**Main contribution**

New insights on Byzantine-tolerant distributed SGD:
- break 2 popular Byzantine-tolerant SGD: coordinate-wise median, and Krum
- attacks with theoretical and empirical guarantees
- a revised definition of Byzantine tolerance for SGD

**Byzantine failures in distributed SGD**

\[
\min_{x \in \mathbb{R}^d} F(x), \quad \text{where } F(x) = \mathbb{E}_{z \sim \mathcal{D}}[f(x; z)].
\]

- \(m\) workers, \(q\) of them are Byzantine, \(q < \frac{m}{2}\)
- Distributed SGD:

\[
x^{t+1} = x^t - \gamma \text{Aggr}\{g_i(x^t) : i \in [m]\},
\]

where Aggr\((\cdot)\) is an aggregation rule (e.g., averaging),

\[
g_i(x^t) = \begin{cases} \text{Arbitrary } & i\text{th worker is Byzantine}, \\ \nabla F_i(x^t) & \text{otherwise.} \end{cases}
\]

For distributed SGD:
- Byzantine failure: loss gets arbitrarily bad
- Traditional Byzantine tolerance: bounded error \(\|\text{Aggr}(\cdot) - \nabla F(x)\|

**Inner product manipulation**

Main idea:
- If the attacker can guarantee the ascent of loss in each iteration, then, after sufficient number of iterations, the loss gets arbitrarily bad
- Negative inner product guarantees the ascent of loss:

\[
\langle \text{Aggr}(\cdot), \nabla F(x) \rangle < 0
\]

- Bounded error of traditional Byzantine tolerance:
  - only guarantee Byzantine tolerance for single iteration
  - does not consider the entire training procedure

A revised version of Byzantine tolerance for distributed SGD:

**Definition 1.** Without loss of generality, suppose that in a specific iteration, the server receives \((m-q)\) correct gradients \(\mathcal{V} = \{v_1, \ldots, v_{m-q}\}\) and \(q\) Byzantine gradients \(\mathcal{U} = \{u_1, \ldots, u_q\}\). We assume that the correct gradients have the same expectation \(E[v_i] = g, \forall i \in [m-q]\). An aggregation rule Aggr\((\cdot)\) is said to be DSSGD-Byzantine-tolerant if

\[
(g, E[\text{Aggr}(\mathcal{V} \cup \mathcal{U})]) \geq 0.
\]

Using the new definition, we can easily attack median and Krum.

In short:
- Large negative Byzantine gradients breaks median
- Small negative Byzantine gradients breaks Krum

**Attacking coordinate-wise median**

- Assume that \(m - 2q = 1\)
- Assume that the correct stochastic gradients are IID
- Assume that the variance of stochastic gradients is lower-bounded
- When \(\max_{i \in [d]} |g_i| < \sqrt{\frac{\epsilon}{m}}\), we show that there exist Byzantine gradients \(\mathcal{U}\) such that \(g, E[\text{Median}(\mathcal{V} \cup \mathcal{U})]) < 0\)
- Toy example: \(\mathcal{V} = \{-0.1, 0.1, 0.3\}\) with the mean 0.1, \(\mathcal{U} = \{-4, -2\}\), Median\((\mathcal{U} \cap \mathcal{V})\) = -0.1

**Attacking Krum**

- Assume \(m - 2q = 3\) (worst case of Krum)
- The correct stochastic gradients are IID
- Assume that the variance of stochastic gradients is lower-bounded and upper-bounded
- Take \(u_1 = u_2 = \cdots = u_q = -\epsilon\), where \(\epsilon\) a positive and small enough
- When \((m-q)\) is large enough, we show that \(g, E[\text{Krum}(\mathcal{V} \cup \mathcal{U})]) < 0\)
- Toy example: \(\mathcal{V} = \{0, 0.02, 0.14, 0.26, 0.38, 0.5\}\) with the mean 0.2167, \(\mathcal{U} = \{-0.1, -0.1, -0.1\}\), Krum\((\mathcal{U} \cap \mathcal{V})\) = -0.1

**State-of-the-art defense techniques**

**Coordinate-wise median:** take median on each coordinate

**Krum:** minimize the sum of Euclidean distances of the neighbours

\[
\text{Krum}(\{\tilde{v}_i : i \in [m]\}) = \tilde{v}_k, \quad k = \arg \min_{i \in [m]} KR(\tilde{v}_i),
\]

\[
KR(\tilde{v}_i) = \sum_{j \neq i} ||\tilde{v}_i - \tilde{v}_j||^2,
\]

where \(i \rightarrow j\) are the indices of the \(m - q - 2\) nearest neighbours of \(\tilde{v}_i\) in \(\{\tilde{v}_j : j \in [m], i \neq j\}\) as measured by squared Euclidean distance.

**Experiments**

- CIFAR-10 image classification dataset
- CNN with 4 convolutional layers followed by 1 fully connected layer
- Median:

\[
\begin{align*}
& \text{CIFAR-10:} \quad m = 25, \quad q = 12, \\
& \text{CNN:} \quad u_1 = u_2 = \cdots = u_{12} = -\frac{1}{13} \sum_{i=1}^{13} v_i \\
& \text{Fail when } \epsilon \text{ is large}
\end{align*}
\]

- Krum: \(m = 25, \quad q = 11\)

\[
\begin{align*}
& \text{Krum:} \quad m = 25, \quad q = 12, \\
& \text{CNN:} \quad u_1 = u_2 = \cdots = u_{11} = -\frac{1}{12} \sum_{i=1}^{11} v_i \\
& \text{Fail when } \epsilon \text{ is small}
\end{align*}
\]

**Related work**