibit 16 relies on it.

INTEGRITY CONSEQUENCE

The reconstructed `document.pdf` is protected through a two-step integrity chain.

First, the SHA-256 of `document.pdf` must match `payload_sha256` recorded in `manifest.json`. **Any alteration to the reconstructed certified payload would produce a different SHA-256 and fail that payload-to-manifest integrity check.**

Second, the SHA-256 of the exact `manifest.json` must match the message imprint contained in `response.tsr`. **Any alteration to the exact manifest would produce a different SHA-256 and fail the RFC 3161 timestamp verification.**

Accordingly, the reconstructed certified reference document is cryptographically linked through its `payload_sha256` to the manifest whose SHA-256 hash matches the message imprint contained in the RFC 3161 timestamp response.

Where a presented digital file is claimed to be a byte-identical copy of the certified reference document, **its file-level identity can be checked directly by computing its SHA-256 hash and comparing that hash with `payload_sha256` recorded in `manifest.json`.** The integrity of that exact `manifest.json` is separately checked by comparing its SHA-256 hash with the message imprint contained in `response.tsr`.

A physical printed copy has no file-level SHA-256. A scanned, photographed, converted or otherwise rendered digital copy will ordinarily have different file-level bytes and therefore a different SHA-256. In either situation, **provided the Certidox QR code remains readable, the QR code provides the linkage used with the verification material to recover the exact certified reference document**, allowing the substantive content and service selections of the presented copy to be examined against that fixed reference.

The verification process therefore provides reproducible cryptographic evidence of the exact certified digital reference document and its cryptographic linkage to the manifest whose SHA-256 hash is independently timestamped through the RFC 3161 response, together with a fixed reference against which a digital or printed copy bearing a readable Certidox QR code associated with that reference can be examined.

REPRODUCIBILITY

The verification described above can be independently repeated from a working copy of the native bundle using the procedures and standard tools identified in the bundle documentation. After preserving the original ZIP unchanged, the reviewing party can first compute the SHA-256 of the native ZIP and verify that it matches the SHA-256 stated under NATIVE FILE below.

The reviewing party can then reconstruct the certified document from a working copy of that bundle by running `python3 decrypt.py` and verify the RFC 3161 timestamp by running `bash verify_timestamp.sh`.

The reviewer can independently calculate the SHA-256 of the reconstructed `document.pdf` and compare it with `payload_sha256` in `manifest.json`, then calculate the SHA-256 of the exact `manifest.json` and compare it with the message imprint contained in `response.tsr`.

The included `decrypt.py` script is provided in readable source form and decrypts `document.enc` using AES-256-GCM with the reading key contained in the bundle.

Successful AES-256-GCM decryption verifies the authentication tag associated with `document.enc`. The independent historical integrity and timestamp evidence for the reconstructed document is provided by the separate SHA-256/RFC 3161 verification described above.

The included `verify_timestamp.sh` script is likewise provided in readable source form and performs RFC 3161 verification using OpenSSL `ts -verify`. A reviewing party can therefore inspect the verification logic before execution and reproduce the relevant operations using independently available tools.

The included scripts are convenience implementations rather than trust anchors; the underlying cryptographic operations can be independently reproduced using standard AES-256-GCM and RFC 3161 verification tools.

An independent reviewer correctly performing those operations on an unchanged copy of the native bundle **will obtain the same reconstructed `document.pdf`, the same payload SHA-256, the same manifest SHA-256 and the same RFC 3161 verification result.**

Except for current certification status, the reconstruction and timestamp verification described in this exhibit do not require access to the Certidox live verification interface or reliance on a Certidox-generated verification display.

For this bundle, successful reconstruction produces `document.pdf`. A successful timestamp verification reports `Verification: OK` and concludes `Timestamp verification passed`.

RELATIONSHIP TO EXHIBIT 15

Exhibit 15 shows the result obtained through the Certidox user interface after scanning the QR code contained in Exhibit 05.

Exhibit 16 provides a separate technical method for reconstructing and verifying the certified reference document without using the Certidox mobile application.

The two exhibits therefore serve different purposes:

- **Exhibit 15: user-facing Certidox verification.**
- **Exhibit 16: independent reconstruction and technical verification of the certified reference document and its associated RFC 3161 timestamp.**

CURRENT STATUS

The bundle does **not** independently establish the current Certidox status of the certification.

A status such as “**Document VALID**” is a live value maintained by Certidox and may change after certification. Verification of current status therefore requires access to the current Certidox status information.

SCOPE AND LIMITATIONS

Exhibit 16 does not, by itself, establish:

- the legal admissibility or evidentiary weight of Exhibit 05;
- the identity, authorship, authority or legal capacity of Laura Bennett independently of the other evidence in the record;
- the factual truth or legal effect of the content of Exhibit 05;
- who created or modified Exhibit 07;
- whether Exhibit 07 reflects a selection actually made by Laura Bennett;
- delivery to or receipt of any related email;
- file-level identity between the certified digital reference document and a printed, scanned, photographed, converted or otherwise rendered copy;
- the current live Certidox certification status; or

- any ultimate conclusion concerning fraud, alteration, authorship, intent or liability.

Those matters are outside the scope of Exhibit 16 and, where relevant, remain for determination from the complete evidentiary record.

NATIVE FILE

Native exhibit:

EXHIBIT_16_Certidox_Independent_Verification_Bundle_for_Exhibit_05.zip

SHA-256 of native ZIP:

7cefe496a376cbfb63c4ebd832d773914b4c33d361f56b26cbb715cd3c39a51d

This SHA-256 identifies the exact native ZIP archive accompanying this explanatory exhibit. A matching SHA-256 provides strong cryptographic evidence that a copy of the native ZIP is byte-for-byte identical to the bundle identified above.