An (Almost) Constant-Effort Solution-Verification Proof-of-Work Protocol based on Merkle Trees

Fabien Coelho

MINES ParisTech

Composed with $\LaTeX$, revision 841

Proof of Work?

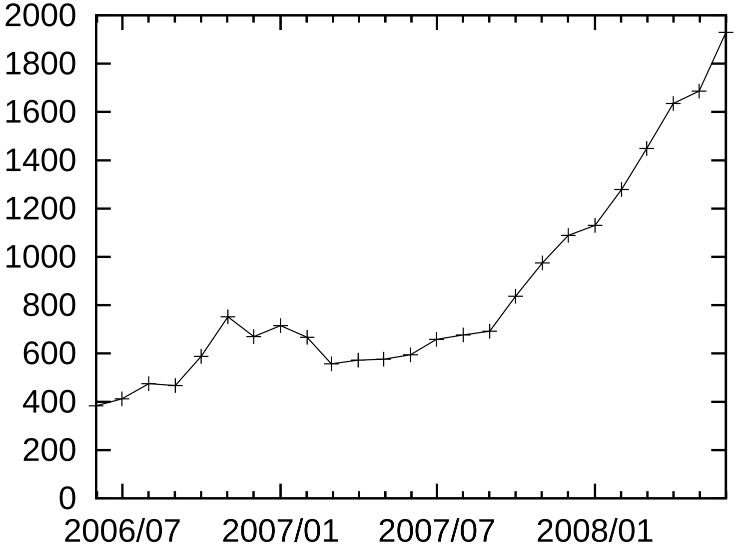
economic measure to deter DOS attacks

Crypto'92 Cynthia Dwork and Moni Naor

Pricing via processing or combatting junk mail

computation stamp for a service

moderately hard for requester, easy check by provider



spams per day received

on my addresses

period last 2 years

HashCash Adam Back 1997

- partial hash inversion $\mathtt{SHA1}(\mathtt{service} - \mathtt{description} : \mathtt{counter})$

  hash starts with n zeros (*e.g.* $n = 22$)

- 2^n hashes on average to compute 1 hash to check

```
To: fabien.coelho@ensmp.fr
Date: Sun, 19 Mar 2006 19:41:30 -0500
From: "Eric S. Johansson" <esj@harvee.org>
Hashcash: 1:25:060320:fabien.coelho@ensmp.fr::8064c52cc126872c:14b3bb
```

