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

Jialai Wang[1], Ziyuan Zhang[1], Meiqi Wang[1], Han Qiu[1], Tianwei Zhang[2], Qi Li[1]
Zongpeng Li[1], Wei Tao[3], Chao Zhang[1]

1. Tsinghua University
2. Nanyang Technological University
3. Ant Group

Outline

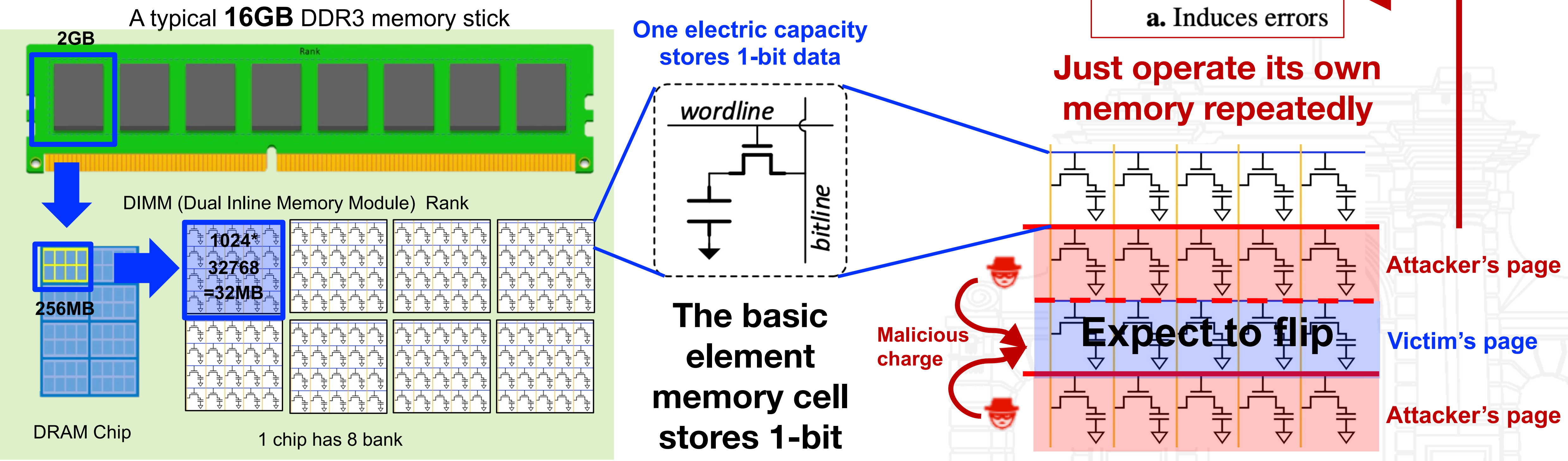
- Background
- Existing defense and their limitations
- Our solution Aegis
- Evaluations and results

Flip bits

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

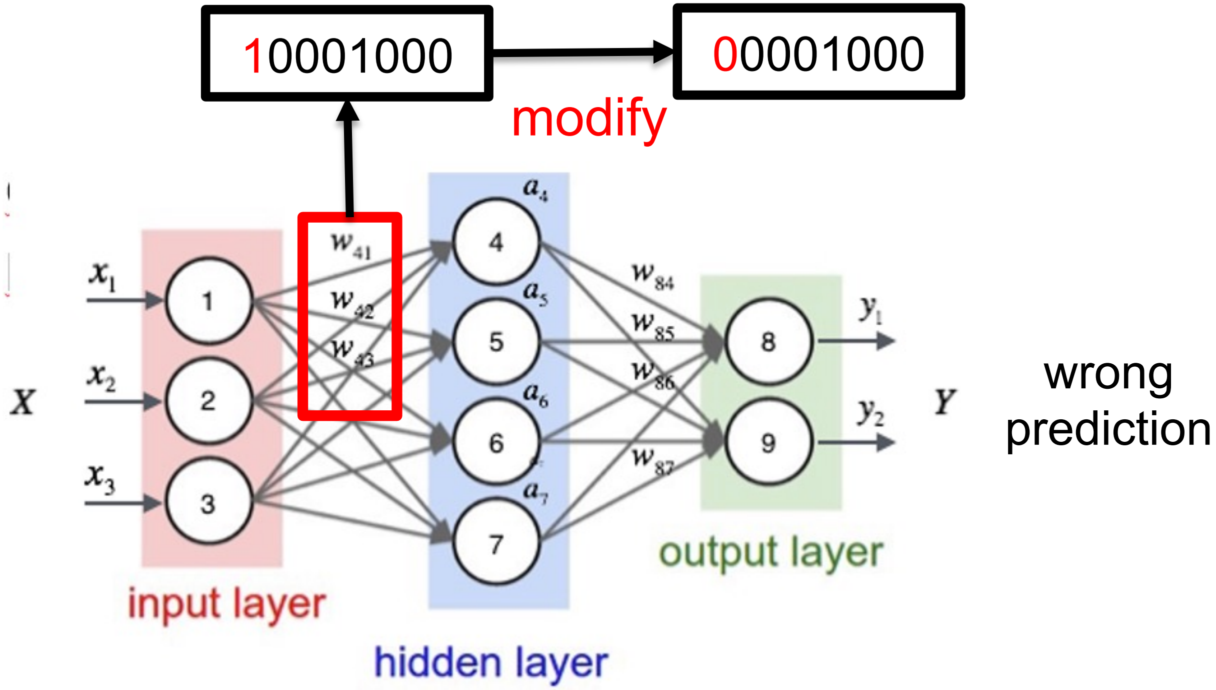
```
code1a:
  mov (X), %eax
  mov (Y), %ebx
  clflush (X)
  clflush (Y)
  mfence
  jmp code1a
```

