Optimal Verification of Operations on Dynamic Sets

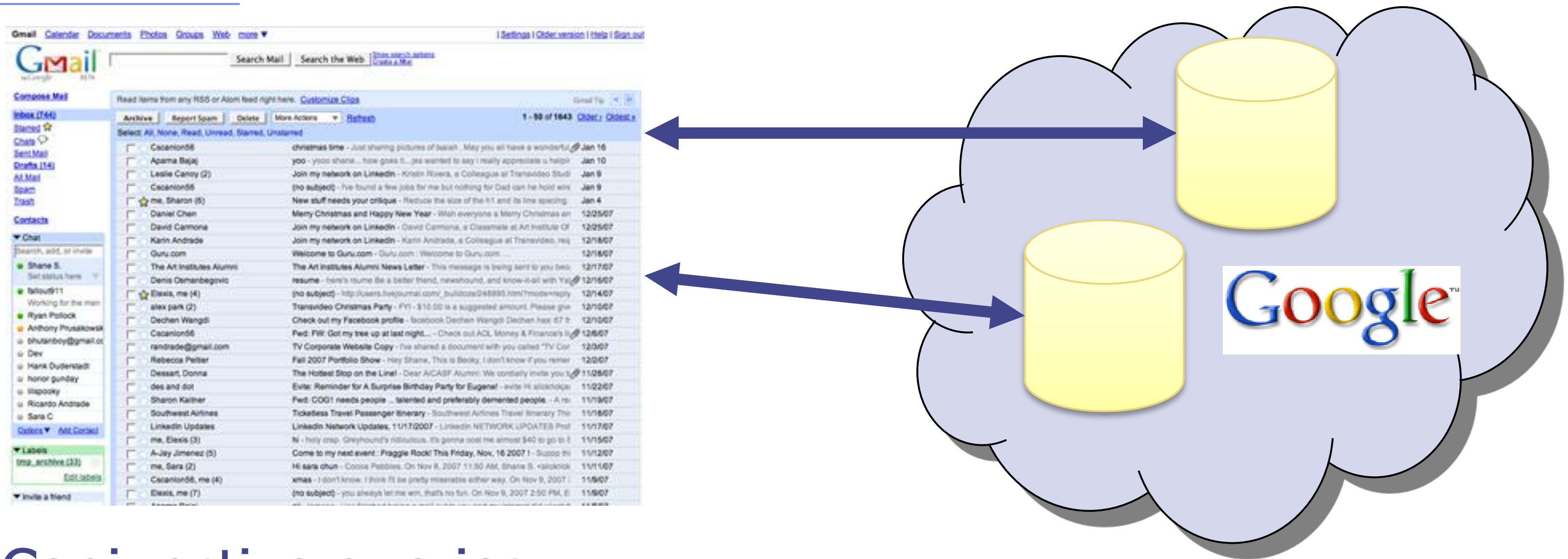
Charalampos Papamanthou, UC Berkeley
Roberto Tamassia, Brown University
Nikos Triandopoulos, RSA Labs & BU

CRYPTO 2011
08/15/11

Data in the cloud

- Data privacy
  - Server wants to learn our data
  - Can we enable the server use encrypted data in a meaningful way?
  - **Computing on encrypted data**
- Data and computations integrity
  - Server wants to tamper with our data
  - Are answers to queries the same as if the data were locally stored?
  - **Authenticated data structures**
  - **Verifiable delegation of computation**

