The Provenance Protocol

A backwards-compatible, cryptographically-secure, method for AI image identification

Boyd Kane[*]

December 31, 2024

Introduction

The Provenance Protocol allows anyone to verify the source of an image, PDF, or other digital media, thereby making AI-generated content impossible to pass off as human-generated content.

This is accomplished in two steps:

1. All content is untrustworthy by default. Everything is assumed to be from a bad actor unless valid provenance is attached.
2. The Provenance Protocol provides a backwards-compatible method of attaching proof-of-creation information to most forms of digital media.

This means that:

- good actors can incrementally begin adding provenance to content that they want to prove their authorship of.
- Content networks (social media sites, news websites, academic publications) can integrate provenance-verification into their platform, making it simple for their users to verify a piece of content was human-generated.

Background

The Rise of AI-Generated Content

In recent years, advancements in generative machine learning techniques have enabled the creation of sophisticated tools capable of generating high-quality digital media, such as images, videos, and text. These tools have been used to

[*] boyd r kane at google's mail service dot com, or @beyarkay online

create fake news articles, deepfakes, and other forms of misinformation that can be difficult to distinguish from “real” (human-generated) content.

The proliferation of AI-generated content has significant implications for various fields, including journalism, academia, and online publishing. The ability to generate convincing yet false information can undermine trust in institutions, manipulate public opinion, and even influence elections.

Current Challenges

Currently, there is no standardized method for verifying the origin of digital media. As a result, it is often impossible to determine whether an image or article was created by a human or an AI algorithm. This lack of transparency makes it difficult to trust online content, especially when it comes to sensitive information or high-stakes decisions.

Furthermore, existing methods for detecting AI-generated content often rely on heuristics or machine learning models that can be easily evaded by sophisticated attackers. These vulnerabilities make it challenging to establish a reliable means of verifying digital media authenticity.

The Need for Provenance

In the face of these challenges, there is a growing need for a standardized method of attaching provenance information to digital media. Provenance refers to the chain of custody or ownership history of an object or piece of content. In the context of digital media, provenance would provide a way to verify that a particular image or article was created by a specific person or organization.

By establishing a widely accepted protocol for attaching provenance information, it becomes possible to build trust in online content and ensure that only authentic human-generated content is disseminated. This has significant implications for various fields, including journalism, academia, and online publishing, where the accuracy and reliability of digital media are essential for informed decision-making.

File Format Metadata

The majority of digital media file formats provide a method for attaching “metadata” to an image, PDF, or other piece of content. Usually, this metadata is used for storing information such as the camera settings used to take the photo or the global coordinates of the where the photo was taken.

Critically, this metadata is stored and transferred along with the rest of the file, and follows the actual “content” of the file wherever it may go. Existing file sharing systems such as social media websites already contain the logic to work with metadata.

Cryptographic Signatures

Asymmetric cryptography is a method of transforming a message into an unreadable form using one key (typically called the secret or private key) in such a way that this unreadable form can be converted back into the original message only by using the public key associated with the original secret key. This is the basis for the secure internet that allows communication without worry about bad actors reading the communication.

The key pair (consisting of the secret key and the public key) is generated together, and then the secret key is kept secret while the public key is made publicly available. If Alice wanted to send Bob a message, Alice would encrypt the message with Bob's public key, and then only Bob's private key would be able to decrypt the message.

The technical implementation of asymmetric cryptography allows for the public key to be used as a secret key, and vice versa. This enables the creation of a cryptographically secure "signature". If Bob wanted to send a message to Alice and also prove to Alice 1. that he sent the message and 2. that the message has not been tampered with, Bob can encrypt the original message[1] with his private key, and send this private-key-encrypted message along with the original message. Alice can then decrypt the private-key-encrypted message using Bob's public key, and compare the decrypted message against the received message. If they match, then Alice must have the message that Bob sent, and Alice can know that Bob did indeed send the message.

Solution

A good actor "Grace" wanting to stake their social reputation to some digital content would cryptographically sign the content and attach the signature to the content using that content's metadata. This good actor then makes the associated public key generally accessible via the internet.

Someone later wanting to verify that this content did indeed come from Grace would extract the provenance metadata, retrieve Grace's public key, and use Grace's public key to verify the signature embedded in the metadata.

