# Regulating Ownership Verification for Deep Neural Networks: Scenarios, Protocols, and Prospects

**Fang-Qi Li[1], Shi-Lin Wang[1], Alan Wee-Chung Liew[2]**

[1] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University,

[2] School of Information and Communication Technology, Griffith University.

Aug 21st, 2021

IJCAI 2021

# Table of Contents


# 01

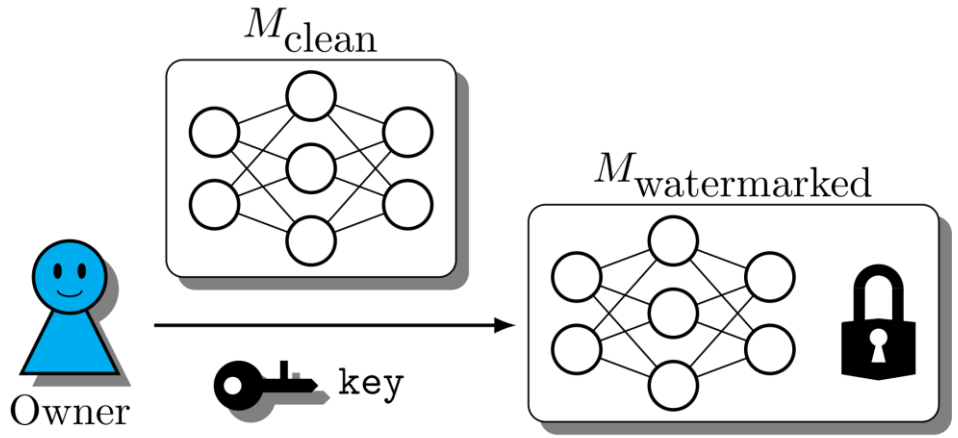
# Introduction

What is the necessity behind deep learning model ownership verification protocols?

# Introduction

**The background:**

Schemes for the ownership verification of deep learning models are technically developing, yet they are only one aspect of DNN IPR.



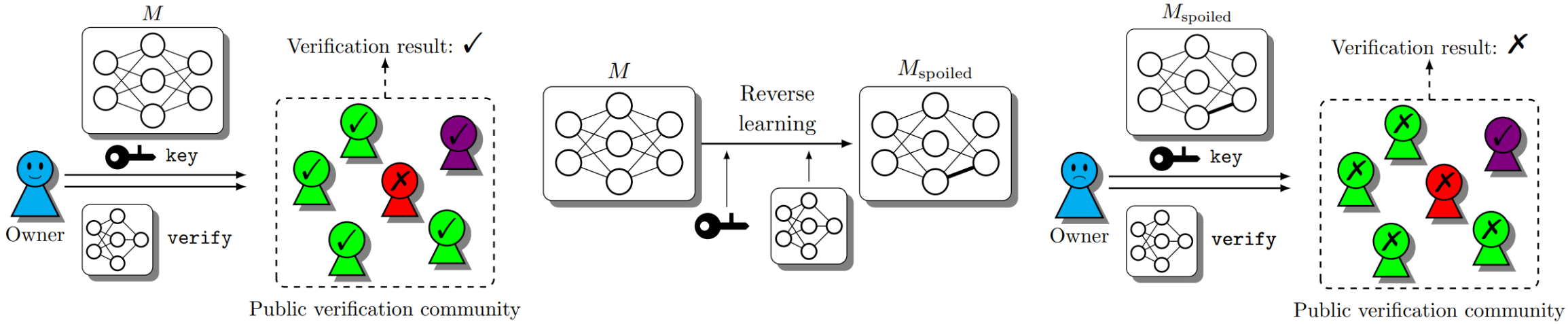
# Introduction

**Problems:**

1. Is it possible to deploy the OV as a service?
2. How to prove the ownership to a third-party *customer*?
3. Can a scheme remain secure against a malicious customer?
4. What about other demands in IPR that go beyond OV?

# Introduction

- Example of a malicious customer: *the spoil attack.*
- What if I prove my ownership to an adversary?




(a) The first verification. (b) The spoil attack. (c) The second verification.

Figure 1: The spoil attack. The blue node is the owner, green nodes are benign agents, the red node is a malicious agent, and the purple one is the eavesdropping adversary.

# Introduction

**Challenges:**

1. Numerous emerging demands in distributed learning models.
2. Convince a third-party is a different task from convincing the owner itself.
3. Malicious customers are hardly considered.

# Introduction

**Solutions:**

1. Incorporating cryptological formulations for provable and practical OV.

2. Devising new metrics to formulate more sophisticated adaptive attacks/industrial demands.


# 02

# Formulation

Involving the cryptological style.

