The 20th ACM ASIA Conference on Computer and Communications Security (AsiaCCS), 2025

ChainMarks: Securing DNN Watermark with Cryptographic Chain

Brian Choi, Shu Wang, Isabelle Choi, Kun Sun

Johns Hopkins University
paloalto Networks
UCLA
George Mason University

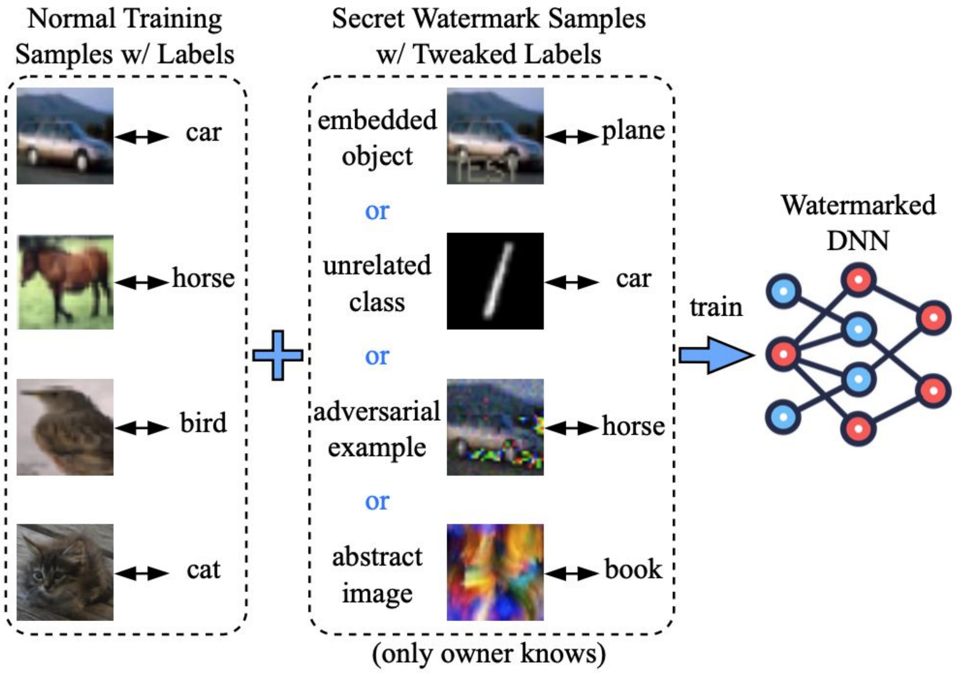
Motivation & Problem

- Value and Vulnerability of DNN Models.
  - **High Value IP:** Developing DNNs is incredibly resource-intensive.
    - massive data collection & curation.
    - expensive, time-consuming training.
    - significant competitive advantage.
  - **The Threat:** Unauthorized use, resale, and model theft are major concerns.
  - **Existing Solution:** Digital watermarking to prove ownership.

Background: Dynamic Watermarking

Dynamic watermarking is a common backdoor-based approach for IP protection.

- The owner creates a secret "trigger set" of inputs and target labels.
- The model is trained on both the original task data and this secret trigger set.
- The final model *<u>behaves normally on standard inputs</u>* but *<u>produces the owner's secret labels when given the trigger inputs</u>*.



Two Core Challenges

- **Security Flaws: The Ambiguity Attack.**
  - Attackers can forge **their own** watermark onto a stolen model.
  - They use optimization techniques (**adversarial learning**) to find a new set of triggers that produce their desired labels.
  - **Ownership dispute**: if two parties can "prove" ownership with two different watermarks, the claims becomes impossible to resolve.
- **Vague Verification: The Unprovable Claim.**
  - The criteria for verifying a watermark is often **unclear and statistically weak**.
  - It is hard to calculate the probability of a random match.
    - Models have highly skewed classification probabilities for random inputs (many classes are never chosen).
  - Existing methods cannot provide high-confidence proof (i.e., a very low p-value).

