Scalable and Efficient Provable Data Possession

Prof. Luigi V. Mancini

Dipartimento
di Informatica

Sapienza
Università di Roma

Via Salaria 113 - 00198 Roma,
Tel: +39 06 49918421
E-mail: lv.mancini@di.uniroma1.it

IC Security Group - http://icsecurity.di.uniroma1.it

Critical infrastructure protection

App & Web Security

Homeland security

Steganography and digital watermarking

PKI and X.509

Vulnerability assessment
Exploit analysis

SPKI/SDSI

Intrusion detection
and prevention

Access control
and trust mgmt.

Secure multicasting

Secure e-commerce
and micro-payments

Ad hoc wireless network

Privacy and anonymity

Protection from malicious code

Network mobility

Two Programs Offered

primo livello

Master in Sicurezza

dei Sistemi e delle Reti Informatiche per l'impresa e la Pubblica Amministrazione

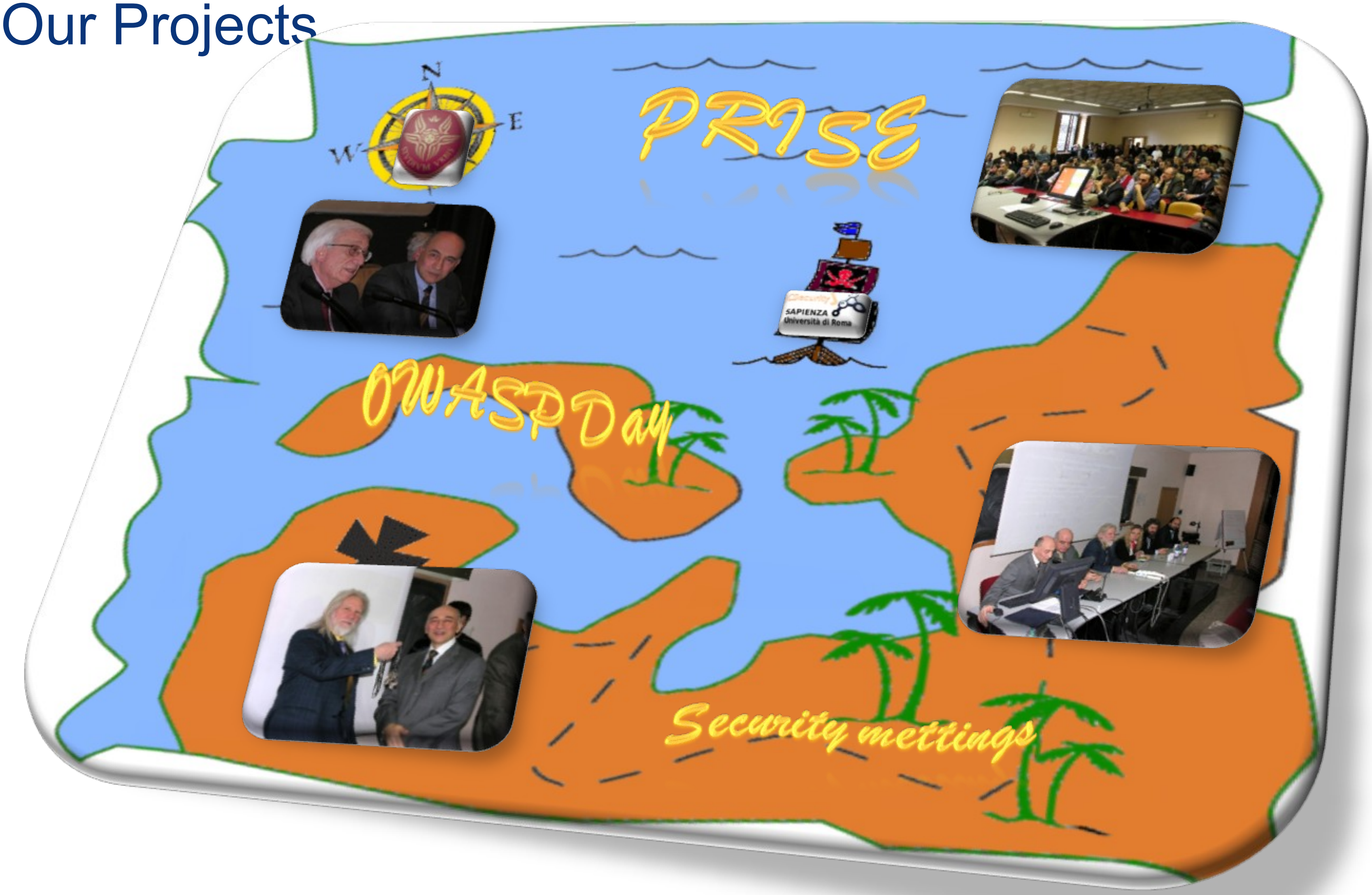
secondo livello

Master in Gestione della Sicurezza

Informatica per l'impresa e la Pubblica Amministrazione

Events Treasure Map

Our Projects



Agenda

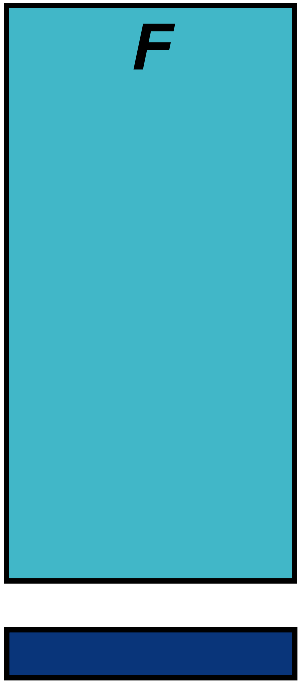
- *Problem definition*
- *Our PDP proposal*
- *Supporting Dinamyc Outsourced Data*