# The cryptological style formulation

- A DNN watermarking scheme WM={Gen,Embed} consists of two modules.
  - One generates identification:

$$\texttt{key} \leftarrow \texttt{Gen}(1^N),$$

    where $N$ is the security parameter.
  - One embeds the identification key into the DNN to be protected:

$$(M_{\text{WM}}, \texttt{verify}) \leftarrow \texttt{Embed}(M_{\text{clean}}, \texttt{key}).$$

- The security parameter is correlated with the security level (e.g., the size of the key space, the number of backdoor triggers, the tuning scale of parameters, etc.).

# The cryptological style formulation

- The *correctness* of a DNN watermarking scheme is formulated as:
  - Integrity:

$$\Pr\{\texttt{verify}(M_{\text{WM}}, \texttt{key}) = 1\} \geq 1 - \epsilon,$$

  - Unambiguity:

$$\Pr\{\texttt{verify}(M_{\text{WM}}, \texttt{key}_{\text{ADV}}) = 0\} \geq 1 - \epsilon,$$

- The error term ε is a function negligible in the security parameter $N$, ε captures the concept of asymptotically security.
- These two properties have to be formally examined for all DNN watermarking schemes.

# The cryptological style formulation

**Challenges:**

1. The toolkit of cryptoanalysts' is insufficient.

2. Some requirements cannot be formally defined or proven by reduction.

3. Some requirements are even contradictive against each other.


# 03

## Scenarios

Basic OV, federated learning, and IP transfer.

# Scenario I: The basic OV

**A naïve centralized solution:**

Use a centralized architecture where an authority is responsible for all OV requests.

The owner submits its evidence (key,verify) to the center, the center publishes the proof to the destinated customer.

The *multiple-owner dilemma* can be handled by authorized time-stamps.

Consequent problems: inflexibility, security risk, heavy burden, etc.

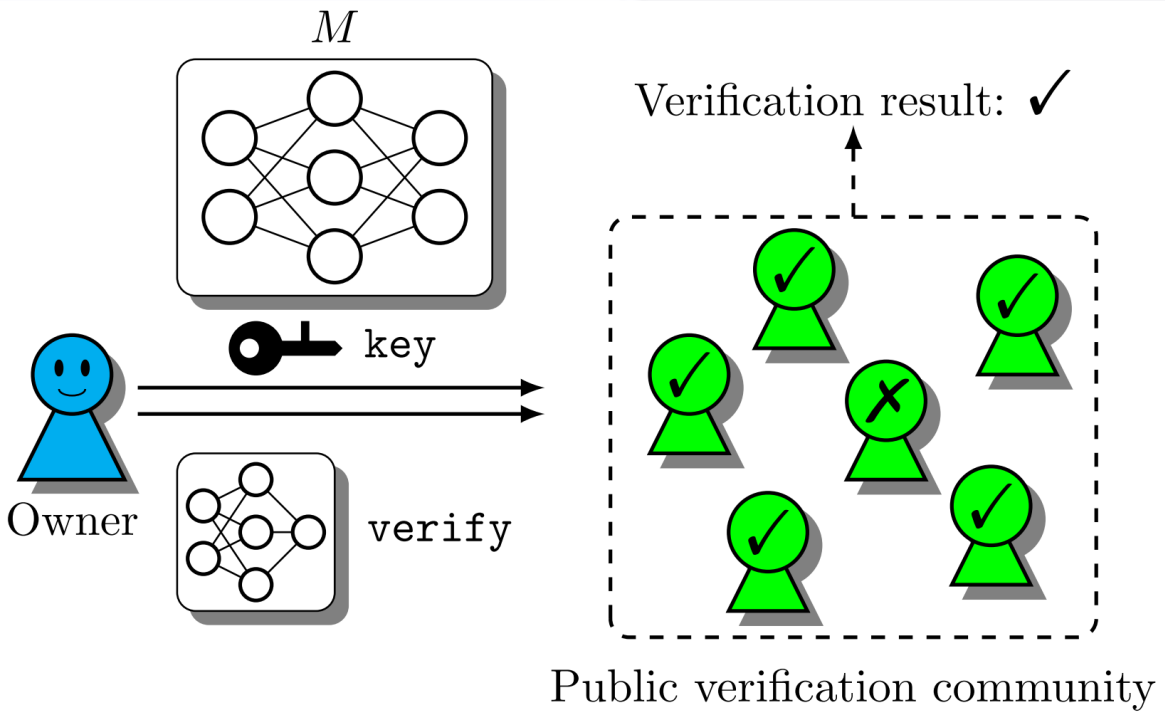
# Scenario I: The basic OV

**A decentralized solution:**

Ownership proof is done by a distributed community under a consensus protocol.

Advantages: flexible, secure, and cheap.