a. Induces errors



Bit-flip attacks (BFAs) against dnns

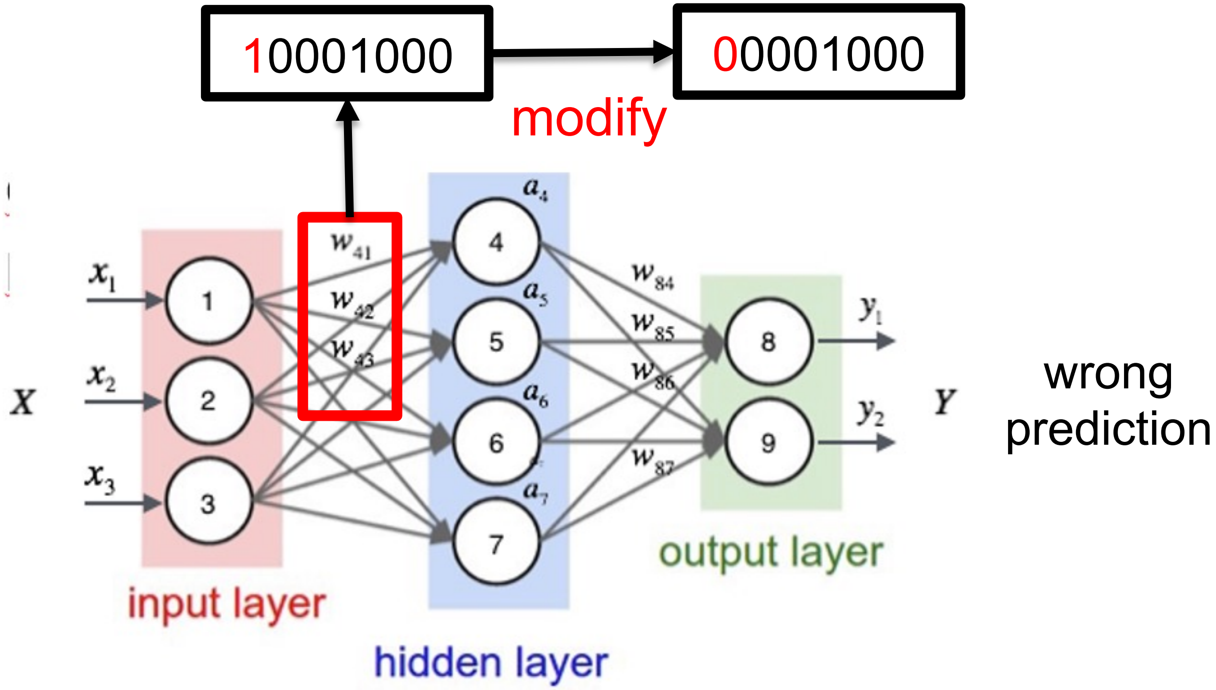
- An example of a bit-flip attack



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

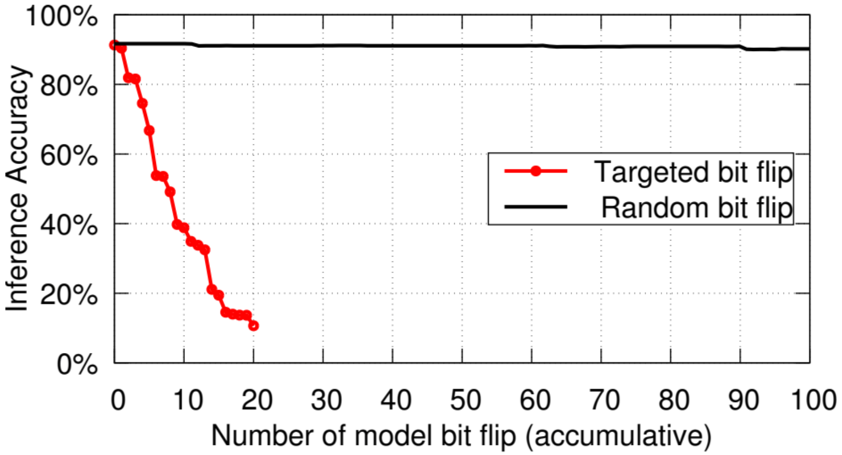
Bit-flip attacks (BFAs) against dnns

- An example of a bit-flip attack
- How many bits need to be flipped?



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

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



Some bits are naturally very critical

