**Algorithm 1: Server Aggregation**

**Inputs:** $\mathbf{w}'_g$: the global model of last FL training round; $\mathcal{W}_l$: the list of local models submitted by each client in the current FL training round.

**Variables:** $\mathcal{A}$: A FedAttacker instance initialized based on the FL configuration file; $\mathcal{D}$: A FedDefender instance that is initialized based on the FL configuration file.

```
Function $server\_aggregation(\mathcal{W}_l)$ begin
    $\mathcal{W}_l \leftarrow before\_aggregation\_process(\mathcal{W}_l, \mathbf{w}'_g)$
    $\mathbf{w}_g \leftarrow before\_aggregation\_process(\mathcal{W}_l, \mathbf{w}'_g)$
    return $after\_aggregation\_process(\mathcal{W}_l, \mathbf{w}_g)$
Function $before\_aggregation\_process(\mathcal{W}_l, \mathbf{w}'_g)$ begin
    if $\mathcal{A}.is\_attack\_enabled()$ then
        if $\mathcal{A}.is\_data\_reconstruction\_attack()$ then $\mathcal{A}.reconstruct\_data(\mathcal{W}_l, \mathbf{w}'_g)$ ;
        if $\mathcal{A}.is\_model\_poisoning\_attack()$ then $\mathcal{W}_l \leftarrow \mathcal{A}.poison\_model(\mathcal{W}_l, \mathbf{w}'_g)$;
    if $\mathcal{D}.is\_defense\_enabled()$ & $\mathcal{D}.is\_defense\_before\_aggregation()$ then
        $\mathcal{W}_l \leftarrow \mathcal{D}.defend\_before\_aggregation(\mathcal{W}_l, \mathbf{w}'_g)$
    return $\mathcal{W}_i$
Function $on\_aggregation\_process(\mathcal{W}_l, \mathbf{w}_g)$ begin
    if $\mathcal{D}.is\_defense\_enabled()$ & $\mathcal{D}.is\_defense\_on\_aggregation()$ then
        return $\mathcal{D}.defend\_on\_aggregation(\mathcal{W}_l, \mathbf{w}_g)$
    return $aggregate(\mathcal{W}_i)$
Function $after\_aggregation\_process(\mathbf{w}_g)$ begin
    if $\mathcal{D}.is\_defense\_enabled()$ & $\mathcal{D}.is\_defense\_after\_aggregation()$ then
        return $\mathcal{D}.defend\_after\_aggregation(\mathbf{w}_g)$
    return $\mathbf{w}_g$
```

**Algorithm 2: Client Training**

**Inputs:** $dataset$: the local dataset of a client.

**Variables:** $\mathcal{A}$: A FedAttacker instance initialized based on the FL configuration file;

```
Function $client\_training(dataset)$ begin
    if $\mathcal{A}.is\_attack\_enabled()$ & $\mathcal{A}.is\_data\_poisoning\_attack()$ then
        $dataset \leftarrow \mathcal{A}.poison\_data(dataset)$
    $\mathbf{w}_l \leftarrow train(dataset)$
    $send\_to\_server(\mathbf{w}_l)$
```