Our Solution: ChainMarks

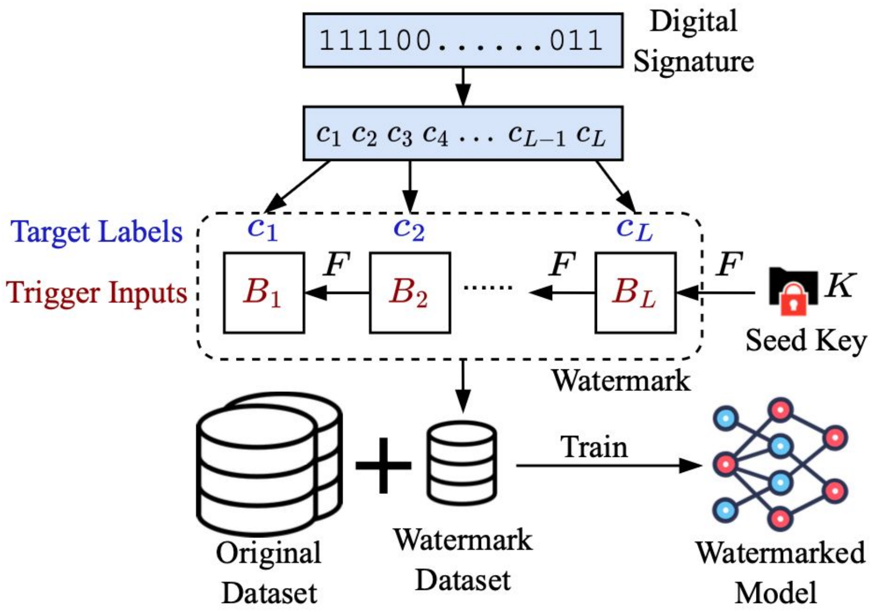
- We propose ChainMarks, a scheme that directly addresses these challenges.
- **Key Ideas**:
  - **Defeat Ambiguity with a Cryptographic Chain**
    - Trigger inputs are not independent but linked sequentially by a one-way hash function.

$$Trigger_n = hash(Trigger_{n+1})$$

    - The structure is computationally infeasible to forge using gradient-based optimization.
  - **Ensure Authenticity with a Digital Signature**
    - The target labels are derived directly from the owner's digital signature.
  - **Provide Rigorous Proof for Decision Threshold**
    - We introduce a two-phase Monte Carlo method to accurately calculate the decision threshold, enabling high-confidence verification.

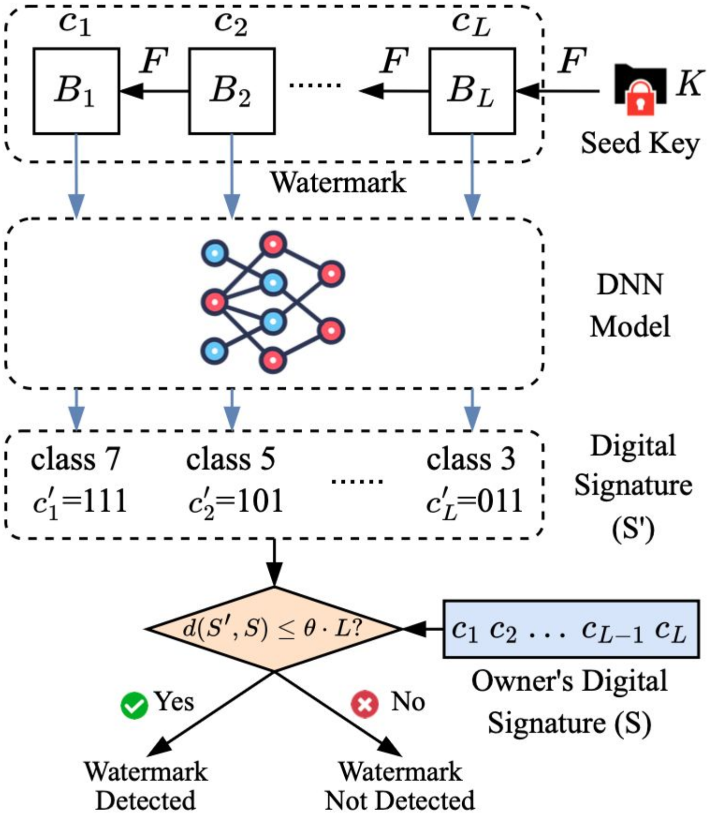
ChainMarks: Watermark Generation & Embedding

- **Generate Trigger Chain**
  - Start with a secret key (K). Repeatedly apply a hash function F to create a chain of trigger inputs B_i: $B_L = F(K), B_{i-1} = F(B_i)$.
- **Generate Target Labels**
  - Convert the owner's Digital Signature (S) into a base-C number (C is #classes).
  - The digits $\{c_i\}$ become the target labels.
- **Embed Watermark**
  - Train the DNN on the original dataset combined with the watermark dataset $\{(B_1, c_1), (B_2, c_2), \ldots\}$.



ChainMarks: Watermark Verification

- **Regenerate Triggers**
  - The verifier regenerates the trigger chain $\{B_i\}$ with the secret Seed Key (K).
- **Query Model**
  - The triggers are fed into the suspect model to to get predicted labels $\{c_i'\}$.
- **Compare Signatures**
  - The Hamming distance $d(S', S)$ between the predicted and original labels is calculated.
- **Decision**
  - If the distance is above a threshold, ownership is confirmed: $d(S', S) \leq \theta \cdot L$.