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

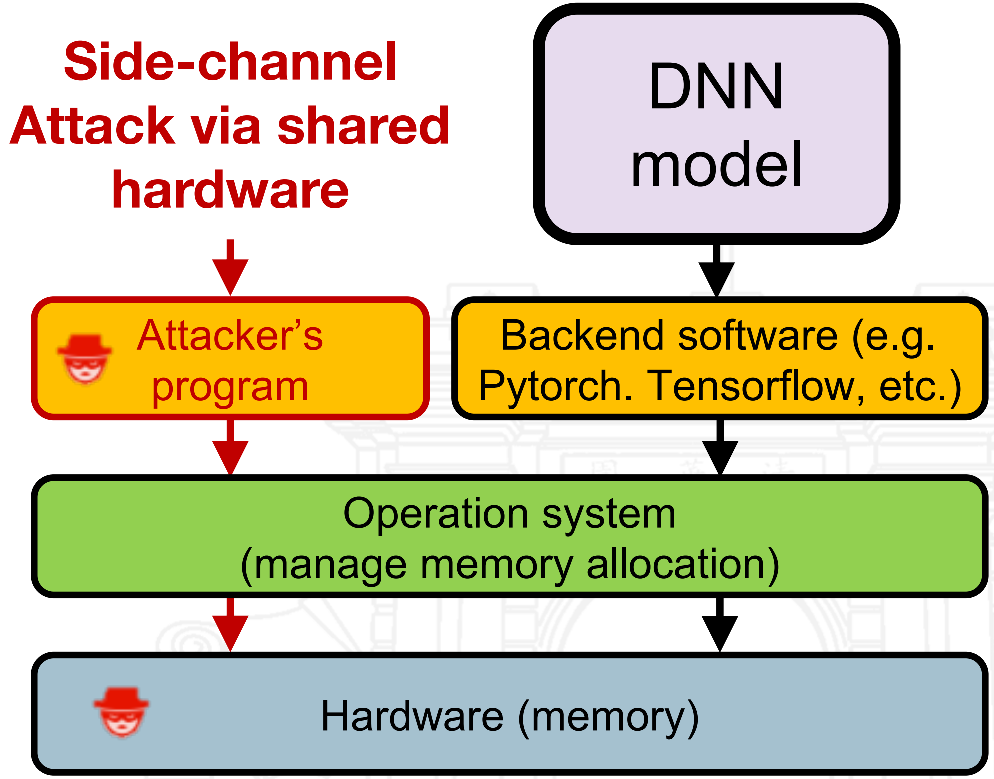
- Two steps for successful attacks
  - 1. Locate a few critical bits out of millions parameters.
  - 2. Flip the bits in real-world devices.

➢ **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)



Malicious modify target DNN's parameters inside the memory

Outline

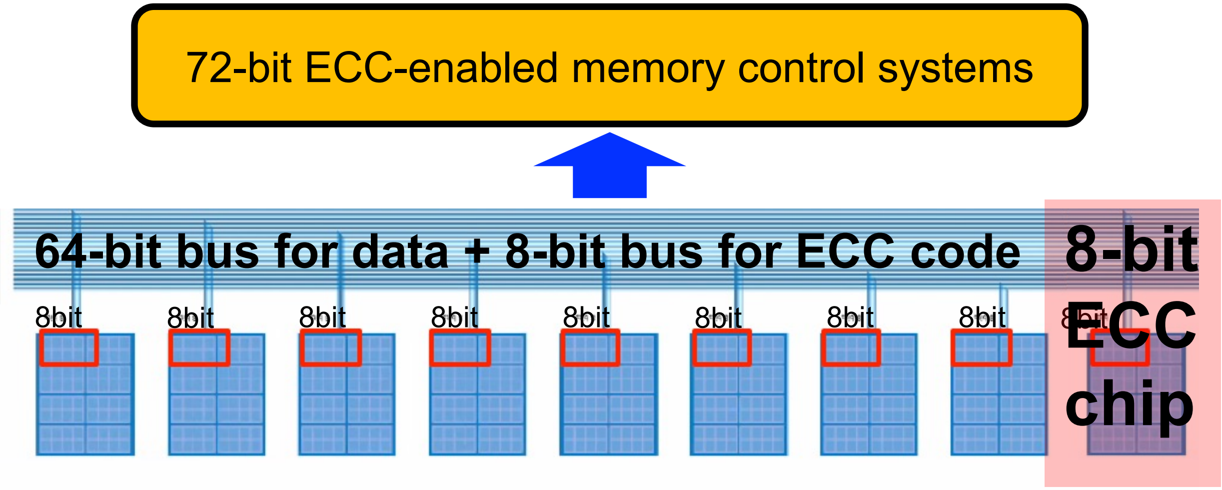
- Background
- Existing defense and their limitations
- Our solution Aegis
- Evaluations and results

