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

Fang-Qi Li[1] , **Shi-Lin Wang**[1*] and **Alan Wee-Chung Liew**[2]
[1]School of Electronic Information and Electrical Engineering,
Shanghai Jiao Tong University, Shanghai, China.
[2]School of Information and Communication Technology,
Griffith University, Gold Coast Campus, Australia.
{solour_lfq, wsl}@sjtu.edu.cn, a.liew@griffith.edu.au

Abstract

With the broad application of deep neural networks, the necessity of protecting them as intellectual properties has become evident. Numerous watermarking schemes have been proposed to identify the owner of a deep neural network and verify the ownership. However, most of them focused on the watermark embedding rather than the protocol for provable verification. To bridge the gap between those proposals and real-world demands, we study the deep learning model intellectual property protection in three scenarios: the ownership proof, the federated learning, and the intellectual property transfer. We present three protocols respectively. These protocols raise several new requirements for the bottom-level watermarking schemes.

1 Introduction

The development of deep learning boosted the application of deep neural networks (DNN). Given abundant data and computing resources, DNNs outperform traditional models in many disciplines such as image processing, natural language processing [Guo *et al.*, 2020], internet of things [Lv *et al.*, 2020], etc. The expense behind a DNN is high. Much data is collected, processed, and labeled. Designing the DNN architecture and tuning its parameters also involves tremendous effort. Therefore, DNNs are the intellectual property (IP) of the legitimate owner.

Ownership verification (OV) is necessary for the Intellectual Property Protection (IPR) of DNNs. To achieve OV, various DNN watermarking schemes have been proposed. A watermarking scheme embeds the owner-dependent watermark into the DNN, whose later revealing proves the owner's identity. Based on the access level at which the suspicious DNN can be interacted with, watermarking schemes can be classified into *white-box* ones and *black-box* ones.

In the white-box setting, the owner has full access to the pirated DNN. The watermark can be encoded into the model's parameters [Uchida *et al.*, 2017] or intermediate outputs [Darvish Rouhani *et al.*, 2019]. The owner can also insert extra modules into the DNN's intermediate layers for OV [Fan *et al.*, 2021]. As for the black-box setting, the owner can only interact with the pirated model through an API. Watermarking schemes for this case usually resort to backdoors [Zhang *et al.*, 2018; Zhu *et al.*, 2020].

In contrast to the variety of proposed watermarking schemes, discussions on the protocol under which the OV is conducted remain scanty. The OV protocol is indispensable for commercializing deep learning models, but established works on OV protocols mainly focused on the secure transmission of watermarks and are highly inflexible [Adi *et al.*, 2018]. So the IPR of DNNs as a service remains a challenge.

Emerging real-world scenarios are providing diversified challenges for the verification protocols. For example, in the model competition, there exists a trusted sponsor with white-box access to all DNNs. While in commercial services where models are deployed as APIs, such a trusted party is unavailable. Distributed learning paradigms such as the Federated Learning (FL) introduce extra security requirements, which go beyond the scope of established watermarking schemes and protocols. Moreover, it is of broad interest to enlarge the coverage of IPR for DNN models to disciplines other than piracy identifying, e.g., secure intellectual property transfer.

To apply established DNN watermarking schemes to real-world scenarios for IPR of DNN models, we formulate these practical demands and propose candidate protocols meeting respective settings. The contributions of this paper are:

- We analyze three real-world scenarios involving IPR of DNNs and formulate respective protocols.
- We show that some security properties of the proposed protocols can be built upon the security of underlying watermarking schemes by reduction.
- We explore several additional requirements for the watermarking schemes introduced by the protocols, which are prospective directions for further research.

2 Properties of Watermarking Schemes

In general, a watermarking scheme $\mathtt{WM}$ is composed of two modules $\{\mathtt{Gen}, \mathtt{Embed}\}$, one generates the watermark:

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

one embeds the watermark into the model to be protected:

$$(M_{\mathrm{WM}}, \mathtt{verify}) \leftarrow \mathtt{Embed}(M_{\mathrm{clean}}, \mathtt{key}).$$

*Shi-Lin Wang is the contact author.