- *Analysis*
- *Related work*
- *Conclusion*

4th ACM Securecomm Conference, Sept. 2008
Joint work with G. Ateniese, R. Di Pietro, G Tsudik

Introduction

- “Storage-as-a-service” becoming a more common business model
  - Client pays server to store file *F*

- Without retrieving file, how can client be sure that server still has it?
  - Or, more generally, can provide it within an agreed response time?
- *Archiving* is a typical case: Client retains only metadata



Adversarial Model

- *Erasing adversary* may fail to store parts of file, or store at less than agreed tier
- *Corrupting adversary* may also modify parts of file
- Motivations:
  - Reduce cost / increase profit
  - Hide “evidence”
  - Change content – though typically detectable by integrity checks
  - Or, just hardware, software, or human error
- Assume that adversary has deleted or corrupted a fraction of file, up to time that test is run

Proofs of Retrievability

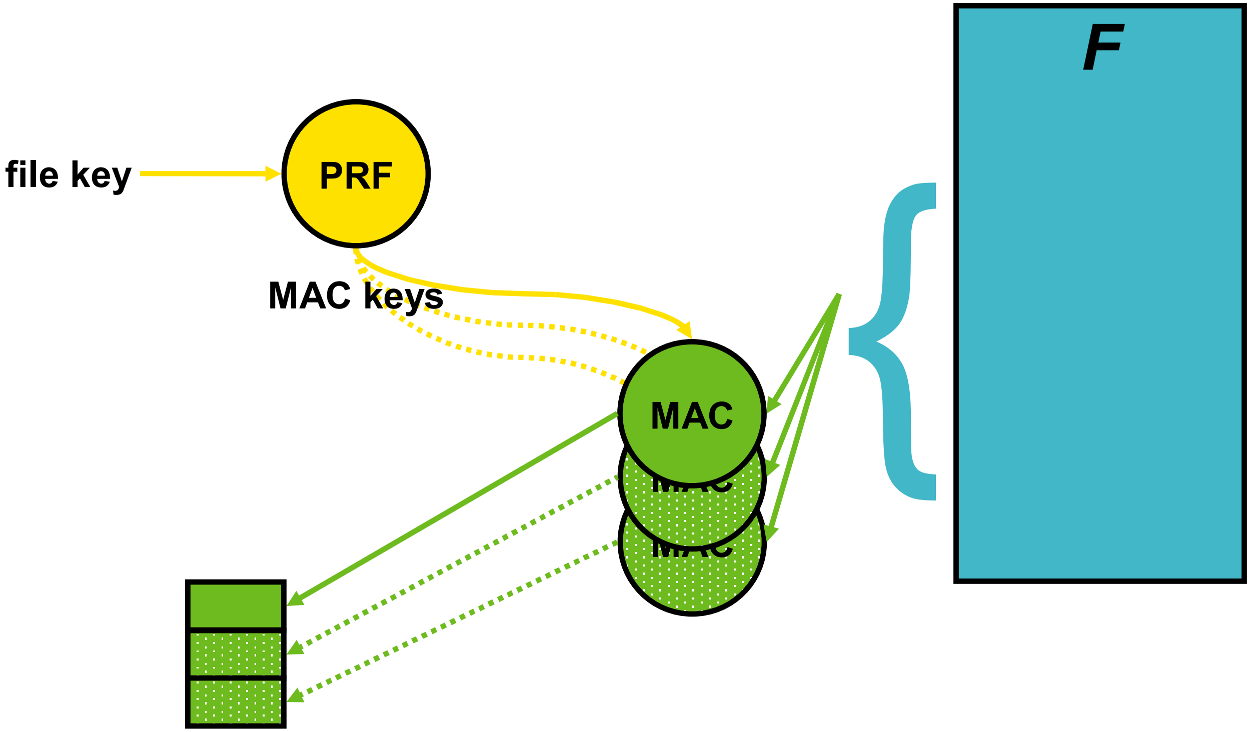
- *Provable Data Possession* (PDP) provides (probabilistic) assurance that a party possesses a file, without actually retrieving it

