Proxychain: Developing a Robust and Efficient Authentication Infrastructure for Carrier-Scale VoIP Networks

Italo Dacosta and Patrick Traynor

Georgia Tech College of Computing

GEORGIA TECH INFORMATION SECURITY CENTER

Performance, Scalability and Security

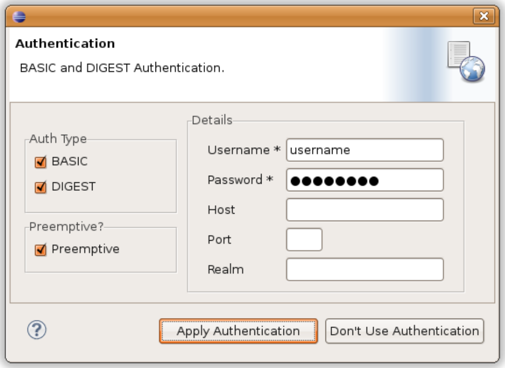
- Finding the right balance between performance /scalability and security is a well-known challenge
- Robust but computationally expensive security mechanisms are difficult to deploy in production environments
  - S-BGP, DNSSEC
- Weaker but more efficient security mechanisms are generally broken and abused
  - WEP, IKE Aggressive mode

Another Example: SIP Authentication

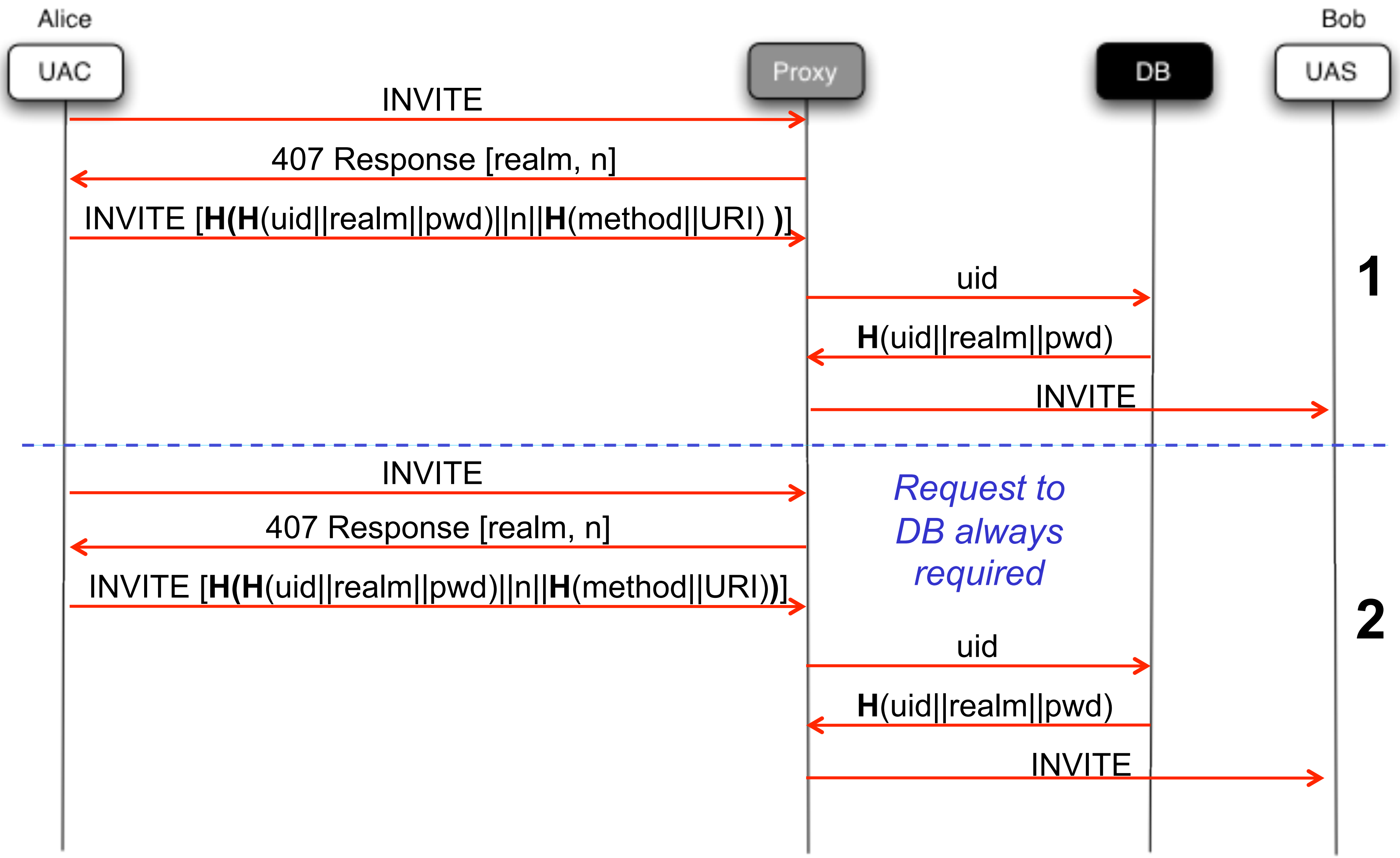
- Session Initiation Protocol (SIP)
  - Establishes, manages and terminates sessions between two or more clients
  - Generally associated with VoIP
- RFC 3261 recommends several security mechanisms: Digest authentication, SSL/TLS, IPsec and S/MIME
- However, **Digest authentication** is typically the only one employed
  - Weaker but more efficient

