COUNTERFOIL: Verifying Provenance of Integrated Circuits using Intrinsic Package Fingerprints and Inexpensive Cameras

USENIX Security '20

Siva Nishok Dhanuskodi
Xiang Li
Daniel E. Holcomb
University of Massachusetts Amherst
Global semiconductor market is $500B/year
Estimated $5B/year lost to counterfeiting

CounterFoil: A practical low-cost scheme to verify provenance

1Kae-Nune and Pesseguiier, 2013
Global Distribution Network

- Consumer
- Counterfeiter
- Manufacturer
Global Distribution Network

- Consumer
- Counterfeiter
- Manufacturer
Buy chips

- chocolate
- computer
  - AAA co.
  - BEST co.
    - Model A
- paint
- potato

Reliable Systems inc.
Global Distribution Network
Counterfeiter

Greedy’s Goods

4 SALE
Global Distribution Network

Counterfeiter

Manufacturer

Consumer

Model A

Model A

Global Distribution Network
Fake tech gear has infiltrated the U.S. government
Countermeasures

• Anomaly detection
  • Look for something unusual
  • E.g. Visual inspection, classification, electronic signatures

• Authenticate provenance
  • Look for something you trust

The output classes considered for this experiment were 0 and 1. The output value 0 represents no defect condition, and the output value 1 indicates that the IC under consideration is defected. The output of the neural network may not be exactly equal to 1 or 0. Due to this, we assign the class based on how close the output is to any of the values. For example, if the output is 0.9, it is assigned class 1. If the output is 0.1, it is assigned class 0.

The backpropagation ANN was implemented using MATLAB platform. The training was done using two images with and without defects for 10 iterations each. Figure 9 shows the images used for training. The ANN was tested for detecting the counterfeit ICs.

The original image had more than 4 million pixels. This means we need 4 million input nodes, where we have only two output nodes. By employing just one hidden layer, we would lose information during the training process. This was resolved by reducing the size of the images to a resolution of 512 x 512 during the preprocessing stage. The number of input nodes for the neural network was then reduced to 262,144.

The number of hidden nodes were kept as a variable to be adjusted depending on the results. One hundred hidden nodes gave us good results. As we increase the number of nodes, the computational time rises. For our experiments, we set the number of hidden nodes as 100. The learning rate parameter was varied between 0.01 and 2. The learning rate of 0.1 gave us the minimum number of iterations required for training.
Biometrics

- Fingerprints [Galton 1895]
- Iris [Daugman, 1993]
- Ear shape [Choras et al., 2004]
- Compact Discs [Hammouri et al., 2009]
- Blank Paper [Clarkson et al., 2009]
- Retina [Hill, 1978]
- Gait Analysis [Nixon et al., 1996]
- Camera Sensors [Lukas et al., 2006]
Transfer molding creates plastic package of ICs by forcing liquified molding compound (75% filler) into cavities around wire-bonded ICs and then allowing to cure and harden.
Overview of CounterFoil

- Consumer
- Manufacturer
- Counterfeiter
- CounterFoil
  - Verifier
  - Enroller
  - Trusted integrated circuit die

- EID
- Fingerprint
- Verified chip in package

- Kpub
- Kpr

sdhanusk@umass.edu
Security and Attack Model

**Enroller**

1. Seal IC die inside package using transfer molding
2. Affix computer-readable label with enrollment id and take image of package
3. Read \( \text{eid} \), extract digitized fingerprint \( f_{\text{eid}} \)
4. Sign \( f_{\text{eid}} \) with private key \( k_{\text{pr}} \) to create \( s(f_{\text{eid}}) \)
5. Record \( e\text{id} || s(f_{\text{eid}}) \) as entry for \( \text{eid} \) in database

**Verifier**

1. Image package surface. Read label from image and fetch its record from database
2. Verify signature \( s(f_{\text{eid}}) \) using public key \( k_{\text{pub}} \)
3. Compare enrolled fingerprint \( f_{\text{eid}} \) against extracted fingerprint \( f_{\text{y}} \)

**Best Co. Chips**

**Trusted Integrated Circuit Die**
Security and Attack Model

Unique fingerprints bound to specific chip instance
- Swapping label will fail

Fingerprints difficult to forge
- Won’t create fingerprints that match enrolled label