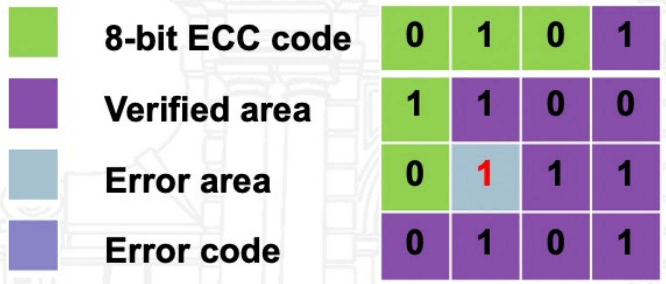
Existing defense and their limitations

- 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

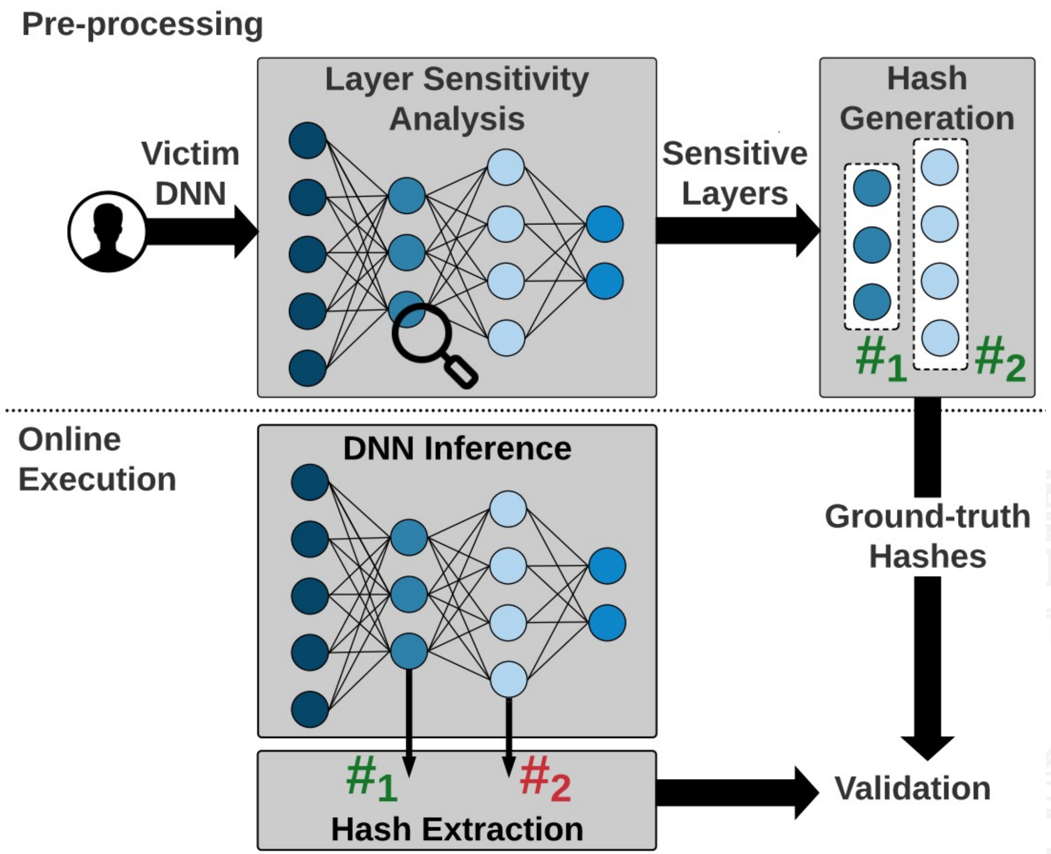




ECC can do nothing when 3-bit or more errors happen

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

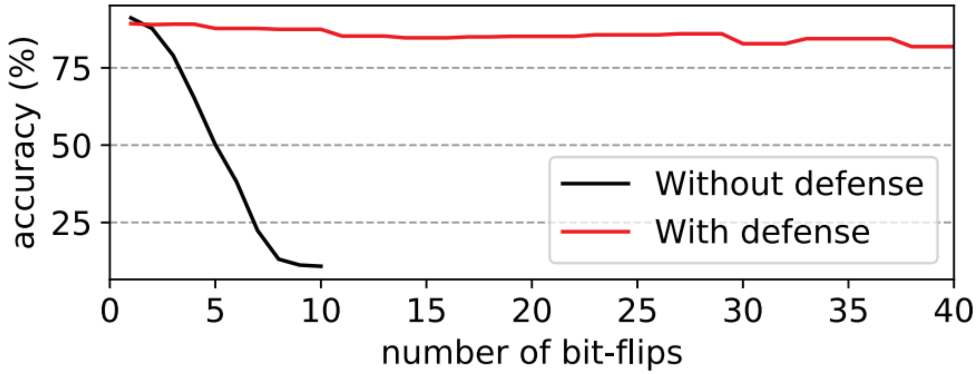


Table 2: Model ACC influence evaluation.

Dataset	Model	BASE ACC (%)	Δ ACC (%) BIN	RA-BNN	SDN	Aegis
CIFAR-10	ResNet32	92.79	-2.26	-1.71	-1.27	**-1.26**