SIP Digest Authentication

- Challenge-response authentication protocol
- Based on cryptographic hash operations (MD5)
- De facto authentication mechanism in SIP



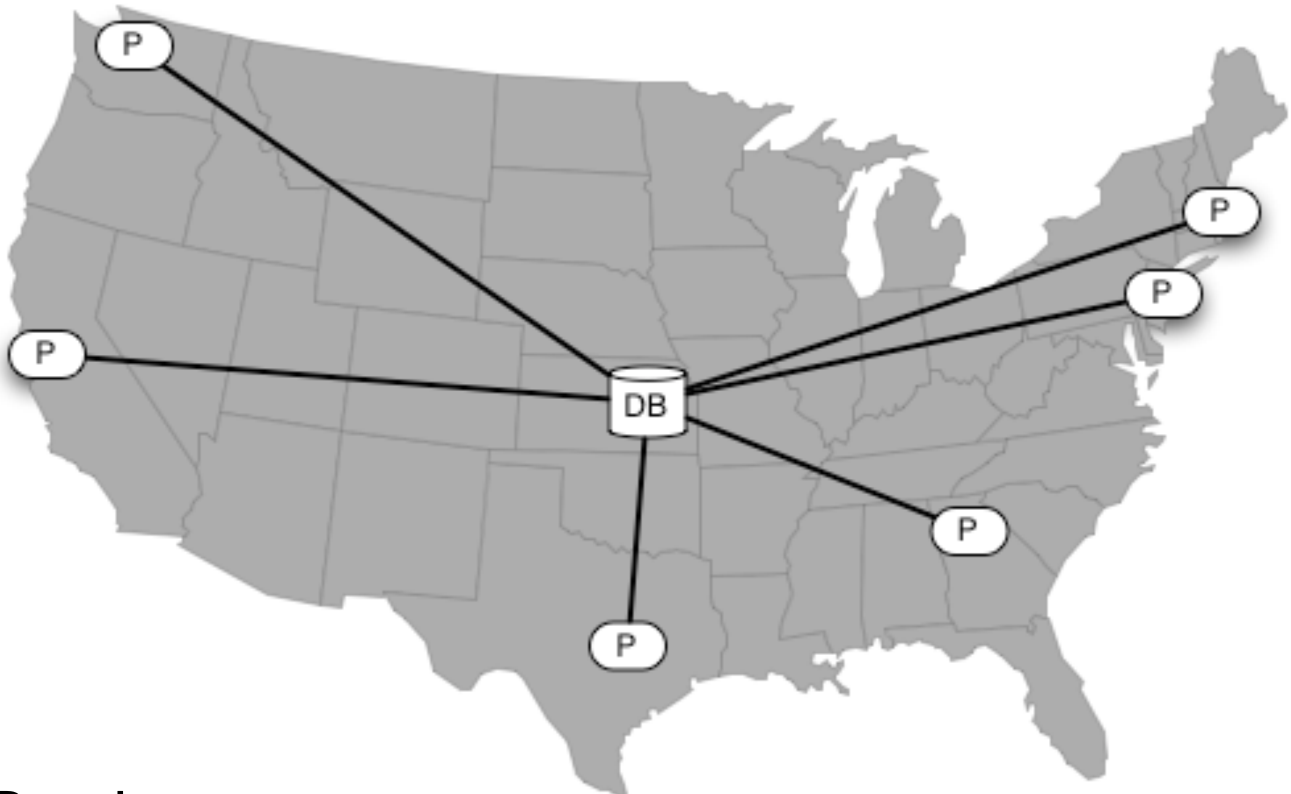
SIP Dialogs with Digest Authentication



Problems with Digest Authentication

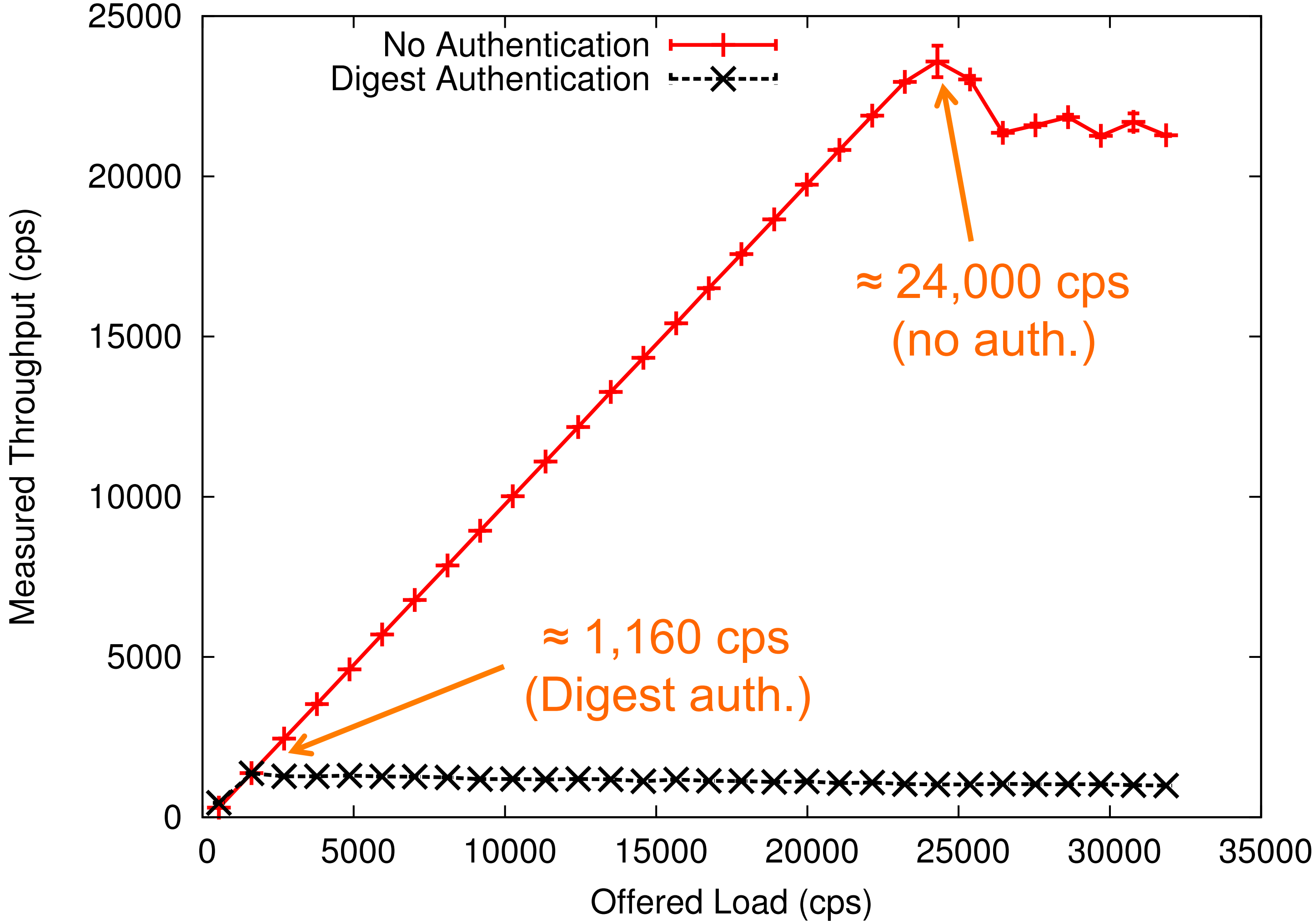
- Inefficient in scenarios with a remote authentication service or database
  - RTT added to each authentication operation
  - One request to the database per authenticated SIP message
  - High load in the database if it is shared by multiple SIP servers
- Considered a weak authentication protocol
  - E.g., No mutual authentication