# Scenario I: The basic OV




**Algorithm 3** The decentralized OV protocol.

**Participants**: The owner, the verification community

**Modules**: A watermarking scheme `WM`, a digital signature scheme, a consensus protocol

1: The owner generates $M_{\text{clean}}$.
2: The owner generates `key`, $M_{\text{WM}}$, and `verify` by `WM`.
3: The owner signs the following message:

$$\langle \texttt{time} \| \texttt{hash(key)} \| \texttt{hash(verify)} \| \texttt{hash(info)} \rangle$$

using the digital signature scheme, where `time` is the current time-stamp, `hash` is a hash function, and `info` descripts the DNN model's architecture.
4: The owner broadcasts the signed message to the community using the consensus protocol.
5: To conduct OV over a DNN $M$, the owner signs and broadcasts $\langle M \| \texttt{key} \| \texttt{verify} \rangle$.
6: An agent retrives the time-stamp by computing `hash(verify)`, and submits `verify`$(M, \texttt{key})$ to the community using the consensus protocol.

# Scenario I: The basic OV

**A decentralized solution:**

By specifying the broadcasted messages, this protocol can address the multiple-owner dilemma as well.

The owner only has to hash the network structure with its evidence and broadcasts the hashed message before publishing its DNN.

# Scenario I: The basic OV

**Challenges:**

The spoil attack?

**Solution:**

Embedding multiple independent watermarks into the model.

Broadcasting all evidence as a Merkle-tree.

# Scenario I: The basic OV

**Requirements beyond basic security demands:**

The watermarking capacity.

How many independent watermarks can be embedded into a DNN model?

# Scenario I: The basic OV

DEFINITION 1. The watermark capacity for a DNN model, $\mathsf{cap}^{\delta}_{\mathsf{WM}}$, is the maximal number of keys that can be correctly embedded by $\mathsf{WM}$ into the model until the DNN model's performance drops by $\delta$ w.r.t. the metric $\mathcal{E}$ defined in its primary task.

The value $\mathsf{cap}^{\delta}_{\mathsf{WM}}$ measures the upper bound of the number of successfully embedded watermarks inside a model. Formally, $\mathsf{cap}^{\delta}_{\mathsf{WM}}$ is the maximal $q$ satisfying the following conditions:

$$\begin{aligned}
(M_1, \mathsf{verify}_1) &\leftarrow \mathsf{Embed}(M_{\text{clean}}, \mathsf{key}_1), \\
(M_2, \mathsf{verify}_2) &\leftarrow \mathsf{Embed}(M_1, \mathsf{key}_2), \\
&\cdots \\
(M_q, \mathsf{verify}_q) &\leftarrow \mathsf{Embed}(M_{q-1}, \mathsf{key}_q), \\
\mathcal{E}(M_q) &\geq \mathcal{E}(M_{\text{clean}}) - \delta,
\end{aligned} \tag{2}$$

where $\mathsf{key}_q$ is generated by $\mathsf{Gen}$ and all $q$ watermarks can be correctly verified.

# Scenario I: The basic OV

**Requirements beyond basic security demands:**

The watermarking independency.

Would spoiling one watermark invalidate others?

# Scenario I: The basic OV

**What if a watermark has low capacity and poor independency?**

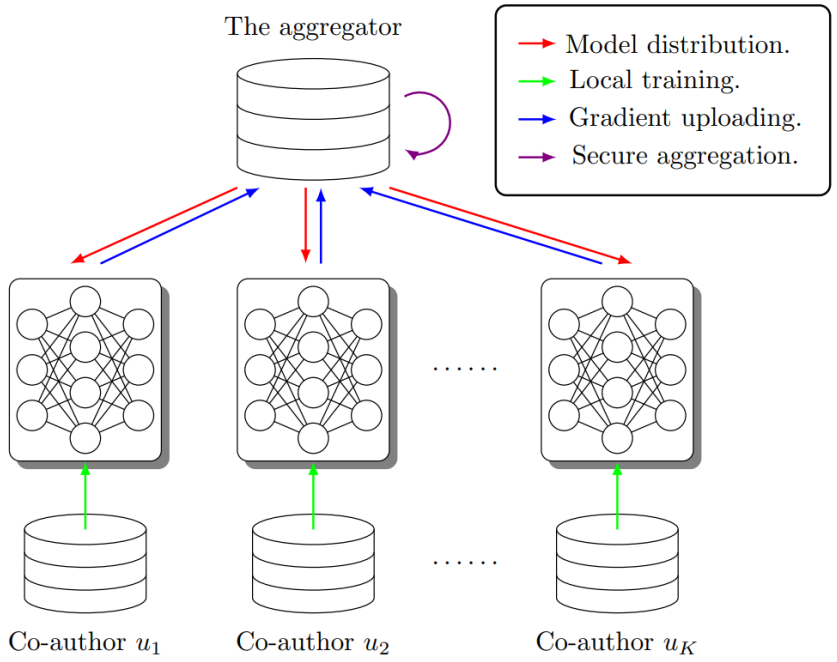
An eavesdropping adversary can pirate the protected model.