The watermark is an identifier $\texttt{key}$ representing the owner's identity and N is the security parameter. The embedding module takes a clean DNN M_{clean} as its input. The module $\texttt{verify}$ returned from $\texttt{Embed}$ is part of the evidence for reconstructing the owner's identity from M_{WM} to achieve OV. As has been outlined in [Li and Wang, 2021], a watermarking scheme has to satisfy the following security requirements.

2.1 Correctness

The module $\texttt{verify}$ can identify the owner's identity from the watermarked model:

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

where ϵ is a function negligible in N. Meanwhile, an adversary's identity cannot pass the verifier:

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

where $\texttt{key}_{\text{ADV}}$ is the adversary's evidence randomly sampled from the key space.

2.2 Robustness

The adversary can tune the pirated model using fine-tuning, neuron-pruning, fine-pruning [Liu *et al.*, 2018], even distillation [Zhang *et al.*, 2021]:

$$M_{\text{tuned}} \xleftarrow{\text{tuning}} M_{\text{WM}}.$$

Under a robust watermarking scheme, such tuning should not affect the accuracy of OV:

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

2.3 Covertness

An adversary should not be able to distinguish a watermarked model from a clean one. Otherwise, the adversary might manage to escape the IP regulation. Formally, we design the following Algo. 1. The watermarking scheme is covert if no efficient probabilistic machine $\mathcal{A}$ can win $\texttt{Exp}_{\mathcal{A}}^{\text{covertness}}$ with a probability significantly higher than $\frac{1}{2}$.

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

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

1: Randomly select $b \leftarrow \{0, 1\}$.
2: Generate M_{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_{clean}.
6: **else**
7: $\quad\mathcal{A}$ is given M_{WM}.
8: **end if**
9: $\mathcal{A}$ outputs $\hat{b}$.
10: $\mathcal{A}$ wins the experiment if $\hat{b} = b$.

Algorithm 2 $\texttt{Exp}_{\mathcal{A}}^{\text{key-pp}}$.

Input: $\mathcal{A}$, N, $\texttt{WM}$, $\texttt{key}_0 \neq \texttt{key}_1$, M_{clean}
Output: Whether $\mathcal{A}$ wins or not

1: Randomly select $b \leftarrow \{0, 1\}$.
2: Generate M_{WM} from $\texttt{WM}(M_{\text{clean}}, \texttt{key}_b, N)$.
3: $\mathcal{A}$ is given M_{WM}, M_{clean}, N, $\texttt{WM}$, $\texttt{key}_0$, $\texttt{key}_1$.
4: $\mathcal{A}$ outputs $\hat{b}$.
5: $\mathcal{A}$ wins the experiment if $\hat{b} = b$.

2.4 Privacy-preserving

The privacy-preserving property suggests that no adversary can identify the model's ownership given only partial information of the owner. One type of privacy-preserving is defined through Algo. 2. If no efficient $\mathcal{A}$ can win $\texttt{Exp}_{\mathcal{A}}^{\text{key-pp}}$ with a probability significantly higher than $\frac{1}{2}$ then $\texttt{WM}$ is *key-privacy-preserving* [Li and Wang, 2021]. Analogously, we can define *verifier-privacy-preserving*.

The key-privacy-preserving properties suggests that the $\texttt{verify}$ module of the watermarking scheme should depend on $\texttt{key}$. Otherwise the privacy is easily breached.

Example 1. *The watermarking scheme of Uchida's replaces U parameters within the clean DNN by special digits. Its keyspace can be defined as $\mathbb{R}^U$ or $\Theta^U \times \mathbb{R}^U$, where Θ is the space of all parameters in the DNN model. Being formulated in the first manner, Uchida's is a key-privacy-preserving scheme. In the second formulation, a legal $\texttt{key}$ includes both the place where the digits are embedded and the digits. The corresponding $\texttt{verify}$ is only a parameter-free comparison operator so it is not a key-privacy-preserving scheme.*

2.5 Overwriting issues

Having known the watermarking scheme, the adversary can embed its identity into the model:

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

So the ownership becomes ambiguous. To cope with this threat, the owner's watermark must not be invalidated, i.e.:

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

In cases where the adversary embeds its watermark into the DNN and *redeclare* the ownership, extra mechanisms, e.g., authorized time-stamp, are necessary to break the tie.