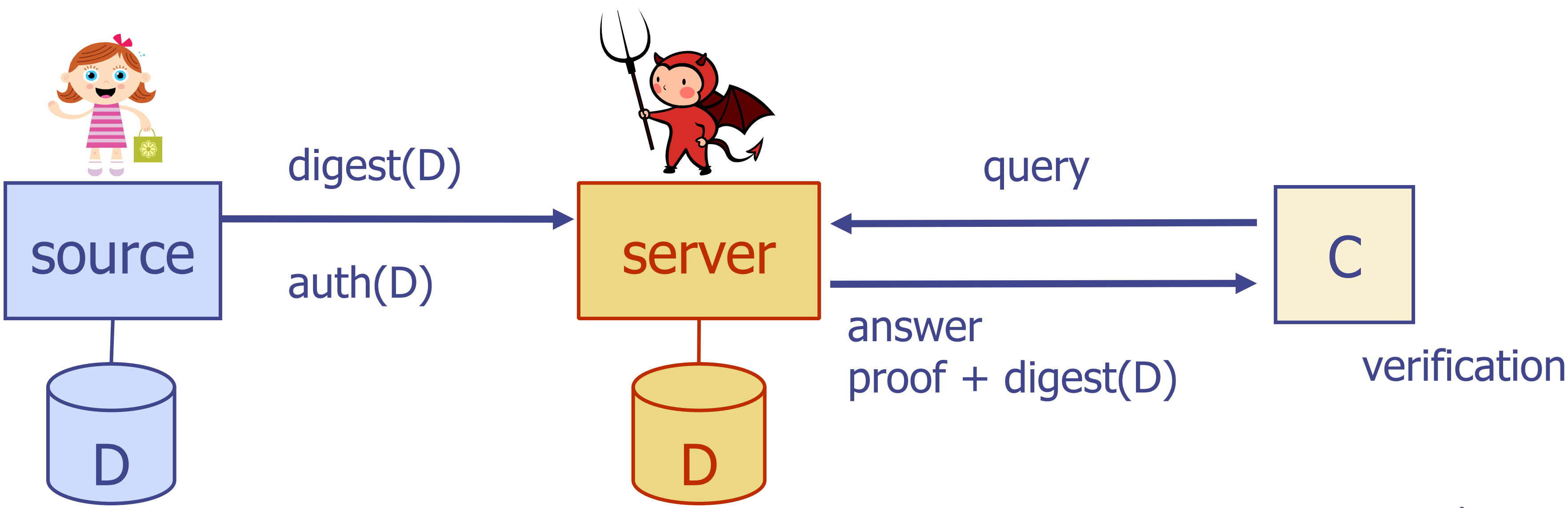
Verifying outsourced computation



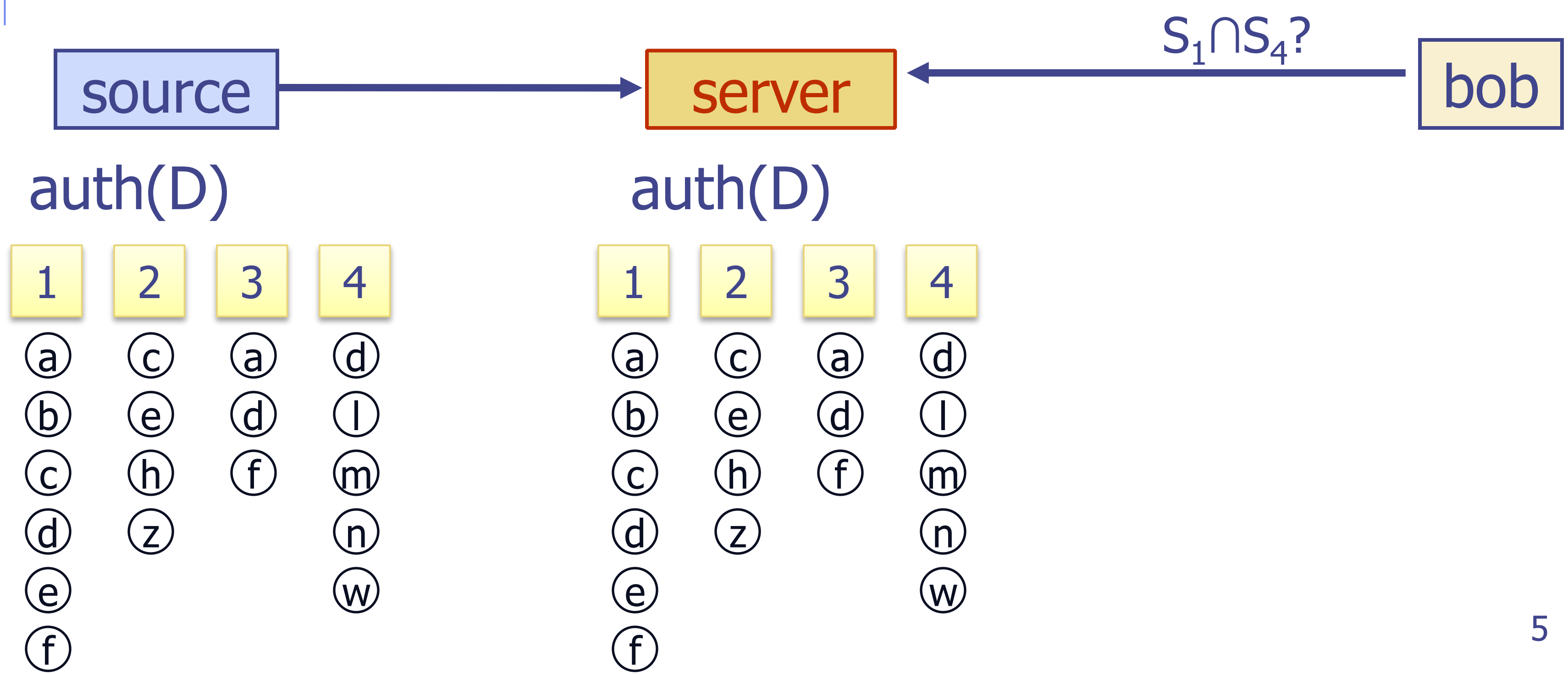
- Conjunctive queries
  - Emails that have the terms “Brown” and “Berkeley”
- Disjunctive queries
  - Emails that have the terms “thesis” or “publication”
- All these queries boil down to **set operations**!