- Objective: Provide “early warning” of deletion, corruption, or other failure to meet service levels, in time to remediate
  - e.g., exclude this server and add another one

- PDP shows (probabilistically) that at time of test, adversary’s state is sufficient (w.h.p.) to enable retrieval – thereby limiting time period during which undetected corruption may occur

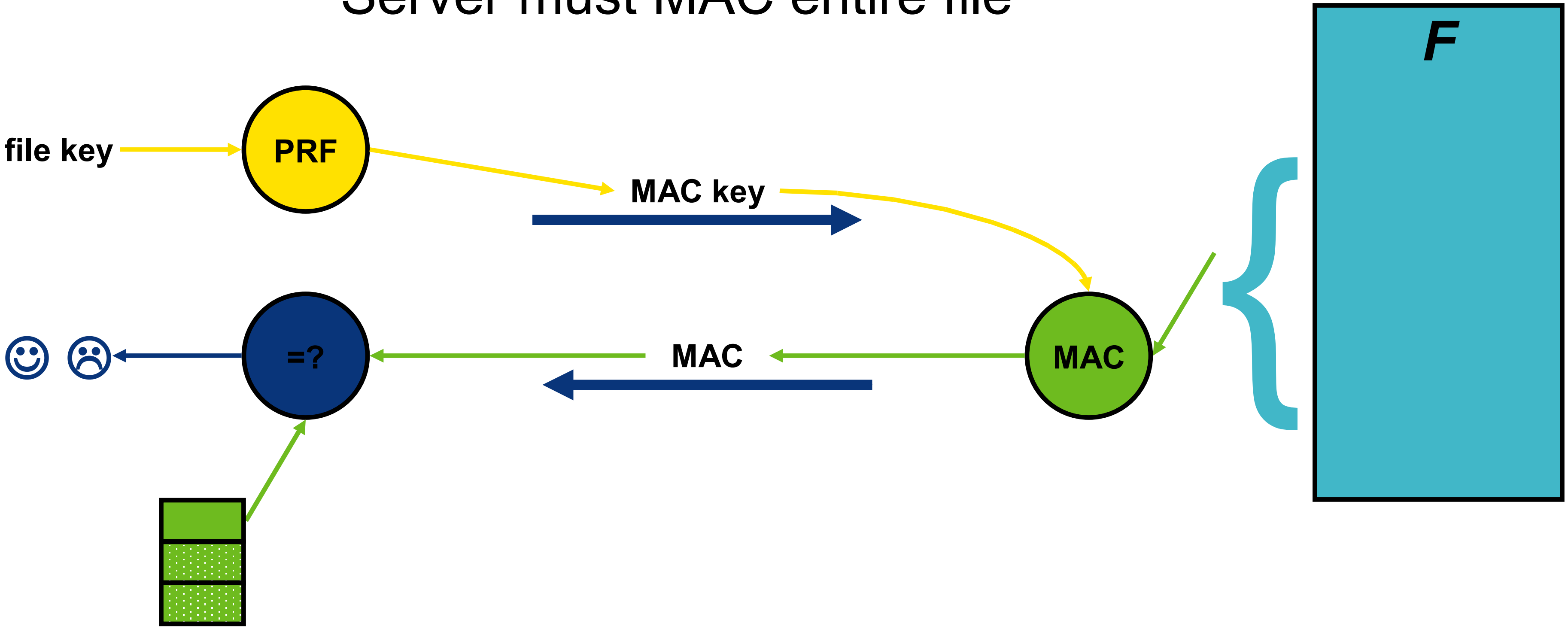
A Simple Approach: Challenge-Response MACs

- Message Authentication Code - MAC
- MAC entire file with different keys, try one at a time



