Mitigating DNSSec Deployment Risks Using Client Puzzles

Sunitha Beeram

George F. Riley

Georgia Tech, College of Engineering

Background

- DNSSec Sig0 Secure Dynamic Updates
  - Insure that update requests are legitimate
  - Uses public key RSA method for validation
  - Extra computational overhead
- Algorithmic Complexity Attacks
  - Denial of Service Attack
  - Overload server with excessive request, requiring excessive aggregate computation
- Client Puzzles
  - Can mitigate Algorithmic Complexity Attacks
  - Require requestors to solve computationally complex puzzles before requests are processed

Research Objectives

- Quantify overhead associated with SIG0 updates
  - Experimental testbed (not simulation)
  - Single Attacker, Single DNS server
- Demonstrate effectiveness of Algorithmic Complexity Attack
- Investigate applicability of Client-Puzzles to mitigate possible Algorithmic Complexity Attacks.

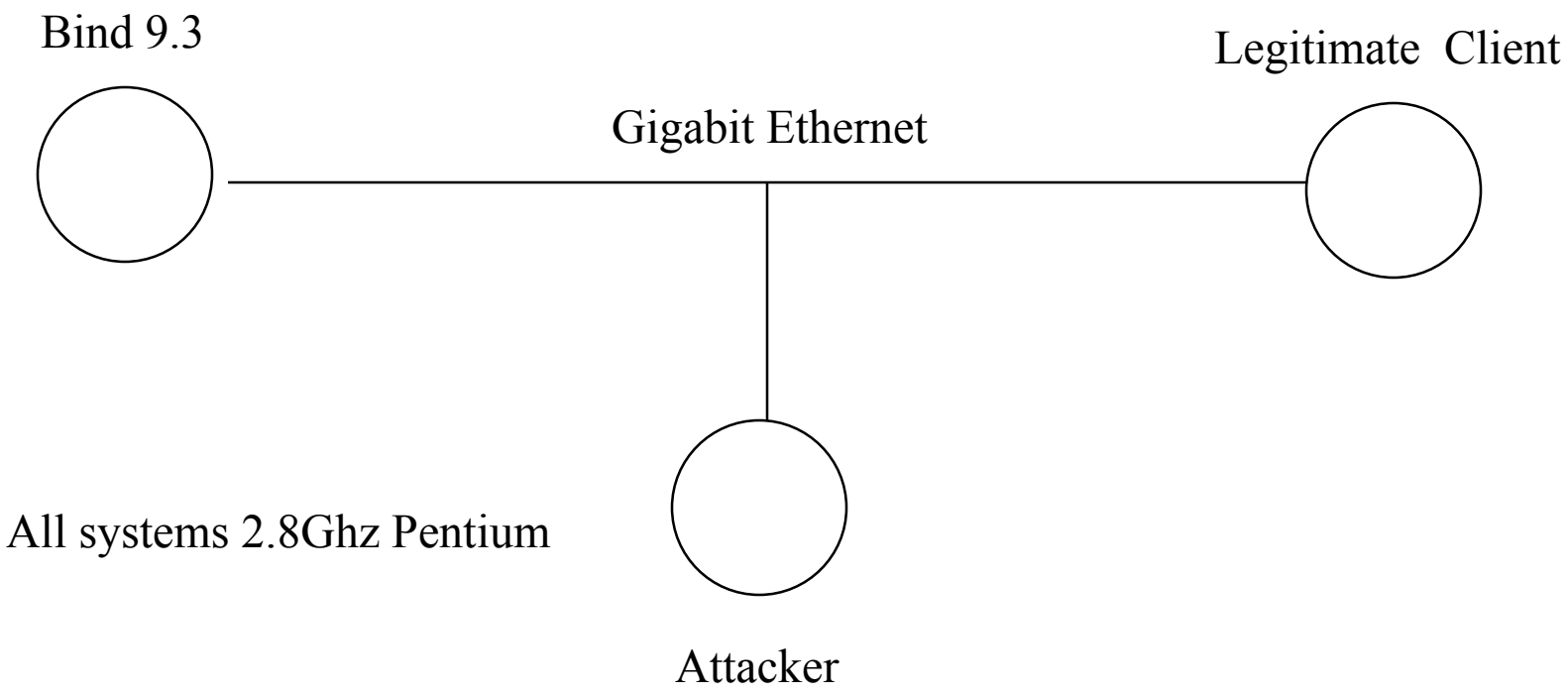
Threat Model

- Adversary can eavesdrop on all messages between client and server
  - Can copy and replay update requests
- Adversary can spoof IP addresses
  - Claim to be someone other than actual identity
- Adversary has one or more compromised systems from which to launch the attack.