Authenticated data structures model

- Complexity
  - Update at source and server
  - Query at server
  - Verification at client
  - Size of proof
  - Space
- Security
  - A poly-bounded adversary cannot construct invalid proofs except with negligible probability
  - Need for computational assumptions

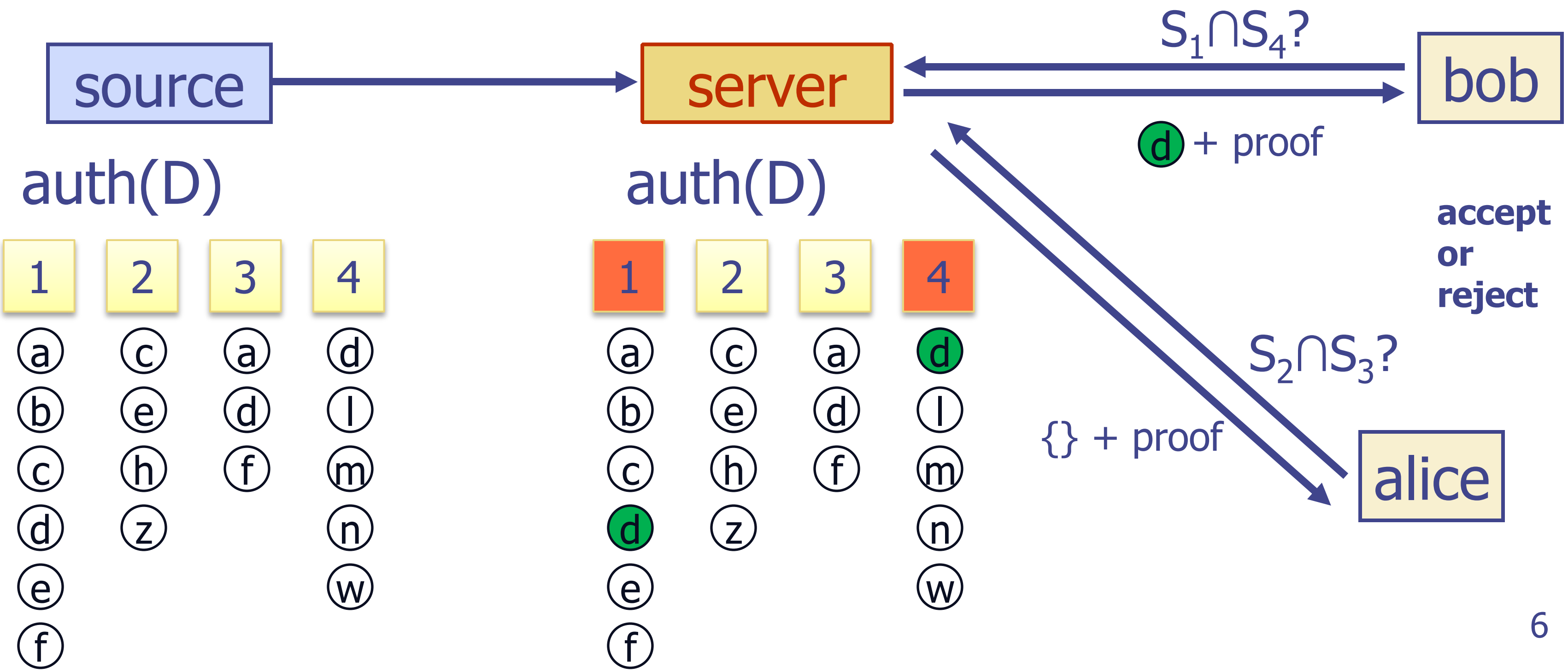


Authenticated sets collection



Queries on sets

- m: number of sets (e.g., m = 4)
- M: sum of sizes of **all** the sets (e.g., M = 6 + 4 + 3 + 5 = 18)
- t: number of queried sets (e.g., t = 2)
- δ: number of elements contained in the **answer** (e.g., δ = 1)
- n: the sum of sizes of the queried sets (e.g., n = 6 + 5 = 11)



Related work and comparison

- Optimal proof size and verification time: O(δ)
- Linear space: O(m + M)
- Efficient queries and updates
- Performance comparison for the intersection of c = O(1) sets

	space	query	proof	assumption
D+04 YP09	m + M	n + log m	n + log m	Generic CR
M+04	m + M	n	n	Strong RSA
PT04	m^c	1	δ	Discrete log
PTT10	m + M	$n \log^3 n + m^\varepsilon \log m$	δ	Bilinear q-strong DH