Attacker cannot obtain $k_{pr}$
- Unable to sign own package fingerprints

enroller $ightarrow$ $k_{pr}$
1. Seal IC die inside package using transfer molding
2. Affix computer-readable label with enrollment ID and take image of package
3. Read $e_{id}$, extract digitized fingerprint $f_{e_{id}}$
4. Sign $f_{e_{id}}$ with private key $k_{pr}$ to create $s(f_{e_{id}})$
5. Record $e_{id} || s(f_{e_{id}})$ as entry for $e_{id}$ in database

trusted integrated circuit die

verifier $ightarrow$ $k_{pub}$
1. Image package surface. Read label from image and fetch its record from database
2. Verify signature $s(f_{e_{id}})$ using public key $k_{pub}$
3. Compare enrolled fingerprint $f_{e_{id}}$ against extracted fingerprint $f_v$

untrusted distribution channel

untrusted database

<table>
<thead>
<tr>
<th>key</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_{id}$ $f_{e_{id}}</td>
<td></td>
</tr>
</tbody>
</table>

verified chip in package

signature valid? $\times$
unverified fingerprint $f_{e_{id}}$

fingerprint match? $\times$
maliciously mislabeled
Why CounterFoil?

- Fingerprints extracted from images of plastic IC packages
- Inexpensive cameras
- Verification in 150ms
- Practical scheme that works with existing IC distribution
- Does not require chain of custody
Experimental Setup

- OpenCV implementation of Computer Vision algorithms
- Crypto++ for processing
- Code/images posted at [1]
- Cameras from two vendors
- Different instances for enrollment and verification
- Two chip models
- Some packaged with the same mold instance

Naïve Fingerprinting

• Use difference in pixel intensity ($\Delta$) to quantify dissimilarity
• Intolerant of misalignment in position, rotation and scale
• Imperfect alignment even with fiducial marks
• Brute-forcing pixel alignment impractically slow
Fingerprinting using Computer Vision

• Identifying features of an image compared rather than each pixel
  • Features used for object detection & matching, motion tracking, stitching panoramic views

• Feature extraction with four prominent algorithms
  • SIFT, SURF, ORB, BRISK

• Keypoints: Position and Descriptor

• Matching keypoint is closest in feature space

• Geometry estimation using RANSAC

• Inlier only if match is nearest in feature space and fits geometry
CounterFoil vs Common Attacker

CounterFoil and Common Attacker

Histograms showing occurrence of inliers for Same Chip and Diff Mold
Sophisticated Attacker

• Can a package be forged to match enrolled fingerprint?
  • Consider an attacker that exactly duplicates manufacturer mold
  • Model this attacker using purchased packages produced by same mold instance
  • Revealed by mold marking
Sophisticated Attacker

- Can a package be forged to match enrolled fingerprint?
  - Consider an attacker that exactly duplicates manufacturer mold
  - Model this attacker using purchased packages produced by same mold instance
  - Revealed by mold marking

- Even perfect mold copy mostly unsuccessful
  - Filler particles
  - Mold buildup
  - Curing process
- In the worst of 4 datasets, need to allow 16% FP rate for 95% TP rate
Suitability for IC Distribution

Inexpensive
- Cheap cameras
- Marker costs are negligible
- 1MB storage per enrolled record

$390  $40

Efficient
- Authenticate in 150ms on Intel Xeon E5-2690
- Could run real-time on pick-and-place (700ms)

Wear-Tolerant
- 95% same mold

Efficiency Graph:
- Different markers and techniques (Read image, ROI using Aruco markers, Digital Signature, ORB, Matching, Others)
- Runtime in seconds
- Inliers count

Graph Data:
- Time in rock tumbler (minutes):
  - 0, 10, 20, 30, 40, 50, 60
- Window size (mm):
  - 1.8, 2, 2.2
## How Broadly Applicable?