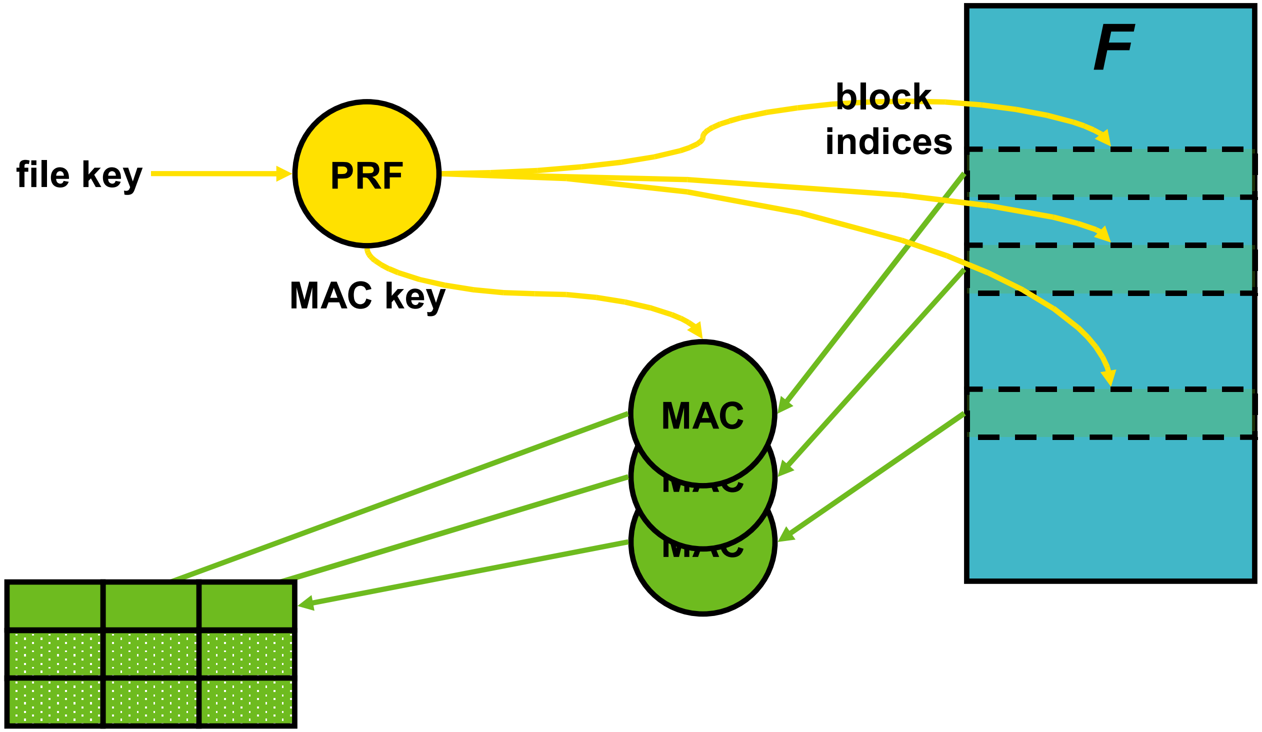
Simple Approach, cont'd

- MAC file with different keys, try one at a time
- # runs limited by client storage
- Server must MAC entire file