	VGG16	93.61	-1.26	-1.19	-1.72	**-0.67**
CIFAR-100	ResNet32	66.13	-4.38	-2.47	-2.54	**-1.96**
	VGG16	72.85	-4.14	-2.08	-1.97	**-1.90**
STL-10	ResNet32	74.80	-4.09	-3.85	-2.80	**-0.90**
	VGG16	79.51	-1.41	-1.39	-1.35	**-1.02**
Tiny-ImageNet	ResNet32	54.58	-11.16	-6.31	-3.87	**-1.92**
	VGG16	60.51	-4.18	-4.07	-0.39	**-0.28**

Outline

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

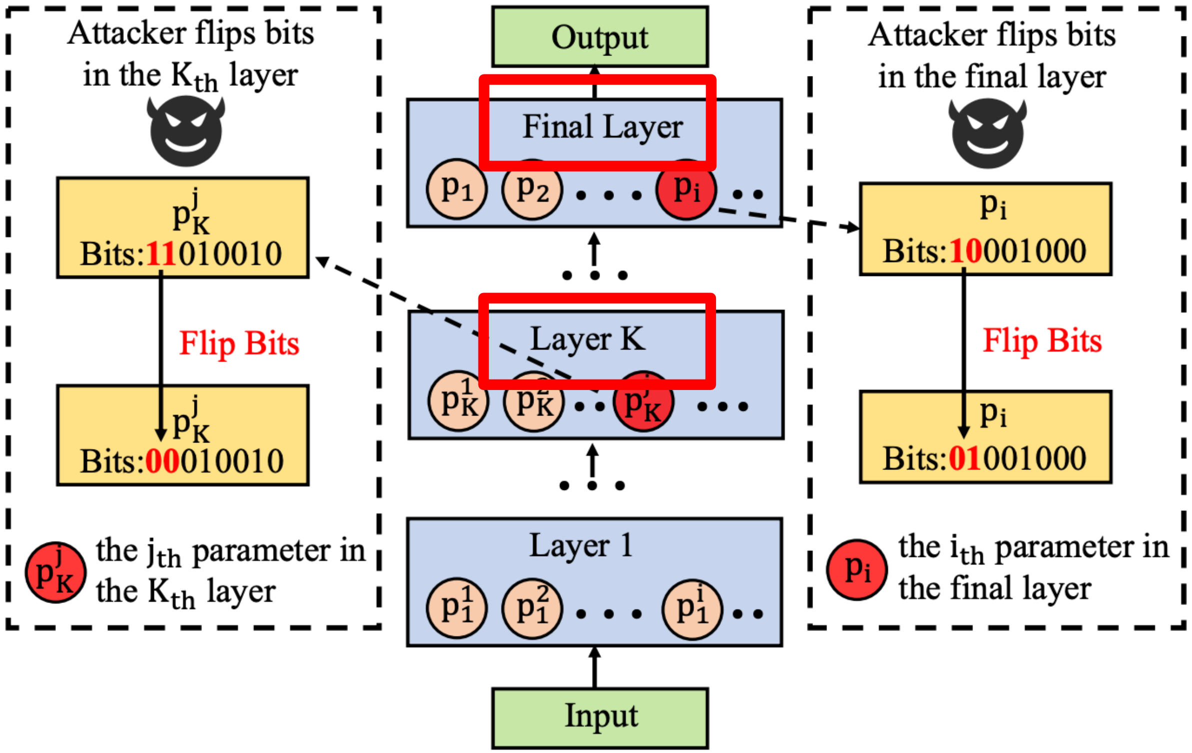
Defense requirements

- 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.)

