Aegis: Mitigating Targeted Bit-flip Attacks against Deep Neural Networks

Jialai Wang¹, Ziyuan Zhang¹, Meiqi Wang¹, Han Qiu¹, Tianwei Zhang², Qi Li¹
Zongpeng Li¹, Wei Tao³, Chao Zhang¹
1. Tsinghua University
2. Nanyang Technological University
3. Ant Group
Outline

- Background
- Existing defense and their limitations
- Our solution Aegis
- Evaluations and results
Flip bits

- **Rowhammer attack**
  - First discovered in 2014
  - Rowhammer becomes easier with smaller chips
  - Nowadays, it can almost change any 1-bit you need

A typical **16GB DDR3 memory stick**

- **DIMM (Dual Inline Memory Module)**
- **Rank**
- **2GB**
- **256MB**

- **1 chip has 8 bank**
- **1024^3 = 32768**
- **32MB**

- **256MB**
- **2GB**

- **One electric capacity stores 1-bit data**

- **The basic element memory cell stores 1-bit**

- **Expect to flip**

- **Just operate its own memory repeatedly**

- **Attacker’s page**
- **Victim’s page**
- **Attacker’s page**

- **1 codela:**
  - mov (X), %eax
  - mov (Y), %ebx
  - clflush (X)
  - clflush (Y)
  - mfence
  - jmp codela

- **a. Induces errors**
Bit-flip attacks (BFAs) against dnns

- An example of a bit-flip attack

BFA: Modify models’ weight parameters through flipping some bits of weights
Bit-flip attacks (BFAs) against dnns

- An example of a bit-flip attack

BFA: Modify models’ weight parameters through flipping some bits of weights

- How many bits need to be flipped?

A “lightweight” DNN contains 100M+ bits, is it matter to flip a few of them?

Some bits are naturally very critical

Wrong prediction
Threat models

- Two steps for successful attacks
  - 1. Locate a few critical bits out of millions parameters.
  - 2. Flip the bits in real-world devices.
Threat models

- Two steps for successful attacks
  - 1. Locate a few critical bits out of millions parameters.
  - 2. Flip the bits in real-world devices.

  - **Attacker’s goal:** Flipping a few bits in memory to maliciously manipulate the DNN model
  - **Attacker’s knowledge:** Knowing the model’s physical address and the model’s weights
  - **Attacker’s capability:** Be able to plant his program in memory and start rowhammer attack
  - **Attacker’s constrains:** Can flip only a few bits with location constraints (attack preparation needs a long time)
Outline

- Background
- Existing defense and their limitations
- Our solution Aegis
- Evaluations and results
Existing defense and their limitations

- **Correction-based approach**
  - Correct the flipped bits
  - Memory enhancement (ECC memory)

- **Detection-based approach**
  - Protect the integrity for the model’s memory
  - Memory hash (HashTAG, ICCAD’21)

- **Model-level defense approach**
  - Enhance the DNN model to tolerant bit flips
  - Our baselines use binary neural network (BNN) to constrain the error
Correction-based approach

- Error correction code (ECC) enabled memory
  - ECC is not an absolutely secure solution against Rowhammer
  - ECC is still not used in DDR3 devices (embedded devices like Nvidia Nano)
  - ECC has special requirements on the whole computer architecture
  - ECC can only recover 1-bit error, detect 2-bit error, and that’s all

![Diagram showing 72-bit ECC-enabled memory control systems](image)

- 64-bit bus for data + 8-bit bus for ECC code

![Matrix showing 8-bit ECC code, Verified area, Error area, and Error code](image)

ECC can do nothing when 3-bit or more errors happen
Detection-based approach

- Detect any malicious modification in the memory
  - E.g. HashTAG, ICCAD’21
  - Hard to signature all parameters
  - Choose “sensitive” layers to protect
  - Using hash to verify during runtime

Protection analysis

- Pros:
  - Lightweight (no modification on the model)
  - No ACC loss if bit flip detected

- Cons:
  - Overhead (can be potentially optimized)
  - Extra trustworthy program (hash) on shared untrustworthy resources
Model-level defense approach