Our solution: Sets and polynomials

- Set X with n elements

  $X = \{x_1,\ldots,x_n\}$
- Set Z is the intersection of X and Y
- The intersection of X and Y is empty, i.e.,
  $X \cap Y = \varnothing$

- Polynomial X(s) in Zp

  $X(s) = (s+x_1)\ldots(s+x_n)$
- Polynomial Z(s) is the GCD of X(s) and Y(s)
- X(s) and Y(s) have GCD equal to 1, i.e.,
  $\gcd(X(s),Y(s)) = 1$

  $\Leftrightarrow$
- There are polynomials P(s) and Q(s) such that

  $P(S)X(s) + Q(s)Y(s) = 1$

Cryptographic tools we use

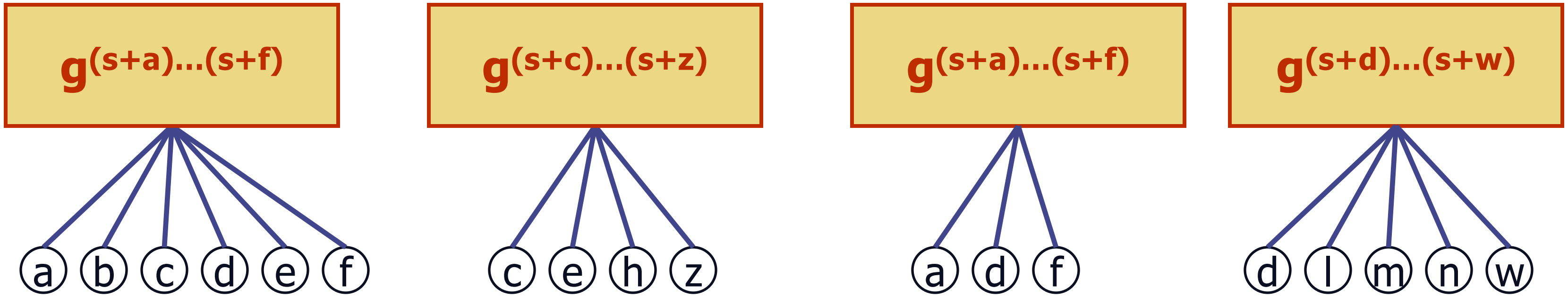
- Two multiplicative groups G and T of prime order p
- g is a generator of G
- A bilinear map e(.,.) from G to T such that
  - $e(g^a, g^b) = e(g,g)^{ab}$ for all a,b in Zp
  - e(g,g) generates T
- Bilinear q-strong Diffie Hellman Assumption
  - Pick a random s in Zp
  - s is the trapdoor
  - Compute $g^s, g^{s^2}, g^{s^3}, \ldots, g^{s^q}$
  - The public key pk are the values $g^s, g^{s^2}, g^{s^3}, \ldots, g^{s^q}$
  - The probability that a PPT Adv can find an a in Zp and output the tuple $(a, e(g,g)^{1/(s+a)})$ is negligible