<table>
<thead>
<tr>
<th>Surface Map</th>
<th>Example Image</th>
<th>Chip Name</th>
<th>Package</th>
<th>Same Chip Inliers $\mu (\sigma)$</th>
<th>Diff. Chip Inliers $\mu (\sigma)$</th>
<th>Area (mm²)</th>
<th>Example Image</th>
<th>Surface Map</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Surface Map 1" /></td>
<td><img src="image2.png" alt="Example Image 1" /></td>
<td>W25Q80EWUXIETR</td>
<td>23-SOT</td>
<td>38.6 (13.2)</td>
<td>3.0 (1.5)</td>
<td>0.454</td>
<td><img src="image3.png" alt="Example Image 3" /></td>
<td><img src="image4.png" alt="Surface Map 2" /></td>
</tr>
<tr>
<td><img src="image5.png" alt="Surface Map 2" /></td>
<td><img src="image6.png" alt="Example Image 4" /></td>
<td>TSV524IQ4T</td>
<td>16-QFN</td>
<td>42.9 (8.4)</td>
<td>4.1 (1.7)</td>
<td>0.315</td>
<td><img src="image7.png" alt="Example Image 5" /></td>
<td><img src="image8.png" alt="Surface Map 3" /></td>
</tr>
<tr>
<td><img src="image9.png" alt="Surface Map 3" /></td>
<td><img src="image10.png" alt="Example Image 6" /></td>
<td>MX25V4006EM11-13G</td>
<td>8-SOIC</td>
<td>58.8 (10.2)</td>
<td>3.8 (1.5)</td>
<td>0.454</td>
<td><img src="image11.png" alt="Example Image 7" /></td>
<td><img src="image12.png" alt="Surface Map 4" /></td>
</tr>
<tr>
<td><img src="image13.png" alt="Surface Map 4" /></td>
<td><img src="image14.png" alt="Example Image 8" /></td>
<td>24LC32A-UMS</td>
<td>8-MSOP</td>
<td>344.3 (44.8)</td>
<td>4.0 (1.3)</td>
<td>2</td>
<td><img src="image15.png" alt="Example Image 9" /></td>
<td><img src="image16.png" alt="Surface Map 5" /></td>
</tr>
<tr>
<td><img src="image17.png" alt="Surface Map 5" /></td>
<td><img src="image18.png" alt="Example Image 10" /></td>
<td>CY7C1335G-100AXC</td>
<td>100-TQFP</td>
<td>280.8 (40.1)</td>
<td>4.7 (1.0)</td>
<td>2</td>
<td><img src="image19.png" alt="Example Image 11" /></td>
<td><img src="image20.png" alt="Surface Map 6" /></td>
</tr>
<tr>
<td><img src="image21.png" alt="Surface Map 6" /></td>
<td><img src="image22.png" alt="Example Image 12" /></td>
<td>ADG419TQ</td>
<td>14-CDIP</td>
<td>358.4 (73.2)</td>
<td>3.9 (1.3)</td>
<td>2</td>
<td><img src="image23.png" alt="Example Image 13" /></td>
<td><img src="image24.png" alt="Surface Map 7" /></td>
</tr>
<tr>
<td><img src="image25.png" alt="Surface Map 7" /></td>
<td><img src="image26.png" alt="Example Image 14" /></td>
<td>ADP125ACPZ-R7</td>
<td>8-LFCSP</td>
<td>18.3 (7.2)</td>
<td>3.2 (1.5)</td>
<td>0.315</td>
<td><img src="image27.png" alt="Example Image 15" /></td>
<td><img src="image28.png" alt="Surface Map 8" /></td>
</tr>
<tr>
<td><img src="image29.png" alt="Surface Map 8" /></td>
<td><img src="image30.png" alt="Example Image 16" /></td>
<td>W25Q80EWUXIE TR</td>
<td>8-USON</td>
<td>12.3 (5.9)</td>
<td>2.1 (1.3)</td>
<td>0.201</td>
<td><img src="image31.png" alt="Example Image 17" /></td>
<td><img src="image32.png" alt="Surface Map 9" /></td>
</tr>
<tr>
<td><img src="image33.png" alt="Surface Map 9" /></td>
<td><img src="image34.png" alt="Example Image 18" /></td>
<td>FAN53540UCX</td>
<td>20-WLCSP</td>
<td>3.2 (2.9)</td>
<td>1.8 (1.4)</td>
<td>0.315</td>
<td><img src="image35.png" alt="Example Image 19" /></td>
<td><img src="image36.png" alt="Surface Map 10" /></td>
</tr>
<tr>
<td><img src="image37.png" alt="Surface Map 10" /></td>
<td><img src="image38.png" alt="Example Image 20" /></td>
<td>2N3440</td>
<td>TO-39</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>2</td>
<td><img src="image39.png" alt="Example Image 21" /></td>
<td><img src="image40.png" alt="Surface Map 11" /></td>
</tr>
</tbody>
</table>
Summary

• Counterfeit ICs threaten critical systems and profits
• CounterFoil verifies IC provenance by using fingerprints from surface texture of plastic packages
• Low cost, high performance and easy to integrate into IC distribution

Thank you!

https://github.com/danholcomb/supply-chain-security