- Enhance the DNN model to tolerant bit flips
  - E.g. BNN, CVPR’20
  - Leverage binarization-aware training
- Pros:
  - Improve model tolerance to bit flips
- Cons:
  - Computation cost (retrain model from scratch)
  - Significant Accuracy Degradation

![Graph showing model accuracy with and without defense](image)

Table 2: Model ACC influence evaluation.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>BASE ACC (%)</th>
<th>BIN</th>
<th>RA-BNN</th>
<th>SDN</th>
<th>Regla</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>ResNet32 92.79</td>
<td>-2.26</td>
<td>-1.71</td>
<td>-1.27</td>
<td>-1.26</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VGG16   93.61</td>
<td>-1.26</td>
<td>-1.19</td>
<td>-1.72</td>
<td>-0.67</td>
<td></td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>ResNet32 66.13</td>
<td>-4.38</td>
<td>-2.47</td>
<td>-2.54</td>
<td>-1.96</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VGG16   72.85</td>
<td>-4.14</td>
<td>-2.08</td>
<td>-1.97</td>
<td>-1.90</td>
<td></td>
</tr>
<tr>
<td>STL-10</td>
<td>ResNet32 74.80</td>
<td>-3.09</td>
<td>-3.85</td>
<td>-2.80</td>
<td>-0.90</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VGG16   79.51</td>
<td>-1.41</td>
<td>-1.39</td>
<td>-1.35</td>
<td>-1.02</td>
<td></td>
</tr>
<tr>
<td>Tiny-ImageNet</td>
<td>ResNet32 54.58</td>
<td>-11.16</td>
<td>-6.31</td>
<td>-3.87</td>
<td>-1.92</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VGG16   60.51</td>
<td>-4.48</td>
<td>-4.07</td>
<td>-0.39</td>
<td>-0.28</td>
<td></td>
</tr>
</tbody>
</table>
Outline

- Background
- Existing defense and their limitations
- Our solution Aegis
- Evaluations and results
Defense requirements

- Our defense solution Aegis:
  - Non-intrusive: Easy to deploy on those off-the-shelf models to make it efficient
  - Platform-independent: Solutions are not restricted to some specific hardware/software platforms
  - Utility-preserving: Solutions have a negligible impact on the model’s inference (speed, ACC, etc.)

The point is to force attackers to flip more bits until impractical
Aegis framework

- Attackers locate the bits to flip \textbf{first by layer then parameters}
  - 1. TBT and TA-LBF consider to flip bits \textbf{only in the last layer}
  - 2. Pro-flip first \textbf{compute the critical layer then locates bits inside}

➢ \textbf{Break the inference pattern:}
  We adopt the multi-exit strategy (SDN) to allow samples exit earlier
➢ \textbf{More than 90\% samples} can exit accurately in shallow layers
Aegis framework

• Attackers locate the bits to flip **first by layer then parameters**
  - 1. **TBT** and **TA-LBF** consider to flip bits **only in the last layer**
  - 2. **Pro-flip** first **compute the critical layer then locates bits inside**

  ➢ **Step 1: break the inference pattern:**
  We adopt the multi-exit strategy (SDN) to allow samples (>90%) exit earlier
  ➢ Targeting the final layer/critical middle layer is pointless
  ➢ Attacker may change to locate adaptive critical layer (**where is the most exit?**)
  ➢ **Step 2: randomly mask internal exits** to make samples uniformly exit
  ➢ Attacker can only consider all layers as the critical layers
Aegis framework

- Attackers locate the bits to flip **first by layer then parameters**
  1. **TBT** and **TA-LBF** consider to flip bits **only in the last layer**
  2. **Pro-flip** first **compute the critical layer then locates bits inside**

**Step 1: break the inference pattern:**
We adopt the multi-exit strategy (SDN) to allow samples (>90%) exit earlier
- Targeting the final layer/critical middle layer is pointless
- Attacker may change to locate adaptive critical layer (**where is the most exit?**)

**Step 2: randomly mask internal exits**
- to make samples uniformly exit
- Attacker can only consider all layers as the critical layers

**Step 3: mimic potential bit-flip attack**
- for a robust training (only parameters in exits) to force the model fit attacks
Experiment setup

