

The background is a dark blue gradient. On the left, there is a circular inset showing a close-up of a circuit board with various components. A magnifying glass is positioned over this inset. Above the magnifying glass, there are two overlapping geometric shapes: a blue parallelogram and a light green parallelogram. In the top right corner, there is a faint, stylized pattern of interconnected lines and squares, resembling a circuit or a data network.

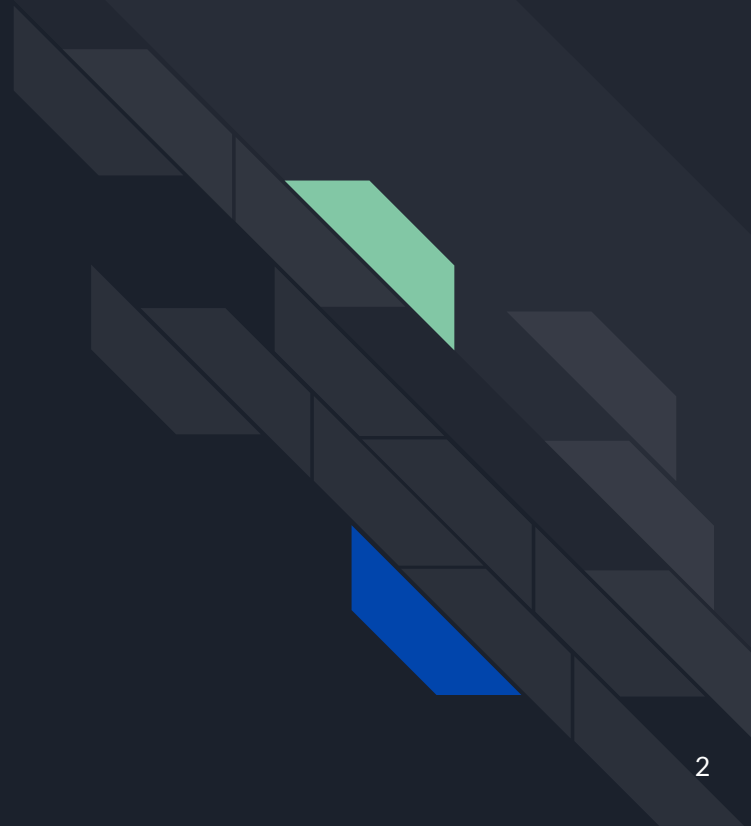
Deep Leakage from Gradients

Faisal Mohamed

Core Idea

Implementation

Results



Core Idea

Implementation

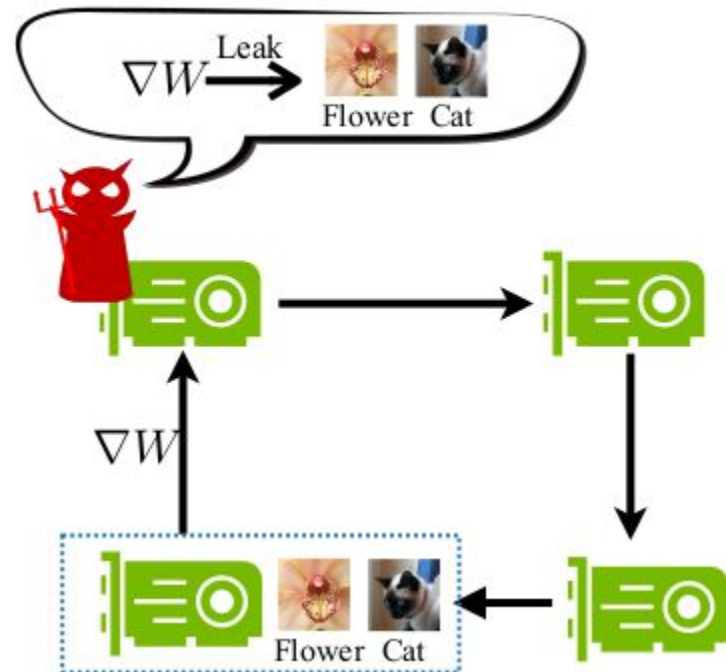
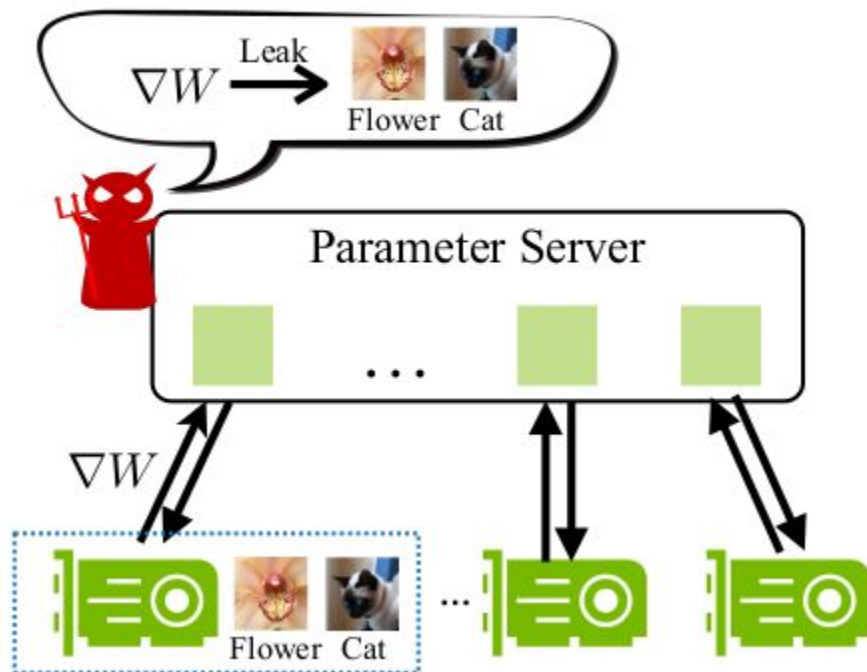
Results



