The Cloud was tipsy and ate my files!

Giuseppe Ateniese
Johns Hopkins University and
Sapienza-University of Rome

Papers

- G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In ACM CCS '07, Full paper in ACM TISSEC 2011.
- G. Ateniese, R. Di Pietro, L.V. Mancini, and G. Tsudik. Scalable and Efficient Provable Data Possession. In SecureComm '08.
- G. Ateniese, S. Kamara, and J. Katz. Proof of Storage from Homomorphic Identification Protocols. Asiacrypt 2009.

Short "elevator pitch"

- “Your files are stored in the Cloud. My company, for $9 per month ($900, for businesses), monitors the Cloud to ensure that the entire content of your digital life is intact and readily available.”
- “What’s cool about it? We do not even know what we are checking! No privacy issues or intellectual property infringement.”

Cloud Storage

- Benefits:
  - Clients with limited resources or expertise can outsource their storage
  - Universal access, independent of location (Gmail, Hotmail, Yahoo, Gdoc, Office, etc.)
  - Data backup/recovery/archival
  - Security (encryption)

Archival Storage Outsourcing

- Electronic records legislation requires:
  - Data be retained for several years
  - Data be available
- Outsourcing data to third parties:
  - Avoids initial setup cost
  - Maintenance and scalability

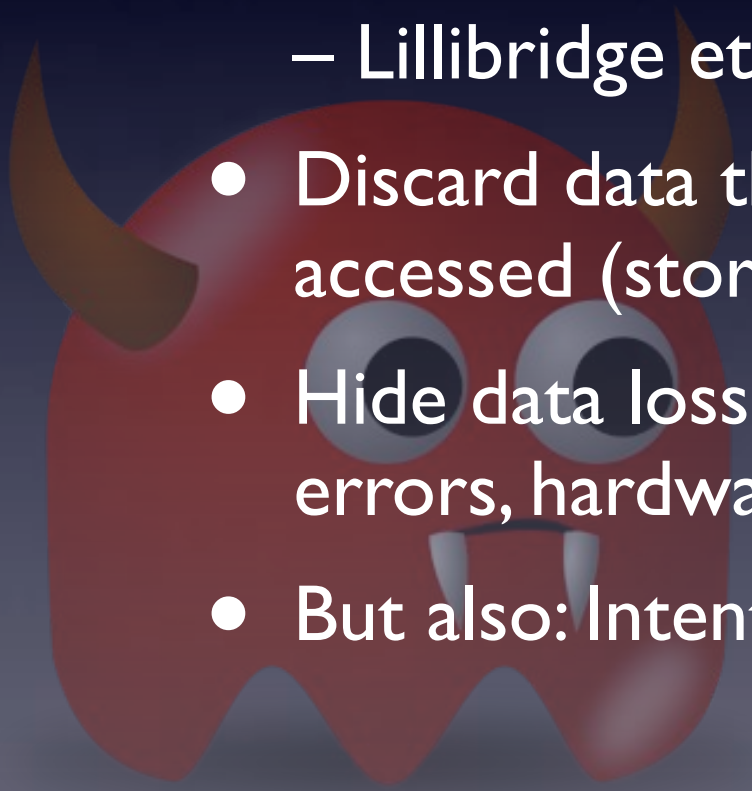
Our focus: Archival Storage

- Remote servers retain tremendous amounts of data
- Only small parts of the data are retrieved
- Data is stored for a long time (forever)

Source: www.loc.gov

Third-parties cannot be trusted

- Remote servers can misbehave:
  - Reduce cost / increase profit ("freeloading" – Lillibridge et al.)
  - Discard data that is not accessed or rarely accessed (stored on secondary tapes, etc.)
  - Hide data loss incidents due to management errors, hardware failures, attacks, etc.
  - But also: Intentionally modify data



Provable Data Possession

- Can my cellphone verify that the entire content of the Library of Congress is stored and available online?
- We provided the first provably-secure and practical PDP schemes
- We showed experimentally that PDP can be used for very large data sets

Review of PDP

- Trivial schemes that do not work:
  - Check data upon retrieval
  - Ask the storage server (google) to MAC the entire archive
  - Ask the storage server to send a subset of randomly-picked file blocks along with their MACs
- Our target: Aggregate MACS and DO NOT send file blocks!

What?