Our Scenario: A Nationwide VoIP Provider



P = SIP Proxies
DB = Authentication database

The Problem: Digest Authentication Performance in Our Scenario



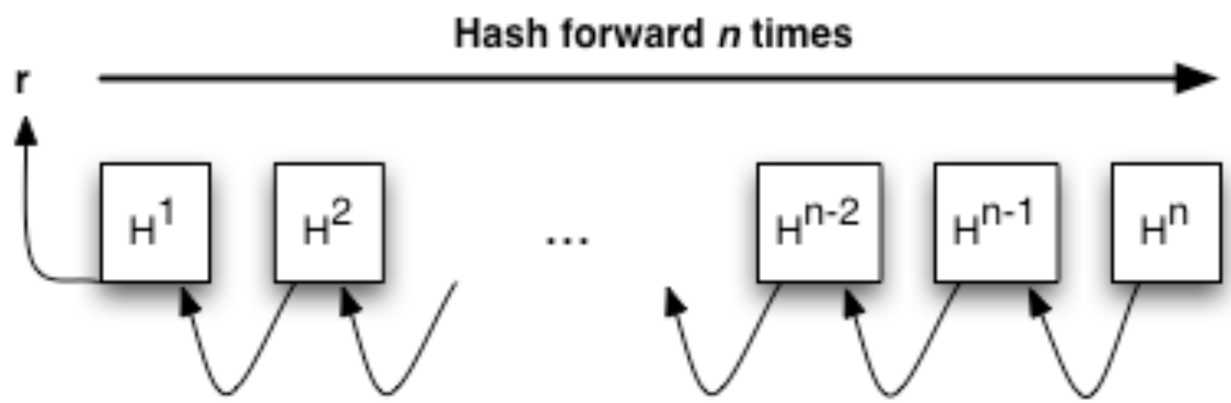
Our Proposed Solution

- Reduce the number of requests to the database by **caching temporary authentication credentials** in the proxies
- Use **hash chains** to build these temporary credentials
  - Take advantage of hash chains properties
- *Caching Digest auth. credentials reduces security!*

Hash Chains Background

- Sequence of **one-time** authentication tokens
- Created by applying a **cryptographic hash function** to a secret value **r** multiple times