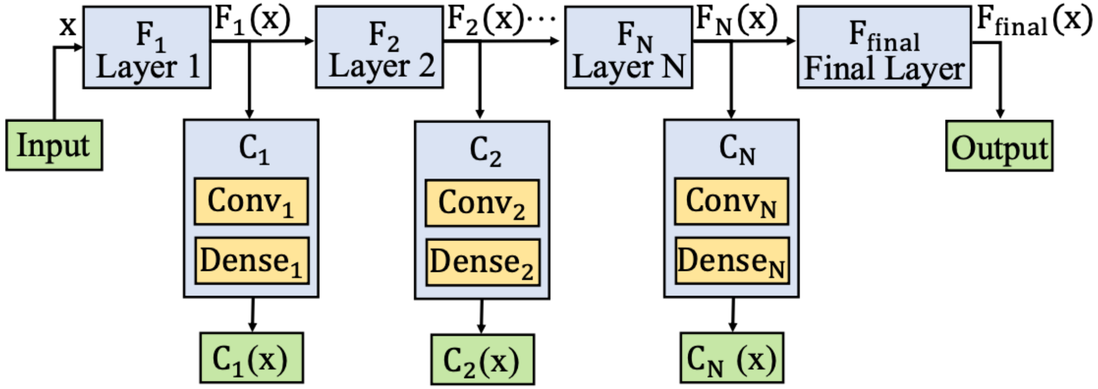
The point is to force attackers to flip more bits until impractical

Aegis framework

- Attackers locate the bits to flip **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**

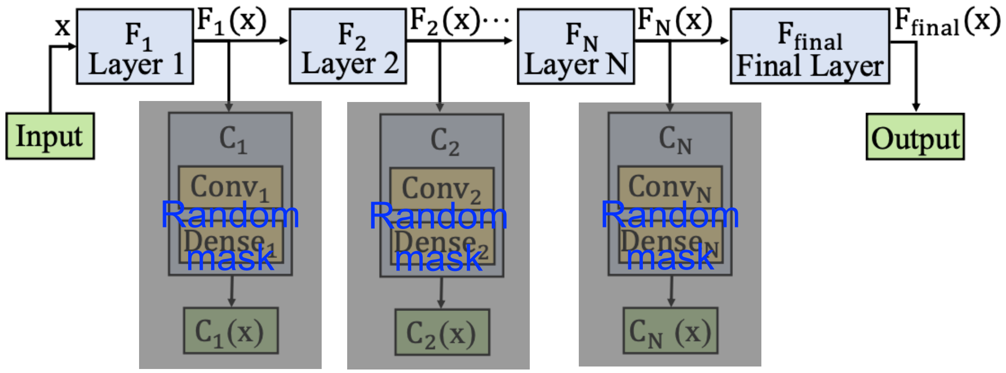


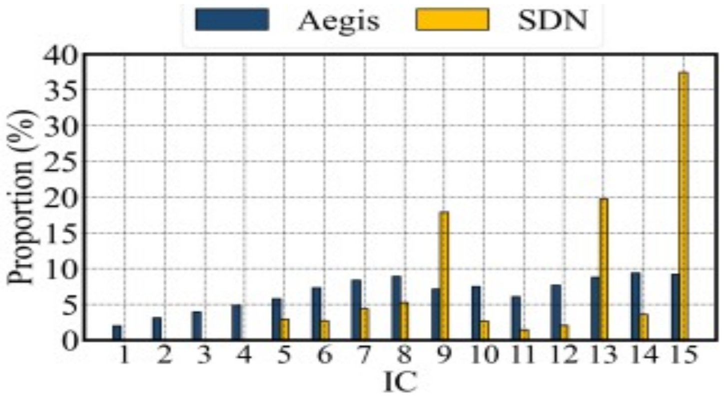
- **Break the inference pattern:**
  We adopt the multi-exit strategy (SDN) to allow samples exit earlier
- **More than 90% samples** can exit accurately in shallow layers