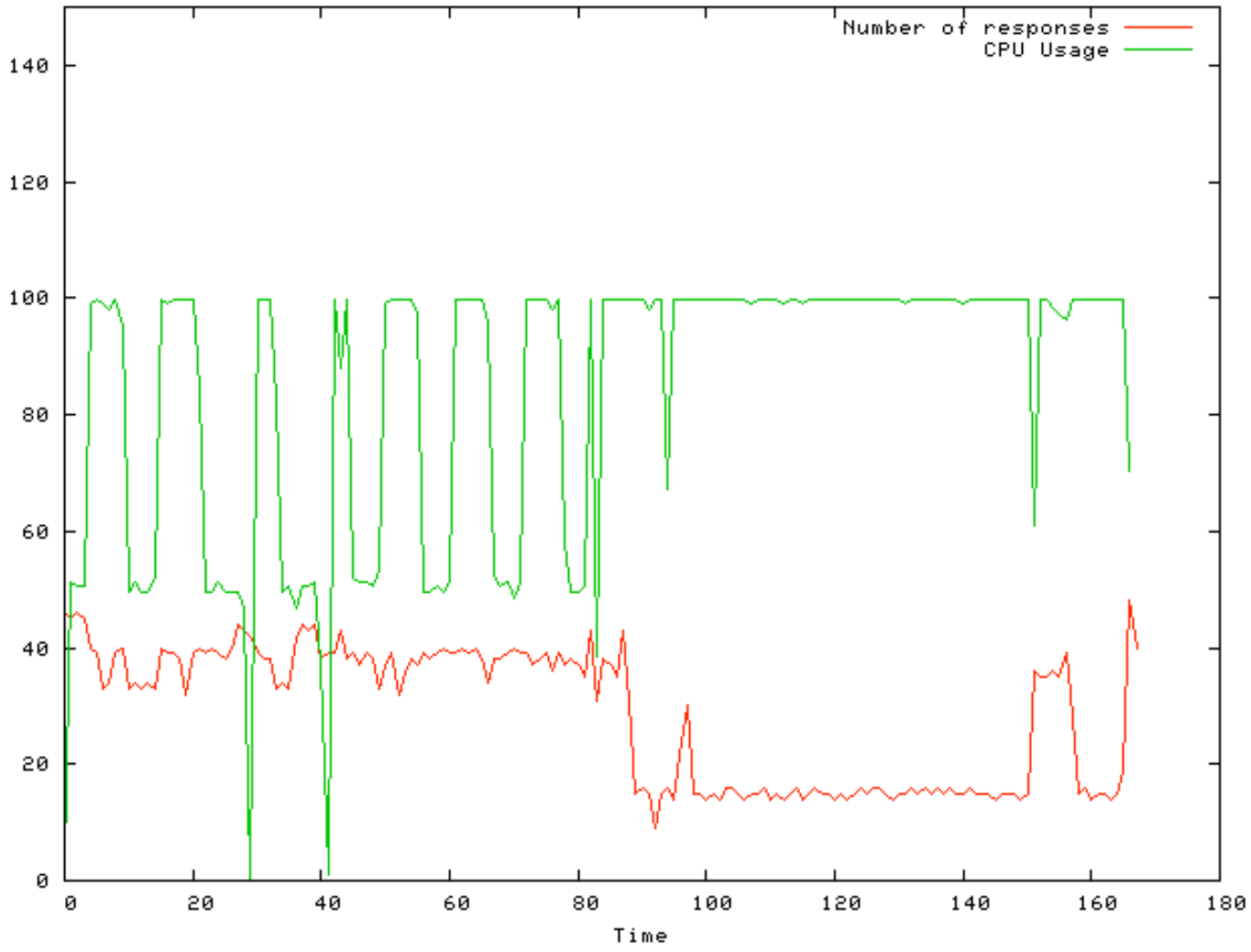
Attack Characteristics

- Bind 9.3.0s20021217
- 2.8 GHz Pentium Processor
- About 200µsec for RSA Signature Verification
- About 5000 verifications per second
- Signature Generation at Client is more expensive
  - Generate a few and replay them repeatedly
  - TTL defaults to 5 minutes