3 Scenarios and Watermarking Protocols

IPR involves proving the ownership to a third-party, which we denote as the *notary*. Embedding and recovering watermarks without clarifying the role of the notary is insufficient for IPR. Top-level protocols, follow which all parties involved in IPR (the owner, the adversary, and the notary) operate, are indispensable. The configuration of the three parties' functionalities varies in different scenarios, so it is necessary to design a specialized protocol for each case. We present practical protocols for three important real-world scenarios:

- An owner proves its ownership over a DNN to a notary.

- Collaborated owners in FL prove their ownership over a DNN to a notary, during which they can recover each other's identity proof and trace potential traitors.
- An owner transfers the IP of its DNN to a third party.

For watermarking schemes, these protocols introduce extra security requirements besides those listed in Section 2.

3.1 Protocols for ownership proof

The centralized OV protocol

The simplest OV protocol is centralized, in which the notary is a verification center responsible for publishing legitimate ownership proofs. This is the case which most established watermarking schemes have assumed. It involves two steps:

1. The owner submits $(\texttt{key}, \texttt{verify})$ and the access to M to the notary.
2. The notary computes $\texttt{verify}(M, \texttt{key})$ and publishes the output.

To use white/black-box watermarking schemes, the owner has to provide the white/black-box access of the suspicious model to the notary. To preserve privacy, the channel between the owner and the notary has to be encrypted. As for a curious notary, a Secure Multi-Party Computation (SMPC) [Bogetoft *et al.*, 2009] protocol can be adopted to protect the owner's data. Using such a protocol, the redeclaration problem can be solved. Instead of generating `key` on its own, the owner queries the notary for time authorization, who would return a key containing the time-stamp back to the owner. Overwriting and redeclaring cannot falsify the time-stamp so the ownership is secured.

Despite its simplicity, this protocol has many defects:

- The proof is valid only within the community that recognizes the notary's credit. It is difficult to accommodate this protocol for a broader range of entities.
- If the notary is compromised then all verifications within the community are at risk.
- Attacks against centralized protocols, such as the Deny Of Service (DOS) can paralyze the protocol.

The decentralized OV protocol

Given the defects of the centralized protocol, we propose a decentralized protocol for OV [Li and Wang, 2021]. Instead of relying on a verification center, we resort to a community of agents distributed across the network. To prove its ownership over a DNN, the owner broadcasts the necessary evidence to the verification community. Then each agent can volunteer to conduct the verification and broadcast the result. The OV is finished by voting through the entire community. To solve the redeclaration dilemma, an owner has to broadcast the hash of the DNN architecture and the evidence under a consensus protocol [Ongaro and Ousterhout, 2014]. Then the entire community would have a consensus on the time-stamp corresponding to ownership. This protocol is outlined in Algo. 3. Its unforgeability and correctness can be reduced to the security of `WM`, that of the digital signature scheme, and the reliability of the consensus protocol.

As in other distributed service systems [Mengelkamp *et al.*, 2018], to motivate the entire community to conduct OV, each correct verification assigns credits to agents that contribute to the proof, with which they can initiate their OV requests. This protocol is immune to attacks that only compromise a single agent. However, the communication traffic is increased. Especially when the owner adopts a white-box watermarking scheme, then each agent has to download the entire model. Since the proof is done on many independent agents, using SMPC thoroughly would be expensive and inefficient. Therefore, an eavesdropping adversary may steal the evidence corresponding to the owner and its model. Then the adversary can *spoil* this specific watermark so the owner can no longer succeed in OV over the new model, this *spoil attack* is illustrated in Fig. 1.

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

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

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 $\texttt{hash}(\texttt{verify})$, and submits $\texttt{verify}(M, \texttt{key})$ to the community using the consensus protocol.