25 bits partial hash inversion

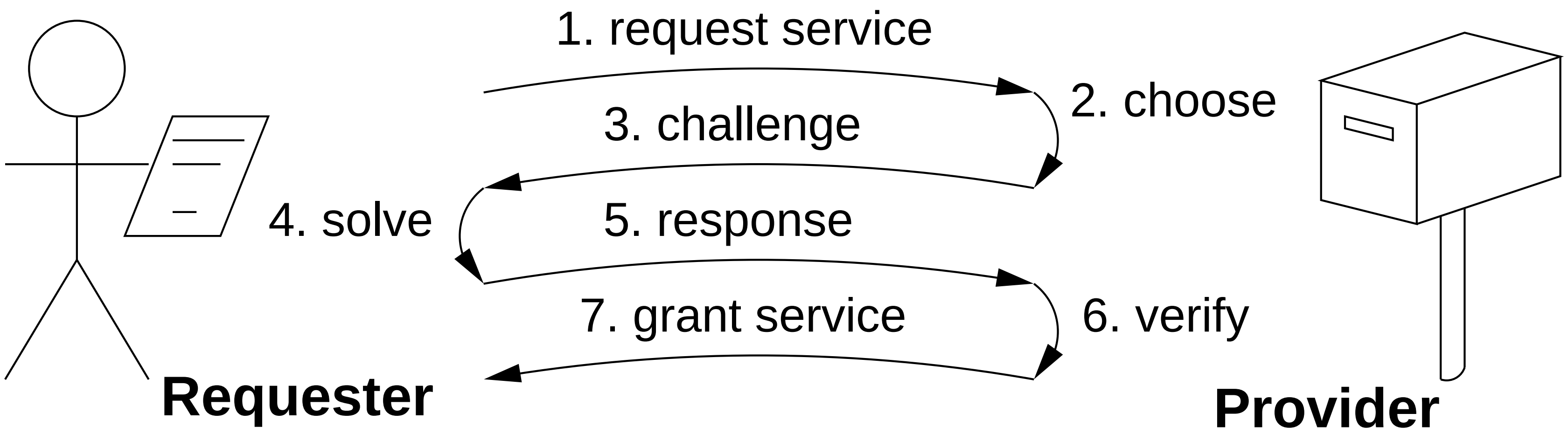
060320 valid until March 20, 2006

fabien.coelho@... dest. address

14b3bb counter is $1,356,731$

$$\mathtt{SHA1}(stamp) = \mathtt{0000006e0dfbac6d6664d4afc028aa767ac98275}$$

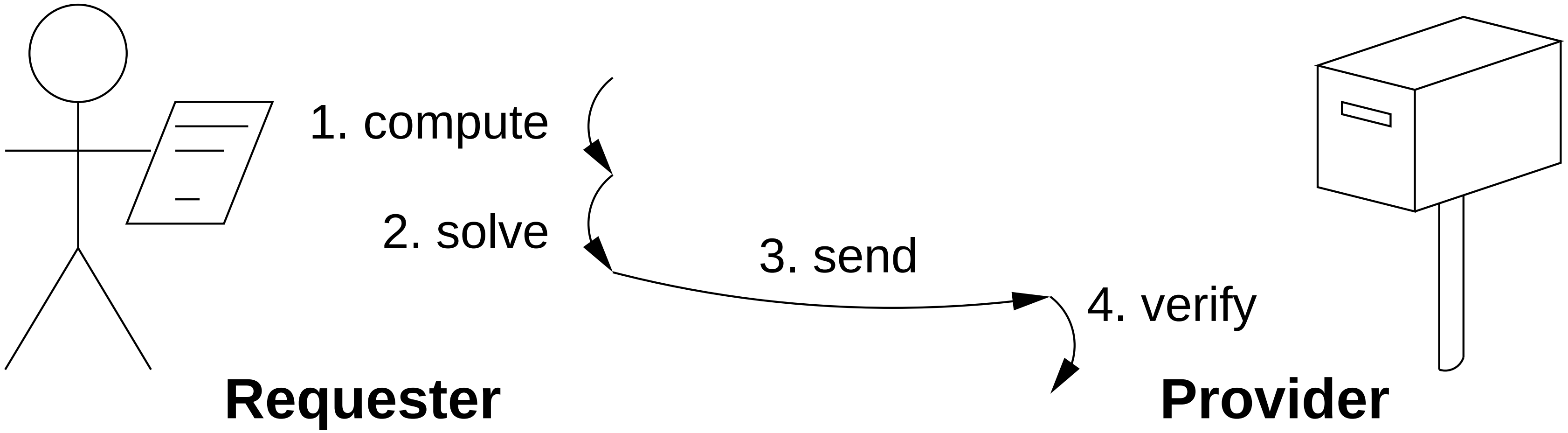
Challenge-Response



interactive bounded schemes, small variance

bounded search, find an item with some property in a finite set

Solution-Verification



one-way schemes as HashCash : must check problem and solution

unbounded probabilistic search, stdev equals average (long tail)

trial success proba $\frac{1}{N}$, $e^{-\frac{i}{N}}$ no-success after i iters, $e^{-4} \approx \frac{1}{50}$

Deterministic bounded solution-verification scheme?

possible? **YES!** Dwork and Naor Crypto'92

integer square root modulo a large prime $p \equiv 3 \bmod 4$

optimality? **NO!** solution p^3, communication p, verification p^2

complexity depends on multiplication/root-squaring algorithm

Better scheme?