Core Idea

- Obtain private training data from publicly shared gradients.
- Raise awareness about the safety of sharing gradients.

Core Idea



Core Idea

Implementation

Results

Implementation

Normal Participant



Differentiable Model
 $F(x, W)$

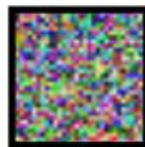
$Pred$

$Loss$

$[0, 1, 0]$

∇W

Malicious Attacker 



Differentiable Model
 $F(x', W)$

$\nabla W'$

Try to match

$Pred'$

$Loss'$

$[0.2, 0.7, 0.1]$

$\partial \mathbb{D} / \partial X$

$$\mathbb{D} = \|\nabla W' - \nabla W\|^2$$

$\partial \mathbb{D} / \partial Y$

Implementation

Algorithm 1 Deep Leakage from Gradients.

Input: $F(\mathbf{x}; W)$: Differentiable machine learning model; W : parameter weights; ∇W : gradients calculated by training data

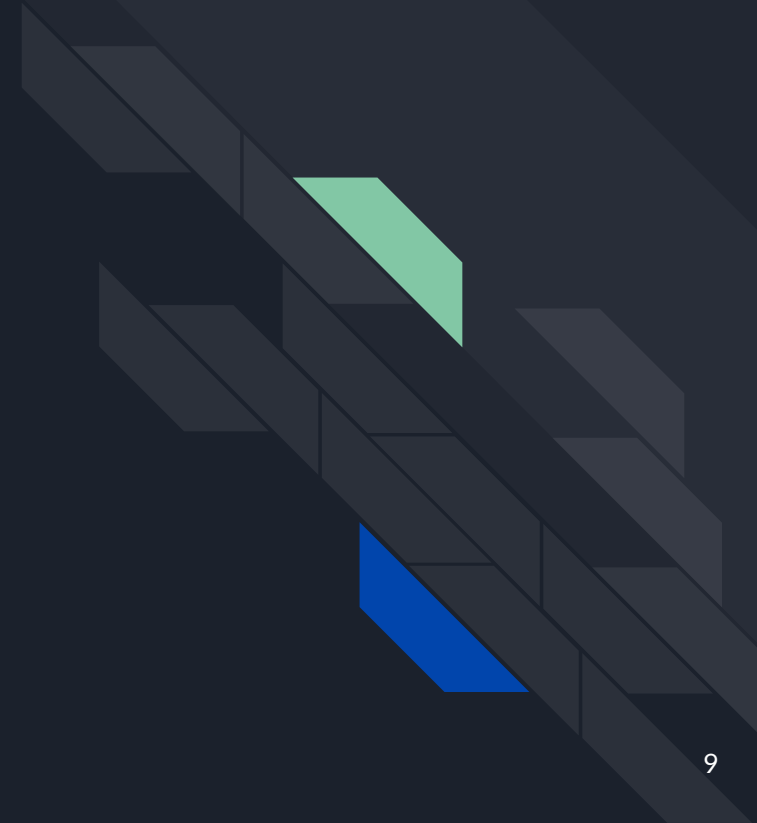
Output: private training data \mathbf{x}, \mathbf{y}

```
1: procedure DLG( $F, W, \nabla W$ )  
2:    $\mathbf{x}'_1 \leftarrow \mathcal{N}(0, 1), \mathbf{y}'_1 \leftarrow \mathcal{N}(0, 1)$  ▷ Initialize dummy inputs and labels.  
3:   for  $i \leftarrow 1$  to  $n$  do  
4:      $\nabla W'_i \leftarrow \partial \ell(F(\mathbf{x}'_i, W_t), \mathbf{y}'_i) / \partial W_t$  ▷ Compute dummy gradients.  
5:      $\mathbb{D}_i \leftarrow \|\nabla W'_i - \nabla W\|^2$   
6:      $\mathbf{x}'_{i+1} \leftarrow \mathbf{x}'_i - \eta \nabla_{\mathbf{x}'_i} \mathbb{D}_i, \mathbf{y}'_{i+1} \leftarrow \mathbf{y}'_i - \eta \nabla_{\mathbf{y}'_i} \mathbb{D}_i$  ▷ Update data to match gradients.  
7:   end for  
8:   return  $\mathbf{x}'_{n+1}, \mathbf{y}'_{n+1}$   
9: end procedure
```

Core Idea

Implementation

Results



Results

