

MOCK HEARING – SIMULATION ONLY – TECHNICAL EXHIBIT

EXHIBIT 18 – CERTIDOX INDEPENDENT VERIFICATION BUNDLE FOR EXHIBIT 04

Exhibit 18 consists of this explanatory document together with the native Certidox verification bundle associated with Exhibit 04.

The bundle allows a reviewing party to reconstruct and examine the certified email content representation 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 and, where indicated, Exhibit 04, a reviewing party can independently:

- **reconstruct the certified email-content representation as `document.json`;**
- calculate the SHA-256 hash of the reconstructed certified payload and **verify that it matches the `payload_sha256` value recorded in `manifest.json`;**
- calculate the SHA-256 hash 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 embedded in Exhibit 04, the reading key, `document.enc`, the reconstructed `document.json` and the corresponding `manifest.json`.**

Except for current live certification status, these procedures do not require access to the original mailbox, the original mail server, 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 Certidox QR code embedded in Exhibit 04 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 email-content representation as document.json.
3. The reconstructed document.json has SHA-256:
5518dcace7d4d9a118835298528ff1d057e03444ea14ce9f72359372546115c1
That value matches payload_sha256 in manifest.json.
4. The exact manifest.json has SHA-256:
389bc50a167c6cd7897935ef08c46cea196e3b1a51cb2a3e4826b13ebd645278
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 15:18:34.82 GMT, corresponding to 10:18:34.82 AM CDT.

Independent verification reconstructs the certified email-content representation associated with Exhibit 04. That representation records Laura Bennett as the From address, smitchell@southbridge-tech.com as the recipient address, the subject “Re: Order ST-AG-2026-0914 – Election Period,” the message date, the certified message body electing **Option B – Predictive Grid Analytics** and expressly stating that **Options C and D remain unselected**, together with protected attachment metadata. 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 with the email to the exact reconstructed certified email-content representation and its associated independently verifiable timestamp evidence.

INDEPENDENT RECONSTRUCTION OF THE CERTIFIED EMAIL CONTENT

Using the encrypted data and the reading key contained in the bundle, the reviewing party can reconstruct the certified email content representation as: `document.json`

The reconstructed JSON has the following SHA-256 hash:
5518dcace7d4d9a118835298528ff1d057e03444ea14ce9f72359372546115c1

The reconstructed certified content includes:

- From: “Laura Bennett” <lbennett@apexgrid-infra.com>;
- To: smitchell@southbridge-tech.com;
- message date: September 24, 2026 at 10:18:30 AM CDT;
- subject: Re: Order ST-AG-2026-0914 – Election Period;
- the complete certified HTML email body, including embedded image data;
- body type: html; and
- two attachment-metadata entries.

The reconstructed certified email content states: “ApexGrid elects Option B – Predictive Grid Analytics.” It further states: “Options C and D remain unselected.” The certified HTML body also contains Laura Bennett’s signature block, including her name, title, organization and email address.

The certified HTML body also contains the earlier September 24 email thread. Within that quoted history, Laura Bennett’s earlier message asks Sarah Mitchell to confirm the extension by return email and states that all other terms of the Order Form would remain unchanged.

The reconstructed From address, recipient address, subject and current-message content correspond to the fields and visible message content displayed through the Certidox verification interface in Exhibit 17. Exhibit 17 also displays two attachment entries corresponding to the two attachment-metadata entries recorded in `document.json`.

RELATIONSHIP TO EXHIBIT 04

Exhibit 04 is the native .eml email file containing the QR code used for the Certidox verification.

The bundle does not reconstruct Exhibit 04 as a byte-for-byte .eml file. Instead, it reconstructs the certified email-content representation stored for verification purposes, including the substantive certified fields and attachment metadata identified above.

Accordingly, the SHA-256 and RFC 3161 verification described in this exhibit apply to the reconstructed certified email-content representation, **not to the Exhibit 04 .eml file as a whole.**

The bundle does not reproduce all transport-level information such as all mail headers, routing information or the complete MIME structure.

QR CODE LINKAGE

The Certidox QR code embedded in Exhibit 04 **can be decoded independently.**

The QR code contains information linking the certified email 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.json`.

The certification identifier recorded in `manifest.json` is:
4fef01769d61f5f4c1f7fc93a61b8394cd55b92059a4956f05ff575efabcb81d

The Certidox QR image data embedded in the reconstructed certified HTML body, when Base64-decoded, yields a PNG that is byte-for-byte identical to the Certidox QR-code PNG embedded in Exhibit 04. The encoded QR image data forms part of the certified HTML body contained in `document.json`, whose SHA-256 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 Certidox QR code embedded in Exhibit 04, the verification bundle, the reading key and the reconstructed certified email-content representation.

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.json`.

The SHA-256 of the exact `manifest.json` contained in this bundle is:
389bc50a167c6cd7897935ef08c46cea196e3b1a51cb2a3e4826b13ebd645278

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`:
5518dcace7d4d9a118835298528ff1d057e03444ea14ce9f72359372546115c1

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

The RFC 3161 response records a `genTime` of September 24, 2026 at 15:18:34.82 GMT, corresponding to September 24, 2026 at 10:18:34.82 AM CDT.

The RFC 3161 response uses SHA-256 and records the manifest hash as its message imprint. Signature and trust-chain validation can be performed using DigiStamp trust material obtained independently of Certidox.

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.json`, **the certified email-content representation 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.

RECORDED CERTIFIER INFORMATION

The exact `manifest.json` 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, personal authorship, authority or legal capacity of Laura Bennett.