Hence the IPR is intractable.

# Scenario II: The federated learning

**Specialized demands in FL IPR:**

1. Independency.
2. Privacy-preserving.
3. Recovery.
4. Traitor-tracing.



# Scenario II: The federated learning

**Table 2: Dependence of the advanced requirements on basic security requirements.**

| **Advanced requirements** | **Basic security requirements** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Correctness | Functionality-preserving. | Security against tuning. | Covertness. | Privacy. | Security against overwriting. | Security against piracy. | Multiple-time verification. |
| Independency. | ✓ | ✓ | × | × | × | ✓ | ✓ | ✓ |
| Privacy-preserving. | × | × | × | ✓ | × | × | × | × |
| Recovery. | ✓ | ✓ | × | × | × | × | × | × |
| Traitor-tracing. | ✓ | ✓ | ✓ | ✓ | × | × | ✓ | ✓ |

✓means relevant. × means irrelevant.

# Scenario II: The federated learning

**FL IPR can be achieved by adopting a different protocol:**

1. The aggregator is the temporal center.
2. Insert latent key for traitor-tracing.

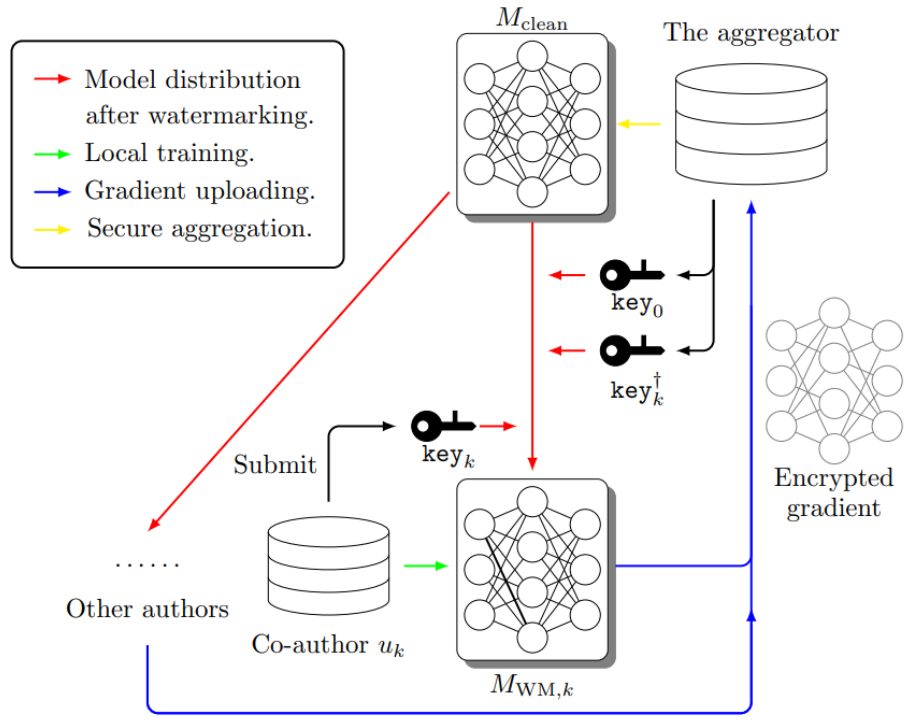
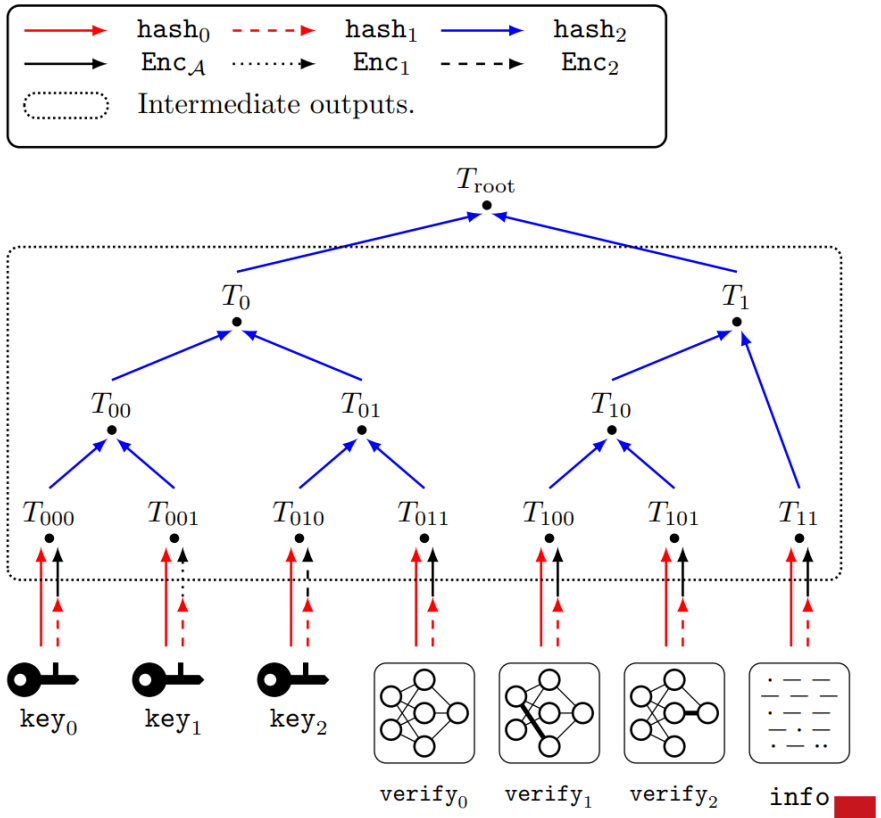