- Attacks, adaptive attacks

<table>
<thead>
<tr>
<th>Attack Method</th>
<th>Attack Type</th>
<th>Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBT (CVPR’20)</td>
<td>Backdoor injection</td>
<td>50+</td>
</tr>
<tr>
<td>ProFlip (ICCV’21)</td>
<td>Backdoor injection</td>
<td>15+</td>
</tr>
<tr>
<td>TA-LBF (ICLR’21)</td>
<td>Sample-wise mislead</td>
<td>10+</td>
</tr>
</tbody>
</table>

- We consider a white-box scenario: **both models and defenses are public**
  - Evaluate both **initial version attacks** and their **adaptive attacks**

- Datasets & model structures:
  - CIFAR-10, CIFAR-100, STL-10, and TinyImageNet-200
  - ResNet-32 and VGG-16

- Baselines
  - BASE, BIN, RA-BNN, SDN

- Metrics
  - **ASR**: attack success rate
Evaluation results (50-bits and 500-bits as limits)

Table 3: Evaluation results of ASR against TBT.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>BASE</th>
<th>BIN</th>
<th>RA-BNN</th>
<th>SDN</th>
<th>Aegis</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>ResNet32</td>
<td>70.7</td>
<td>94.8</td>
<td>74.5</td>
<td>16.3</td>
<td>19.9</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>71.1</td>
<td>90.4</td>
<td>82.9</td>
<td>42.8</td>
<td>36.0</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>ResNet32</td>
<td>95.8</td>
<td>99.8</td>
<td>23.5</td>
<td>20.5</td>
<td>10.8</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>65.9</td>
<td>58.4</td>
<td>47.4</td>
<td>53.8</td>
<td>10.6</td>
</tr>
<tr>
<td>STL-10</td>
<td>ResNet32</td>
<td>100.0</td>
<td>72.5</td>
<td>29.4</td>
<td>47.1</td>
<td>13.0</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>64.1</td>
<td>99.7</td>
<td>88.0</td>
<td>9.0</td>
<td>10.5</td>
</tr>
<tr>
<td>Tiny-ImageNet</td>
<td>ResNet32</td>
<td>100.0</td>
<td>63.3</td>
<td>31.4</td>
<td>65.8</td>
<td>27.9</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>69.7</td>
<td>72.3</td>
<td>40.2</td>
<td>48.5</td>
<td>10.1</td>
</tr>
</tbody>
</table>

Table 4: Evaluation results of ASR against TA-LBF.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>BASE</th>
<th>BIN</th>
<th>RA-BNN</th>
<th>SDN</th>
<th>Aegis</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>ResNet32</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>5.5</td>
<td>6.3</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>57.6</td>
<td>100.0</td>
<td>100.0</td>
<td>1.1</td>
<td>0.3</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>ResNet32</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>35.0</td>
<td>16.4</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>56.4</td>
<td>100.0</td>
<td>100.0</td>
<td>19.4</td>
<td>4.4</td>
</tr>
<tr>
<td>STL-10</td>
<td>ResNet32</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>47.7</td>
<td>9.6</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>81.4</td>
<td>99.7</td>
<td>98.7</td>
<td>0.3</td>
<td>2.0</td>
</tr>
<tr>
<td>Tiny-ImageNet</td>
<td>ResNet32</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
<td>71.1</td>
<td>26.1</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>51.8</td>
<td>98.1</td>
<td>90.7</td>
<td>27.2</td>
<td>17.5</td>
</tr>
</tbody>
</table>