The Crux: How to Set the Decision Threshold θ?

Threshold must be statistically robust to prevent attackers from matching it by pure chance.

- **Question:** What is the probability that ***a random seed key and random signature*** would produce *m or more* matches on a given model?
- **Difficulty:** This probability depends on *model's classification behavior* for random, noise-like inputs.
- **Observation:** This behavior is extremely skewed.
  - For random inputs, some classes are predicted frequently, while others are never predicted.

Dataset	Avg. Prob.	Min Prob.	Max Prob.	Prob. Stdev	# of classes never hit
CIFAR-10	0.1	0	0.9962	0.2987	5
CIFAR-100	0.01	0	0.9433	0.0399	49

Skewed probability distribution across different classes for DNN models trained on CIFAR-10/CIFAR-100

For a ResNet-18 on CIFAR-100, 49 out of 100 classes were **never** hit by 10 million random inputs.

Our Method: Two-Phase Monte Carlo Estimation

Standard estimation fails due to the zero-hit classes. Our two-phase approach solves this.

Phase 1: Initial Distribution & Zero-Hit Set

- Feed a large number N (e.g., 10 million) of random inputs into the model.
- Calculate initial probabilities p_i for all classes i that were hit.
- Identify the set of classes U that had **zero hits**.

Phase 2: Estimate Probability of the Zero-Hit Set

- Feed more random inputs until a class in U is hit for the **first time**.
- The number of trials required to get this first hit gives us an accurate estimate of the total probability mass p_U for the entire zero-hit set.

From Probability Profile to Secure Threshold

Once we have the accurate classification probabilities P_{c_i} for each target label c_i .

- **Model the Guessing Attack**: The number of matches M in L trials follows a Poisson Binomial Distribution.
- **Calculate Success Probability**: The probability of getting at least m matches out of L candidates is

$$P(M \geq m) \approx \Phi(\frac{L+0.5-\mu}{\sigma'}) - \Phi(\frac{L-0.5-\mu}{\sigma'})$$

- **Set the Threshold:**
  - define a desired security level (e.g., p-value < 10^{-7}), which is max acceptable probability for a guessing attack to succeed.
  - find the minimum number of matches m needed to achieve this p-value.
  - obtain decision threshold: $\theta = 1 - (m/L)$.

Experimental Setup

- **Datasets**: CIFAR-10, CIFAR-100
- **Models**: ResNet-18, ResNet 28x10
- **Baseline Schemes**:
  - *Adi et al.* (abstract images)
  - Content-based (masked images)
  - Noise-based (Gaussian noise)
  - Unrelated-images
- **Attacks Evaluated** (17 total):
  - Watermark Ambiguity Attack
  - 16 Watermark Removal Attacks:
    - Input Preprocessing
    - Model Modification
    - Model Extraction

Black-box watermarking schemes in evaluation.

Scheme	Category	Verification	Capacity
ChainMarks	model dependent/independent	black-box	multi-bit
Adi	model dependent/independent	black-box	multi-bit
Content	model independent	black-box	zero-bit
Noise	model independent	black-box	zero-bit
Unrelated	model independent	black-box	zero-bit

Watermark removal attacks in our evaluation.

Attack	Category	Param. Access	Data Access
Adaptive Denoising	Input Preprocessing	White-box	None
JPEG Compression			
Input Quantization			
Input Smoothing			
Adversarial Training	Model Modification		Domain
Fine-Tuning (RTLL, RTAL)			
Weight Quantization			
Weight Pruning			
Regularization			
Fine-Tuning (FTLL, FTAL)			Labeled Subset
Transfer Learning	Model Extraction	Black-box	Domain
Retraining			
Cross-Architecture Retraining			
Adversarial Training (From Scratch)			

Results: Test Accuracy and Watermark Accuracy

- The impact of watermark embedding on model test accuracy is negligible, typically under 1%.
- After watermark removal or ambiguity attacks, the watermark accuracy decreases; however, the number of remaining valid watermarks is sufficient for ownership verification.

Accuracy	**Accuracies (CIFAR-10/CIFAR-100)**				
	ChainMarks	**Adi**	**Content**	**Noise**	**Unrelated**
Test Accuracy w/o WM embedding	0.923/0.691	0.921/0.692	0.915/0.684	0.913/0.685	0.914/0.682
Test Accuracy w/ WM embedding	0.915/0.683	0.916/0.685	0.91/0.681	0.911/0.678	0.909/0.676
Test Accuracy after Attack	0.78/0.68	0.77/0.69	0.56/0.52	0.81/0.73	0.53/0.51
WM Accuracy after Embedding	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0
WM Accuracy after Attack	0.67/0.34	0.69/0.37	0.58/0.33	0.73/0.41	0.64/0.35

Test and watermark (WM) accuracy before/after watermark embedding and after watermark attacks.