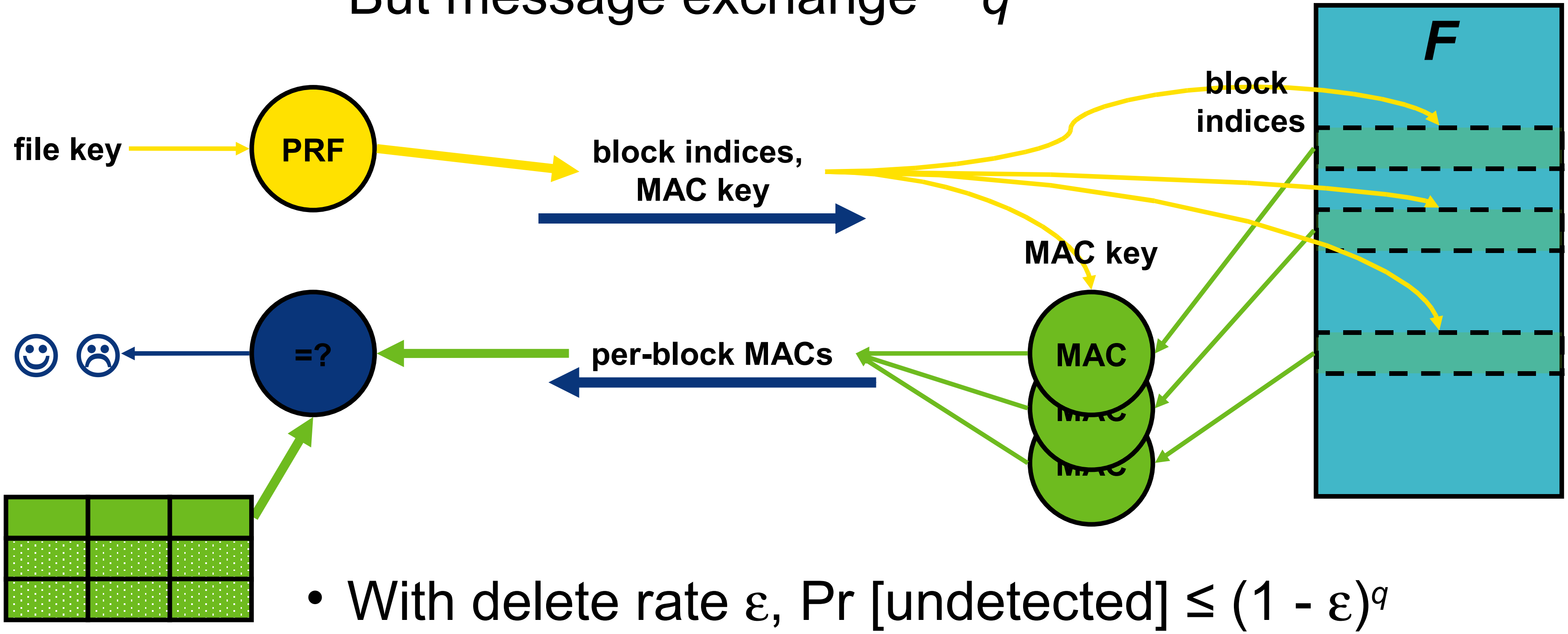
Per-Block MACs

- MAC selected q blocks



Per-Block MACs, cont'd

- MAC q selected blocks
- Server work now only *q* MACs / run
- But message exchange ~ *q*



- With delete rate ε, Pr [undetected] ≤ $(1 - \varepsilon)^q$

Our PDP proposal

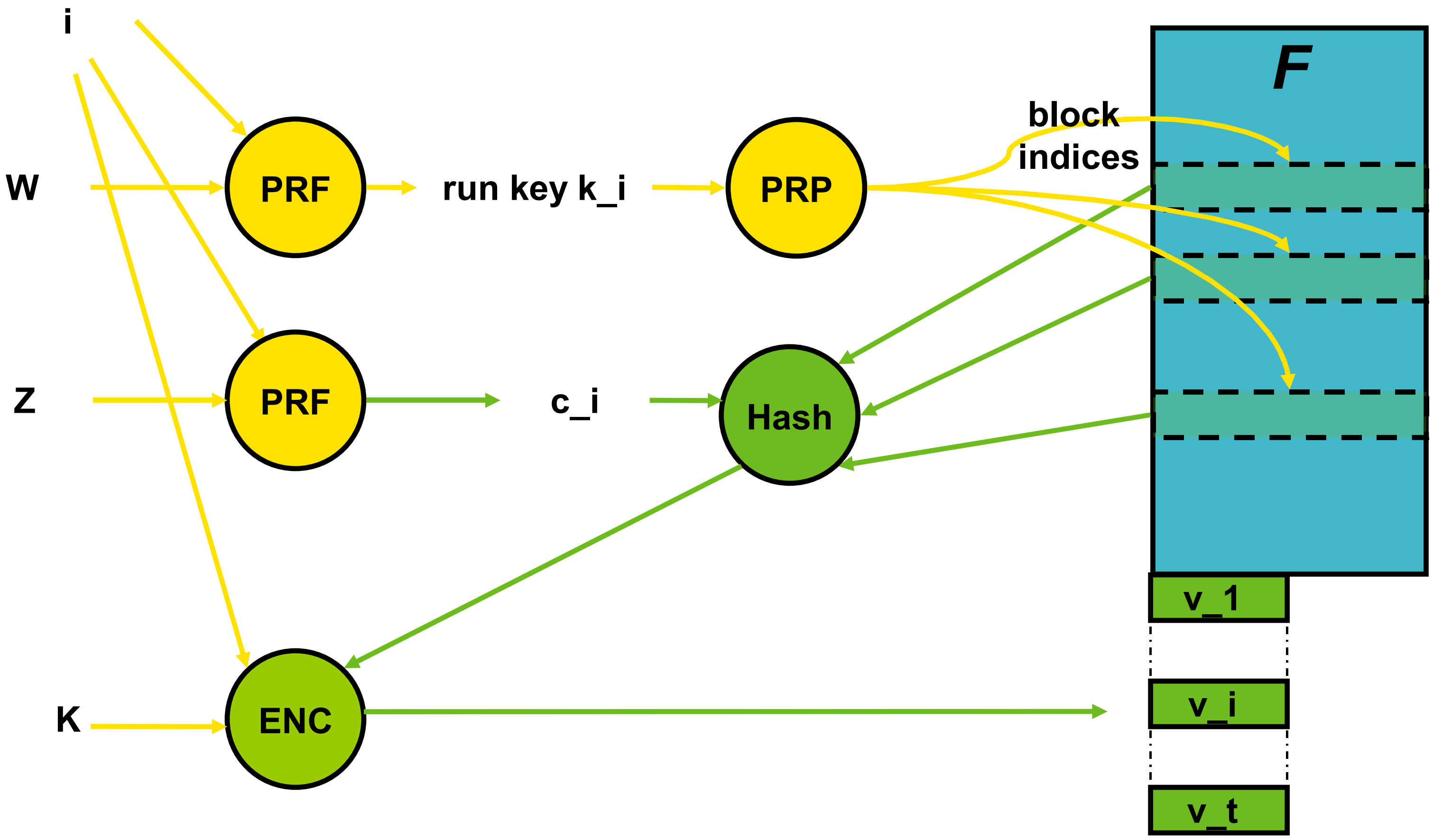
- Set up phase:
  - Generates t token
  - A token is computed over the hash of r blocks
  - Tokens can be stored either on OWN or on OUT
  - Each token is spent (cannot be reused) to perform one check