Figure 3: The `Merkle-Sign` watermarking framework for FL.

# Scenario II: The federated learning

**FL IPR can be achieved by adopting a different protocol:**

1. All identification keys are organized into a Merkle-tree.
2. This ensures recovery and decentralized OV.



# Scenario II: The federated learning

**Table 3: Evaluation of watermarking capacity, the maximal number of correctly embedded watermarks when the classification error rate doubles. `WF` represents `Wonder Filter`, `M-S` represents `MTL-Sign`.**

| Dataset | MLP | | | | | |
|---|---|---|---|---|---|---|
| | Accuracy | Uchida's | ATGF | Rand | WF | M-S |
| MNIST | 91.92% | 36 | 23 | 17 | 43 | 65 |
| Fashion | 82.90% | 45 | 75 | 50 | 66 | 95 |
| CIFAR10 | 25.92% | — | — | — | — | — |
| CIFAR100 | 13.26% | — | — | — | — | — |

| Dataset | Shallow CNN | | | | | |
|---|---|---|---|---|---|---|
| | Accuracy | Uchida's | ATGF | Rand | WF | M-S |
| MNIST | 97.71% | 110 | 12 | 13 | 17 | 55 |
| Fashion | 85.40% | 90 | 17 | 19 | 36 | 65 |
| CIFAR10 | 61.45% | 8 | 7 | 7 | 33 | 90 |
| CIFAR100 | 30.03% | — | — | — | — | — |

| Dataset | ResNet-18 | | | | | |
|---|---|---|---|---|---|---|
| | Accuracy | Uchida's | ATGF | Rand | WF | M-S |
| MNIST | 99.60% | 8,750 | 350 | 373 | 453 | 9,503 |
| Fashion | 93.81% | ≥10,000 | 519 | 513 | 588 | ≥10,000 |
| CIFAR10 | 89.10% | ≥10,000 | 582 | 572 | 663 | ≥10,000 |
| CIFAR100 | 62.59% | ≥10,000 | 612 | 610 | 797 | ≥10,000 |

| Dataset | ResNet-50 | | | | | |
|---|---|---|---|---|---|---|
| | Accuracy | Uchida's | ATGF | Rand | WF | M-S |
| MNIST | 99.72% | ≥10,000 | 417 | 411 | 494 | ≥10,000 |
| Fashion | 95.25% | ≥10,000 | 580 | 540 | 669 | ≥10,000 |
| CIFAR10 | 91.50% | ≥10,000 | 600 | 612 | 773 | ≥10,000 |
| CIFAR100 | 67.70% | ≥10,000 | 710 | 712 | 779 | ≥10,000 |

# Scenario III: The IP transfer

**How to prevent a malicious seller from selling a DNN to a customer and redeclaring its ownership?**

A seller has to prove that its model is *clean* when transferring the model's IP to a customer.

# Scenario III: The IP transfer

**Problem: A distinguisher on the model's freedom from an watermark breaches the watermark's *covertness*.**

**Algorithm 1** $\texttt{Exp}_{\mathcal{A}}^{\text{covertness}}$.

**Input**: $\mathcal{A}$, $N$, $\texttt{WM}$, $M_{\text{clean}}$
**Output**: Whether $\mathcal{A}$ wins or not

1: Randomly select $b \leftarrow \{0, 1\}$.
2: Generate $M_{\text{WM}}$ from $\texttt{WM}(M_{\text{clean}}, N)$.
3: $\mathcal{A}$ is given $N$ and $\texttt{WM}$.
4: **if** $b = 0$ **then**
5: $\quad$ $\mathcal{A}$ is given $M_{\text{clean}}$.
6: **else**
7: $\quad$ $\mathcal{A}$ is given $M_{\text{WM}}$.
8: **end if**
9: $\mathcal{A}$ outputs $\hat{b}$.
10: $\mathcal{A}$ wins the experiment if $\hat{b} = b$.

