Rabbit-Mix

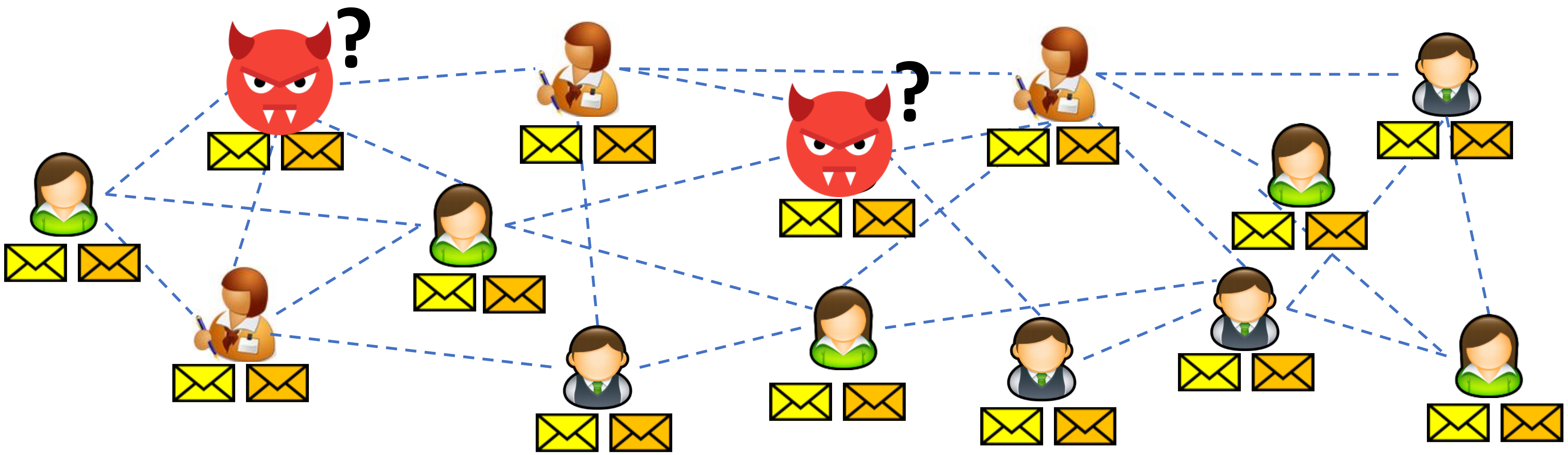
Robust Algebraic Anonymous Broadcast from Additive Bases

USENIX Security '24
Philadelphia, PA, USA

Chongwon Cho (Stealth Software Technologies Inc.)
Samuel Dittmer (Stealth Software Technologies Inc.)
Yuval Ishai (Technion)
Steve Lu (Stealth Software Technologies Inc.)
Rafail Ostrovsky (UCLA)

Anonymous Broadcast

- Senders
  - Senders have messages
- Receivers
- **Goal**: All honest receivers receive all m's from all honest senders (i.e., broadcast)
  - Adversary cannot interrupt honest participants
  - Adversary cannot link messages with messages (sender-anonymity)



Main Approaches

- Onion routing
  - [Tor, Dandelion, …]
- Mixing network
  - [Loopix, Miranda, …]

→

- Practical scalability and latency
- Vulnerable to traffic analysis
  - Statistical analysis
  - AI/ML analysis
- Subtle privacy/security definitions

- MPC-based mixing network
  - Dining Cryptographer network
  - [Riposte, MCMix, AsynchroMix, PowerMix, Blinder, Express, Sabre, RPM]

→

- Relatively less practical scalability and latency
- Cleaner privacy/security definitions
- Cryptographic security guarantees of non-linkage

Possible Applications

- Anonymous Anti-censorship Public Bulletin/Posting Systems
  - Human-rights violation reporting
  - Whistleblowing – Governmental/Organizational Corruption
  - Public journalism movement under Oppressive Regimes
  - .....