- Verification phase

Provides support for: block modification, deletion, and append

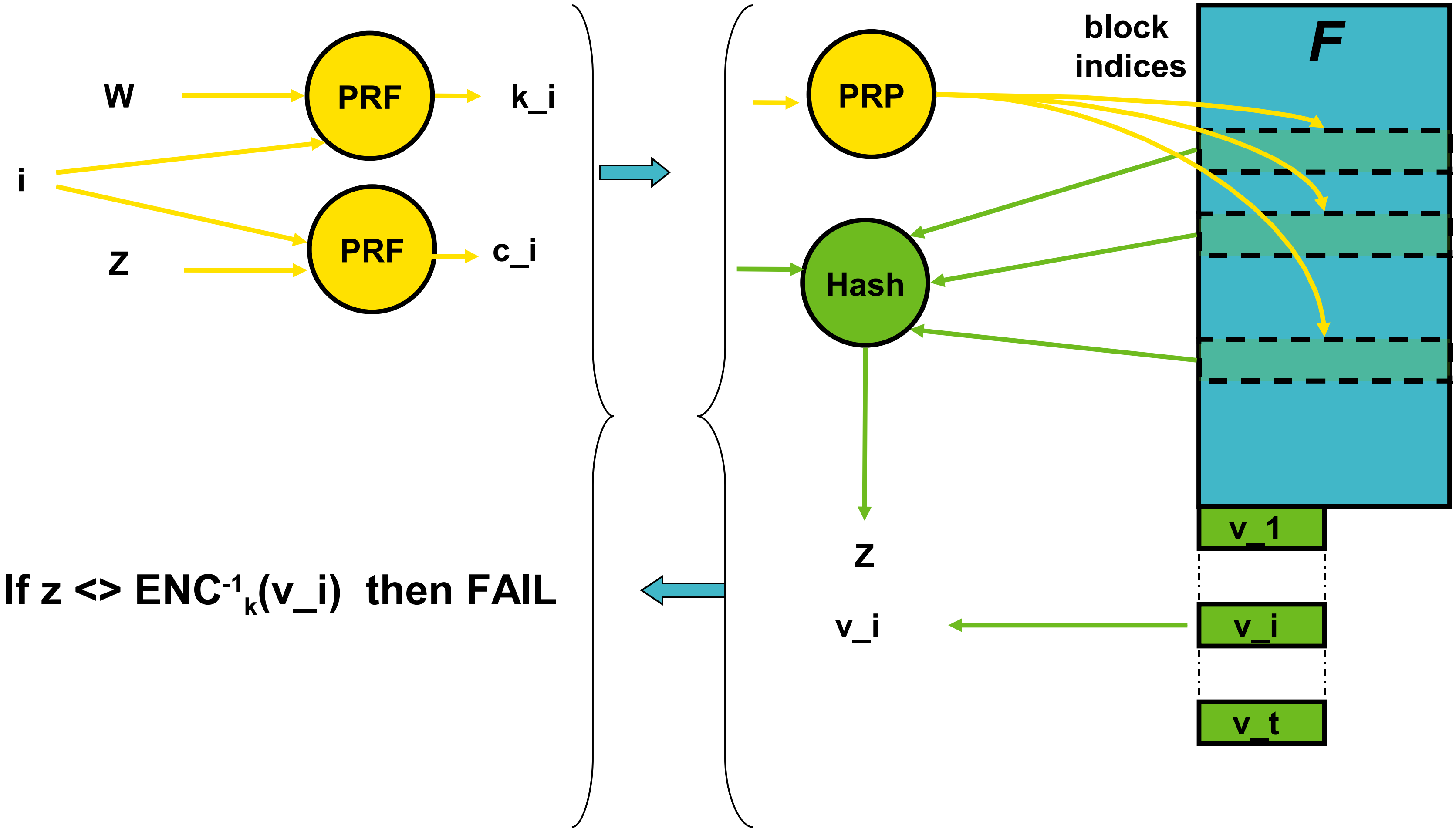
Our PDP proposal

Set-up Phase (to generate token i)