1. bounded solution
2. small proof
3. quick verification

Outline

- Proof of Work and optimality
- Lamport signature and Merkle tree
- bounded scheme and feedback proof
- attack cost lower bound
- iterative attack
- conclusion

Measures

effort solution work from the requester $E(w)$

communication volume from requester to provider $C(w)$

checking work computation by provider w

work ratio requester work to provider work $\frac{E(w)}{w}$

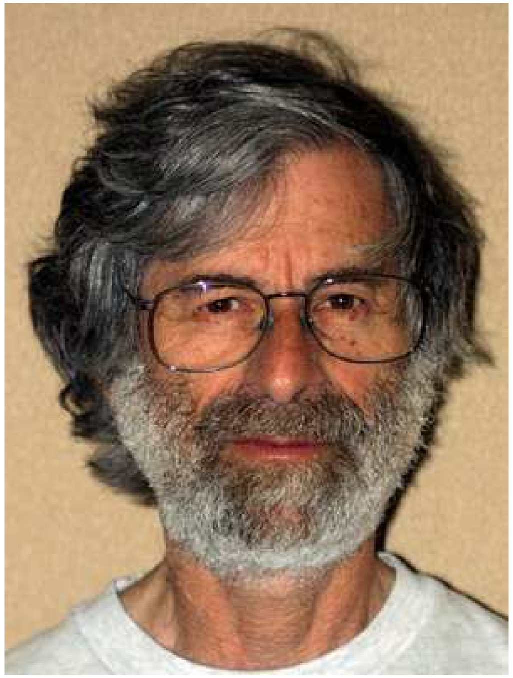
Two Optimality Criteria

communication volume is minimum $C(w) = \log\left(\frac{E(w)}{w}\right)$

computation check is minimum $C(w) = w$

verification is linear in the received data

Lamport signature scheme



- Alice publishes the hashes of two secrets

$$x_0 = h(s_0), \quad x_1 = h(s_1)$$

- Bob proposes: *would you marry me?*
- Alice one-bit answer is signed:

no by returning s_0

yes by returning s_1

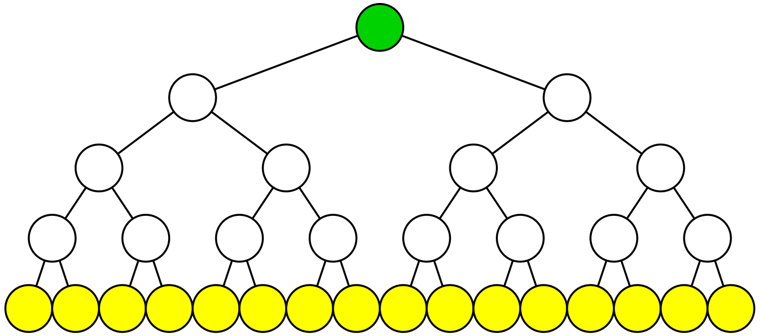
- Bob checks with published hashes

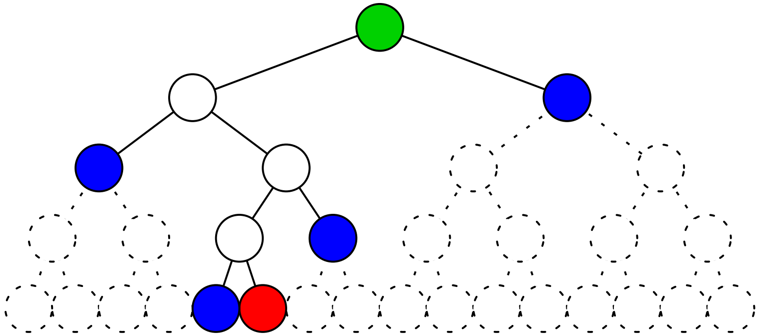
Requires publishing a lot of hashes. . .