Discussion: the spoil attack

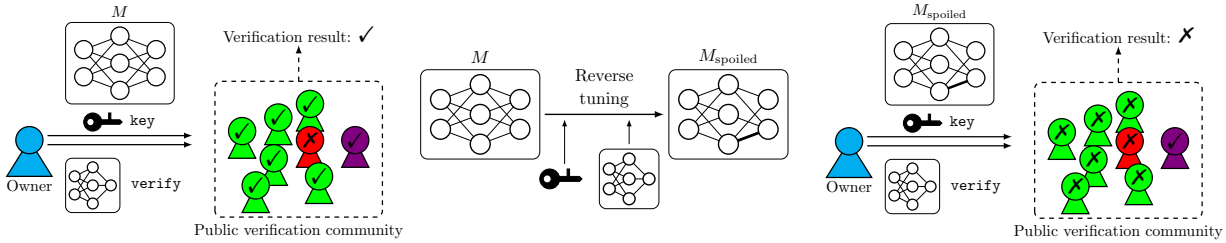
The spoil attack, as an additional threat in the decentralized OV protocol, has seldom entered the concern of designers of DNN watermarking schemes. Consequently, almost all established watermarking schemes are vulnerable against the spoil attack, which fact challenges the applicability of the decentralized OV protocol.

Backdoor-based watermarking schemes can be spoiled by fitting the DNN to randomly shuffled labels on the triggers. For white-box watermarking schemes, the adversary can spoil the watermark by tuning the model reversely.

Theoretically, the security against the spoil attack can be defined through Algo. 4. The scheme `WM` is secure against the spoil attack iff no efficient adversary can win $\texttt{Exp}^{\text{spoil}}_{\mathcal{A},\delta}$ with non-negligible probability for a given δ.

Such proof is intractable for almost all established watermarking schemes. It is unknown whether a scheme provably secure against the spoil attack exists or not.

As a substitute, we can improve traditional watermarking schemes against the spoil attack by simply embedding multiple watermarks into the DNN to be protected. Since each round of OV only exposes one watermark, such configuration can resist the spoil attack. But inserting multiple watermarks



(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.

Algorithm 4 $\texttt{Exp}^{\text{spoil}}_{\mathcal{A},\delta}$.

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

1: Generate M_{WM}, $\texttt{key}$ and $\texttt{verify}$ from $\texttt{WM}(M_{\text{clean}}, N)$.
2: $\mathcal{A}$ is given M_{WM}, $\texttt{key}$, N, $\texttt{WM}$ and $\texttt{verify}$.
3: $\mathcal{A}$ outputs M_{spoiled}.
4: $\mathcal{A}$ wins the experiment if $\texttt{verify}(M_{\text{spoiled}}, \texttt{key}) = 0$ and M_{spoiled}'s performance declines for no larger than δ compared with M_{WM}.

also calls for additional requirements, namely the watermarking capacity and independence.

As defined in [Li *et al.*, 2021], the $(\delta, \texttt{WM})$-*watermarking capacity* for a DNN model M, $\texttt{cap}^{\delta}_{\texttt{WM}}$, is the maximum number of watermarks that can be embedded and verified correctly before M's performance declines for δ. This OV service can survive $\texttt{cap}^{\delta}_{\texttt{WM}}$ rounds of spoil attacks by sacrificing the performance for at most δ.

The other aspect is: spoiling one watermark should not affect other ones. Otherwise, spoiling one watermark might invalidate others that have not been exposed and decrease the times of correct OV. To evaluate the *watermarking independence* against the spoil attack of $\texttt{WM}$ w.r.t. a DNN M, we firstly insert Q watermarks into M using $\texttt{WM}$. Then we spoil a random watermark and denote the number of watermarks that can still be correctly verified as r. The higher the *watermarking independence score* $\frac{r}{Q}$ is, the more robust $\texttt{WM}$ is against the spoil attack.