Our PDP proposal

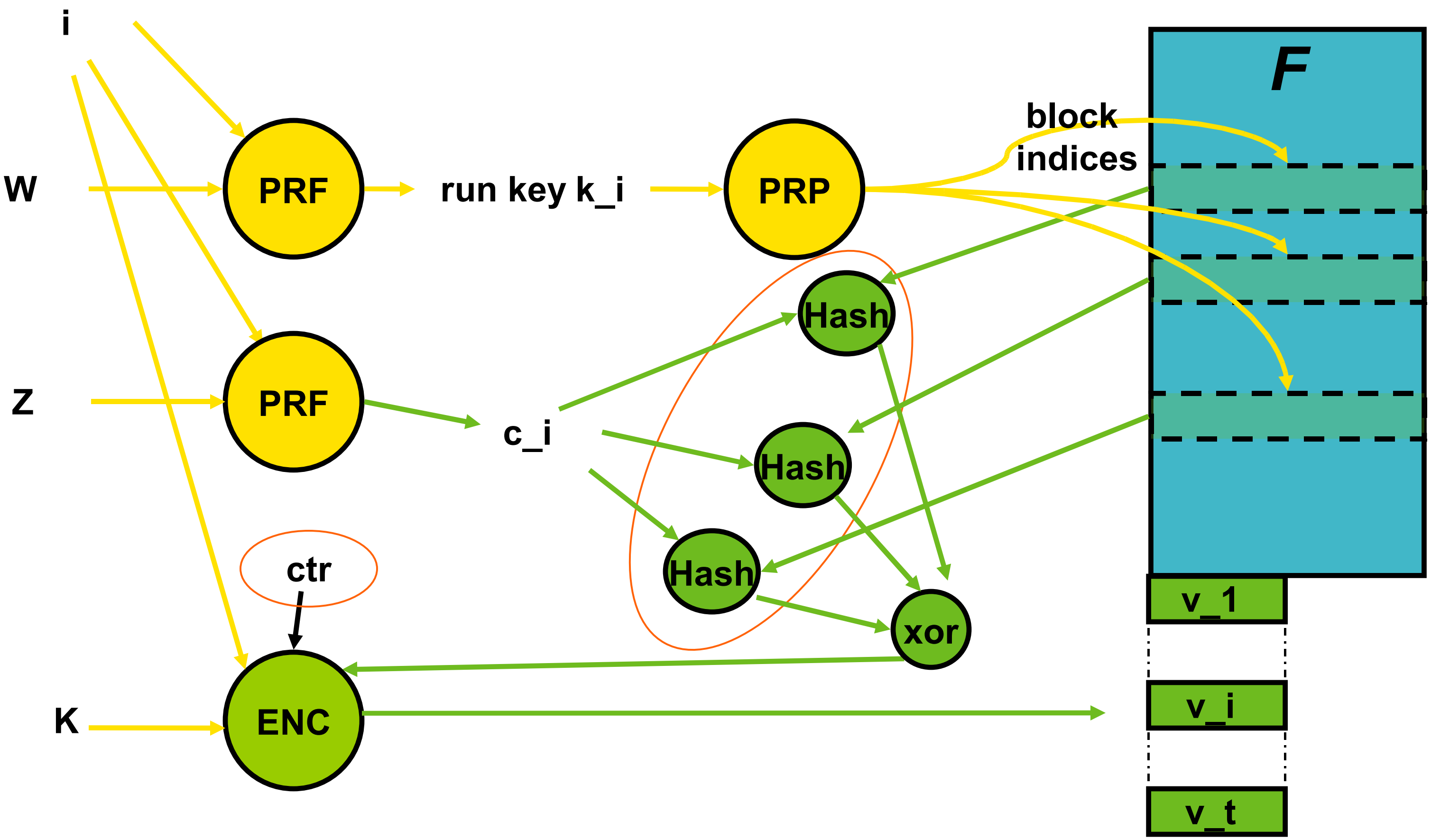
Verification Phase (to consume token i)



If z <> ENC^{-1}_{k}(v_i) then FAIL

Supporting Dinamyc Outsourced Data

Block Update (block i)



Supporting Dinamyc Outsourced Data

Block Update

- Updating one block, requires to update all the verifiers that used that block (on the average, just), but…

- OWN cannot recall and modify those blocks only, otherwise OUT could
- Hence, OWN has to recall and modify all of the verifiers

- (modification is just cheap re-encryption)

Supporting Dinamyc Outsourced Data

Block Deletion and block append

The same idea of block modification:

- All the verifiers have to be sent to OWN, that has to modify all of them

- Two levels of modifications:
  - First level: verifier do not encomprises a modified block→ Just re-encrypt the block;
  - Second level: verifier encomprises a modified block→ modify the verifier accordingly, and re-encrypt it.

