CERTIFICATION OF AUTHENTICITY OF ELECTRONIC EVIDENCE

Fed. R. Evid. 902(13) and 902(14); 28 U.S.C. § 1746

ProofSnap generated this document from the contents of the evidence package identified below. It is an unsigned template. It has no effect until a qualified person with personal knowledge reviews it, completes the blank fields, and signs it. Where a blank below already contains text, that text was taken from the record of this capture and ProofSnap has not verified it: check it, and correct it if it is wrong. Read every statement before you sign. You, not the software, make this certification. File the completed and signed copy with the court as a separate document accompanying the evidence package. Do not place the signed copy back inside the package: the package's integrity rests on the SHA-256 hash values recorded in manifest.json, so any change to this file would make its hash value no longer match the recorded value, and verification would report the package as altered.

I. Declarant

1. My name is _____

2. My position or title is _____

3. My employer or organization is _____

4. My qualifications to make this certification, and the basis of my familiarity with the process described in Section III, are as follows:

(State facts rather than conclusions. For example: I performed the capture described in Section III, I am familiar with how the capture system works, and I use it in the ordinary course of my work.)

5. I have personal knowledge of the matters stated in this certification, and if called as a witness, I could and would testify competently to them.

II. The records this certification covers

6. This certification covers the electronic evidence package (referred to below as the package) identified as follows:

Evidence ID:	ps_e41dffe2-eda0-4652-bf04-76bfcd1f3ff4
Source URL:	https://www.arabnews.com/node/2655769/middle-east
Captured (UTC):	2026-08-25T06:53:54Z (UTC)
Capture software:	ProofSnap Chrome Extension version 1.19.4
ProofSnap account:	f5RWV90LkzU4BYrpmGKxAyDcid83

Files hashed before this certification was written: 32

The hash values of those files are set out in Section IV, subject to the length limit stated there. Not every file in the package carries a hash value there, because a file created after those values were computed cannot carry one: that is the case for the signature over the manifest, for the public key that verifies it, and for the timestamp files. Two files created after the hash values were computed are nonetheless hashed in manifest.json: chain_of_custody.json and this certification itself. The file verification/VERIFICATION_GUIDE.txt explains how the timestamps and their validation data are checked.

III. The process that generated the records (Rule 902(13))

7. The records were generated by an electronic process running in a Chrome browser on the computer I used to make the capture. The system performed the following steps, in this order:

(a) It obtained the current time from independent network time sources and compared those times against each other.

(b) It loaded the source URL in a browser tab and recorded the HTTP response headers returned by the server, and the addresses to which the hostname resolved, as reported by two independent DNS resolvers queried separately.

(c) It saved the page as displayed: a screenshot of the page, the page HTML with its stylesheets, and the text content of the page as rendered. Where it added a band of its own to the top of the screenshot, recording the address, the time and the identifier of this capture, it did so before the hash values in Section IV were computed, so those values are of the annotated image; the page content below the band is unchanged, and chain_of_custody.json records the hash of the image as it was before the band was added.

(d) It recorded the changes it made to the page in order to capture it, such as hiding a cookie banner or a floating header, and it recorded how much of the page the screenshot covered.

(e) It computed a SHA-256 hash value and a SHA-512 hash value for every file that existed when manifest.json was written, and wrote those values into manifest.json together with the length of each of those files in bytes.

(f) It generated an RSA-4096 key pair for this capture and signed manifest.json with the private key. The signature is manifest.sig, and the public key is publickey.pem.

The system also wrote a step-by-step log of the capture (forensic_log.json) in which each entry carries the hash value of the entry before it, so that the removal or alteration of an entry can be detected. The log is sealed before the hash values in step (e) are computed, because manifest.json records a hash value for the log itself; the steps that follow the seal are evidenced by manifest.sig and by the timestamp files rather than by entries inside the log.

8. The records identified in paragraph 6 were generated by the process described in paragraph 7. Based on my familiarity with that process and on my observation of this capture, that process produces an accurate result, and it produced an accurate result here.

9. The screenshot reaches the end of the page. It was assembled from successive views of the page, and enough of them were taken to cover it. The counts are recorded in chain_of_custody.json in the package.

IV. Digital identification of the records (Rule 902(14))

10. Each file included in the count "Files hashed before this certification was written" stated in paragraph 6 was identified by a SHA-256 hash value at the time the file was produced. A SHA-256 hash value is computed from the entire contents of a file. Changing a single byte of a file changes its hash value, and no two different files are known to share a SHA-256 hash value.