**Algorithm 5** $\texttt{Exp}_{\mathcal{P}}^{\text{clean}}$.

**Input**: $\mathcal{P}$, $N$, $\texttt{WM}$, $M_{\text{clean}}$
**Output**: Whether $\mathcal{P}$ wins or not

1: Generate $M_{\text{WM}}$ from $\texttt{WM}(M_{\text{clean}}, N)$.
2: Randomly select $b \leftarrow \{0, 1\}$.
3: $\mathcal{P}$ is given $N$ and $\texttt{WM}$.
4: **if** $b = 0$ **then**
5: $\quad$ $\mathcal{P}$ is given $M_{\text{clean}}$.
6: **else**
7: $\quad$ $\mathcal{P}$ is given $M_{\text{WM}}$.
8: **end if**
9: $\mathcal{P}$ outputs $\hat{b}$.
10: $\mathcal{P}$ wins the experiment if $\hat{b} = b$.


# 04

# Conclusion & prospects

# Prospects: Flexible and provably secure watermark

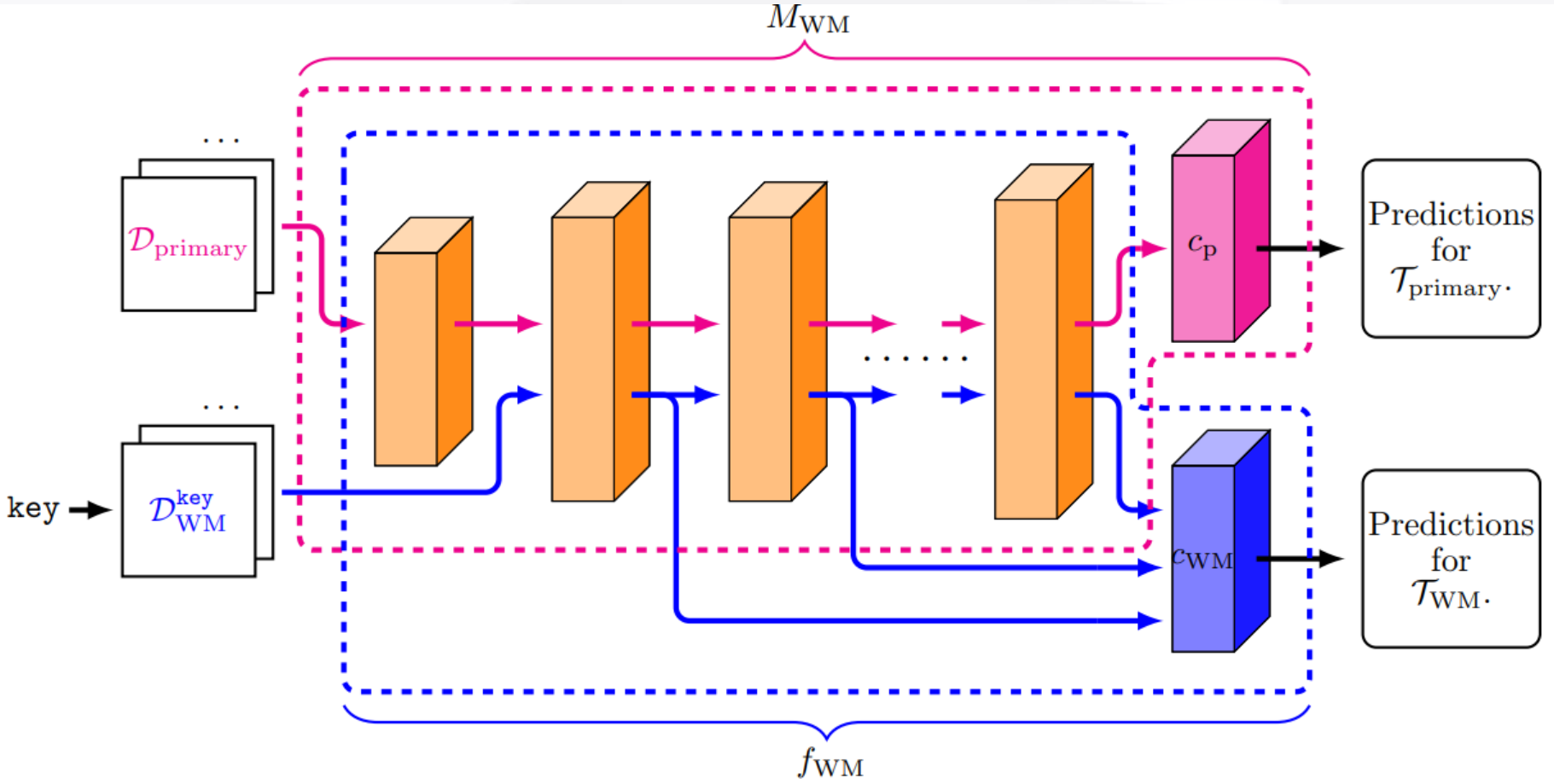



Figure 1: Architecture of the MTL-based watermarking scheme component in `MTLSign`. The orange blocks are the backbone, the pink block is the backend for $\mathcal{T}_{\text{primary}}$, the blue block is the classifier for $\mathcal{T}_{\text{WM}}$.

# Prospects: Flexible and provably secure watermark

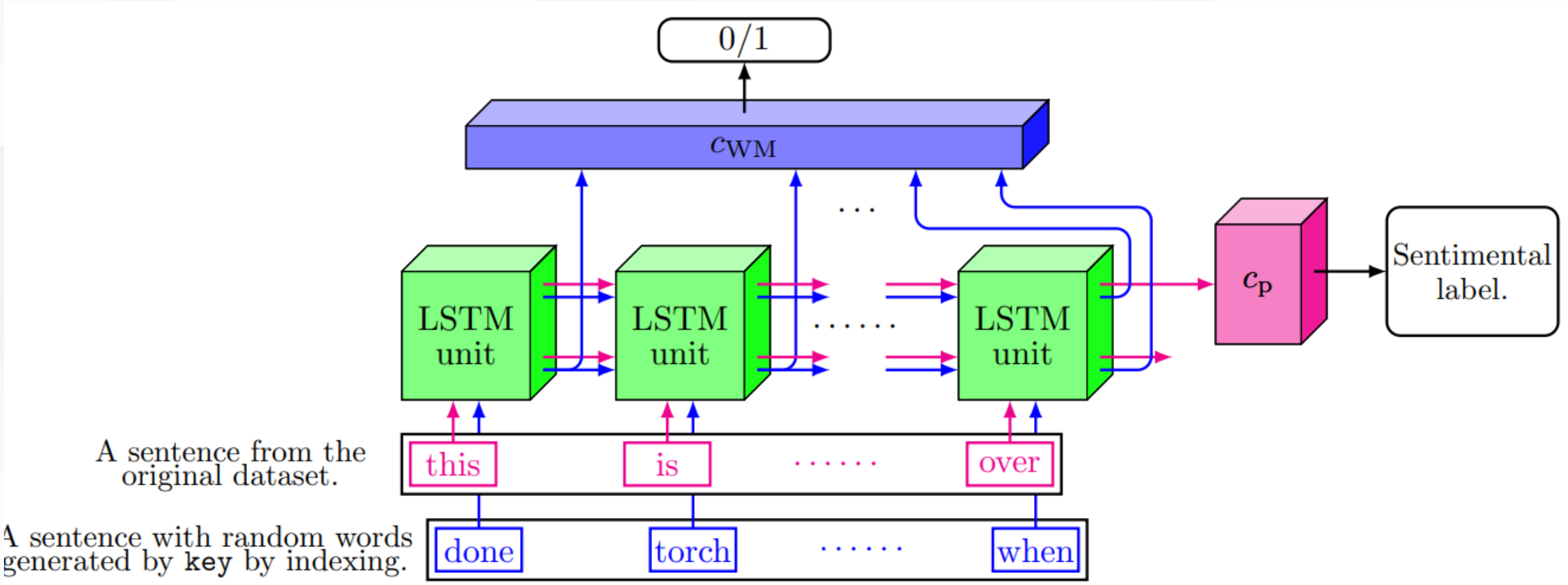



Figure 3: The network architecture for sentimental analysis.