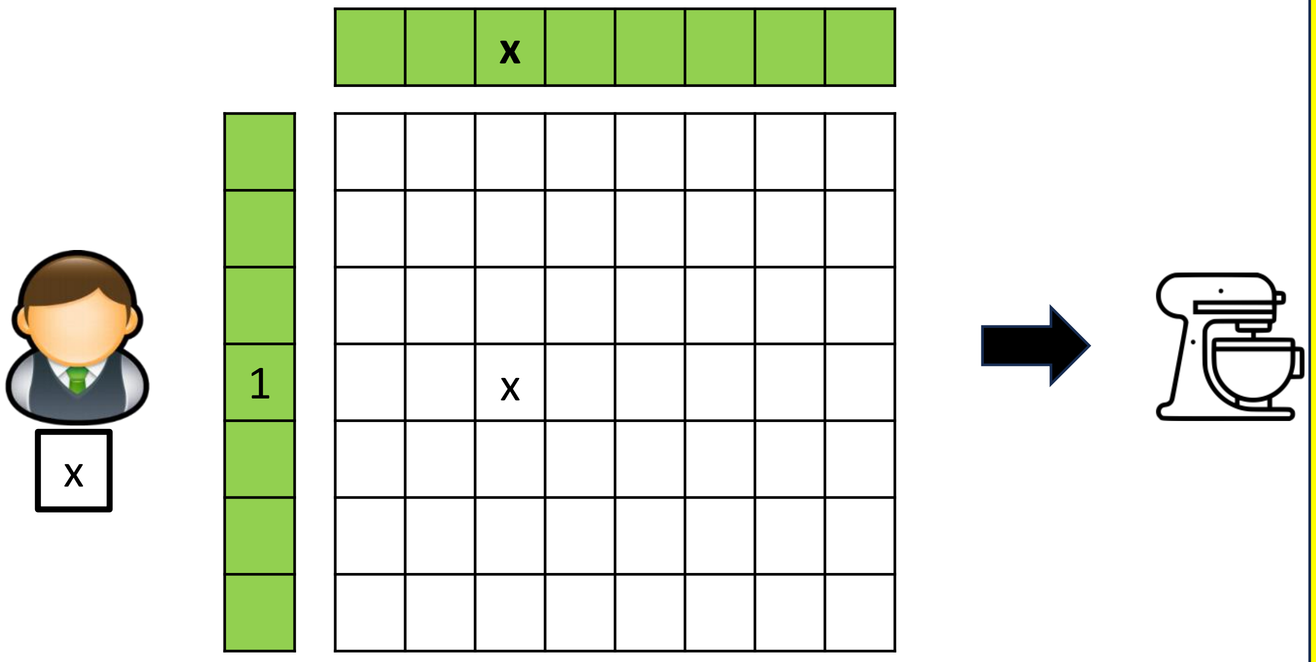
Robust Anonymous Broadcast

- Functionality in Client-Server Multiparty Computation (MPC) paradigm
- N clients
  - Each client has message m_i
- n servers - a mixing committee
- Clients submit m_i to the mixing committee (e.g., secret-shares m_i to servers)
- n servers executes a MPC protocol to mix (e.g., shuffle) and output messages to the clients
- **Goals:**
  - Small constant round complexity
  - Sublinear communication from clients to servers
  - Negligible message delivery failure
  - Efficient offline computation
    - e.g., the MPC prep for the main MPC is as efficient as the main MPC

MPC Approaches and Previous Works

- Private Writing [Riposte, Blinder, ...]
  - A client sends ~ $5\sqrt{N}$
  - Server computation O(N^2)
  - 95% of messages expected to be output (5% needs to be resent)

- **Decompression**: Tensor of two vectors - O(N^2)
- Collisions between messages will destroy messages
- Use two tables each bigger by factor of 2.7

