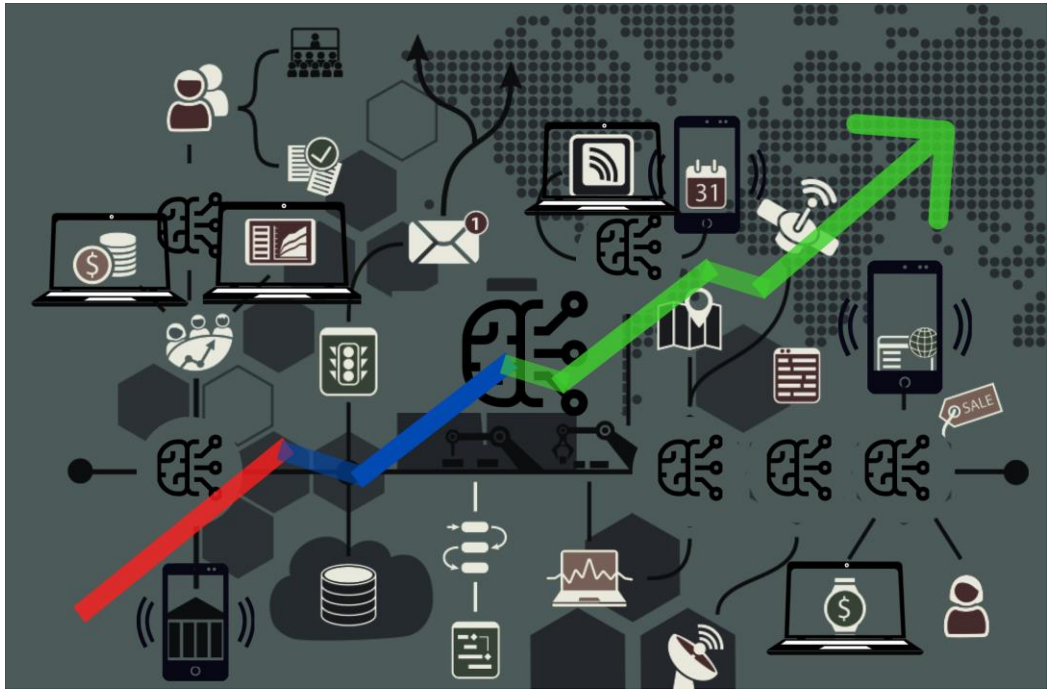
SoK: All You Need to Know About On-Device ML Model Extraction- The Gap Between Research and Practice

Tushar Nayan[1], Qiming Guo[1], Mohammed Al Duniawi[1], Marcus Botacin[2], Selcuk Uluagac[1], Ruimin Sun[1]

[1]Florida International University, [2]Texas A&M University

The rise of on-device ML

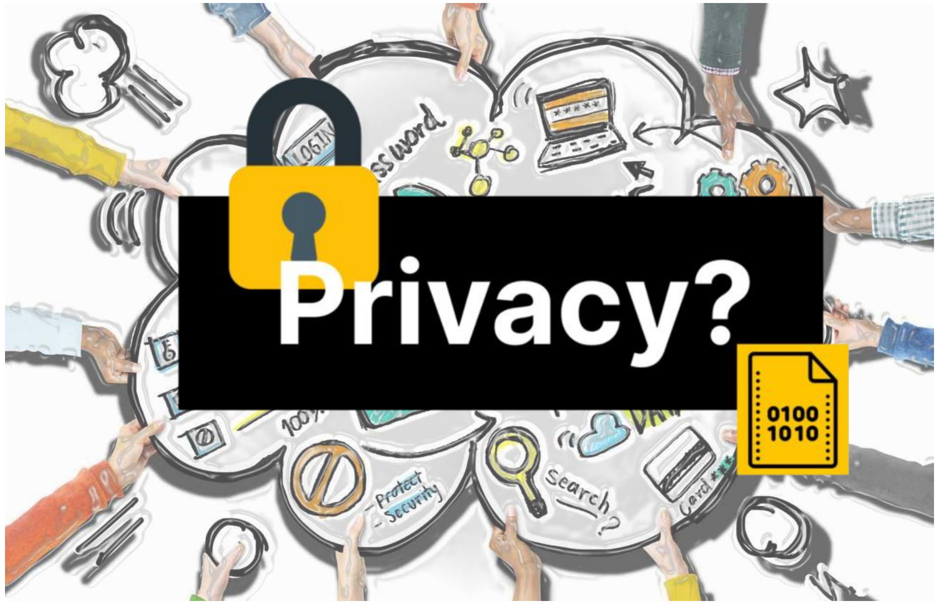
- There is a rising trend of on-device ML



- On-device ML offers many benefits for IoT devices.
  - Stronger user privacy
  - Real-time analysis
  - Better user experiences, optimized performance, and intelligent edge decision-making

ML model extraction attacks

- **On-Device ML Brings Security Challenges:** *Model theft and extraction attacks risks.*
  - Financial & Security Implications
  - Privacy Concerns



Defending these attacks

Defender's perspective
Advanced Encryption Standard (AES), Homomorphic Encryption (HE), Trusted Execution Environment (TEE), Data transformation, and various algorithm-based protection techniques.