Experimental Setup



Results of Attack



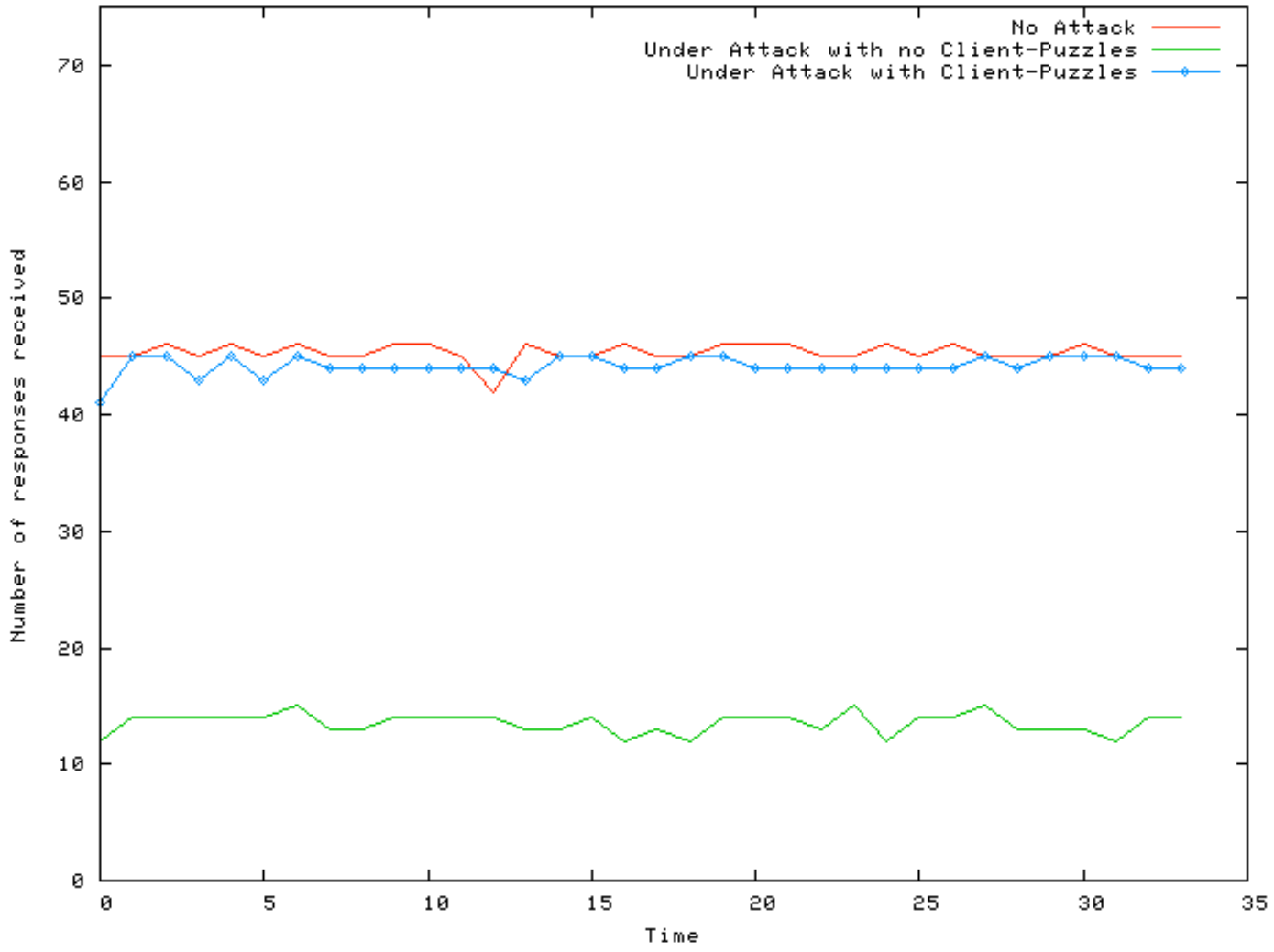
Client Puzzles

- Puzzle generation should easy for server
- Minimal state (near zero) on server
- Puzzle solutions easy to verify on server
- Puzzle solutions cannot be precomputed by client
- Puzzles solutions cannot be replayed by client
- Puzzle difficulty can be varied depending on load
- No puzzles at all when average load is low
- Many puzzle types exist
  - We used reverse hashing of variable number of bits

