1. PROBLEM

- A Denial of Service (DoS) attack is an attempt to make a resource unavailable to its intended users. When the attack is originated from multiple hosts that are distributed in the network, it is called Distributed Denial of Service (DDoS) attack.
- Cryptographic puzzles (or client puzzles) have been proposed to defend against DoS attacks by requiring clients to "pay" for their request via solving a moderately hard computational problem.

2. GOALS

- Use puzzle-based defense to stop DoS and DDoS attacks.
- Do useful work, while stopping the attack, by using puzzles that actually contribute to useful and meaningful services or applications.

3. SYSTEM ARCHITECTURE

- We propose productive puzzles based DoS mitigation system to achieve the goal doing useful work while mitigating DoS and DDoS attacks. A system that deploys productive puzzles involves 3 parties:
  - **Protected Server**: the party that is being protected against DoS attacks, for example, a Web server;
  - **DoS Protection Provider**: the productive puzzle based DoS mitigation system;
  - **Beneficiary**: a service or project that provides computational tasks/jobs that will be transformed into productive puzzles.

The overall architecture of the system and the relations between these parties are described below.

4. PRODUCTIVE PUZZLES

- A productive puzzle a set of tasks with known solutions, a set of unsolved tasks, IDs of these tasks, plus a keyed hash valued that is used as an integrity protection.
- The puzzle difficulty \( c \) is computed by the protected server as a function of server load and client's contribution to the load. Thus, clients that cause heavy load must solve harder puzzles.

5. APPLICATIONS

A wide range of applications, services, and scientific projects can potentially become the beneficiary of the proposed system. A short list of examples are:

- Distributed computing applications: indexing, distributed sort, document clustering, distributed data mining, graphics/map rendering, ...
- Volunteer computing projects: SETI@home, Folding@home, BURP, GIMPS, BLAST, ...
- Others: Geographical data processing, statistical machine learning, BitCoin, ...

Characteristics of applications that are amenable to the proposed computing model:

- Non real-time
- Task-level parallelism exist in the application
- Each task by itself is inherently sequential
- Tasks exhibit lower data to computing ratio