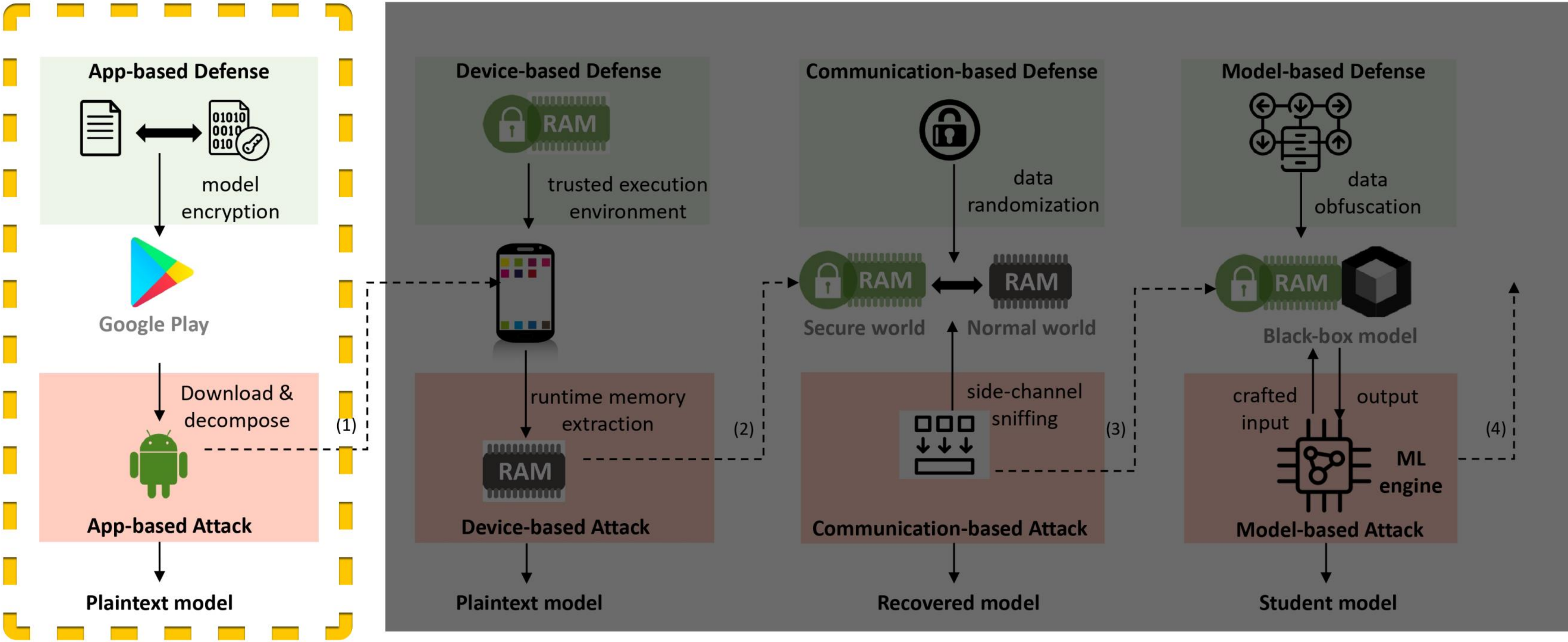
- Despite advances in model extraction security, efforts remain *fragmented* and *ad-hoc*.
- This gap impedes the development of *comprehensive security techniques*.

Our work aims to -

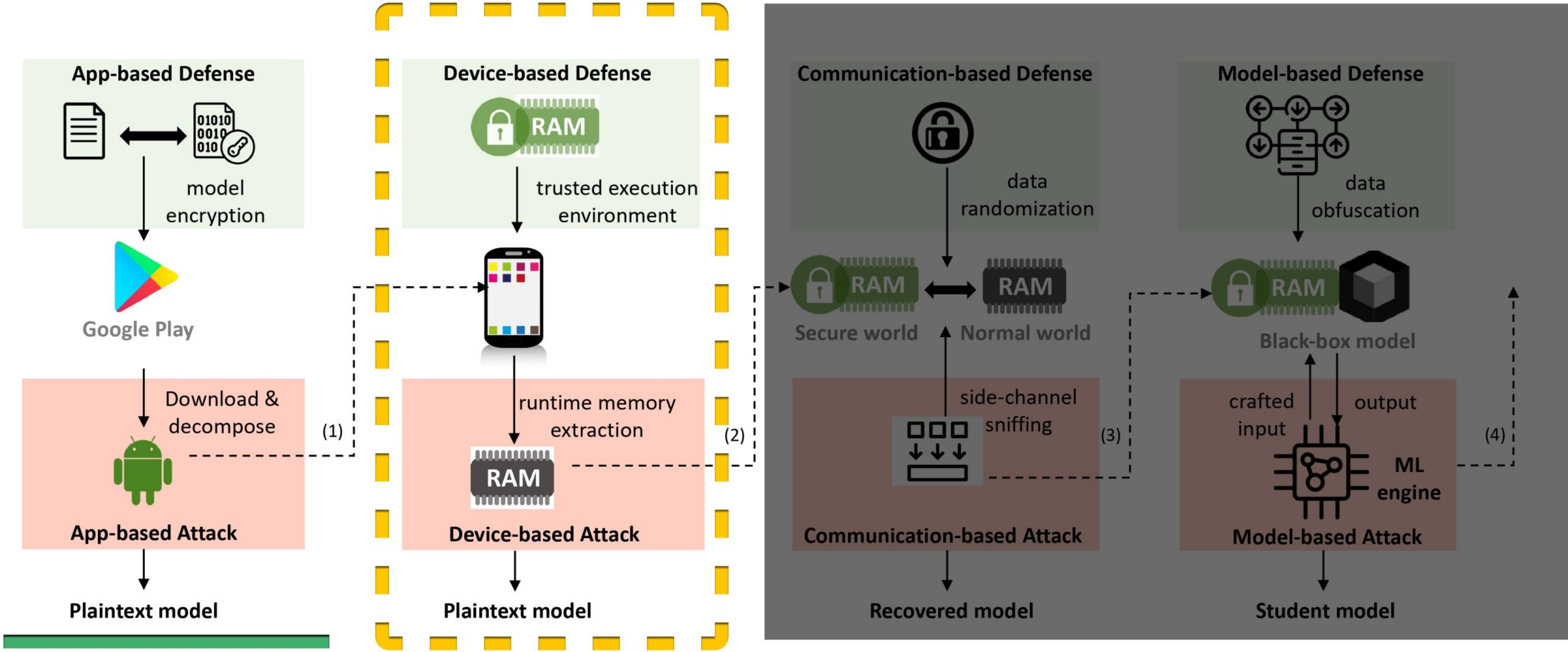
Systematize existing studies in model extraction attacks and defenses based on different threat levels.