$$H^n(r) = H(\ldots H(H(r))\ldots)$$



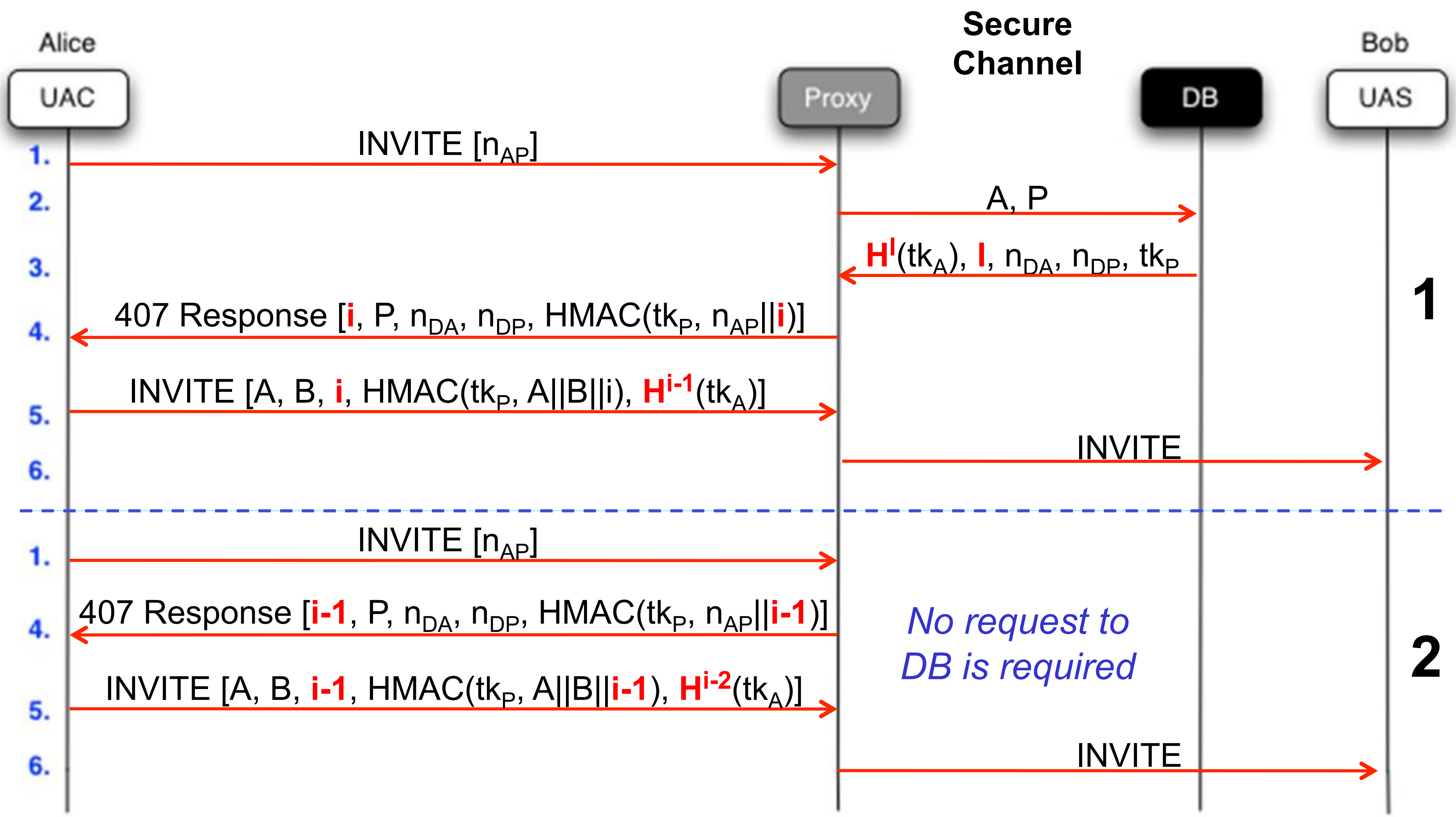
Methodology

- Design and implementation of new SIP authentication protocol: **Proxychain**
- Experimental evaluation
  - Call throughput
  - Bandwidth utilization
  - CPU utilization
- Results analysis

Proxychain Design Goals

- Efficiency
  - Faster authentication operations
- Scalability
  - Support larger number of users and proxies
- Security
  - Provide more security guarantees