Results: Robustness Against Attacks

Key Finding:

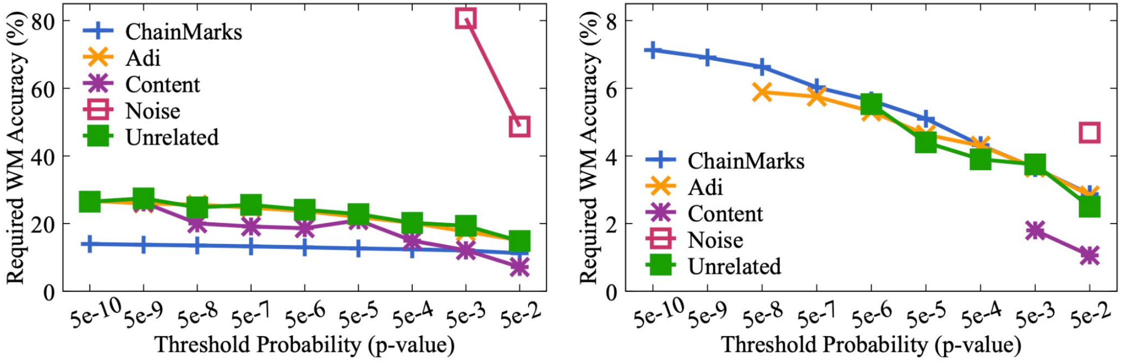
- ChainMarks is the only scheme that successfully resists the Watermark Ambiguity Attack. All other baselines are vulnerable.
- Against the 16 removal attacks, ChainMarks demonstrates comparable or superior robustness to the state-of-the-art.

Attack Types	Robust (-) or Vulnerable (V) for CIFAR-10 / CIFAR-100				
	ChainMarks	Adi	Content	Noise	Unrelated
WM Ambiguity Attack	-/-	V/V	V/V	V/V	V/V
Adaptive Denoising	-/-	-/-	-/-	-/-	-/-
JPEG Compression	-/-	-/-	-/-	-/-	-/-
Input Quantization	-/-	-/-	-/-	-/-	-/-
Input Smoothing	-/-	-/-	-/-	-/-	-/-
Adversarial Training	-/-	-/-	-/-	-/-	-/-
Fine-Tuning (RTAL)	-/-	-/-	-/-	-/-	-/-
Fine-Tuning (RTLL)	-/-	-/-	-/-	-/-	-/-
Fine-Tuning (FTAL)	-/-	-/-	V/V	-/-	V/V
Fine-Tuning (FTLL)	-/-	-/-	-/-	-/-	-/-
Weight Quantization	-/-	-/-	-/-	-/-	-/-
Weight Pruning	-/-	-/-	-/-	-/-	-/-
Regularization	-/-	V/-	V/-	-/-	V/-
Retraining	-/-	-/-	V/V	V/-	V/V
Transfer Learning	V/V	V/V	V/V	V/V	V/V
Cross-Architecture Retraining	-/-	-/-	V/-	-/-	V/-
Adversarial Training	-/-	-/-	-/-	-/-	-/-

Robustness of different watermarking schemes against 17 attack types (threshold probability p=0.01)

Results: Higher Security & Marginal Utility

- **Higher Security Guarantee**
  - ChainMarks allows verification with much smaller p-values (e.g., 5×10^{-10}). Other methods fail to compute a threshold at these high security levels.
- **Higher Marginal Utility**
  - ChainMarks provides a much greater increase in confidence for every percentage point of watermark accuracy retained after an attack.



(a) models on CIFAR-10. **(b) models on CIFAR-100.**

Required watermark accuracy $(1-\theta)$ vs. threshold probability p, for different watermarking schemes.

Takeaways

We introduced **ChainMarks**, a new paradigm for DNN watermarking.

- **Solves the Ambiguity Problem:** The cryptographic chain makes it computationally infeasible for an attacker to forge a valid watermark, providing unambiguous ownership proof.

- **Robust by Design**: The use of out-of-distribution, noise-like triggers provides strong resilience against a wide range of watermark removal attacks.

- **Quantifiable & High-Confidence Verification:** Our two-phase Monte Carlo method allows for the calculation of precise decision thresholds, enabling ownership claims with extremely high statistical confidence (low p-values).

ChainMarks offers a practical, secure, and robust solution for protecting high-value intellectual property in deep learning models.

Thank you!

Contact: shuvwang@gmail.com

ChainMarks: Securing DNN Watermark with Cryptographic Chain

Brian Choi[1], Shu Wang[2], Isabelle Choi[3], Kun Sun[4]

[1] Johns Hopkins University [2] Palo Alto Networks, Inc. [3] UCLA [4] George Mason University