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**

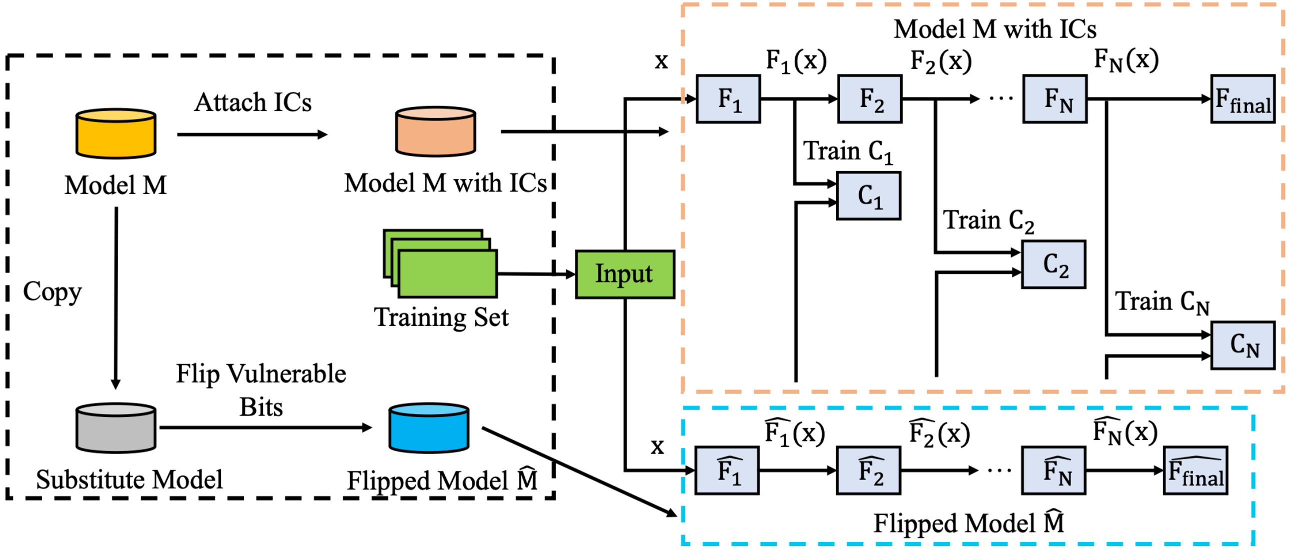




- ➢ **Step 1: break the inference pattern:** We adopt the multi-exit strategy (SDN) to allow samples (>90%) exit earlier
- ➢ Targeting the final layer/critical middle layer is pointless
- ➢ Attacker may change to locate adaptive critical layer (**where is the most exit?**)
- ➢ **Step 2**: randomly mask 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

TBT (CVPR'20)	Backdoor injection	50+
ProFlip (ICCV'21)	Backdoor injection	15+
TA-LBF (ICLR'21)	Sample-wise mislead	10+

  - 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.

Dataset	Model	ASR (%)				
		BASE	BIN	RA-BNN	SDN	Aegis
CIFAR-10	ResNet32	70.7	94.8	74.5	**16.3**	19.9
	VGG16	71.1	90.4	82.9	42.6	**36.0**
CIFAR-100	ResNet32	95.8	99.8	25.5	20.5	**10.8**
	VGG16	65.9	58.4	47.4	53.8	**10.6**
STL-10	ResNet32	100.0	72.5	29.4	47.1	**13.0**
	VGG16	64.1	99.7	88.0	**9.0**	10.5
Tiny-ImageNet	ResNet32	100.0	63.3	31.4	65.8	**27.9**
	VGG16	69.7	72.3	40.2	48.9	**10.1**

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

Dataset	Model	ASR (%)				
		BASE	BIN	RA-BNN	SDN	Aegis
CIFAR-10	ResNet32	100.0	100.0	100.0	**3.5**	6.3
	VGG16	57.6	100.0	100.0	1.1	**0.3**
CIFAR-100	ResNet32	100.0	100.0	100.0	38.0	**16.4**
	VGG16	56.4	100.0	100.0	19.4	**4.4**
STL-10	ResNet32	100.0	100.0	100.0	47.7	**9.6**
	VGG16	81.4	99.7	98.7	**0.3**	2.0
Tiny-ImageNet	ResNet32	100.0	100.0	100.0	71.1	**20.1**
	VGG16	51.8	98.1	90.7	27.2	**17.3**

Table 5: Evaluation results of ASR against ProFlip.

Dataset	Model	ASR (%)				
		BASE	BIN	RA-BNN	SDN	Aegis
CIFAR-10	ResNet32	96.9	99.4	90.6	47.3	**19.8**
	VGG16	88.2	78.6	84.6	70.5	**28.9**
CIFAR-100	ResNet32	89.8	100.0	82.9	58.3	**19.2**
	VGG16	80.0	80.4	76.5	64.9	**20.3**
STL-10	ResNet32	77.4	52.4	91.2	58.1	**33.9**
	VGG16	87.2	96.0	90.3	19.9	**18.7**
Tiny-ImageNet	ResNet32	99.1	82.5	80.4	75.0	**20.1**
	VGG16	88.2	44.1	39.2	26.8	**15.6**

Table 6: Evaluation results of ASR against adaptive TBT.