3.2 The OV protocol for federated learning

In the basic OV protocol, each DNN has a unique owner. The development of distributed learning paradigms, especially FL [Yang *et al.*, 2019], has changed this assumption. In FL, many parties coordinated by an aggregator cooperate to train one deep learning model without interchanging local data as illustrated in Fig. 2. Each participating party should be able to verify its ownership over the model independently. The privacy of each owner must not be breached, concretely, an owner can not falsify himself as another owner. Considering the collaboration of all owners, it is desirable that when one owner undergoes severe spoil attacks so its identity information is erased from the model, other owners can help it recover the ownership proof. Moreover, if one party betrays its co-authors, pirates the intermediate model and claims it to be its product, then the honest parties can correctly identify this traitor.

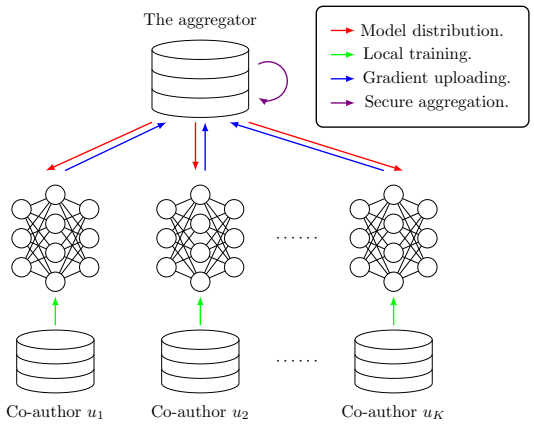


Figure 2: The client-server architecture for FL.

These four requirements, *independent verification*, *privacy-preserving*, *recovery*, and *traitor-tracing*, mark the characteristics for OV in FL. To reduce the communication traffic between the owners and the verification community, achieve the recovery property, and ensure traitor-tracing, a modified version of the basic decentralized OV protocol, $\texttt{Merkle-Sign}$, has been proposed [Li *et al.*, 2021]. Its representative features are:

- As shown in Fig. 3, in training, the aggregator embeds its key ($\texttt{key}_0$), a surveillance key ($\texttt{key}^{\dagger}_k$) into the intermediate model distributed to the k-th author to achieve traitor-tracing. When training terminates, the aggregator embeds the identity information of all authors into the model and broadcasts the hashed message as in the decentralized OV protocol.
- When broadcasting messages to the verification community, the hashed value of the identity proof for all owners are associated into a Merkle-tree [Li *et al.*, 2013] so owners can build correlations between their evidence to enable recovery.

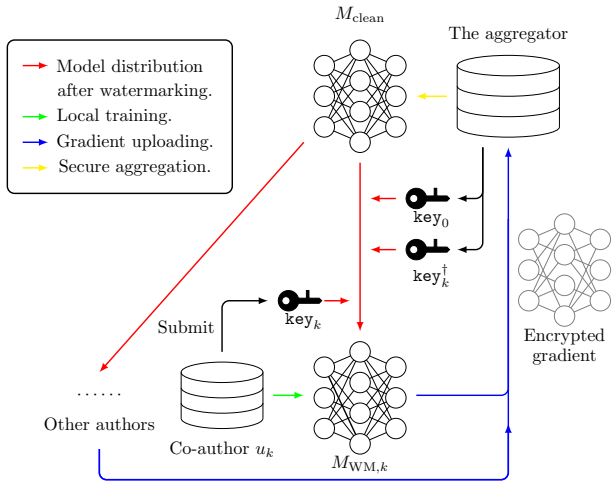


Figure 3: The `Merkle-Sign` watermarking framework for FL.

The analysis in [Li *et al.*, 2021] showed that the security of this protocol and four characteristics can be reduced to the computational hardness of the cryptological primitives within.

The watermarking scheme for `Merkle-Sign` also has to have a large watermarking capacity and independence. In addition, the embedding process is expected to be efficient and exert only slight modification to the entire DNN model. Otherwise, the model might fail to converge.

Discussion: aggregatable watermarks

In `Merkle-Sign`, the aggregator is in charge of embedding watermarks into the DNN model. Therefore this protocol needs a trusted aggregator and is not completely decentralized as in recent configurations of secure FL [Wei *et al.*, 2020].