Model Extraction: Security Design

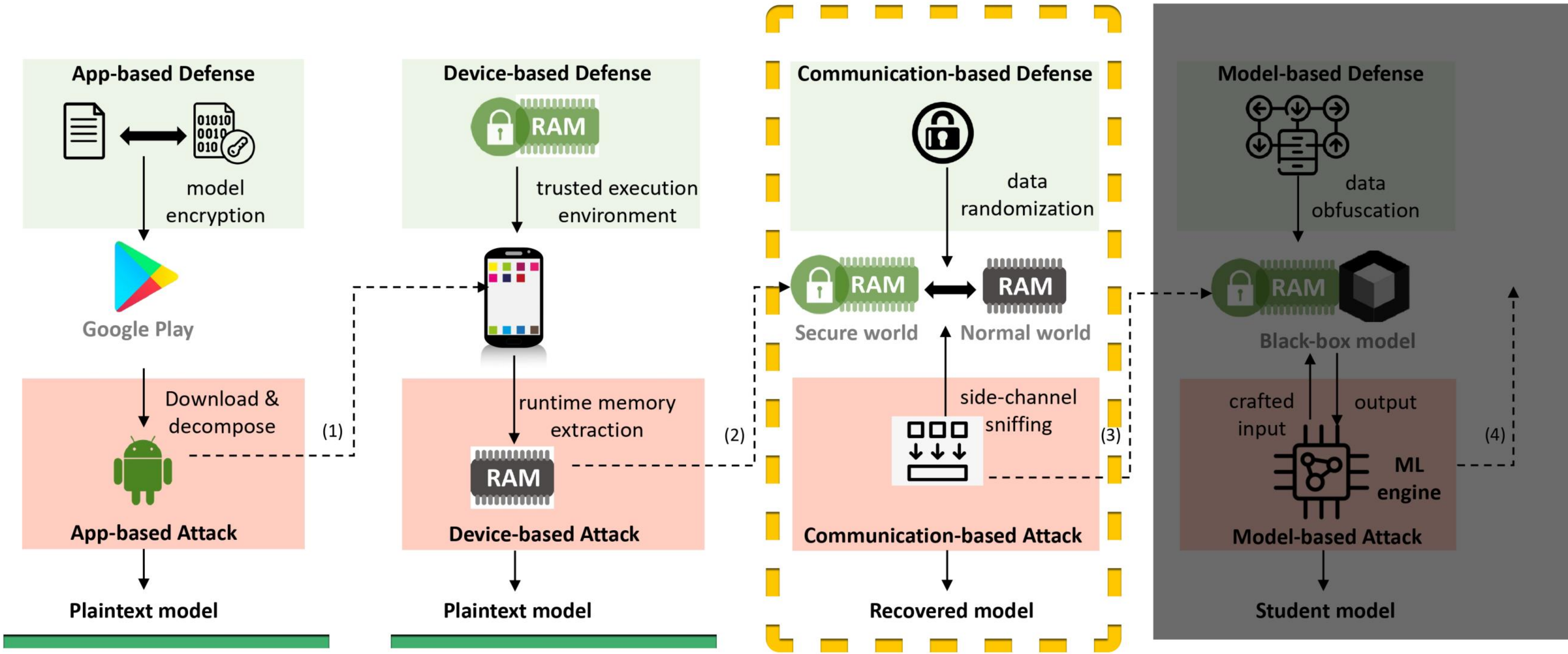
App-based Attack & Defense



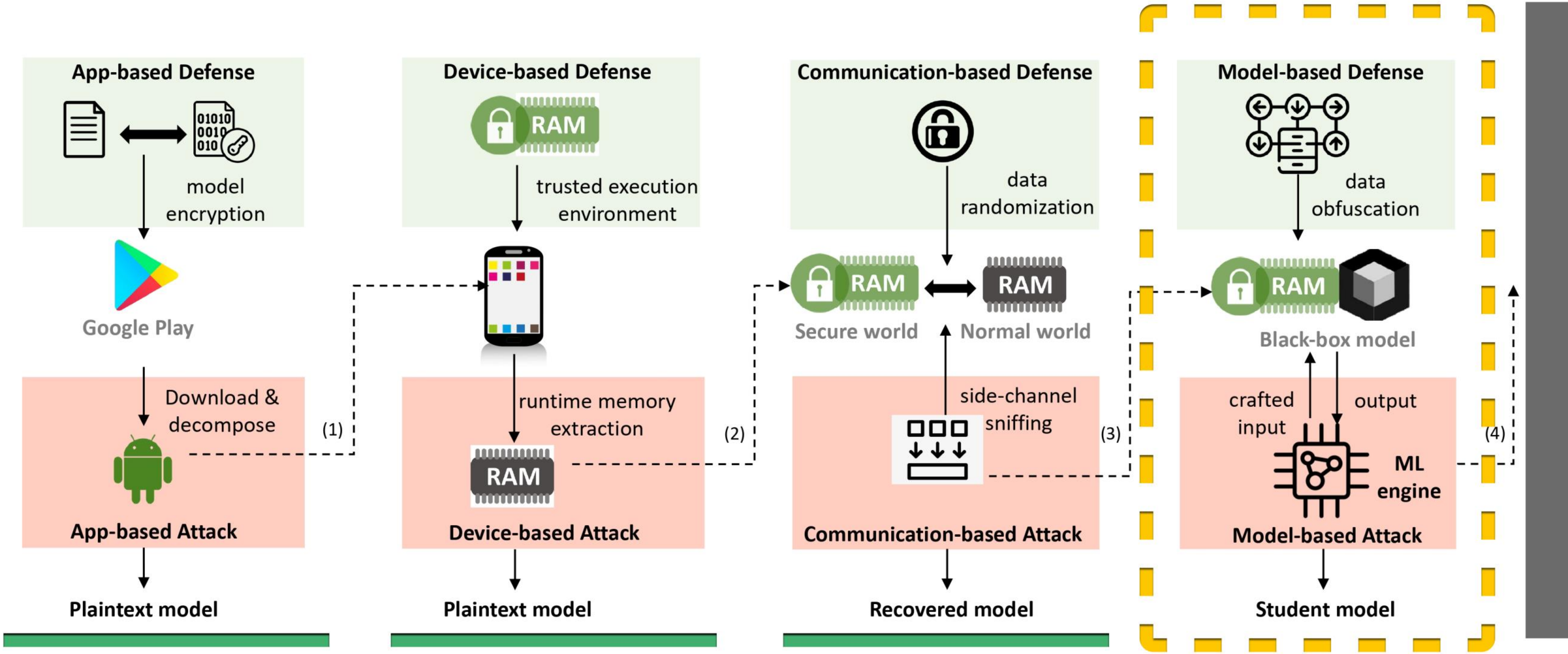
Device-based Attack & Defense



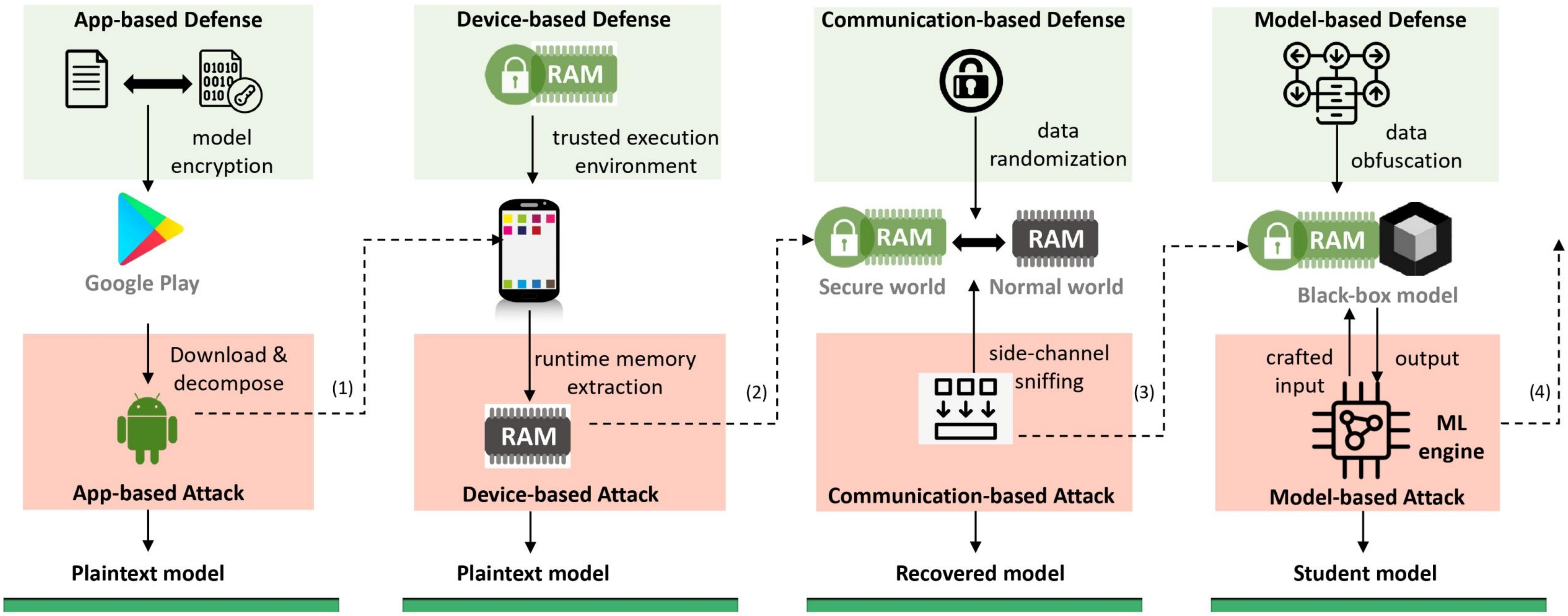
Communication-based Attack & Defense



Model-based Attack & Defense



Threat Models Category



Survey of Existing Literature on Model Extraction Attacks & Defenses

Existing Model Extraction Attacks

Title	Category	Target	Method	Open-source	Reproduced	ML Framework
First Look	App	Whole	Decompile	Yes	Yes	Multiple
SmartAppAttack	App	Whole	Decompile	Yes	Yes	Multiple
Mind'21	App, Device	Whole	Decompile, mem. searching	Yes	Yes	Multiple
Understanding'22	App, Device	Whole	Decompile, API hooking	No	N/A	Multiple
DeepRecon	Comm.	Arch.	Cache (Fl.&Re.)	Yes	No	TensorFlow
CSI NN	Comm.	Arch.,Layer,Weight	timing and electromagnetic	No	N/A	General
Cache Telepathy	Comm.	Arch.	Cache (Pr.&Pr.,Fl.&Re.)	No	N/A	General
Open DNN box	Comm.	Arch.,Weight	Power Feature	No	N/A	General
Reverse CNN	Comm.	Arch.,Weight	Memory Access	No	N/A	General
GANRED	Comm.	Arch.	Cache Attack	No	N/A	General