Proxychain SIP Dialogs



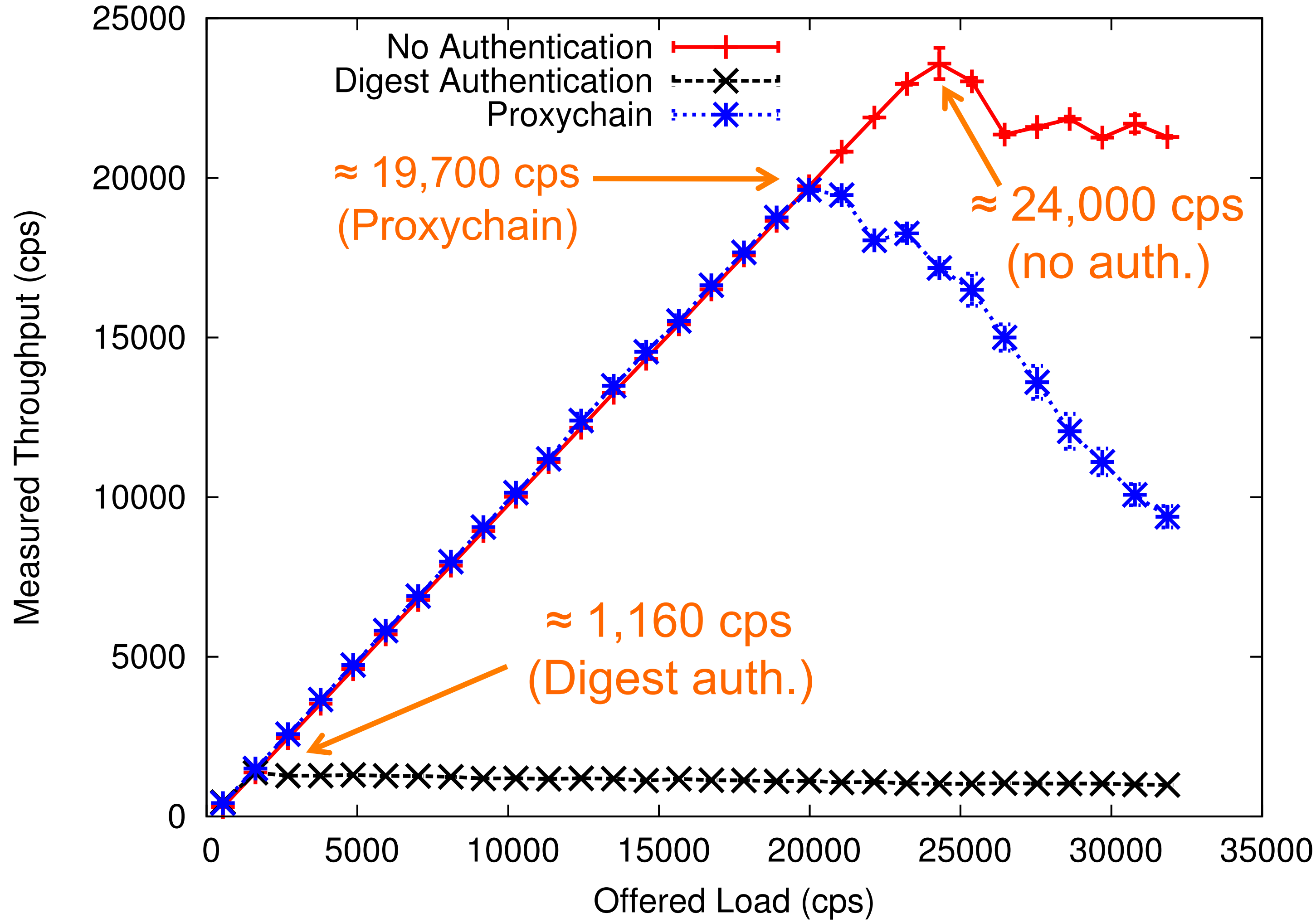
Proxychain implementation

- Modifications to proxy, database and client software
  - Implemented in C language
  - Relatively small when compared to original code base
- Total credential size (MD5): **134 bytes**
  - Only ≈26 MB of proxy's memory required for storing 200,000 users credentials