DISTINCT TIME VALUES

The certified email record contains distinct time values that serve different purposes:

- **Message date:** 2026-09-24T15:18:30.000Z, recorded in `document.json`, corresponding to September 24, 2026 at 10:18:30 AM CDT;
- **Certidox certification-event time:** 2026-09-24T15:18:32Z, recorded in `manifest.json`, corresponding to September 24, 2026 at 10:18:32 AM CDT; and
- **DigiStamp RFC 3161 `genTime`:** September 24, 2026 at 15:18:34.82 GMT, recorded in `response.tsr`, corresponding to September 24, 2026 at 10:18:34.82 AM CDT.

The native Exhibit 04 .eml file contains a separate Date header corresponding to September 24, 2026 at 10:18:32 AM CDT. That native-header value is distinct from the 10:18:30 AM CDT message-date field recorded in `document.json`.

The certified email-content view reproduced in Exhibit 17 displays 10:18:30 AM CDT, corresponding to the message-date field recorded in `document.json`. The verification-result screen in Exhibit 17 displays 10:18:34 AM CDT. The independent verification bundle does

not, by itself, establish which underlying source field or interface display transformation produced that latter value, and no independent technical conclusion in Exhibit 18 relies on it.

INTEGRITY CONSEQUENCE

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

First, the SHA-256 of `document.json` 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 email-content representation 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.

The SHA-256 verification described above applies to the reconstructed `document.json`, not to Exhibit 04 as a whole. Exhibit 04 is a native `.eml` file and its file-level SHA-256 is therefore not the object of this verification. Likewise, a printed email, screenshot, PDF rendering, forwarded copy or other transformed representation will ordinarily have different file-level bytes, or in the case of a physical printout no file-level SHA-256 at all.

In those situations, **provided the Certidox QR code remains readable**, independent decoding of the QR code provides the reading key used with the verification bundle to **reconstruct the exact certified email-content representation**, allowing the corresponding content and certified fields of the presented email or copy to be examined against that fixed reference.

The verification process therefore provides **reproducible cryptographic evidence of the exact certified email-content representation** 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 Exhibit 04 or another representation bearing the associated readable Certidox QR code 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 content 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.json` 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 plaintext content 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.json`, 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 original mailbox, the original mail server, the Certidox live verification interface or reliance on a Certidox-generated verification display.**

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

RELATIONSHIP TO EXHIBIT 17

Exhibit 17 shows the result obtained through the Certidox user interface after scanning the QR code contained in Exhibit 04.

Exhibit 18 provides a separate technical method for reconstructing and verifying the certified email content without using the Certidox mobile application.

The two exhibits therefore serve different purposes:

- **Exhibit 17:** user-facing Certidox verification.
- **Exhibit 18:** independent reconstruction and technical verification of the certified email-content representation and its associated RFC 3161 timestamp.

ATTACHMENT INFORMATION

The reconstructed `document.json` contains two attachment-metadata entries:

1. filename: 144-6AB53980-3-239465C0
MIME type: image/png
byte size: 12,184
SHA-256:
be874381a9eadb453b0a097d427f48d6ed84f02aa1ede71cfce8dd43c84ccb1f
2. filename: Southbridge_ApexGrid_Enterprise_Order_Form_Updated.pdf
MIME type: application/pdf
byte size: 150,418
SHA-256:
e90e117a63524859e834c4038c587de907c5e1b0c29367e213727cf93c59cd9e

The attachment files themselves are not included in this verification bundle as separate attachment payloads. The attachment metadata, including the recorded SHA-256 values, forms part of `document.json` and is protected through the payload-to-manifest-to-RFC-3161 integrity chain described above.

The first attachment-metadata entry is an image/png entry embedded within the quoted earlier email in the certified HTML body. Its recorded SHA-256 matches the corresponding decoded PNG bytes. Independent decoding of that PNG yields a Certidox verification identifier different from the Exhibit 04 certification identifier. **It must therefore not be confused with the Certidox QR code for Exhibit 04.**

The Certidox QR code embedded in the Exhibit 04 message itself decodes to the certification identifier
4fef01769d61f5f4c1f7fc93a61b8394cd55b92059a4956f05ff575efabcb81d and the reading key contained in `reading_key.txt`.

The second metadata entry records the PDF attachment. Independently calculating the SHA-256 of Exhibit 05 produces the same value, e90e117a63524859e834c4038c587de907c5e1b0c29367e213727cf93c59cd9e. **For purposes of this mock hearing, and consistent with the Joint Stipulation, this matching SHA-256 establishes that Exhibit 05 is byte-for-byte identical to the PDF attachment** whose SHA-256 was recorded in the certified email-content representation associated with Exhibit 04.

A matching filename alone would not establish attachment identity; the SHA-256 comparison provides the byte-level linkage.

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 18 does not, by itself, establish:

- delivery to or receipt of the email by Southbridge Technologies, Inc.;
- the complete native .eml file, complete transport history or complete MIME structure of Exhibit 04;
- that the recorded From field, by itself, proves which human physically initiated the transmission;
- the personal authorship, authority or legal capacity of Laura Bennett;
- the factual truth or legal effect of the certified message content;
- the content or identity of an attachment that has not been separately supplied for SHA-256 comparison;
- attachment identity based solely on a matching filename;
- the current live Certidox certification status;
- the authenticity, provenance or authorship of Exhibits 06 or 07;
- the legal admissibility or evidentiary weight of Exhibit 04; or
- any ultimate conclusion concerning fraud, alteration, intent or liability.

Those matters are outside the scope of Exhibit 18 and, where relevant, remain for determination from the complete evidentiary record.

NATIVE FILE

Native exhibit:

EXHIBIT_18_Certidox_Independent_Verification_Bundle_for_Exhibit_04.zip

SHA-256 of native ZIP:

0cf5339cc16c25e9772aabb924a5f36fd4559ecaa5b933386a9d09b5db90a61b

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.**