DeepEM	Comm.	Arch.,Layer,Weight	EM Attack	No	N/A	General
StealingNNTiming	Comm.	Arch.,Weight	Timing Attack	No	N/A	General
HuffDuff	Comm.	Arch.,Weight	Timing Attack	No	N/A	General
Hermes Attack	Comm.	Whole Model	PCIe traffic	No	N/A	TensorFlow
Leaky DNN	Comm.	Arch.	GPU Context-Switching	No	N/A	TensorFlow
ScanChainSteal	Comm.	Model Weight	Scan-chain Infrastructure	No	N/A	TensorFlow
DeepSniffer	Comm.	Model Arch.	Memory, Bus snooping	Yes	Yes	PyTorch
DeepSteal	Comm.	Functionality	Memory Access (rowhammer)	Yes	Yes	PyTorch
ML-Doctor	Model	Model Weight	Inference Attacks	Yes	Yes	Pytorch
Hyperparameters	Model	Hyperparameters	Hyperparameter Stealing	No	N/A	General
Reverse BlackBox	Model	Arch., Optm.,etc	Adversarial Example	No	N/A	Pytorch
Activethief	Model	Model Weight	Active Learning	Yes	No	TensorFlow
ML-Stealer	Model	Functionality	Prediction Stealing	No	N/A	General
KnockoffNets	Model	Functionality	Functionality stealing	Yes	Yes	Pytorch
SimulatorAttack	Model	Functionality	black-box attack	Yes	Yes	TensorFlow,Pytorch

Note that Pr.&Pr. means Prime+Probe, and Fl.&Re. means Flush+Reload.