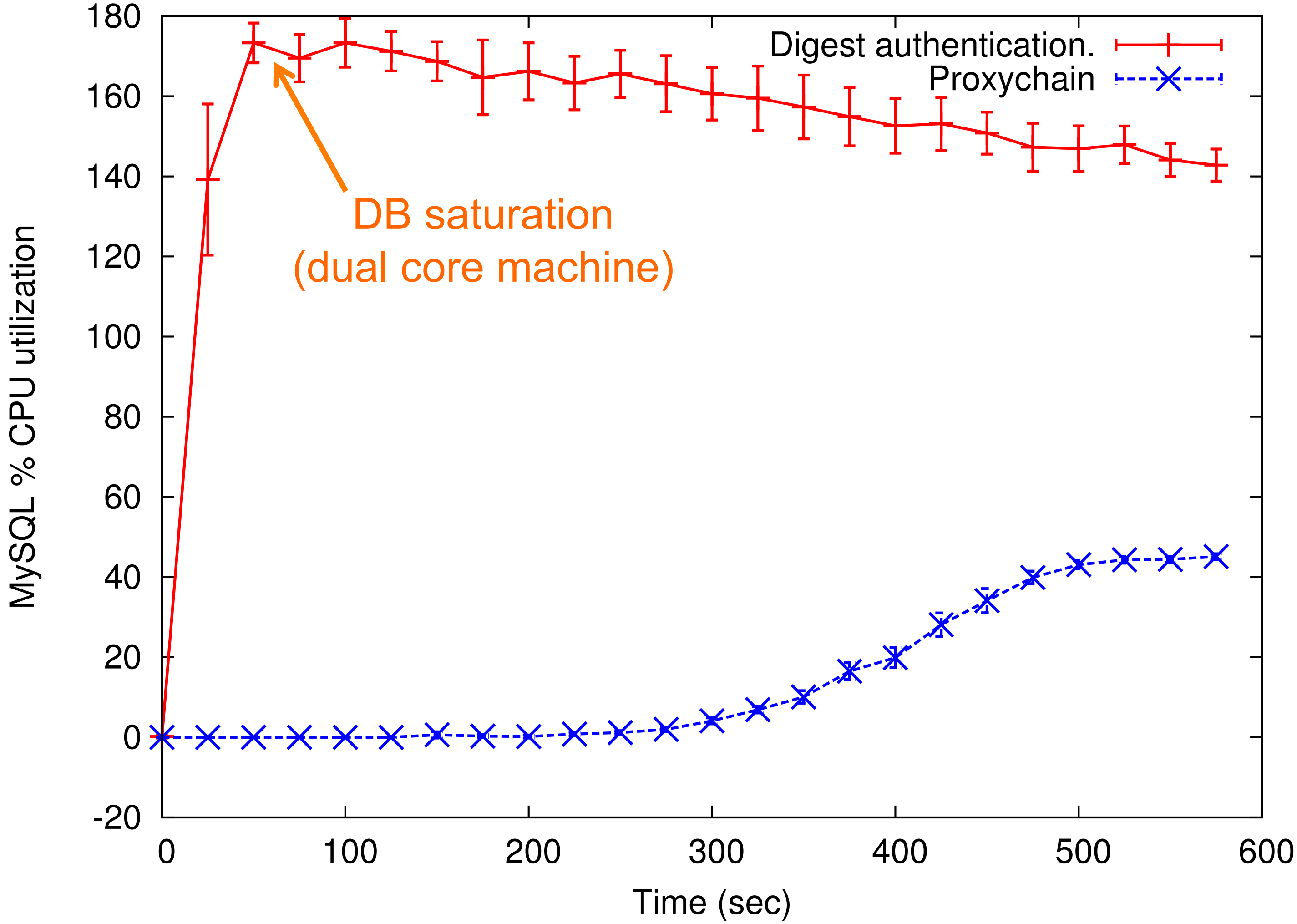
Experimental Setup

- **Planetlab** for obtaining real RTT values
- **GT Emulab testbed** for database and proxies
  - **OpenSIPS** for proxies
  - **MySQL** for the database
- Nine high-capacity servers for generating SIP call traffic
  - **SIPp** as the SIP traffic generator