Bilinear-map accumulator

- G and T of order p have a map e(.,.)
- $X=\{x,y,z,r\}$ in Z_p
- Base $g \in G$, generator of G
- Secret $s \in Z_p$
- Digest
  - **$D = g^{(x+s)(y+s)(z+s)(r+s)}$**
- Witness for x
  - **$W_x = g^{(y+s)(z+s)(r+s)}$**
- Verification
  - **$e(D,g) = e(W_x, g^{(x+s)})$?**
- Security: q-strong Diffie-Hellman assumption
- [Nguyen (05)]

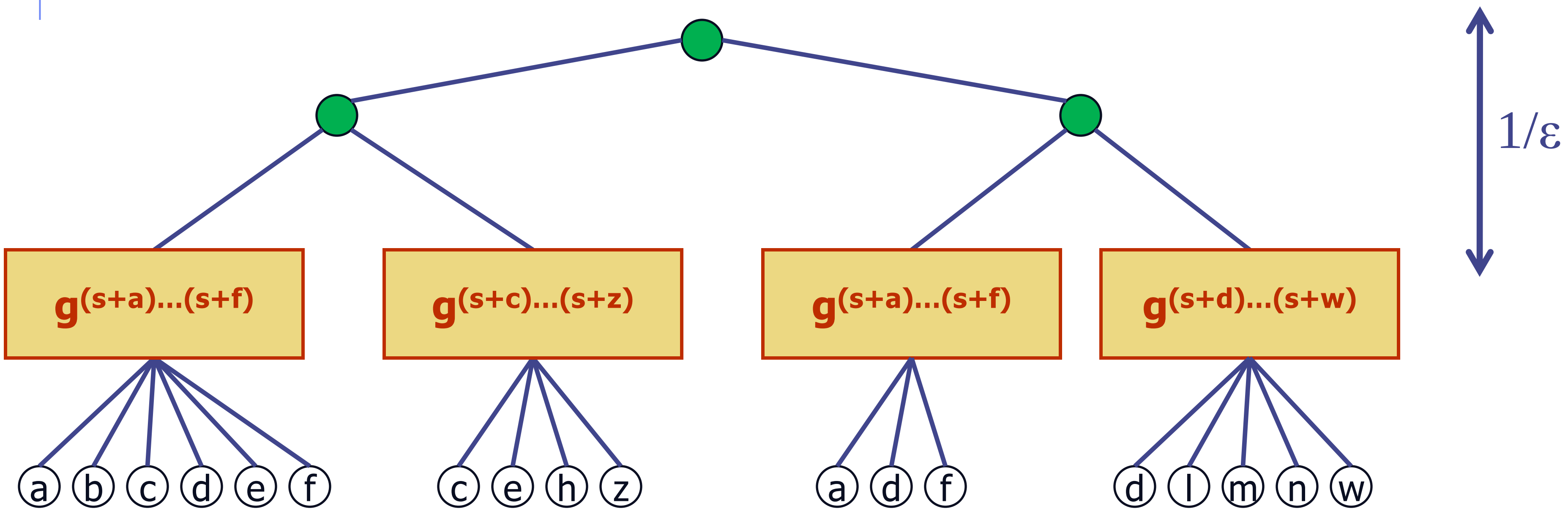
Our construction

- Compute the accumulation value for every set

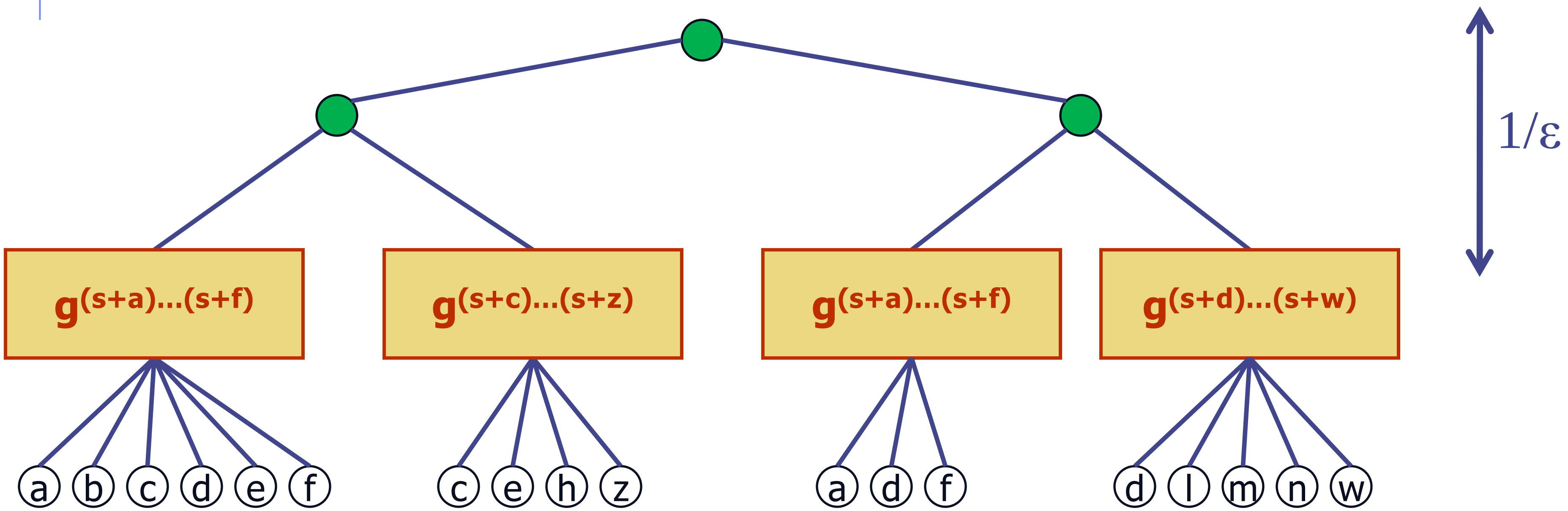


Our construction

- Compute the accumulation value for every set
- Build an *accumulation tree* on top [CCS 2008]
  - $O(1/\varepsilon)$ levels and $O(m^{\varepsilon})$ internal degree
  - $O(m^{\varepsilon}\log m)$ query, $O(1)$ update and $O(1)$ proof
- The accumulation values protect the integrity of the set elements
- The accumulation tree protects the integrity of the acc. values