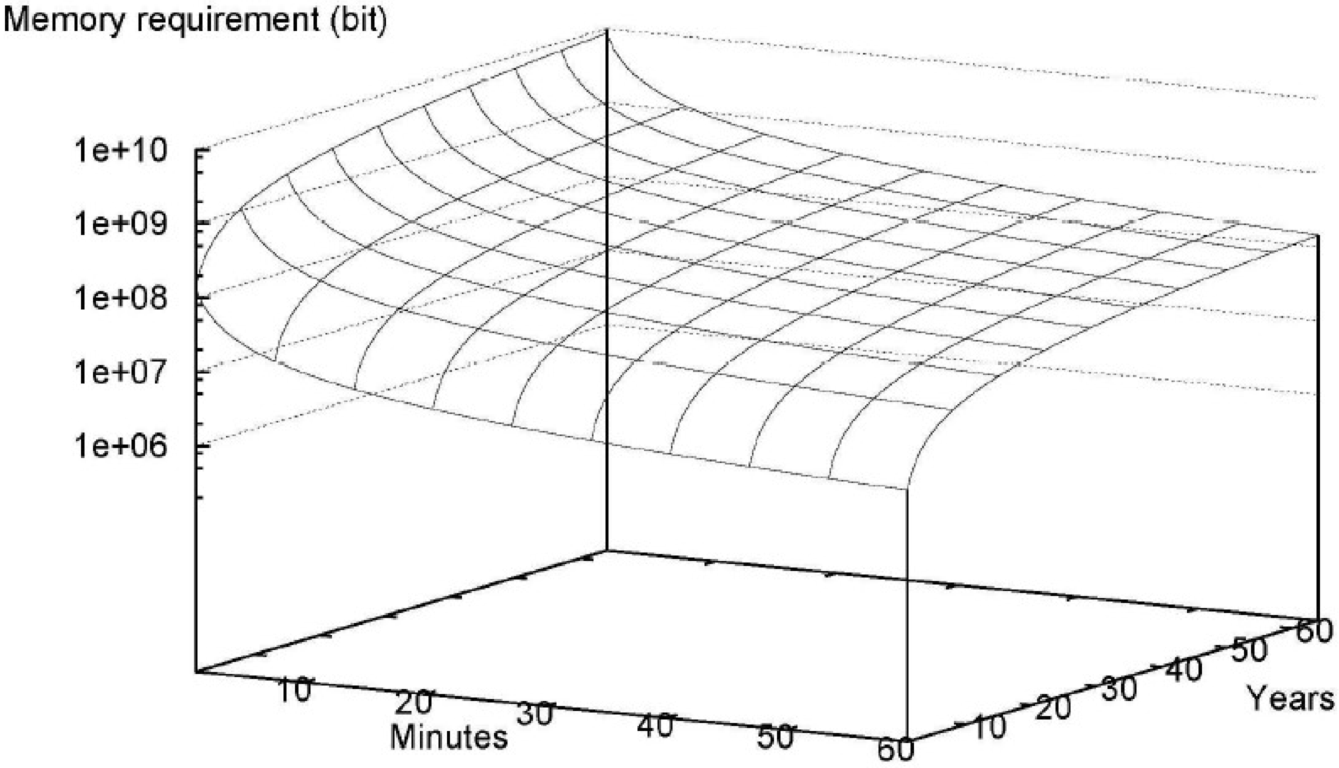
Analysis

Limited number of verifications (t)

- Back on the envelop computations:
  - The Web Capture project:
    - as of May 2007, about 70 Terabytes of data;
    - checking this content every 15 minutes for the next 16 years would require only 1 Mbyte of extra storage per year!
    - Could be even stored directly at OWN.

Analysis

Limited number of verifications (t)



Analysis

Computation

The haviest operation is the set up phase:

- t × r PRP;
- 2t PRF,
- t AEK invocations;
- t hashes, each over a string of size (r x |b|) size, where |b| is the block size.

Analysis

Computation

Plugging in real figures:

- SHA requires just 20 machine cycles/byte
- OWN outsources 2^{37} bytes of data, i.e., 128-GB.
- Each data block is 4-KB (|b| = 2^{12}) and

  d = 2^{37}/2^{37} =2^{25}
- OWN: one daily verification for the next 32 years, i.e., t = 32 × 365 = 11, 680.
- OWN: 99% detection probability, with 1% of the blocks being missing or corrupted;

Computation

Plugging in real figures:

Hence

- r = 29 (see Equation 1);
- The total number of hashes is 11, 680;
- Setup time (t hash computations) is about -on a 1 GHz CPU-: 11, 680 × 0.04 = 467 sec (less than 8 minutes).

Related Work

G. Ateniese et al. (CCS 2007) - Provable Data Possession -

(An elegant RSA variant construction)

- Store homomorphic tag for every block
- Client runs challenge-response protocol on *q* samples -removes limited number of verification, but set up is costly-

A. Juels and B. Kaliski (CCS'07) - PORs: Proofs of Retrievability for Large Files -

Conclusion

- Very light-weight and provably secure PDP scheme.
- The first scheme to support Dynamic Operations on Outsourced data (block update, block deletion, and append);
- It surpasses prior work on several counts: storage, bandwidth, and computation.

Grazie!

E-mail

mastersicurezza@di.uniroma1.it

Web

http://mastersicurezza.uniroma1.it/