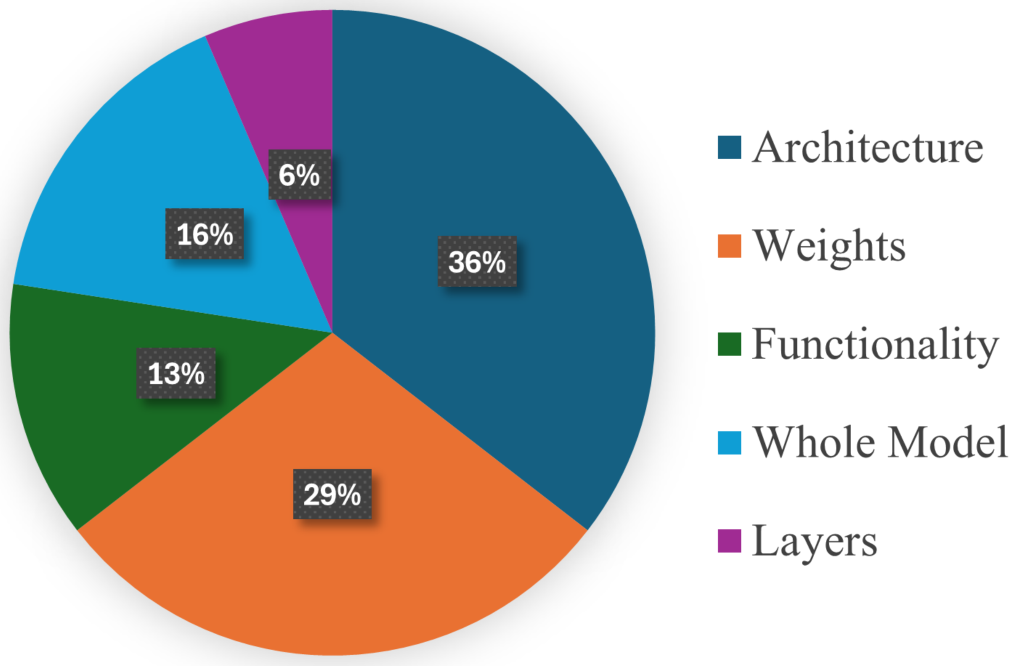
Existing Model Extraction Defense

Title	Category	Target	Method	Open-source	Reproduced	ML Framework
TFSecured*	App	Whole	Encryt.	Yes	Yes	TensorFlow
MindSpore*	App,Model	Whole	Encryt.,Obfu.,DP	Yes	Yes	MindSpore
Knox*	App	Whole	Encryt.	Yes	Yes	Multiple
MACE*	App	Whole	Obfu.,Convert	Yes	Yes	TensorFlow,Caffe,ONNX
m2cgen*	App	Whole	Convert	Yes	Yes	Multiple
MindDB*	App	Whole	Convert	Yes	Yes	Multiple
MMGuard	App	Whole	Encrypt, node insertion	Yes	Yes	TensorFlow
MyTEE	Device	Whole	TEE	Yes	No	General
SANCTUARY	Device	Whole	TEE	Yes	Yes	General
OMG	Device	Whole	TEE	No	N/A	TFLite
DarkneTZ	Device	layer,output	TEE	Yes	Yes	General
Graviton	Device	Whole	TEE	No	N/A	Caffe
ObfuNAS	Comm.	Arch.	Obfu.	Yes	Yes	PyTorch
ShadowNet	Device,Comm.	layer,weight	Transform	Yes	Yes	Darknet, TFLite
Slalom	Comm.	layer,weight	Transform	Yes	No	TensorFlow
E2DM	Comm.	Whole	HE	No	N/A	TensorFlow
NPUFort	Comm.	Weight	Secure Hardware	No	N/A	General
NeurObfuscator	Comm.	Arch.	Obfu.	Yes	Yes	PyTorch
Mitigating'19	Comm.	Functionality	Oblivious shuffle, ASLR, etc.	No	N/A	General
NNReArch	Comm.	Arch.	EM Obfu.	No	N/A	General
Misinformation	Model	Weight	Adaptive Misinformation	Yes	Yes	PyTorch
PredictionPoison	Model	Weight	Perturbation	Yes	Yes	PyTorch
PRADA	Model	Weight	Extraction Detection	Yes	Yes	PyTorch
SteerAdversary	Model	Weight	Gradient redirection	Yes	Yes	PyTorch
LDA-DP	Model	Weight	DP	No	N/A	General

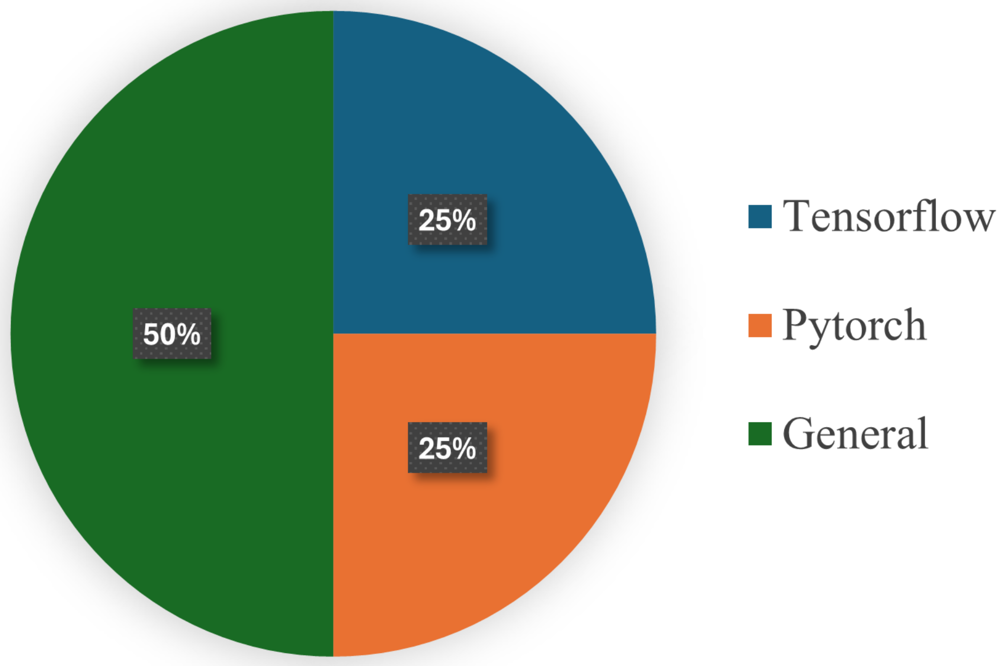
*Note: title with * means the project is maintained by industry community*

Existing Model Extraction Attacks

Aspect	Examples
Common Attack Targets	Architecture, Weights, Functionality, Whole Model, Layers
Targeted ML Frameworks	General, TensorFlow, PyTorch
Common Attack Methods	Decompile, Memory Access, Cache Attacks, Timing Attacks, Black-Box Attacks