- Blockless verification:
  - Force the storage server to perform certain computations on the file blocks
  - Later verify that those computations are correct via authentication tags
- Aggregate MACs:
  - Homomorphic authentication tags
  - Public verifiability (we introduced this notion)

RSA 101

$$N = pq, p = 2p' + 1, q = 2q' + 1$$
$$ed \equiv 1 \bmod \phi(N)$$

$$PK = (e, N) \qquad SK = (d, p, q)$$

$$Sign = H(m)^d \bmod N \text{ (H() is a random oracle)}$$
$$Check : (Sign)^e = H(m)$$

RSA-based TAGS

$$t_i = (H(W_i) \cdot g^{m_i})^d \bmod N$$

Single Block

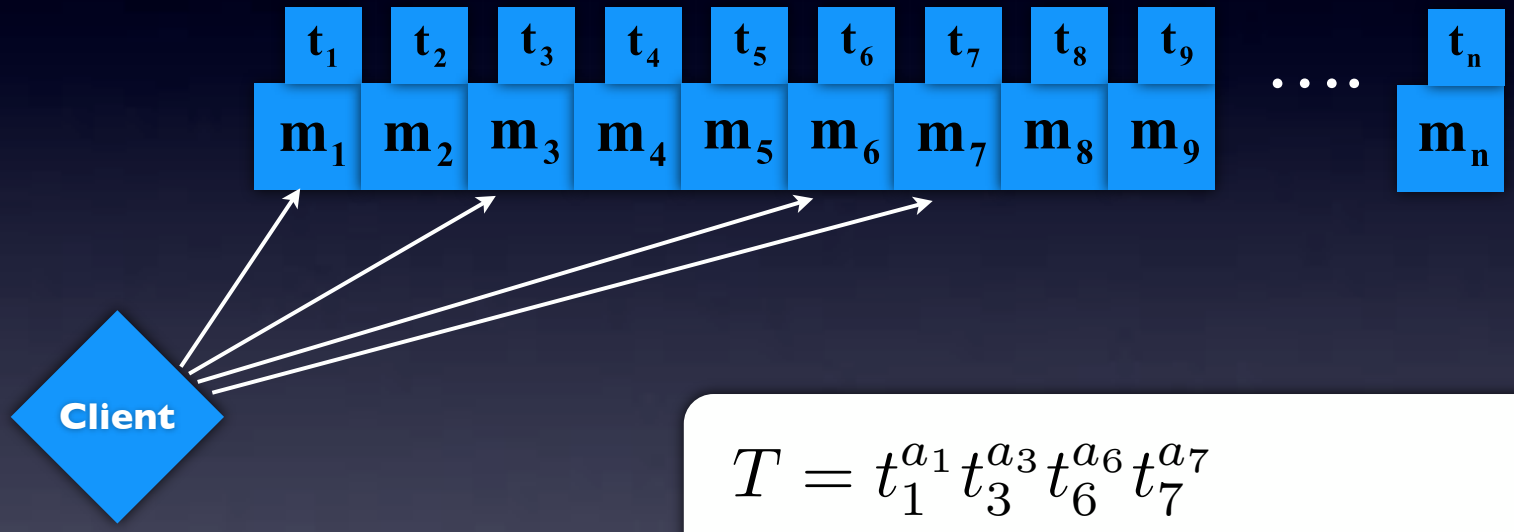
$$\xrightarrow{\quad i \quad}$$

$$t_i = (H(W_i) \cdot g^{m_i})^d$$

$$\xleftarrow{\quad (m_i, t_i) \quad}$$

Check: $0 \leq m_i < e$; $\dfrac{t_i^e}{H(W_i)} = g^{m_i}$

Challenge-verification



$$T = t_1^{a_1} t_3^{a_3} t_6^{a_6} t_7^{a_7}$$
$$M = a_1 m_1 + a_3 m_3 + a_6 m_6 + a_7 m_7$$

$$\text{Check: } 0 \leq M < e; \frac{T^e}{H(W_1)^{a_1} H(W_3)^{a_3} H(W_6)^{a_6} H(W_7)^{a_7}} = g^M$$