Merkel tree

- (binary) hash tree
- aggregate many hashes
  - tree leaves are hashes of secrets
  - build binary tree $\mathrm{n} = h(\mathrm{left}||\mathrm{right})$
  - publish only root hash n_0
- with Lamport signature

  intermediate hashes show that a leaf belongs to the tree



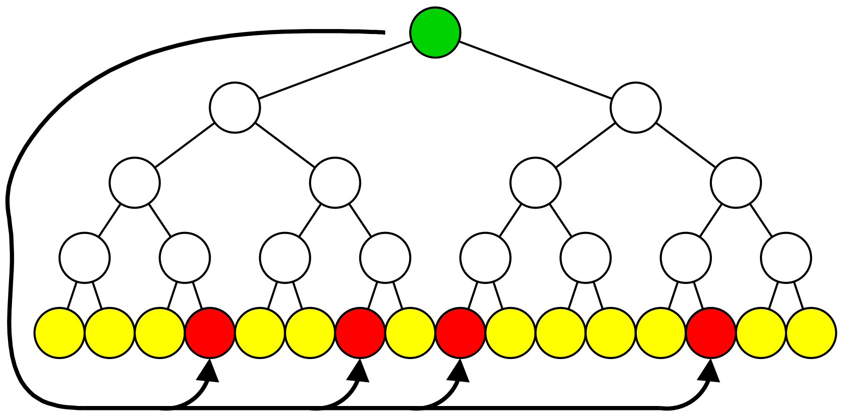


WORK: Merkle tree

- bounded $2N$ hash computations
- D service description `hobbes@comics:20080611:0001`
- $s = h(D)$ service hash `617afdd5b0c61464f33c24d25762ee3b` 1
- $h_s(x) = h(x\|s)$ service-dependent hash function
- $N = 2^d$ number of leaves from tree depth
- $n_{N-1+i} = h_s(i)$ hashes for each leaf number i N
- $n_i = h_s(n_{2i+1}\|n_{2i+2})$ internal node hashes, root hash n_0 $N-1$

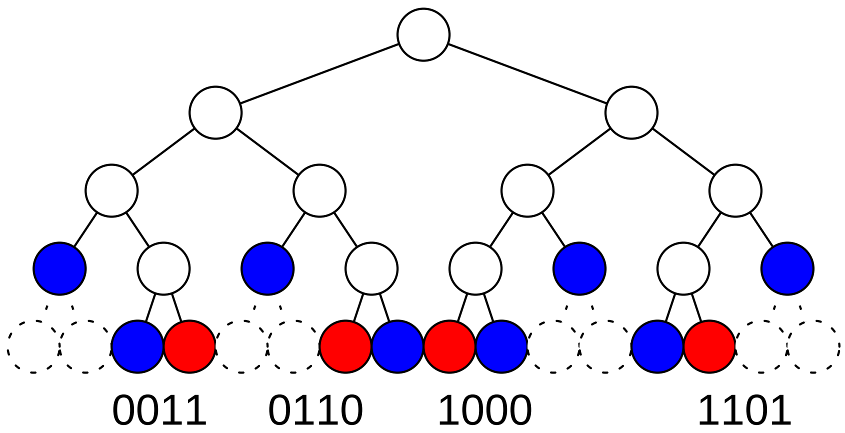
PROOF

- a subset of P leaves selected from n_0
- $r = \mathcal{S}(n_0)$ pseudo-random generator seed
- $\ell_j = \mathcal{G}(r, j)$ pseudo-random leaf numbers to return in $\frac{N}{P}$-size chunks
- **feedback**: selected leaves depend on the *whole* computation



Communication

- send proof that leaves belong to the Merkle tree
- D, ℓ_j for $j \in (0 \ldots P-1)$, inner hashes
- volume is about $P \cdot \log_2(N)$



(Fast) Verification

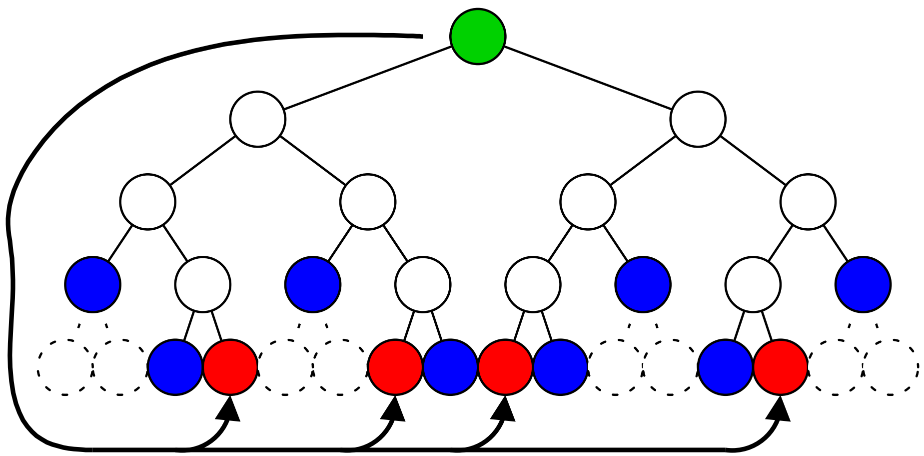
- **consistency** of selected leaves

  recompute ℓ_j from provided data

- $s = h(D)$, $n_{N-1+\ell_j} = h_s(\ell_j)$,

  $n_0 = \ldots$, $r = \mathcal{S}(n_0)$, re-derive ℓ_j from r

- costs $P \cdot \log_2(N)$ computations



How many leaves?

Choice of Parameters

tree depth $d = 22, \quad N = 2^{22}$

hash function strong cryptographic

to avoid inversions or collisions

hash size m may vary

small in lower tree $m \approx 24$

large in upper tree and for service $m \approx 160$

PRNG seed $r = h_s^P(n_0)$ (P compositions)

number of proofs $P = 8 \cdot \log_2(N)$

induces $w = \mathcal{O}(\ln(N)^2)$, proof volume is 11KB

Why is this P okay?

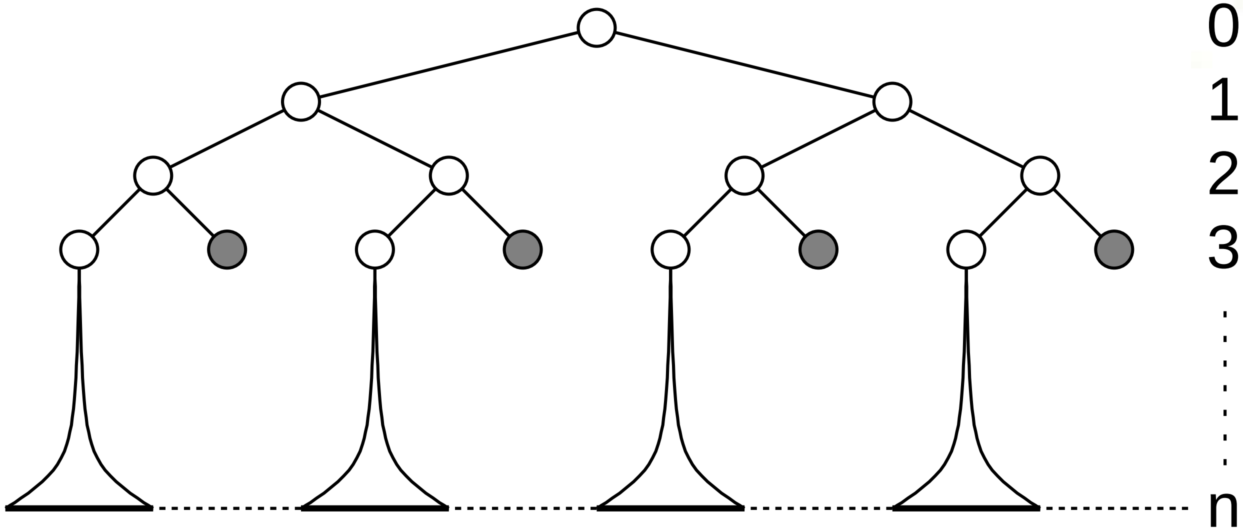
Partial tree attacks

fraction f of actual leaves plus fake hashes

valid feedback probability f^P per trial

mix of iterative/extension strategies

constant f or increasing f



Attack cost lower bound

target a valid accepted partial tree

strong hypothesis any mixed strategy!

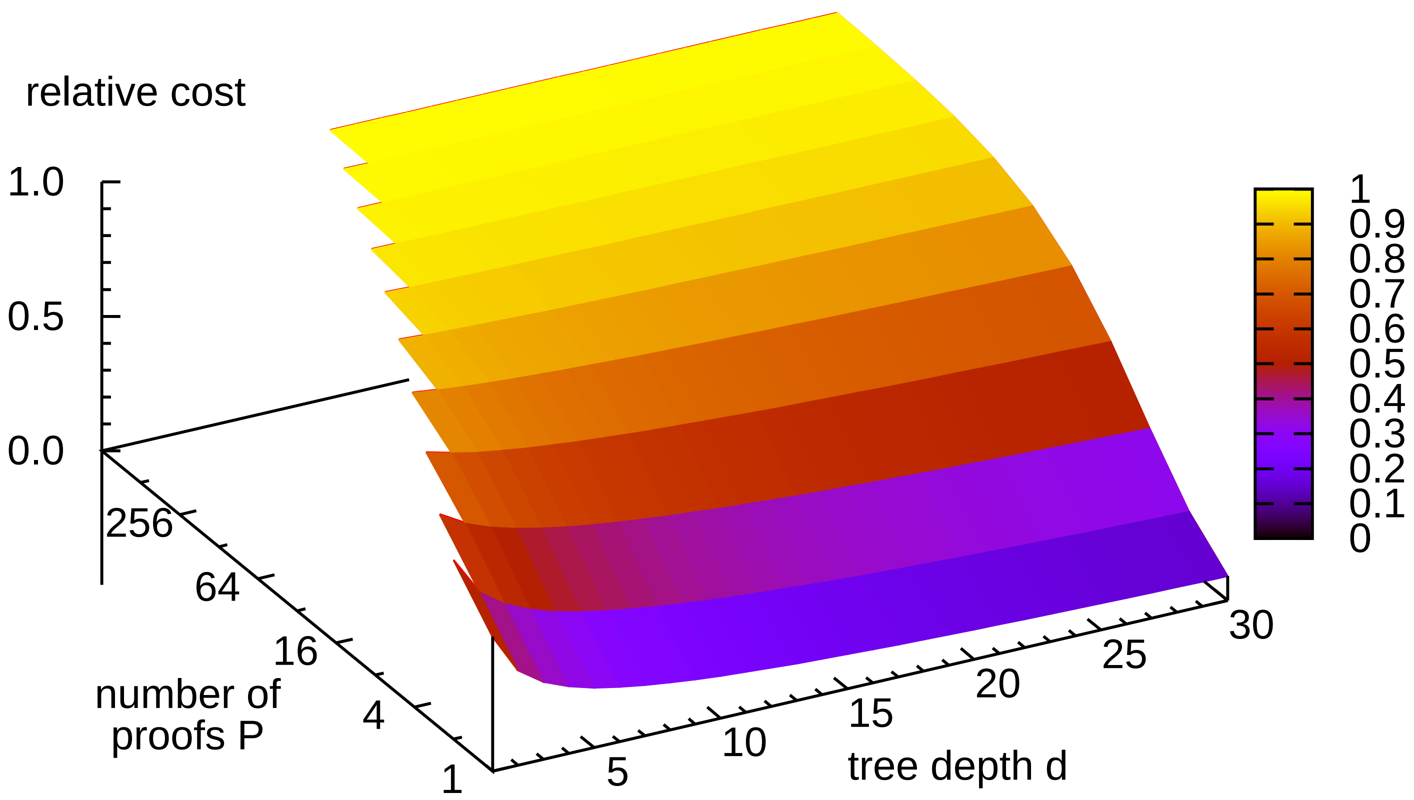
every leaf tested at no added cost

$$\mathcal{C}(N,P) \geq \left(\frac{1}{N}\right)^{\frac{1}{P+1}} \cdot \frac{P}{P+1} \cdot (2N)$$

lower bound 90% of full $2N$ cost with $d \geq 7$

$$\mathcal{C}(N) \geq \left(\frac{1}{2}\right)^{\frac{1}{8}} \cdot \frac{8 \cdot \log_2(N)}{8 \cdot \log_2(N) + 1} \cdot (2N) \geq 0.9 \cdot (2N)$$

Lower bound relative to full cost



Iterative attack

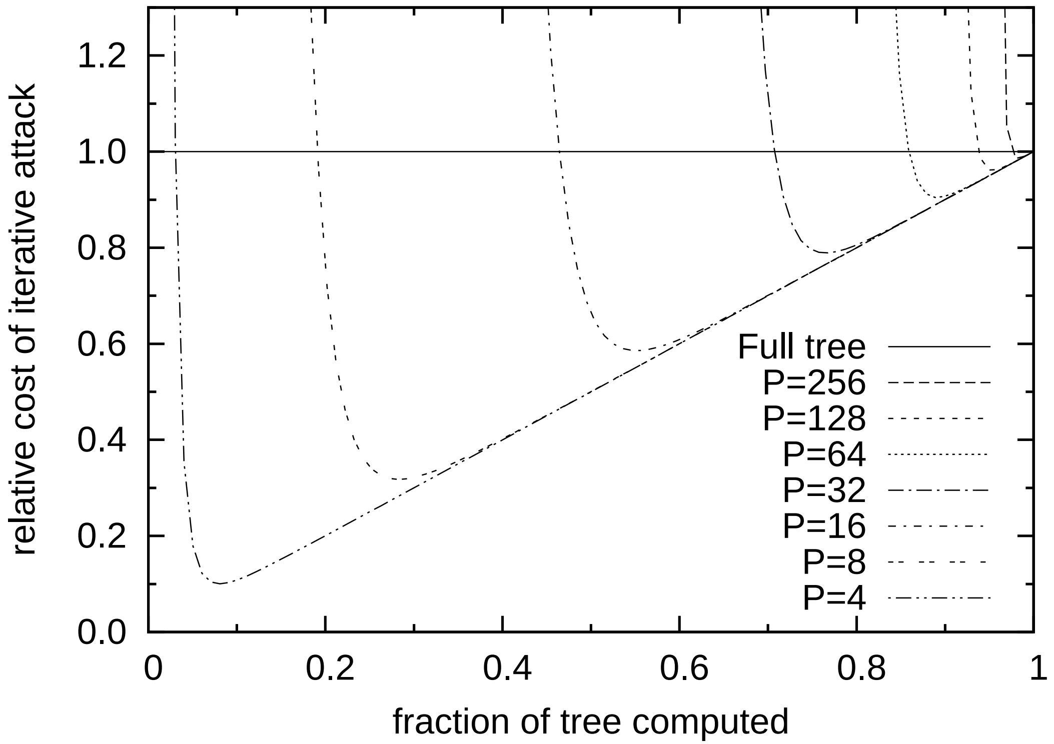
- iterations at constant f
- partial tree + iterative cost

$$\mathcal{C}_{\text{iter}}(f,N,P) \approx 2Nf + (P + \log_2(P) + 1)\frac{1}{f^P}$$

- optimal fraction f

$$\mathcal{F}(N,P) = \sqrt[P+1]{\frac{P\,(P + \log_2(P) + 1)}{2N}}$$

Relative cost of iterative attack



best fraction

$$\mathcal{F}(2^{22}, 256) = 0.981$$

relative cost

$$\mathcal{C}(0.981, 2^{22}, 256) = 0.989$$

Contributions

optimality criteria for POW schemes

1. communication optimal
2. computation optimal

   vs DOS attack on POW

bounded solution-verification POW

effort is $e^{\sqrt{w}}$

computation optimal, not communication optimal

conservative lower bound on attack cost

at least 90% of the full cost

interative attack with a small 1% gain

the attack is probabilistic, thus unbounded

Conclusion

- bounded solution-verification scheme
- solution work is well known, null or small variance (almost)
- but verification is probabilistic!

Future work in POW?

- not the ultimate solution against spams. . .
- try to publish about memory-bound POW functions