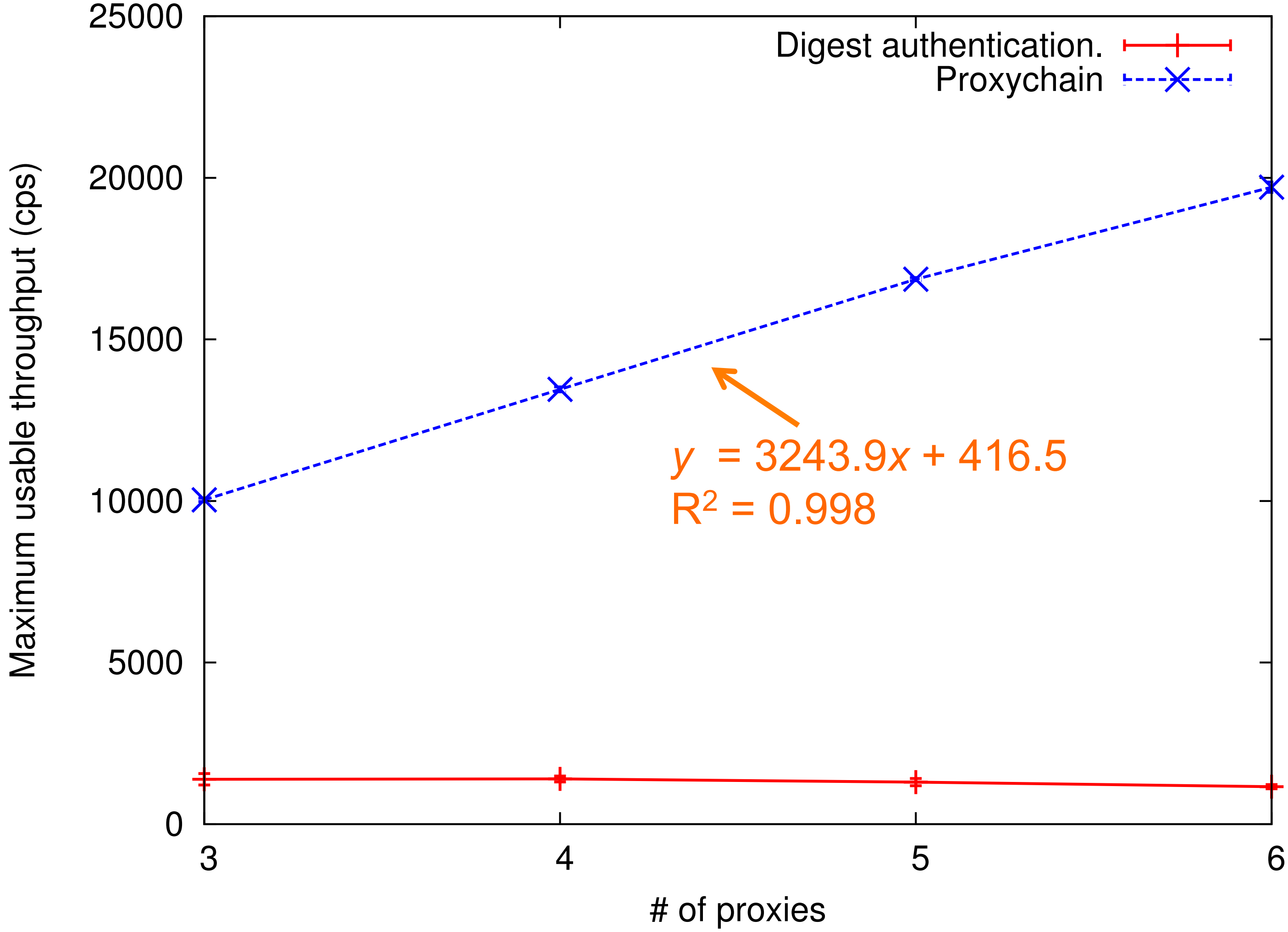
Results: Call Throughput



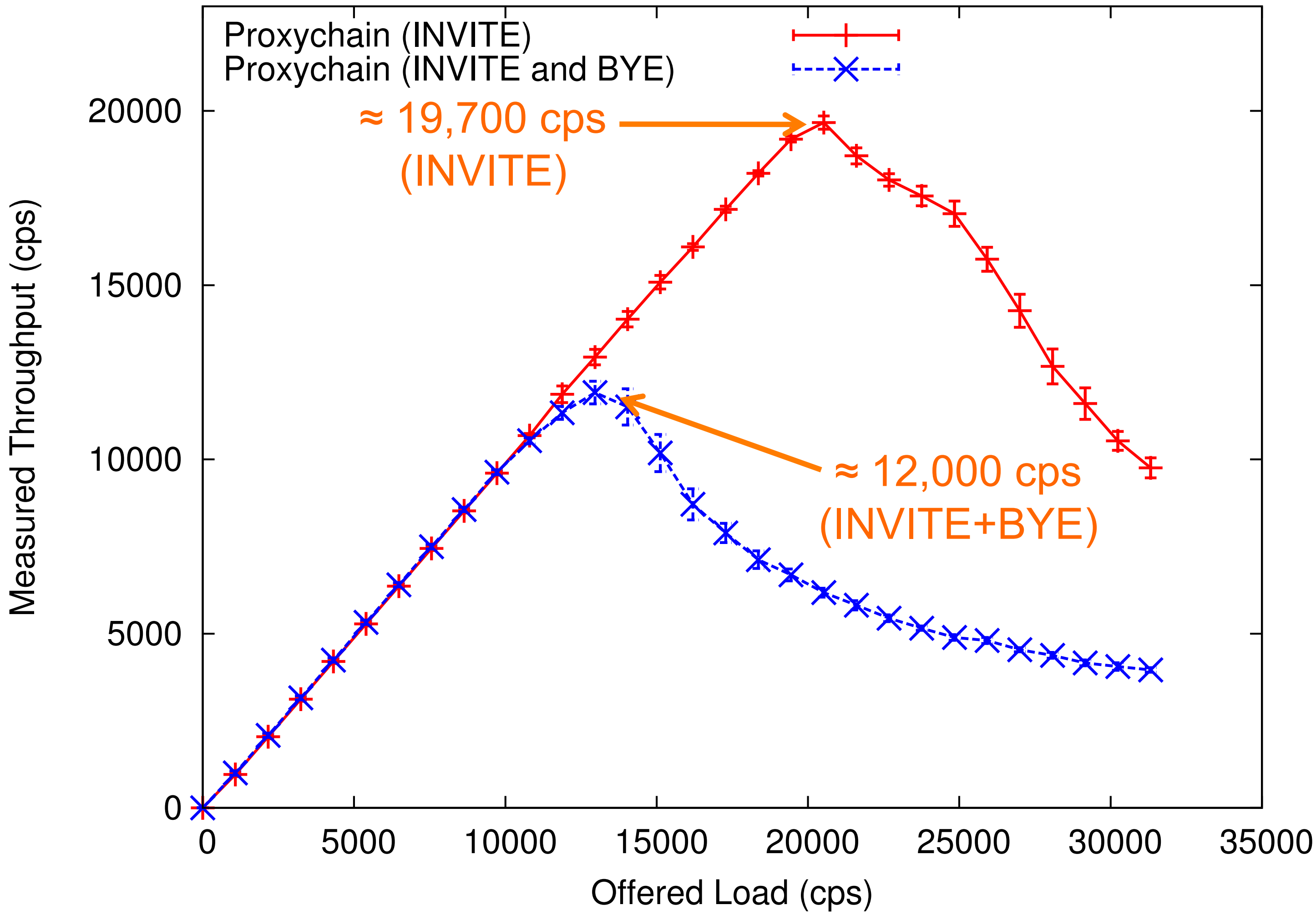
Results: Database CPU Utilization



Results: Scalability



Results: INVITE and BYE Authentication



Discussion: Performance and Scalability

- Proxychain **reduces the effects of network latency**, allowing higher call throughput

- Lower load to the database allows **more scalability** and **lower HW requirement**

Discussion: Performance and Scalability

- Hash chains allow **constant storage space**
  - Dynamic reprovisioning (future work)

- Key assumption: each proxy caches most of its users' credentials (>75%)
  - Pre-fetching mechanism
  - Cache eviction policies (future work)

Discussion: Security

- **Security improvements** over Digest authentication and hash chain protocols
  - **Efficient mutual authentication**, additional security verifications
- Protection against passive and active attackers
  - Stealing credentials from a proxy **does not allow user impersonation** (only affects mutual authentication)

Conclusions

- Proxychain simultaneously provides a robust, scalable and efficient authentication mechanism for carrier-scale SIP providers without additional HW
- Even non-carrier level infrastructures with centralized authentication service can benefit from Proxychain
- The key concepts behind Proxychain can be applied to authentication protocols in other domains

Questions?

Contact: idacosta@gatech.edu