Features

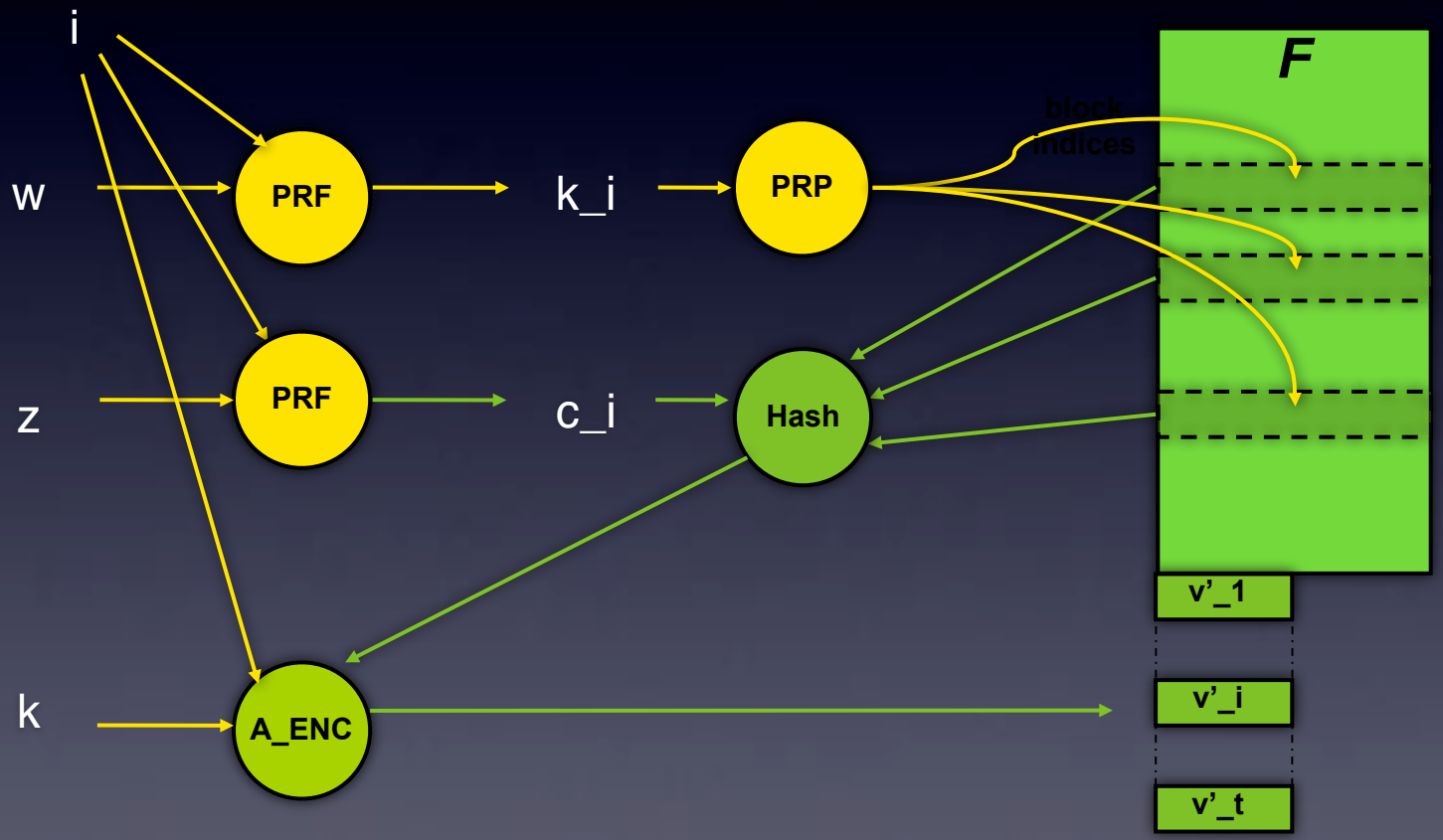
- Public verifiability
- Unbounded number of challenges
- Public data (no encryption)

Efficient Dynamic PDP

Features

- Simple design
- Very efficient
- Dynamic data
- Public data (no encryption is required)
- But: No public verifiability and limited number of challenges

Tokens



v'_i=A_ENC(i, H(c_i, D[i_1], ... , D[i_r]))

Discussion

- Bandwidth-Storage Tradeoff
  - The client can store some or all tokens locally
  - In most practical cases, all tokens require 10-20MB of space (it's just a single hash per challenge)
- Limited number of verifications
  - The number of tokens does not depend on the size of the database
  - Checking several terabytes every 15 minutes for the next 16 years would require only 1 Mbyte of extra storage per year!

Compiler and schemes based on weaker assumptions

PDP from Homomorphic Sigma Protocols

- We introduce a compiler that transforms a homomorphic sigma protocol into a PDP
- The transformation does not require random oracles
- In the Random Oracle Model, we introduce the first PDP scheme based on Factoring

Conclusions

- Provable Data Possession is cool :)
- Scheme providing public verifiability and unbounded challenges
- Scheme to support dynamic operations on outsourced data (block update, deletion, and append)
- Scheme based on Factoring
- Open problems: Full privacy (zero-knowledge), Efficient solutions for multiple storage servers