11. The hash values below were computed by the system at the time of capture. They are reproduced from manifest.json, which lists every file the package carries a hash value for. Any person who has the package can recompute these values from the files and compare them with the values listed there.

metadata.json
SHA-256: 76b3870bb90b092963e9f1fb266075f4f00a3dac87ce01da91e19727be7173a6

page.html
SHA-256: ec84861551c0be529731fe9bfd52f6c4aabf325fea7c5ec3236df017e9b1d9fe

screenshot.jpeg
SHA-256: bf41177408410e50ed51dbccbca230d9075cd1425b528f70e221079a07fcb851

evidence.pdf
SHA-256: 3f2de001a85c559c91a5e7771d033cef57fba40353c80f85badd2373847dc791

provenance_certificate.pdf
SHA-256: be6ff79da00a6ee4a51fec982209079e3c23dbb48872fedb9885a3758b5d5c4c

domtextcontent.txt
SHA-256: 5ddb735aa5846d356b90bec9866206dc6432d8073b7f5ba96c744d440143cd49

c2pa.json
SHA-256: 4f905aac228b5c4608d30721e15940722316945c5c57f19d2531dbf8391d93d9

images/001_4743007-217510164.jpg
SHA-256: 42dc35ee97ca51e272cfdcb23fa9c5eda2a299e01e6b3d0c074368477ba2b3ad

images/002_4743227-561507573.jpg
SHA-256: 5af5170727b712dd079dba5db6331594d6c0362bc56c35a840e8a47610e40b60

images/003_4744083-93996089.jpg
SHA-256: daa0c358b866669aa49da018dc87ca2fe8959d5d9d78fe243df75404ff199b4d

images/004_job4.jpg
SHA-256: e60f6dcd7a8a16869b0edef459135dd0ef7e0c024a0ceaea594d895ebef7956f

images/005_4744095-775220797.jpg
SHA-256: 9a23c3cc0052f115cd2c7e3c3524ca5a24a10844e3493550cbe38afc744d08d8

images/006_4744096-550046716.jpg
SHA-256: 14e102bec3e39484181bce45c86ee09ee1a475c27986587a21d29e337ee97c2f

images/007_4744101-1865854095.jpg
SHA-256: 3a242149480b7115030cdd2a1a5ec0856a5db8468d072de69aa72f774257ee62

images/008_4744105-1500648103.jpg
SHA-256: a6fb98f2768711f3eb732c81166706aeecb7e11dfc8ad9f399f6c1f401ef9e0c

images/009_4744109-1518984547.jpg
SHA-256: 7ae7fae008a2174d94ebf72c20596d9064d60810f548310ee655764d3772b974

images/010_4743851-2041867929.jpg
SHA-256: a0dbeb03b9306e7f18f2570ed44e926577bbda63c9344c0200bd2b7e519022b3

images/011_4743930-1639330254.jpg
SHA-256: 511e1d3dc9369066ca89aed073ba216e49c75479a9dcd595cdf135cfe383af27

images/012_4744093-1047710634.jpg
SHA-256: 33839f1841ecddf22a151ee9db0bd095fe982ab280e298c32defaff6b9d78e82

images/013_4744092-74082202.jpg
SHA-256: 33839f1841ecddf22a151ee9db0bd095fe982ab280e298c32defaff6b9d78e82

images/014_4570111-birth-yhffcixx.jpg
SHA-256: 746a8c1a3cecf43298c61641e8621699fae8a1ec4247c8373aaa2e051dc298e5

images/015_4570110-birth-yhffcixx.jpg
SHA-256: 746a8c1a3cecf43298c61641e8621699fae8a1ec4247c8373aaa2e051dc298e5

images/016_45-logo-mobile-new.webp
SHA-256: 5e4c0e55654d34580aefa552c14e8868e716948d0e0f8eca935ee191c36d02de

images/017_45-logo-mobile-new.webp
SHA-256: 5e4c0e55654d34580aefa552c14e8868e716948d0e0f8eca935ee191c36d02de

The remaining 8 files and their hash values are listed in manifest.json in the package. They are omitted here only to keep this document to a readable length.

12. The file manifest.json was signed with the RSA-4096 private key generated for this capture, using RSASSA-PKCS1-v1_5 with SHA-256. The signature is manifest.sig, and the corresponding public key is publickey.pem. The commands for verifying the signature and recomputing every hash value are set out in verification/VERIFICATION_GUIDE.txt in the package, and the scripts verification/verify.sh and verification/verify.ps1 run those commands.

V. Time

13. The SHA-256 hash value of manifest.json was submitted to the OpenTimestamps protocol, which commits that value to the Bitcoin blockchain. Where the submission returned a proof, that proof is the file manifest.json.ots in the package; where it did not, that file is absent and no OpenTimestamps proof is offered. When the commitment has been confirmed in a Bitcoin block, the proof shows that manifest.json, and therefore every hash value listed in it, existed no later than the time of that block.

VI. Declaration

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed on (date): _____

Executed at (city and state, or country): _____

Signature: _____

Printed name: _____

VII. Notice to the adverse party and limits of this certification

Rules 902(13) and 902(14) each incorporate the notice requirement of Rule 902(11). Before the trial or hearing, the proponent must give an adverse party reasonable written notice of the intent to offer the record, and must make the record and this certification available for inspection, so that the party has a fair opportunity to challenge them. Serving that notice is the proponent's responsibility.

Self-authentication under Rule 902 concerns authenticity only. It relieves the proponent of the need to offer extrinsic evidence of authenticity. It does not make a record admissible, it does not affect an objection based on hearsay, relevance, or any other rule, and it does not establish who wrote, published, or controlled the captured content. A hash value shows that a file has not changed since the hash value was computed; it says nothing about whether the captured content is true. ProofSnap is software: it certifies nothing and testifies to nothing, and every statement in this document is the declarant's. ProofSnap does not provide legal advice, and this template is not legal advice.