Programming Model

MapReduce Programming Model.
- Workers: including mappers and reducers. Contribute computation resources to execute tasks assigned by master
- Master: Job management, task scheduling and load balancing among workers

Adversary Model.
- Lazy Cheaters: The goal is to save computation resources and maximize their profit by performing more tasks in the same period of time
- Malicious Cheaters: The behavior of a malicious worker is either arbitrary or strategic.

Result Verification

A. Watermark Injection
- The basic idea of watermark injection is to inject indistinguishable marks into normal documents.
- Assuming that if the marks we injected are all computed correctly, the workers have executed the tasks honestly. Otherwise, workers, one or many, have not conducted the task honestly.

B. Random Sampling
- Classic PageRank Model
  \[ PR(i) = \frac{1 - d}{N} + d \sum_{u \in B(i)} \frac{PR(u)}{L(u)} \]
- The PageRank values of the in-nodes of the sampled nodes and the out-degree of these in-nodes need to be correct as well.

Experimental Analysis

Watermark Injection
- \( P_1 \) - percentage of documents with injected watermarks; \( p_2 \) - proportion of words in each documents encoded with watermarks
- Among all the 16801 result points, only less than 0.5% of them have a detection rate less than 95%
- More than half of the simulations have their detection rate larger than 99%

Random Sampling
- Intentional miscalculations of PageRank values were 1% to 5%, represented by five different colors as shown
- Detection rates increase dramatically with lower sampling ratios, in particular when sampling ratios are less than 0.15% in the weighted approach or less than 0.3% in the naïve-based method
- Weighted experiments reach higher detection rate of over 80%