To transfer the responsibility of watermark embedding from the aggregator to the owners, it is necessary that: the verification process remains valid to the aggregator's model aggregation. We denote the aggregator's combinator as $\mathcal{O}$, it can be model average, ensembling, etc. For K independent owners, this requirement can be formulated as:

$$\begin{aligned}
&\forall k, \texttt{key}_k \leftarrow \texttt{Gen}(1^N), M_{\text{WM},k} \leftarrow M_{\text{clean},k}, \\
&\quad (M_{\text{WM},k}, \texttt{verify}_k) \leftarrow \texttt{Embed}(M_{\text{WM},k}, \texttt{key}_k), \\
&M_{\text{WM}} \leftarrow \mathcal{O}(M_{\text{WM},1} \cdots M_{\text{WM},K}), \\
&\forall k, \Pr\{\texttt{verify}_k(M, \texttt{key}_k)1\} \geq 1 - \epsilon.
\end{aligned} \tag{1}$$

In which $M_{\text{clean},k}$ is the model distributed to the k-th author from the aggregator in each epoch. If the watermarking scheme `WM` satisfies (1) then we define it as an *aggregatable* scheme. An aggregatable watermarking scheme can improve `Merkle-Sign` into a completely decentralized protocol regarding both model training and OV.

3.3 The Protocol for DNN IP transfer

Apart from OV, IPR in DNN commercialization includes many other aspects, among which an important one is the transferring of DNN as IP. For the purchaser of a DNN, it is important that the deep learning model he paid for only contains his identity information. Otherwise, the seller might unilaterally cancel the transaction by redeclaring the ownership over this DNN from the seller's watermarks hidden within. Therefore, it is necessary to convince the purchaser that the sold DNN product is free from any watermark. Concretely, such convincing is possible only if there exists an algorithm $\mathcal{P}$ that can win the experiment defined in Algo. 5 with probability one.

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

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

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

Notice that in $\texttt{Exp}_{\mathcal{P}}^{\text{clean}}$, $\mathcal{P}$ is given neither `key` nor `verify` since the seller might hide them from the purchaser. We assumed that the watermarking scheme and the security parameter have been agreed on within the community where the transaction takes place. To conduct a DNN IP transfer the purchaser runs $\mathcal{P}$ on the model transmitted by the seller. If the output is zero then the purchaser is convinced that the sold model is free from any watermark. Then the purchaser can treat this model as its M_{clean}, deploy services by it, and protect it as his IP by protocols in Section 3.1.

It is remarkable that the existence of a distinguisher $\mathcal{P}$ winning $\texttt{Exp}_{\mathcal{P}}^{\text{clean}}$ is contradictive to the fundamental property of covertness defined in Section 2.3. So a watermarking scheme designed for one purpose is not necessarily a option in another scenario.

4 Experiments and Discussions

The evaluation of watermarking schemes w.r.t. basic security requirements has been presented in [Chen *et al.*, 2018]. To examine the adaptivity of current watermarking schemes to the real-world settings and corresponding protocols, we are interested in additional requirements listed in Table 1.

4.1 Settings

We adopted ResNet-50 [He *et al.*, 2016] as the backbone DNN architecture. Experiments were conducted on three datasets: MNIST [Deng, 2012], CIFAR10, and CIFAR100 [Krizhevsky *et al.*, 2009]. To evaluate the adaptivity of established DNN watermarking schemes to the presented protocols, we considered five candidates: Uchida's, random trigger (`Rand`), `Wonder Filter` (`WF`), `ATGF`, and `MTL-Sign` (`M-S`). In Uchida's, we adopted $U = 20$. For random trigger and `WF`, we adopted the configuration in [Zhang *et al.*, 2018] and [Li *et al.*, 2019]. As for `ATGF`

Protocol \ Metric	Performance decline due to the spoil attack **(A)**	Watermark capacity **(B)**	Watermark independence score **(C)**	Time consumption of watermark embedding **(D)**	Performance decline in FL due to watermarking **(E)**
Decentralized OV	✓	✓	✓	–	–
`Merkle-Sign`	✓	✓	✓	✓	✓
DNN IP transferring	✓	×	✓	–	–

Table 1: Additional security requirements. ✓denotes necessity, – denotes irrelevance, and × denotes negativity.

and `MTL-Sign`, the initialization in [Li *et al.*, 2021] and [Li and Wang, 2021] were used. All experiments were conducted under the `PyTorch` framework.