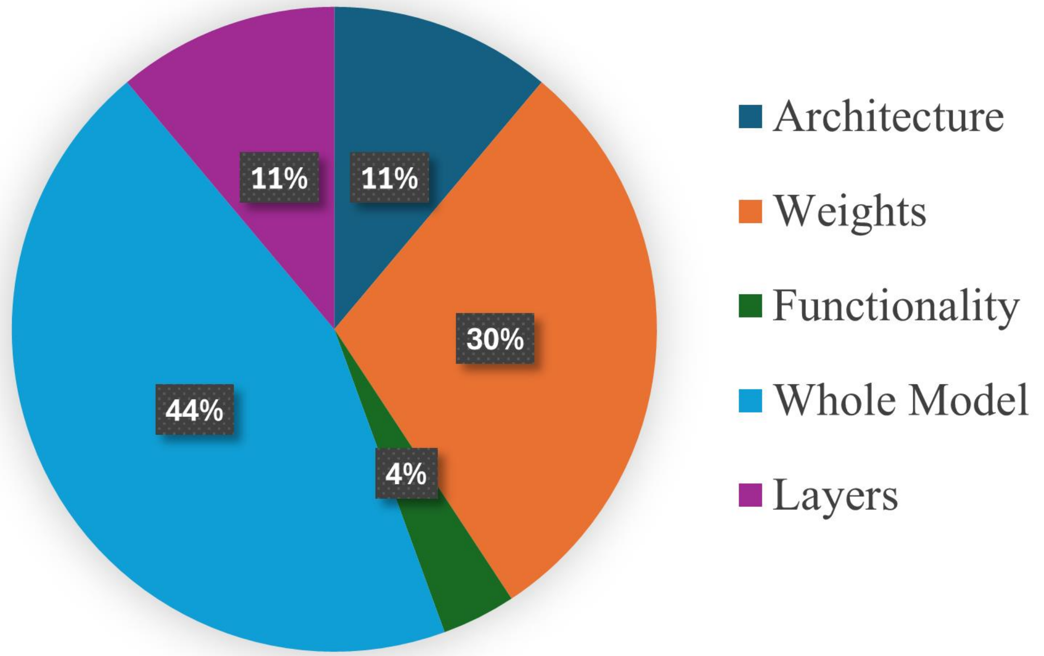
Common Targets for Attack Projects



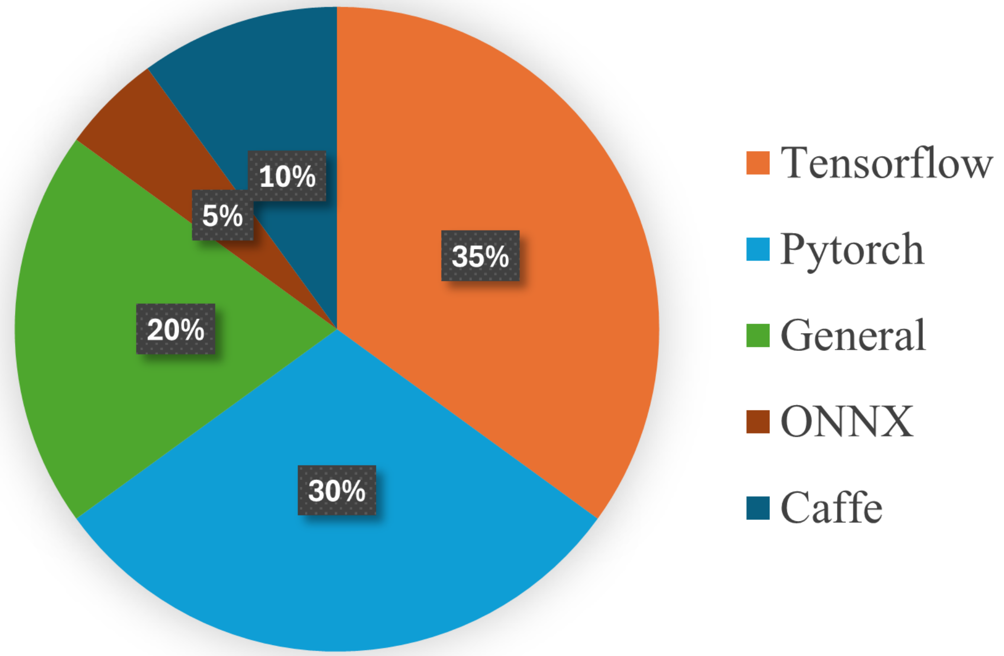
Targeted ML Framework

Existing Model Extraction Defense

Aspect	Examples
Common Attack Targets	Architecture, Weights, Model Functionality, Whole Model, Layer
Targeted ML Frameworks	TensorFlow, PyTorch, General, Caffe, ONNX
Typical Defense Methods	Encryption, Obfuscation, TEE, Transform, Misinformation/Perturbation