Both the signing and verification processes are standardised and given a simple interface via the Provenance Protocol, but they can be performed manually using well-established cryptographic techniques and standardised metadata formats.

Technical Details

The provenance protocol provides a standardised way of embedding an ED25519 signature into the metadata of various file formats and of verifying that signature

[1] Note that in practice, the original message is passed through a hashing function to create a fixed-length message "digest", and then this digest is what is encrypted.

against it's public key counterpart.

The exact method of embedding a signature into a file format's metadata will necessarily depend on the file format. Common image file formats (JPEG, PNG) are known to allow either arbitrary metadata or provide a "User Comments" metadata field which can be used for the purpose of conveying provenance information. Long-term work on the Provenance Protocol includes allocating a dedicated `provenance` metadata field in all common metadata formats.

Given a file format that allows the embedding of an arbitrary sequence of bytes as metadata, the provenance signature line requires the following information (with fields separated by single space `0x20` bytes:

1. The bytes `7e 7e f0 9f 94 8f` indicate the start-of-provenance.
2. A single space character `0x20`.
3. The semantic version number of the provenance protocol, for example `0.1.0`.
4. A single space character `0x20`.
5. The internet URL where the verifier can find the public key associated with this signature.
6. A single space character `0x20`.
7. The base64-encoded signature of the piece of content with all provenance information removed.
8. A single space character `0x20`.
9. The bytes `f0 9f 94 8f 7e 7e 0a` indicate the end-of-provenance.

Note the explicit newline `\n` byte, and the use of `f0 9f 94 8f` (the "lock with ink pen" emoji).

For example, the provenance line might look something like:

```
+--------+-------------------------+-------------------------+--------+--------+
|00000000| 7e 7e f0 9f 94 8f 20 30 | 2e 31 2e 30 20 68 74 74 |~~×××× 0|.1.0 htt|
|00000010| 70 73 3a 2f 2f 65 78 61 | 6d 70 6c 65 2e 63 6f 6d |ps://exa|mple.com|
|00000020| 2f 6b 65 79 2e 70 75 62 | 20 73 69 67 6e 61 74 75 |/key.pub| signatu|
|00000030| 72 65 2d 67 6f 65 73 2d | 68 65 72 65 20 f0 9f 94 |re-goes-|here ×××|
|00000040| 8f 7e 7e 0a             |                         |×~~_    |        |
+--------+-------------------------+-------------------------+--------+--------+
```

The internet URL must provide the public key which can verify the signature of the piece of content.

In order to verify a piece of content, the base64 signature is extracted from the metadata and decoded from base64. The public key is then retrieved from the given URL. The public key is then used to verify the signature and the content via standard cryptographic signature verification techniques.

JPEG metadata

(work in progress)

PNG metadata

(work in progress)

PDF metadata

(work in progress)

Using the Reference Implementation

(work in progress)

A reference implementation is provided to allow users to easily attach provenance information to content. This is provided as a command-line interface, with plans for a web UI in the near future.

The binary for the command-line interface is called `pvnc`. Modifying a document to add provenance to it:

```
$ pvnc sign \
    <DOCUMENT_IN> \
    --secret-key-path <SECRET_KEY_PATH>\
    --url <PROVENANCE_URL>
```

Verifying the provenance of an image:

```
$ pvnc verify <DOCUMENT_PATH>
```

Considerations

(work in progress)

- Stripping of metadata: Many social media websites strip metadata from images, destroying the provenance information.
- Bad actors can still sign AI generated content as their own. At it's core, the provenance protocol is a method for centralising trust in a way that is easier for humans to verify manually.

Security and Bug Reports

Please report all security vulnerabilities to boydrkane at google's mail server dot com, and file all bug reports via the GitHub repository `beyarkay/provenance-rs`.

Conclusion

The Provenance Protocol uses proven technology to dramatically reduce the problem of generative AI misinformation and content attribution. It does not suffer the arms-race style problems faced by AI discrimination techniques. It can be applied immediately to many file formats. It is backwards compatible with existing systems and imposes minimal compute and storage requirements. It is simple to implement and understand, and a reference implementation is provided. Good actors can adopt the protocol immediately as well as incrementally. It aligns the incentives such that good actors have a means of proving their ownership of the content.