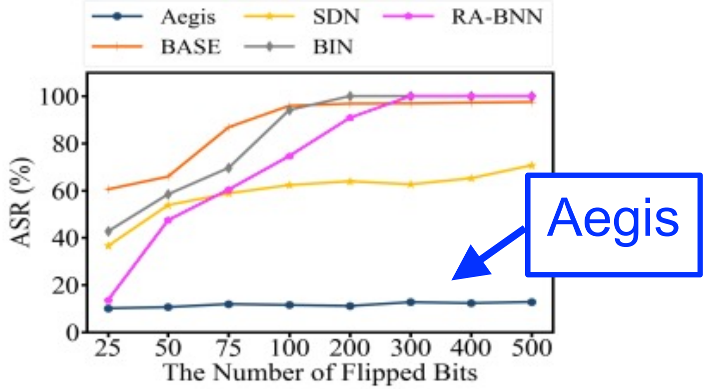
Dataset	Model	ASR (%)		
		BASE	SDN	Aegis
CIFAR-10	ResNet32	70.7	37.2	**31.1**
	VGG16	71.1	86.5	**58.1**
CIFAR-100	ResNet32	95.8	79.3	**49.7**
	VGG16	65.9	85.9	**44.8**
STL-10	ResNet32	100.0	35.0	**31.8**
	VGG16	64.1	93.0	**27.0**
Tiny-ImageNet	ResNet32	100.0	96.3	**28.2**
	VGG16	69.7	63.4	**54.4**

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

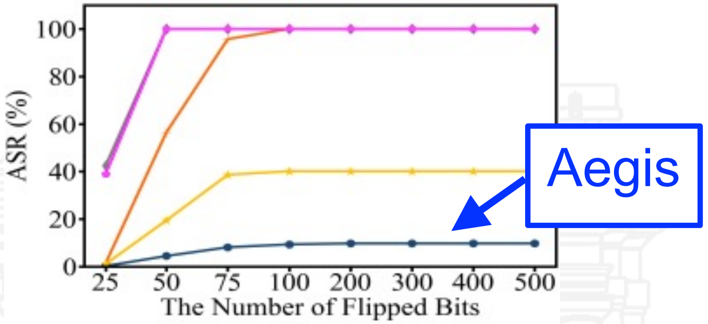
Dataset	Model	ASR (%)		
		BASE	SDN	Aegis
CIFAR-10	ResNet32	100.0	99.1	**60.8**
	VGG16	70.2	89.3	**50.3**
CIFAR-100	ResNet32	100.0	100.0	**26.4**
	VGG16	56.4	78.2	**44.8**
STL-10	ResNet32	100.0	100.0	**10.2**
	VGG16	81.4	89.9	**26.8**
Tiny-ImageNet	ResNet32	100.0	100.0	**16.2**
	VGG16	51.8	90.4	**15.0**

Table 8: Evaluation results of ASR against adaptive ProFlip.

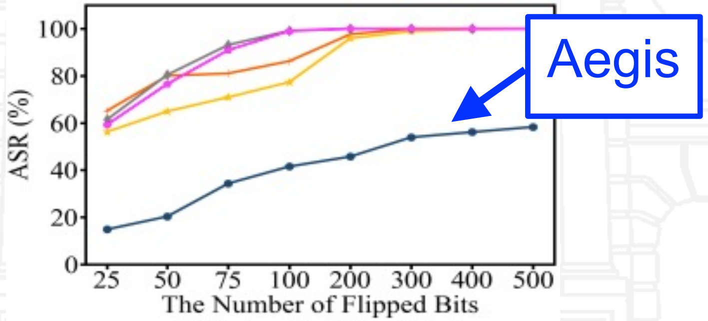
Dataset	Model	ASR (%)		
		BASE	SDN	Aegis
CIFAR-10	ResNet32	96.9	74.2	**38.4**
	VGG16	88.2	79.1	**43.6**
CIFAR-100	ResNet32	89.8	69.1	**25.8**
	VGG16	80.0	92.4	**33.7**
STL-10	ResNet32	77.4	57.8	**41.3**
	VGG16	87.2	87.5	**34.5**
Tiny-ImageNet	ResNet32	99.1	64.4	**36.1**
	VGG16	88.2	73.1	**40.8**



a) Under TBT Attack



b) Under TA-LBF Attack



c) Under ProFlip Attack

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