4.2 Evaluations of extra security requirements

The metric **(A)** reflects the damage of the spoil attack to the DNN model. The higher **(A)** is, the less likely an adversary is willing to conduct a spoil attack. Metrics **(B)**, **(C)**, **(D)**, and **(E)** have been introduced in Section 3. We conducted spoil attacks against five watermarking schemes as described in Section 3.1. To compute **(B)**, we adopted δ as the error rate of classification of the clean model. To compute **(C)**, we adopted $Q = 50$. To compute **(E)**, we included 200 independent authors in FL, and the aggregator used the model average for DNN model combination. The evaluations of **(A)** to **(E)** in all datasets are presented in Table 2, 3, and 4. The optimal scheme w.r.t. each metric is highlighted.

Metric \ Scheme	Uchida's	`Rand`	`WF`	`ATGF`	`M-S`
(A)	0.1%	0.0%	0.0%	0.0%	**0.7%**
(B)	≥**1,000**	111	194	117	≥**1,000**
(C)	94.1%	30.2%	41.3%	**94.3%**	79.5%
(D)	21ms	312ms	320ms	303ms	**750ms**
(E)	0.1%	0.3%	**0.0%**	**0.0%**	**0.0%**

Table 2: Evaluation of extra security requirements w.r.t. MNIST.

Metric \ Scheme	Uchida's	`Rand`	`WF`	`ATGF`	`M-S`
(A)	0.2%	0.1%	0.1%	0.2%	**4.5%**
(B)	≥**1,000**	312	473	300	≥**1,000**
(C)	**95.3%**	41.0%	36.1%	90.4%	78.0%
(D)	20ms	321ms	336ms	300ms	**798ms**
(E)	**0.0%**	1.1%	1.3%	1.1%	0.3%

Table 3: Evaluation of extra security requirements w.r.t. CIFAR10.

4.3 Discussions

We observed that `M-S` is optimal regarding **(A)**, since spoiling this watermark would result in the largest decrease in the DNN's normal performance. For **(B)**, it is found that white-box schemes significantly outperformed backdoor-based ones. As for **(C)**, only one backdoor-based scheme, `ATGF`, had a comparable performance as white-box schemes. The embedding time **(D)** for `M-S` is the longest, followed by that for backdoor-based schemes. Uchida's is the easiest scheme regarding overwriting. All schemes had little impact on the convergence of the DNN model in FL **(E)**. Although Uchida's and `M-S` have met all the requirements of the decentralized OV protocol and `Merkle-Sign`, they are white-box schemes and the corresponding communication traffic is high.

Metric \ Scheme	Uchida's	`Rand`	`WF`	`ATGF`	`M-S`
(A)	0.2%	0.9%	0.7%	1.1%	**8.2%**
(B)	≥**1,000**	412	479	410	≥**1,000**
(C)	**98.2%**	21.2%	12.9%	90.4%	77.5%
(D)	19ms	458ms	433ms	495ms	**784ms**
(E)	**0.0%**	3.4%	5.6%	4.1%	0.3%

Table 4: Evaluation of extra security requirements w.r.t. CIFAR100.

Requirements from different protocols are sometimes contradictive against each other, e.g., the watermark capacity **(B)**. The first two protocols require the influence of the watermark to be as small as possible so many watermarks can be embedded into the DNN, in this case a large watermark capacity is desirable. While in DNN IP transferring, it is preferable that the watermark exerts large impact to the model so a clean model and a watermarked model can be accurately differentiated, so the watermark capacity is better to be small.

5 Conclusion

To explore the applicability of IPR for deep learning models by watermarking as a service, this paper studies three scenarios and presents their respective protocols. Our analysis shows that these protocols demand extra properties other than those discussed in designing ordinary watermarking schemes, among with some are even against each other. Moreover, empirical studies show that current watermarking schemes cannot meet all requirements in practical protocols. Therefore, it is necessary to formulate protocols for more real-world scenarios as well as to design watermarking schemes that meet new security properties.

