PhotoProof: Cryptographic Image Authentication for Any Set of Permissible Transformations

Assa Naveh, Eran Tromer
Photographic images as evidence of physical scenes

- Legal
- Journalism
- Politics
- Social
- Shaming/extortion
- Dating
- Cats

Trustworthy evidence?
More photographic “evidence”
Image authentication scheme

Creator
Prover

Verifier
Viewer

image
proof
Naive solution?

• Camera signs images
  – Implemented by Nikon and Canon
  – (Very badly)

• Problem:
  Any change to the image will invalidate the signature

• Some changes may be considered legitimate:
  crop, rotate, jpeg-compress, resize, grayscale…
Goals

- Enable any specified set of permissible transformations
- Support a sequence of permissible transformation.
- Soundness
- Succinctness
  - proofs are short and can be quickly verified.
- Proofs should be zero-knowledge
  - e.g., reveal no information about removed details.
Use-case: dating sites

- Photos of users should reflect how they really look.
- They may go through some permissible editing, like cropping.
- Cropped parts are not leaked (zero knowledge).
### Image authentication: prior works

<table>
<thead>
<tr>
<th>Paper</th>
<th>Approach</th>
<th>JPEG</th>
<th>Rotate</th>
<th>Crop</th>
<th>Scale</th>
<th>Bright., contrast</th>
<th>Flip</th>
<th>Flexible spec</th>
<th>Small error</th>
<th>ZK</th>
<th>Size overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Schneider Chang 1996], [Lin Chang 2001]</td>
<td>robust hash (JPEG coefficients)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>O(1)</td>
</tr>
<tr>
<td>[Fridrich Goljan 2000]</td>
<td>robust hash</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>O(1)</td>
</tr>
<tr>
<td>[Sun Sun Yao 2002]</td>
<td>semifragile watermark (SVD)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>O(img)</td>
</tr>
<tr>
<td>[Seo Haitsma Kalker Yoo 2004]</td>
<td>robust hash (Radon)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>O(img)</td>
</tr>
<tr>
<td>[Feng Liu 2008]</td>
<td>robust hash (Bayesian)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>O(1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[Walsh 2012]</td>
<td>TPM + Nexus OS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>O(edits)</td>
</tr>
<tr>
<td>This work</td>
<td>proof-carrying data</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>O(1)</td>
</tr>
</tbody>
</table>
PhotoProof

Setup stage: system administrator defines permissible transformations and distribute keys.
Approach: Proof-Carrying Data [Chiesa Tromer 2012]

- Diverse network, containing untrustworthy parties and unreliable components.
Integrity via Proof-Carrying Data

- Every message is augmented with a proof attesting to its compliance with a prescribed policy.
- Compliance can express any property that can be verified by locally checking every node.
- Proofs can be verified efficiently and retroactively.
System administrator specifies his notion of **correctness** via a compliance predicate $C$ (incoming, local inputs, outgoing) that must be locally fulfilled at every node.
C-compliance

local input (program, human inputs, randomness)

in \rightarrow C \rightarrow accept / reject \rightarrow out

C-compliant distributed computation

C-compliant output
PhotoProof: IA Scheme from PCD

• The main idea:
  – Think of an image “life cycle” as a distributed computation.
  – Express local permissible edit as a compliance predicate.
  – Use PCD to enforce compliance of the image’s history from signed to current version.
PhotoProof: IA Scheme from PCD

- The main idea:
  - Think of an image “life cycle” as a distributed computation.
  - Express local permissible edit as a compliance predicate.
  - Use PCD to enforce compliance of the image’s history from signed to current version.

- An authentic image is a signed image which went through permissible edits exclusively.
PhotoProof prototype

• PCD implementation: 
  `<libsark>` library by SCIPR Lab 
  based on [Valiant 08] [Chiesa Tromer 2010] [Parno Gentry Howell Raykova 2013] [Bitansky Canetti Chiesa Tromer 2013] [Ben-Šasson Chiesa Tromer Virza 2014] …

• Supports RGB images

• Prototype’s supported transformations: 
  – identity
  – rectangular crop
  – horizontal and vertical flip
  – transpose
  – brightness/contrast adjustments
  – free rotation
System architecture

1\textsuperscript{st} level, Python
- user interface
- image processing
- input/output serialization
- signature verification ("outside the SNARK")