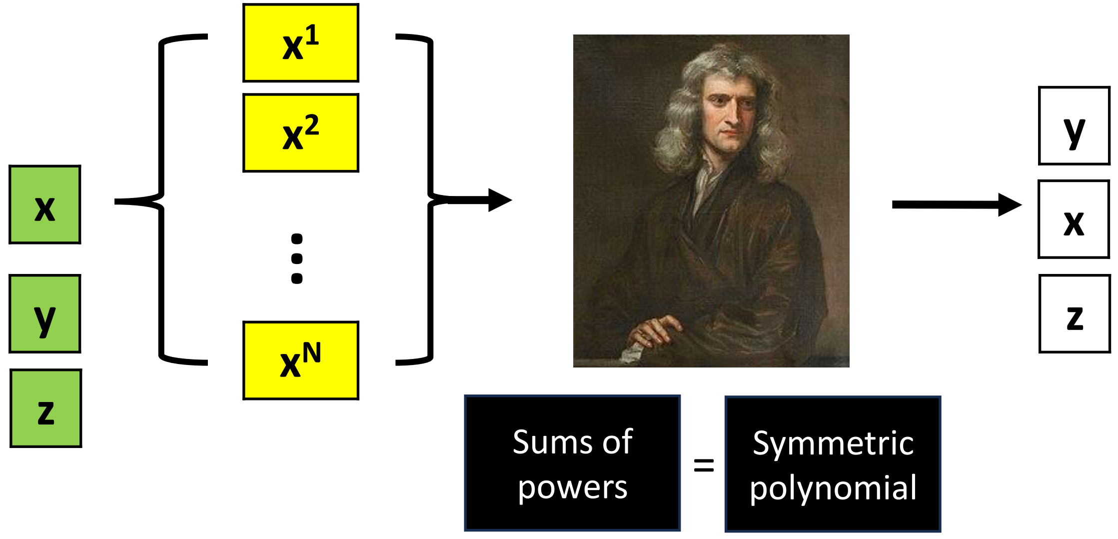


Bigger by factor of 22 for 99.9%

MPC Approaches and Previous Works

- Private Writing [Riposte, Blinder, ...]
  - A client sends ~ $5\sqrt{N}$ (redundancy)
  - Server computation $O(N^2)$
  - 95% of messages expected to be correct (5% needs to be resent)
- Newton's Identity [PowerMix]
  - A client sends 1
  - Server computation $O(N^3)$
  - 100% of messages expected to be correct

Decompression takes $O(N^3)$



Rabbit-Mix

- Goal 1: 100% message delivery rate
  - Newton's Identity

- Goal 2 & 3: Sublinear communication from Clients & $O(N^2)$ total server computation
  - Additive 2-basis

For simple additive 2-basis example, suppose N = 100

x

$O(N^3)$ computation
[Previous]

x^1
x^2
⋮
x^{100}

x

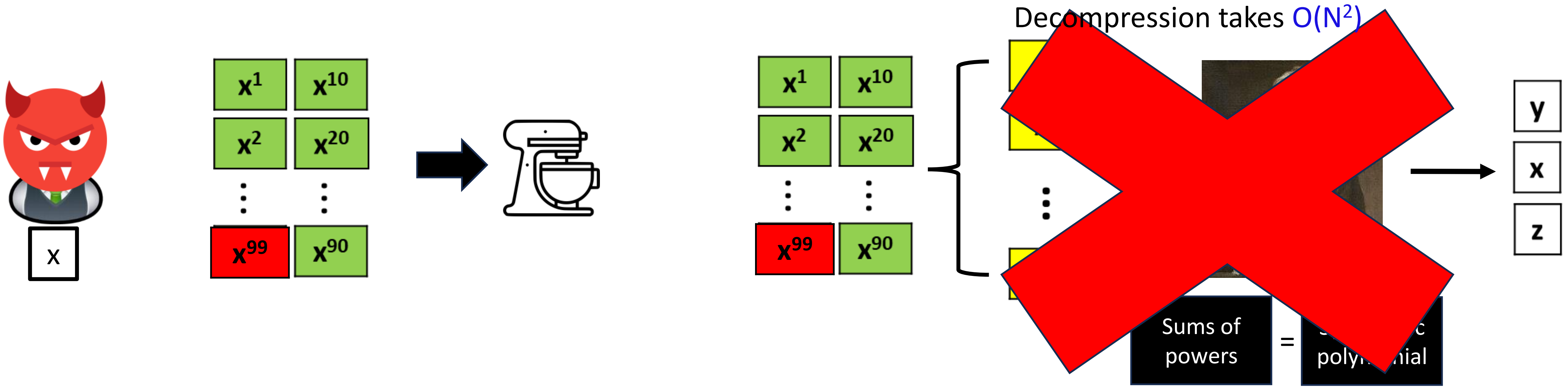
Optimized Additive 2-basis
[Mrose79]
- 7% reduction in size
- $2\sqrt{N}-1 \rightarrow$ **$1.87\sqrt{N}$**
(3X Better than Blinder)

$O(N^2)$ computation
→ Depth-1 local multiplications of two elements
[Our approach]

Rabbit-Mix

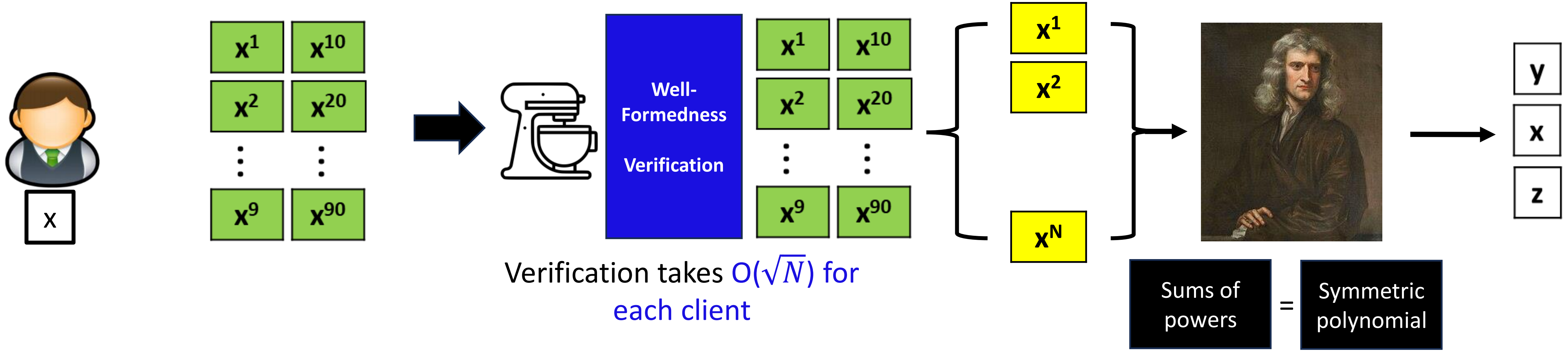
- Goal 1: 100% message delivery rate
  - Newton's Identity
- Goal 2 & 3: Sublinear communication from Clients & $O(N^2)$ total server computation
  - Additive 2-basis

Not Done Yet!!



Rabbit-Mix

- Goal 1: 100% message delivery rate
  - Newton's Identity
- Goal 2 & 3: Sublinear communication from Clients & $O(N^2)$ total server computation
  - Additive 2-basis
- Goal 4: Sieving malicious clients
  - Linear sketch for Additive 2-basis [BBCGI19, Blinder, ...]

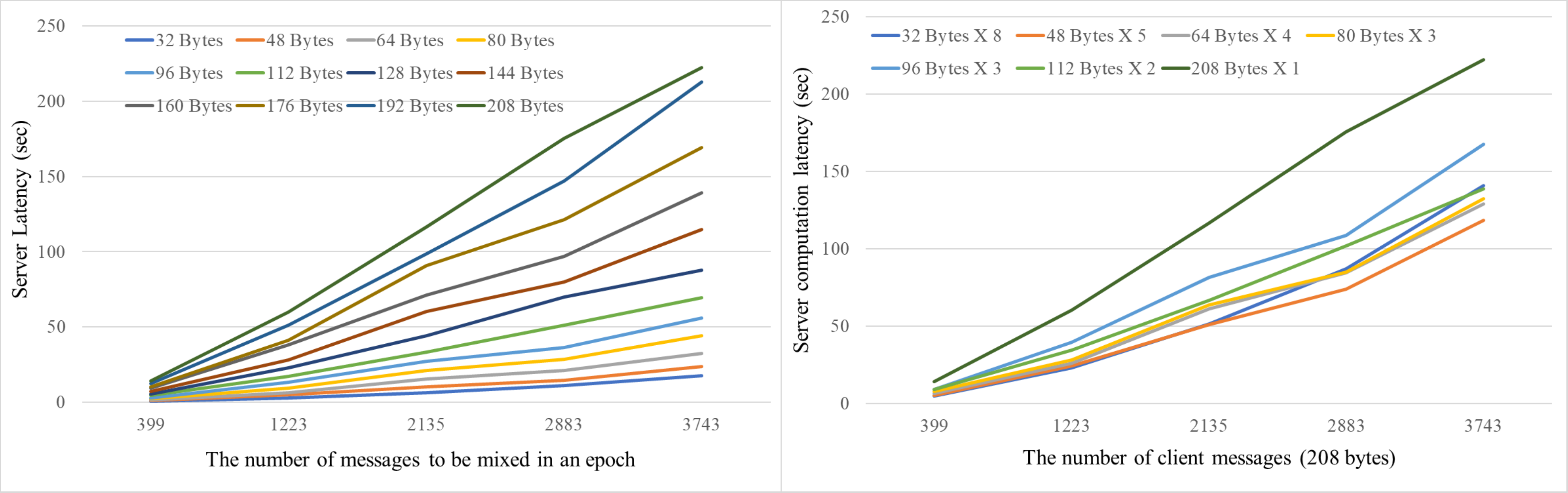


Implementation & Benchmarks

- Proof-of-concept
- C++ with NTL library
- Boost Asio for networking
- No parallelization used

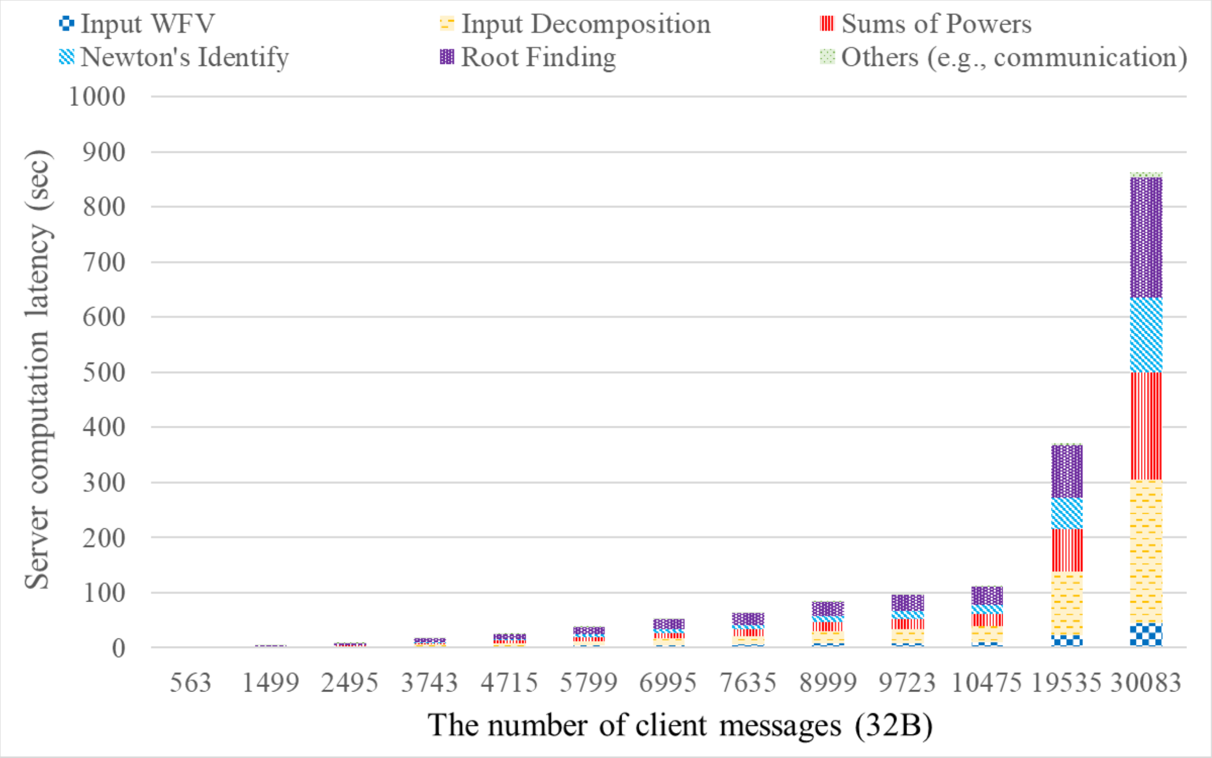
- 6 AWS EC2 instances – c5d.9xlarge
  - 5 servers
  - 1 client – submit all messages

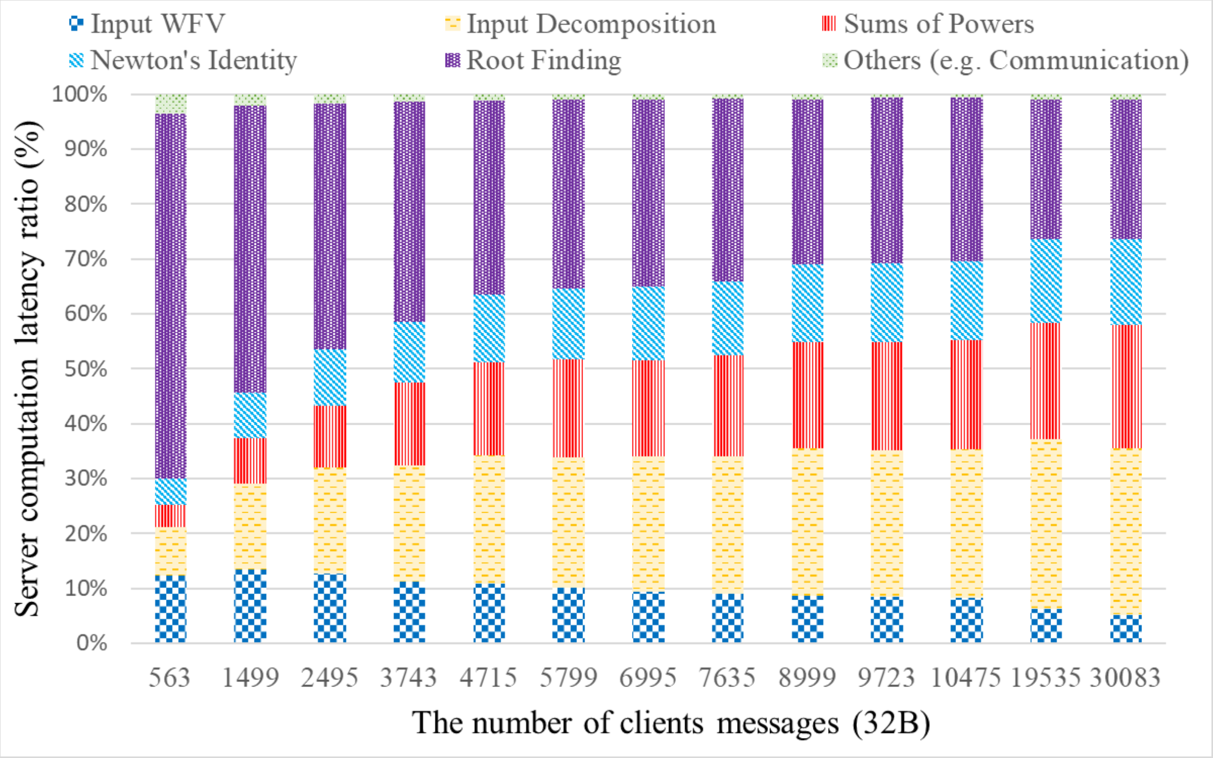
End-to-end latency with various primes and # of messages



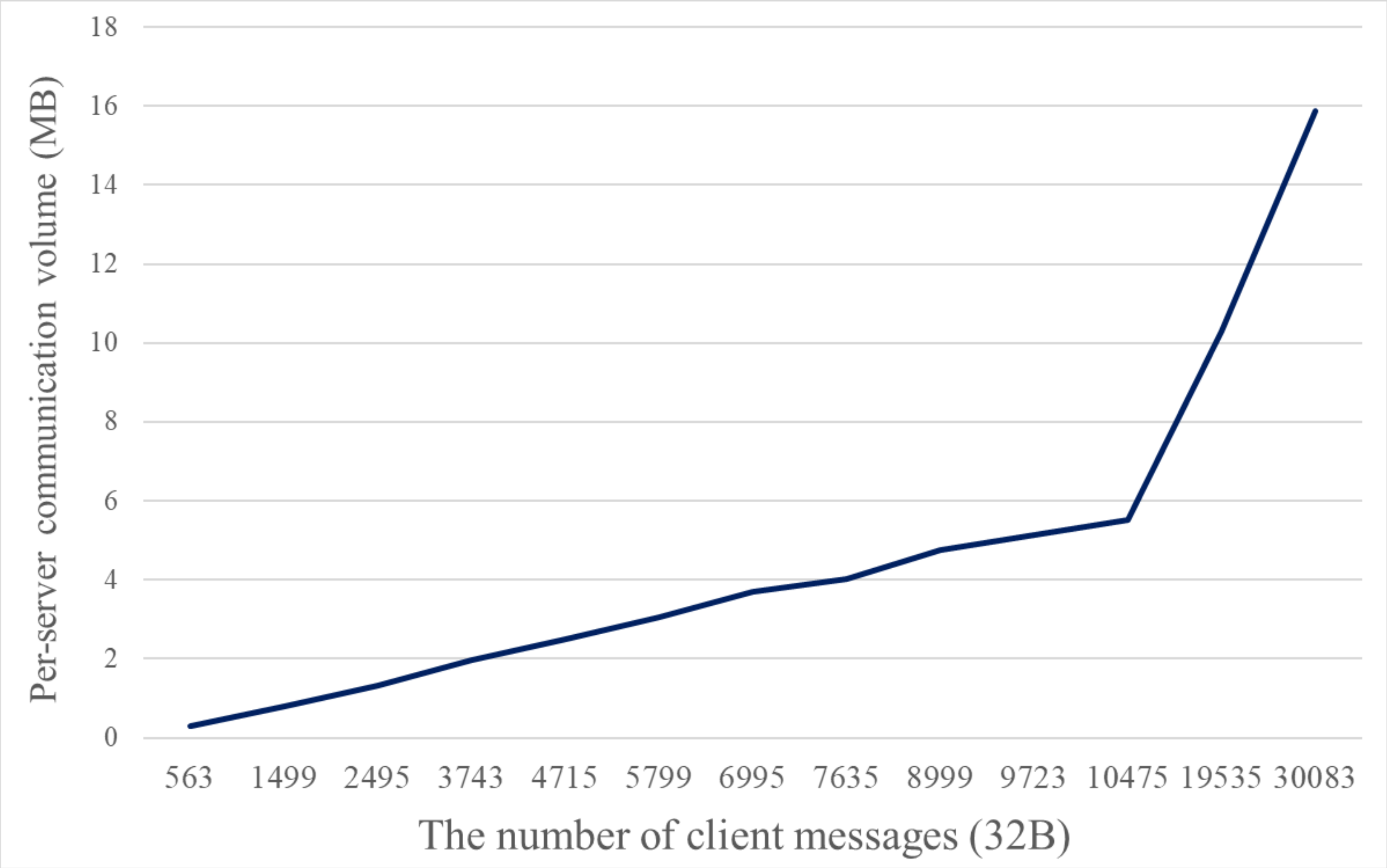
STEALTH SOFTWARE TECHNOLOGIES

Extensive Latency and Subprotocols (32 byte messages)

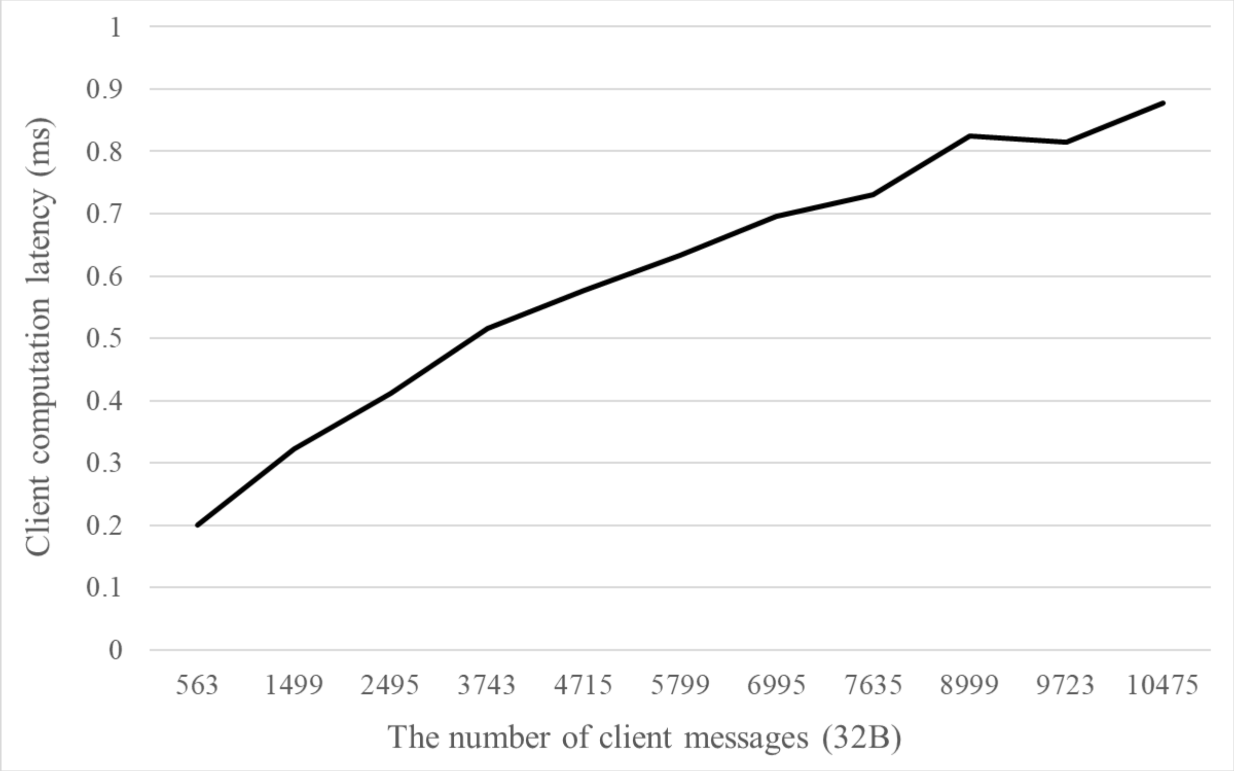


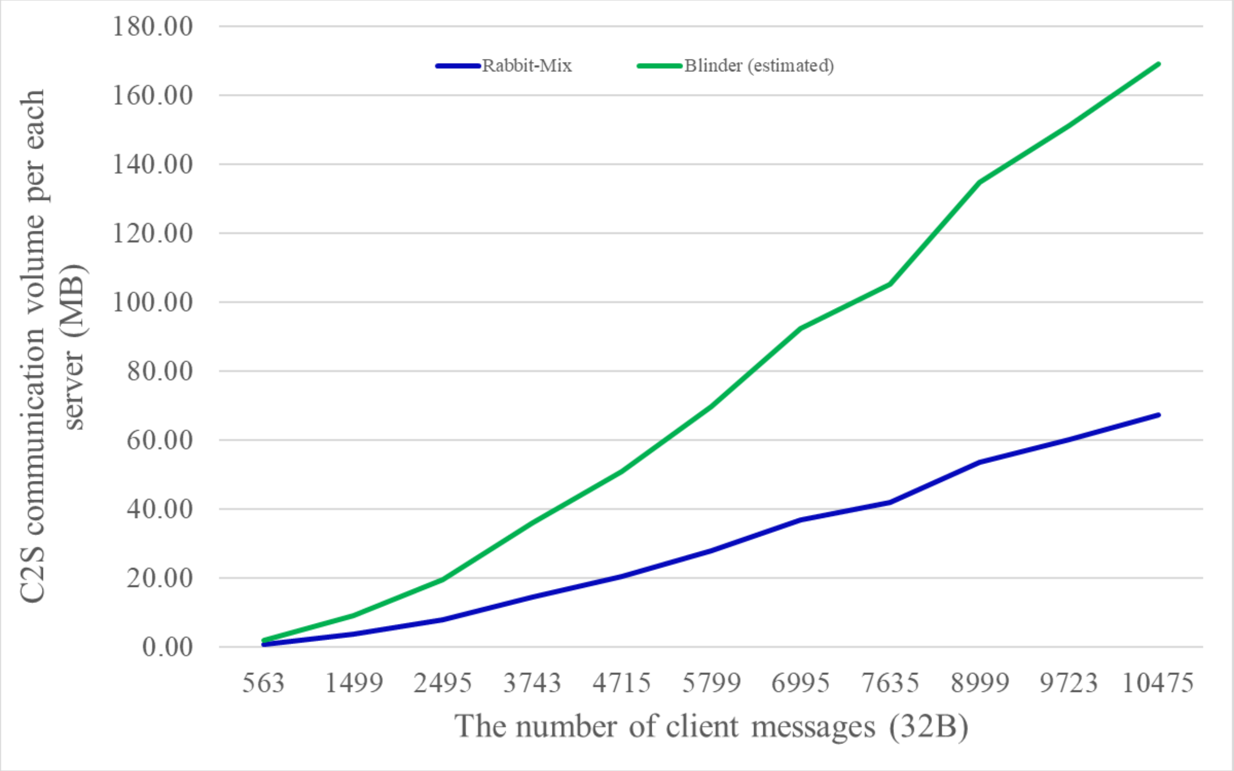


Server communication (32 byte messages)



Client computation and communication (32 byte messages)





Conclusion

- Improve client’s communication by 3X in comparison to Blinder (and other Private-Writing based protocols)
  - No need handle collisions
- Improved concrete server computation efficiency
  - 3X less operations in comparison to Blinder
- Efficient linear sketches for additive 2-basis
  - Verifying well-formedness over known arithmetic progressions
- Proof-of-concept implementation and benchmarks