Common Targets for Defense Projects

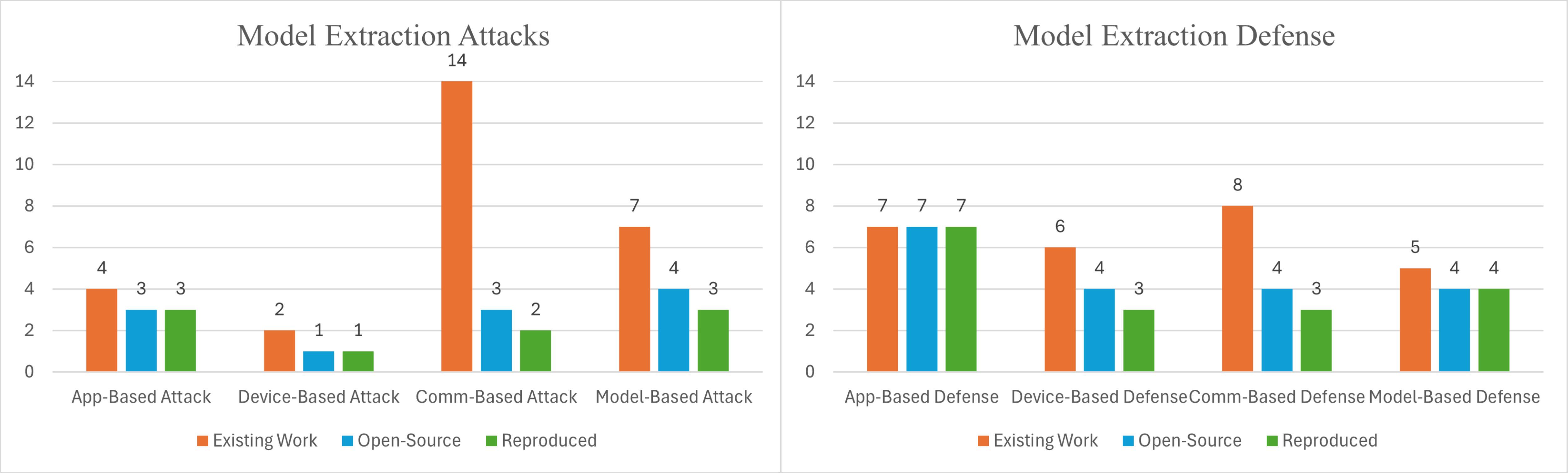


Targeted ML Framework

Evaluation

1. **Research Reproducibility:**
   - Can model extraction attack and defense research be practically replicated?
2. **Effectiveness:**
   - Are the existing model extraction attacks and defenses effective with real-world applications?
3. **Performance Metrics**:
   - What are the computational complexity and power consumption involved?