Acknowledgements

This work presented in this paper was supported by National Natural Science Foundation of China (61771310).

References

[Adi *et al.*, 2018] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1615–1631, 2018.

[Bogetoft *et al.*, 2009] Peter Bogetoft, Dan Lund Christensen, Ivan Damgård, Martin Geisler, Thomas Jakobsen, Mikkel Krøigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, et al. Secure multiparty computation goes live. In *International Conference on Financial Cryptography and Data Security*, pages 325–343. Springer, 2009.

[Chen *et al.*, 2018] Huili Chen, Bita Darvish Rouhani, Xinwei Fan, Osman Cihan Kilinc, and Farinaz Koushanfar. Performance comparison of contemporary dnn watermarking techniques. *arXiv preprint arXiv:1811.03713*, 2018.

[Darvish Rouhani *et al.*, 2019] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. Deepsigns: an end-to-end watermarking framework for ownership protection of deep neural networks. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 485–497, 2019.

[Deng, 2012] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[Fan *et al.*, 2021] Lixin Fan, Kam Woh Ng, Chee Seng Chan, and Qiang Yang. Deepip: Deep neural network intellectual property protection with passports. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[Guo *et al.*, 2020] Jian Guo, He He, Tong He, Leonard Lausen, Mu Li, Haibin Lin, Xingjian Shi, Chenguang Wang, Junyuan Xie, Sheng Zha, et al. Gluoncv and gluonnlp: Deep learning in computer vision and natural language processing. *J. Mach. Learn. Res.*, 21(23):1–7, 2020.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Li and Wang, 2021] Fangqi Li and Shilin Wang. Secure watermark for deep neural networks with multi-task learning. *arXiv preprint arXiv:2103.10021*, 2021.

[Li *et al.*, 2013] Hongwei Li, Rongxing Lu, Liang Zhou, Bo Yang, and Xuemin Shen. An efficient merkle-tree-based authentication scheme for smart grid. *IEEE Systems Journal*, 8(2):655–663, 2013.

[Li *et al.*, 2019] Huiying Li, Emily Willson, Haitao Zheng, and Ben Y Zhao. Persistent and unforgeable watermarks for deep neural networks. *arXiv preprint arXiv:1910.01226*, 2019.

[Li *et al.*, 2021] Fangqi Li, Wang Shilin, and Alan Wee-Chung Liew. Towards practical watermark for deep neural networks in federated learning. *arXiv preprint arXiv:2105.03167*, 2021.

[Liu *et al.*, 2018] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 273–294. Springer, 2018.

[Lv *et al.*, 2020] Zhihan Lv, Liang Qiao, Jinhua Li, and Houbing Song. Deep-learning-enabled security issues in the internet of things. *IEEE Internet of Things Journal*, 8(12):9531–9538, 2020.

[Mengelkamp *et al.*, 2018] Esther Mengelkamp, Benedikt Notheisen, Carolin Beer, David Dauer, and Christof Weinhardt. A blockchain-based smart grid: towards sustainable local energy markets. *Computer Science-Research and Development*, 33(1):207–214, 2018.

[Ongaro and Ousterhout, 2014] Diego Ongaro and John Ousterhout. In search of an understandable consensus algorithm. In *2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14)*, pages 305–319, 2014.

[Uchida *et al.*, 2017] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 269–277, 2017.

[Wei *et al.*, 2020] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[Yang *et al.*, 2019] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[Zhang *et al.*, 2018] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 159–172, 2018.

[Zhang *et al.*, 2021] Jie Zhang, Dongdong Chen, Jing Liao, Weiming Zhang, Huamin Feng, Gang Hua, and Nenghai Yu. Deep model intellectual property protection via deep watermarking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[Zhu *et al.*, 2020] Renjie Zhu, Xinpeng Zhang, Mengte Shi, and Zhenjun Tang. Secure neural network watermarking protocol against forging attack. *EURASIP Journal on Image and Video Processing*, 2020(1):1–12, 2020.