# Prospects: Solve overwriting without time-stamps

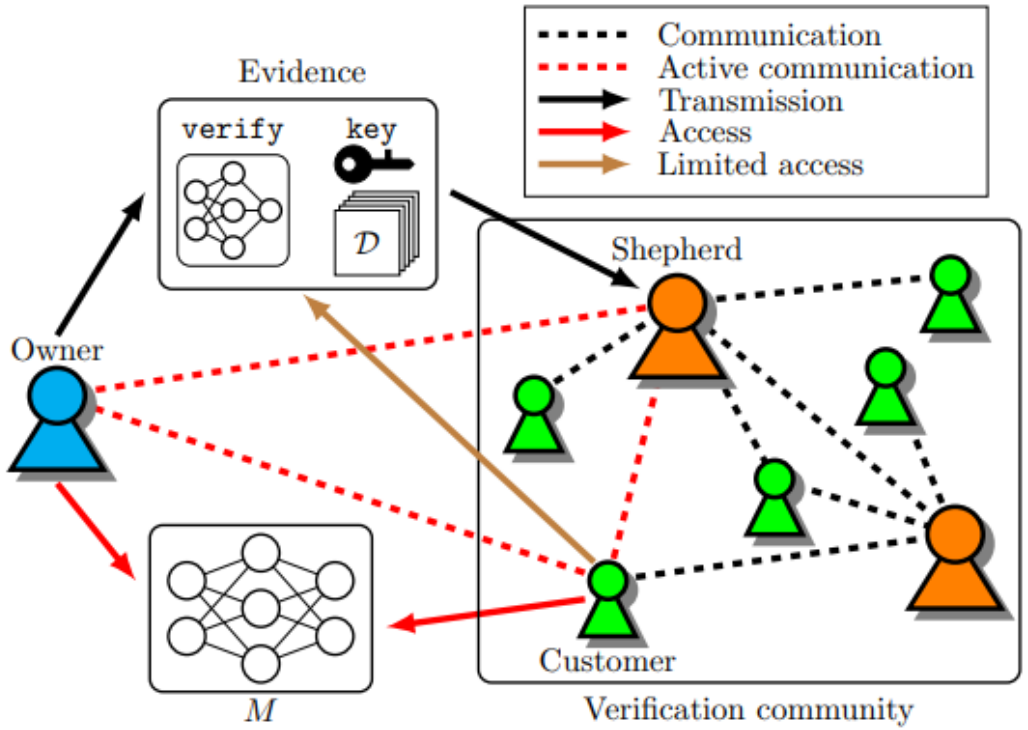



Figure 1: The semi-centralized verification protocol.

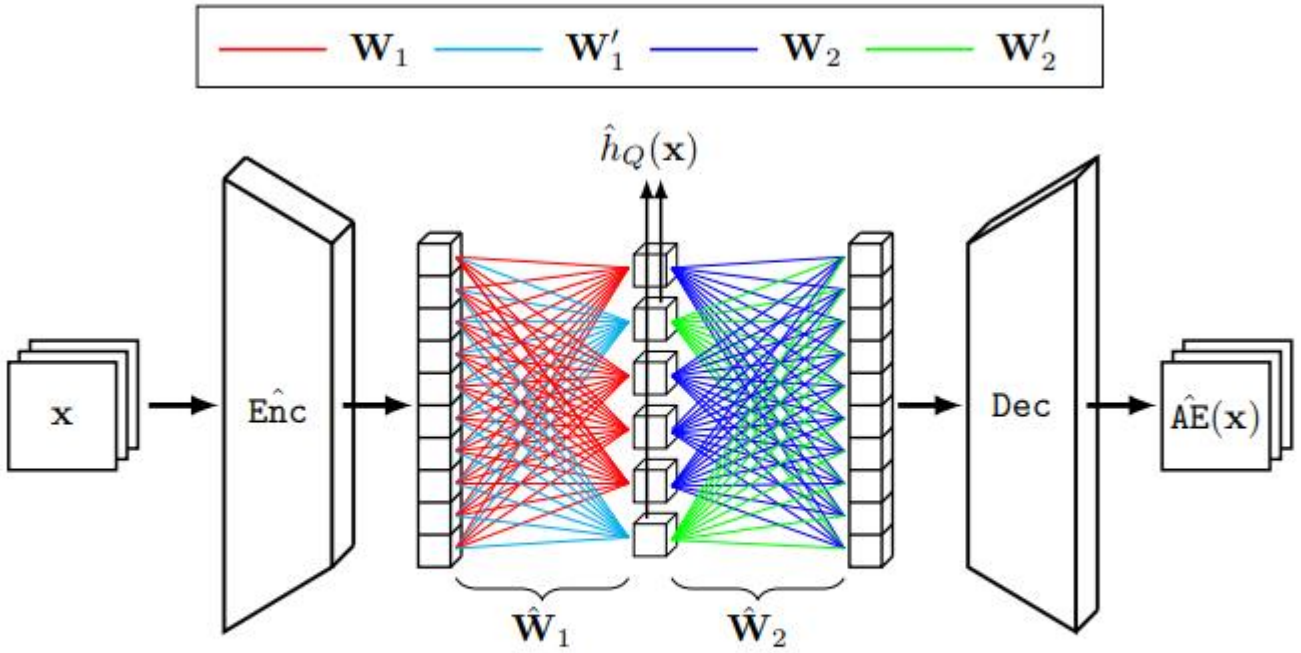



Figure 2: The autoencoder architecture of spectral steganography, $\hat{\mathrm{AE}}$.

上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

# Thank you for listening!

饮 水 思 源　爱 国 荣 校