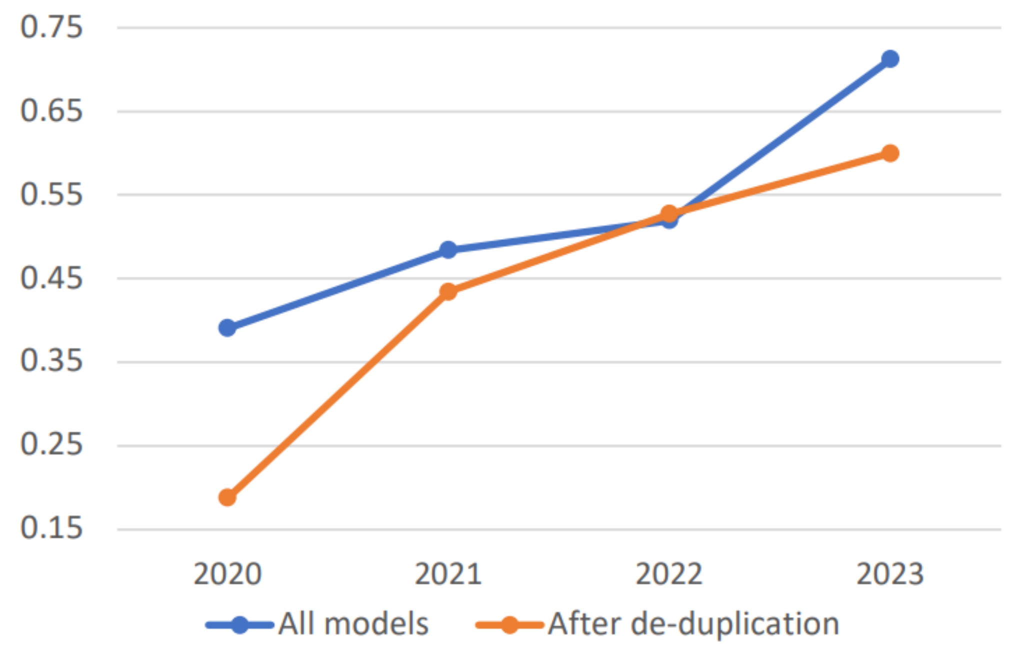
Reproducibility: Attacks & Defenses



Note: Y-axis represents the number of projects

Effectiveness of Model Extraction Attacks

- **APKs Collection:** Gathered ~ 210K APKs from AndroZoo (2020-2023).
- **Model Extraction:** Used *ModelXray*, extracted 16.5K models.
- **De-duplication:** Identified 3K unique model files.



The ***success rate*** of **app-based attacks** (e.g., ModelXray) in the past four years

Model Extraction Attacks: Findings

- **Compatibility Issues**
  - Device-Based Attack (e.g., ModelXtractor) fails with *app instrumentation issues and model buffer identification*.
  - Comm-Based Attacks (e.g., DeepSniffer and DeepSteal) fail with *log incompatibility* and *requires retraining per device*.
  - Model-Based Attack (e.g., ML-Doctor) falters with real-world models due to *model format issues*.
- **High computational demands**
  - Especially for accurate model inference and extraction.
  - Effectiveness depends on dataset complexity

Model Extraction Defenses: Findings

- **Encryption effectiveness is limited.**
  - App-based Defenses (e.g., AES)
- **Expensive setup is required.**
  - Device & Comm-based Defenses (e.g., ShadowNet) requires to *transforms models* - MobileNet and AlexNet.
  - May *reduce defense accuracy*, and may *incur hardware compatibility*.
- **Model format and scalability issues.**
  - Model-based Defenses (e.g., Prediction-Poison, Adaptive Misinformation) achieve <1% accuracy loss but are are *limited to PyTorch models*.

Computation Complexity

Projects	Time Complexity	Factors on which it depends
DeepSniffer	$O(k + f(n) + b * n)$,	kernel classes, sequence model, and search algorithm
DeepSteal	f $O(RowHammerAttacks) + O(W + T * B)$	leaked weights, training iterations and batches.
ML-Doctor	$O(m * d * e)$	number of queries, network size, and epochs for training a student model
AES	$O(m)$	model size, key and block size, and the number of rounds
ShadowNet	$O(TEE + r * l)$	TEE, transformation of linear layers
AM and PP	$O(g * h)$	worst-case perturbation and updating model parameters

Power Consumption

- **Power Analysis:** Intel Performance Counter Monitor (PCM) tool.
- We monitored power consumption in real-time.

Project	Model	Before (J)	After (J)
DeepSniffer	ResNet-18	0.45	29.98
ML-Doctor	a simple CNN	0.70	33.81
AES	ResNet-18	0.41	3.28
PP	LeNet	0.42	33.47
AM	LeNet	0.77	29.24

Power consumption of different projects

Conclusion

- Provided a systematic review of knowledge concerning on-device ML model extraction attacks and defenses.
- Not all attacks are practical or scalable in real-world scenarios.
- Many defense mechanisms are limited in deployment and effectiveness.

Project code

Questions?

Tushar Nayan

tnaya002@fiu.edu

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED