CSci 8271
Security and Privacy in Computing
Day 3: Proofs of Retrievability

Stephen McCamant
University of Minnesota

What is a PoR for?

- Suppose I asked a cloud service to store 1TB of data for me
- How can I efficiently tell if they still have all my data today?
- Proof of Retrievability (PoR) successful audits should imply I can get all the data

Time-space tradeoff

- Impossible for a PoR to have small extra storage and very fast verification
- Intuition: if much of the data is not checked, you need error correction, but that needs extra data
- Proof by contradiction: randomly flipping a few bits would likely be undetected corruption

Linear algebra core

- Treat the data as a matrix M over a finite field, consider product with vectors u and x:
  - u is a long-term client secret
  - x is a random challenge per audit
- $u^\top(Mx) = (u^\top M)x$
- Client verification is just a dot product

External storage and verification

- Vector updates are just linear
  - But we also keep a Merkle tree, verifies reads and writes
- The vector can be kept on the server if it's encrypted
  - Also with a Merkle tree for integrity across updates
- Using a larger finite field and exponentiation gives a public-key-style anyone-verifies version

Cloud performance results

- Computation/storage trade-off is favorable for Google Cloud
- Server computation is less than a SHA256 checksum (~10min/TB 1 core)
- Parallelizable, 8 cents (private) or 16 cents (public) for 1TB
  - Versus at least $40/month for extra storage