Table 5: Evaluation results of ASR against ProFlip.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>BASE</th>
<th>BIN</th>
<th>RA-BNN</th>
<th>SDN</th>
<th>Aegis</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>ResNet32</td>
<td>96.9</td>
<td>99.4</td>
<td>90.6</td>
<td>47.3</td>
<td>19.8</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>88.2</td>
<td>78.6</td>
<td>84.6</td>
<td>70.5</td>
<td>28.9</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>ResNet32</td>
<td>89.8</td>
<td>100.0</td>
<td>82.9</td>
<td>58.3</td>
<td>19.2</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>80.0</td>
<td>80.4</td>
<td>76.5</td>
<td>64.9</td>
<td>20.3</td>
</tr>
<tr>
<td>STL-10</td>
<td>ResNet32</td>
<td>77.4</td>
<td>52.4</td>
<td>91.2</td>
<td>38.1</td>
<td>33.9</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>87.2</td>
<td>96.0</td>
<td>90.3</td>
<td>19.9</td>
<td>18.7</td>
</tr>
<tr>
<td>Tiny-ImageNet</td>
<td>ResNet32</td>
<td>99.1</td>
<td>82.5</td>
<td>80.4</td>
<td>75.0</td>
<td>20.1</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>88.2</td>
<td>44.1</td>
<td>39.2</td>
<td>26.5</td>
<td>15.6</td>
</tr>
</tbody>
</table>

Table 6: Evaluation results of ASR against adaptive TBT.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>BASE</th>
<th>BIN</th>
<th>Aegis</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>ResNet32</td>
<td>70.7</td>
<td>37.2</td>
<td>31.1</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>71.1</td>
<td>86.5</td>
<td>58.1</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>ResNet32</td>
<td>95.8</td>
<td>79.3</td>
<td>49.7</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>65.9</td>
<td>85.9</td>
<td>44.8</td>
</tr>
<tr>
<td>STL-10</td>
<td>ResNet32</td>
<td>100.0</td>
<td>35.0</td>
<td>31.8</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>64.1</td>
<td>27.0</td>
<td>27.0</td>
</tr>
<tr>
<td>Tiny-ImageNet</td>
<td>ResNet32</td>
<td>100.0</td>
<td>96.3</td>
<td>28.2</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>69.7</td>
<td>63.4</td>
<td>54.4</td>
</tr>
</tbody>
</table>

Table 7: Evaluation results of ASR against adaptive TA-LBF.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>BASE</th>
<th>BIN</th>
<th>Aegis</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>ResNet32</td>
<td>100.0</td>
<td>99.1</td>
<td>60.8</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>70.2</td>
<td>89.3</td>
<td>50.3</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>ResNet32</td>
<td>100.0</td>
<td>100.0</td>
<td>26.4</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>56.4</td>
<td>78.9</td>
<td>44.8</td>
</tr>
<tr>
<td>STL-10</td>
<td>ResNet32</td>
<td>100.0</td>
<td>100.0</td>
<td>10.2</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>81.4</td>
<td>89.9</td>
<td>26.8</td>
</tr>
<tr>
<td>Tiny-ImageNet</td>
<td>ResNet32</td>
<td>100.0</td>
<td>100.0</td>
<td>16.2</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>51.8</td>
<td>90.4</td>
<td>15.0</td>
</tr>
</tbody>
</table>

Table 8: Evaluation results of ASR against adaptive ProFlip.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>BASE</th>
<th>BIN</th>
<th>Aegis</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>ResNet32</td>
<td>96.9</td>
<td>74.2</td>
<td>38.4</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>88.2</td>
<td>79.1</td>
<td>43.6</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>ResNet32</td>
<td>89.8</td>
<td>69.3</td>
<td>25.8</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>80.0</td>
<td>92.4</td>
<td>35.7</td>
</tr>
<tr>
<td>STL-10</td>
<td>ResNet32</td>
<td>77.4</td>
<td>57.8</td>
<td>41.3</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>87.2</td>
<td>87.5</td>
<td>34.5</td>
</tr>
<tr>
<td>Tiny-ImageNet</td>
<td>ResNet32</td>
<td>99.1</td>
<td>64.4</td>
<td>36.1</td>
</tr>
<tr>
<td></td>
<td>VGG16</td>
<td>88.2</td>
<td>73.1</td>
<td>40.8</td>
</tr>
</tbody>
</table>
Discussion and Conclusion

- Additional costs brought by protection
  - **Model size:** additional 10-20% parameters
  - **ACC drop:** 0.3-1.9% accuracy drop
  - **Inference speed:** accelerate 45-60%

- Conclusion of Aegis:
  - A non-intrusive, platform-independent, utility-preserving defense to mitigate bit-flip attacks
  - The point is to make the attack impractical to deploy on real-world devices
Questions