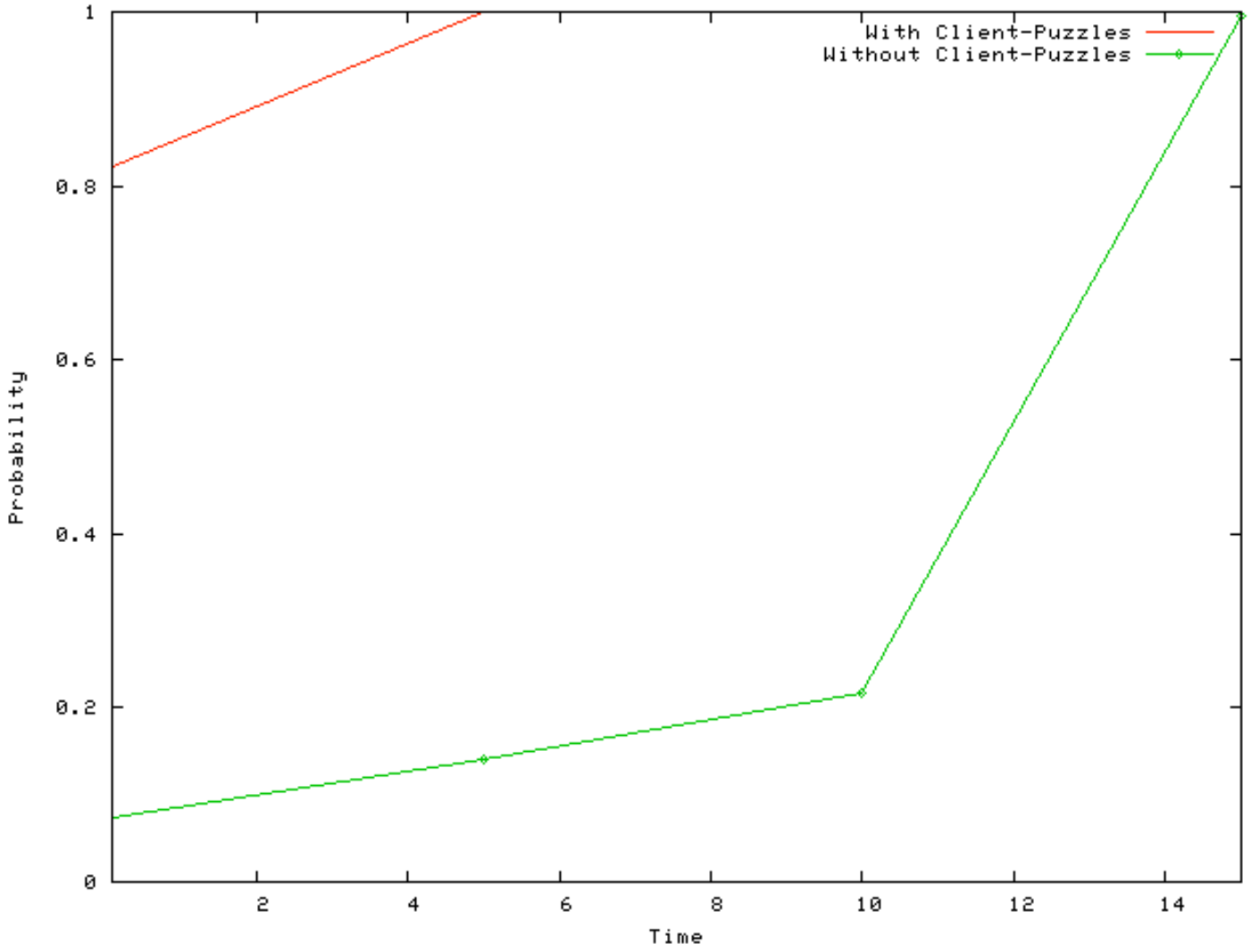
Maintaining Puzzle State at Server

- Use Bloom Filter to track *correctly solved* puzzles
- Bloom filter cleared periodically
  - Or use ping-pong filters
- If puzzle solution already in the filter:
  - Assume client is replaying previous request
  - Ask client to solve another puzzle
  - Effect of false positive minimal

Results with Puzzles Enabled



Results with Puzzles Enabled



Conclusions

- DNSSec deployment increases risk of Algorithmic Complexity Attack
- 1Gbps aggregate bandwidth for attacker is sufficient to overload DNSSec server
- Client Puzzles solution is viable
  - Minimal impact on legitimate clients
  - Easy to implement and deploy
- Bloom filter at server protects against replay attack

Questions?