2\textsuperscript{nd} level, C++

3\textsuperscript{rd} level, C++
libsark’s PCD

4\textsuperscript{th} level
C++ code
Gadget construction

PhotoProof main application
Challenges

• Formalization and analysis
  – Raises deep problems in definition of PCD
• For our compliance predicate, we need to express image transformations as arithmetic circuits.
• Example: rotate transform.
Case Study: Rotate
Case Study: Rotate

\[(x_0, y_0)\]  
\[
\begin{pmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{pmatrix}
\]

\[(x_1, y_1)\]
Case Study: Rotate

- We need to design an arithmetic circuit that checks rotation:

- The size of the circuit affects the PCD running time.

  ![Diagram showing input image, output image, auxiliary input, and is rotate output with 0/1 value]
Images are assumed to be of fixed (maximal) size - $N \times N$

Naively, each output pixel depends on $O(N)$ input pixels, which sums to $O(N^3)$ circuit size.
To perform the rotation, we use rotation by shearing [Paeth 1986]:

The rotation matrix can be decomposed to 3 shearing operations

\[
\begin{pmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{pmatrix}
= \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ b & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}
\]

where \(a = -\tan \frac{\theta}{2}\) and \(b = \sin \theta\).

Shears can be efficiently performed with barrel-shifters with \(O(N \log N)\) gates.
Case Study: Rotate

Nondeterminism

Our circuit can make use of any nondeterministic \texttt{advice} that can help it reach a decision.

We can use this to avoid calculating the trigonometric functions. Instead we can give it $a$ and $b$ and only check they are consistent with some angle.

To do these, the prover provides $c = \sin \frac{\theta}{2}$ and $d = \cos \frac{\theta}{2}$ as \texttt{advice}, and the compliance predicate checks that:

\begin{align*}
c^2 + d^2 &= 1 \\
da &= c \\
2cd &= b
\end{align*}
Challenges and dilemmas in formalizing

• How to combine the signature scheme in the IA definition?
  – require a signature scheme or, more generally, an “originality decider”?

• Who edits the image – inside $P$ or in a separate algorithm?

• Who generates the secret signing key?
  – in reality cameras will already have signing keys.

• How to “convert” digital signatures to PCD proofs?
  – we do this in a user-transparent way.

• What is the correct proof-of-knowledge notion and how to prove it?
### Performance

<table>
<thead>
<tr>
<th>$N$</th>
<th>$#\Pi$</th>
<th>Generator (s)</th>
<th>Prover (s)</th>
<th>Verifier (s)</th>
<th>$pk$ (MB)</th>
<th>$vk$ (MB)</th>
<th>Proof size (kB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>171,815</td>
<td>16.9</td>
<td>15.9</td>
<td>0.09</td>
<td>144.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>706,959</td>
<td>32.9</td>
<td>30.7</td>
<td>0.1</td>
<td>255.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>2,966,167</td>
<td>83.2</td>
<td>91.1</td>
<td>0.14</td>
<td>635.1</td>
<td>2.7</td>
<td>2.67</td>
</tr>
<tr>
<td>128</td>
<td>12,531,999</td>
<td>367</td>
<td>423</td>
<td>0.5</td>
<td>2601.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PhotoProof prototype running times and key sizes for $N \times N$ images. Average and (normalized) standard deviation ($\sigma$) are over 10 iterations each. $\#\Pi$ is the size of the generated compliance predicate. Tests were ran on 4-core 3.4GHz Intel i7-4770 desktop with 32GB RAM.
Additional/future capabilities

• PhotoProof plugin for GIMP/PhotoShop
• Protect metadata to authenticate geo-tags, face tags, copyright messages etc.
• Include (and protect) the image edit-history in the metadata to avoid creating unwanted changes from many small permissible edits.
• Losslessly embed proofs in images.
• Use certificate chains to allow for multiple signing keys and revocation.
PhotoProof: discussion

- Longstanding open problem solved (modulo performance)
- Demonstrates the power of Proof-Carrying Data in tracking and enforcing authenticity for digital media

Open/problems:
- Performance
- Attacks at the sensor/scene level
- Analogously enforcing provenance in:
  - text (e.g., tracking citations)
  - audio (e.g., proving authenticity of a recording)
  - databases (e.g., tracking use of sensitive or unreliable information)