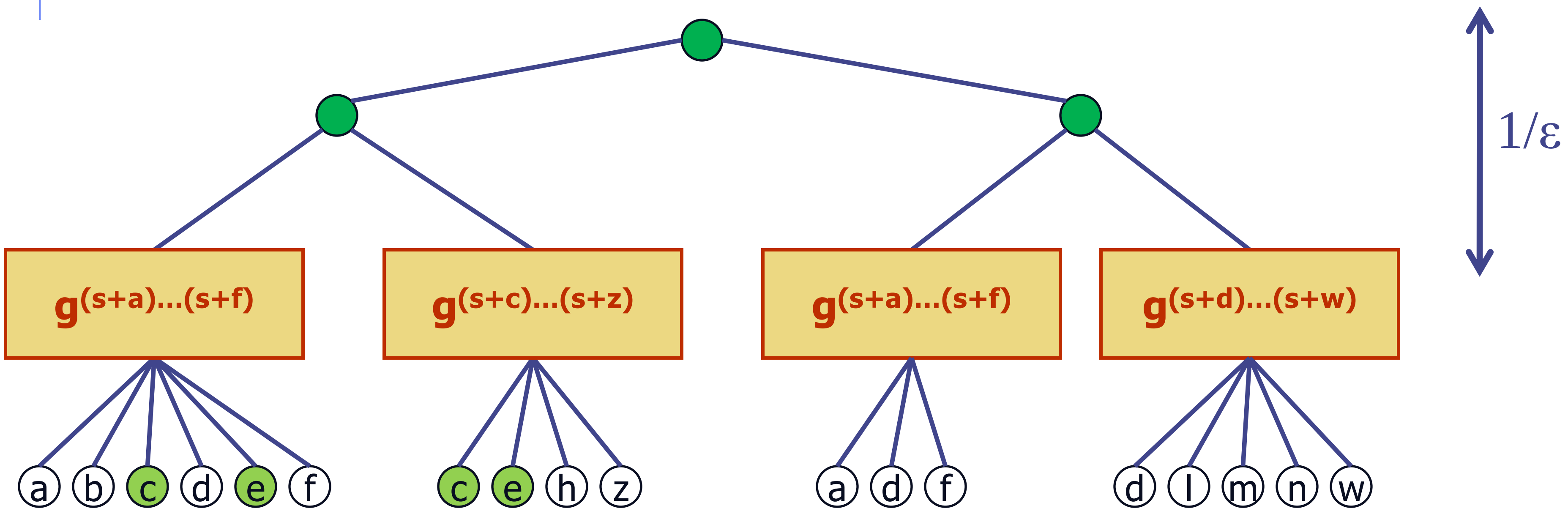


Proof of intersection $I = S_1 \cap S_2$



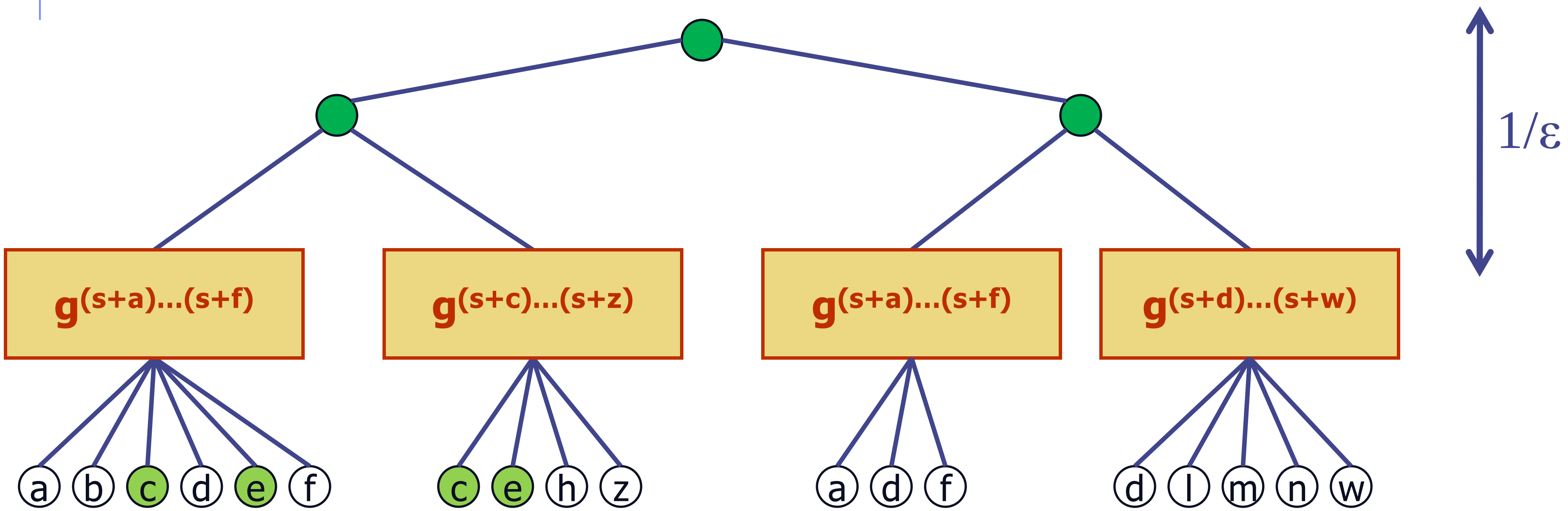
Proof of intersection $I = S_1 \cap S_2$

- **Elements of intersection** {c,e}



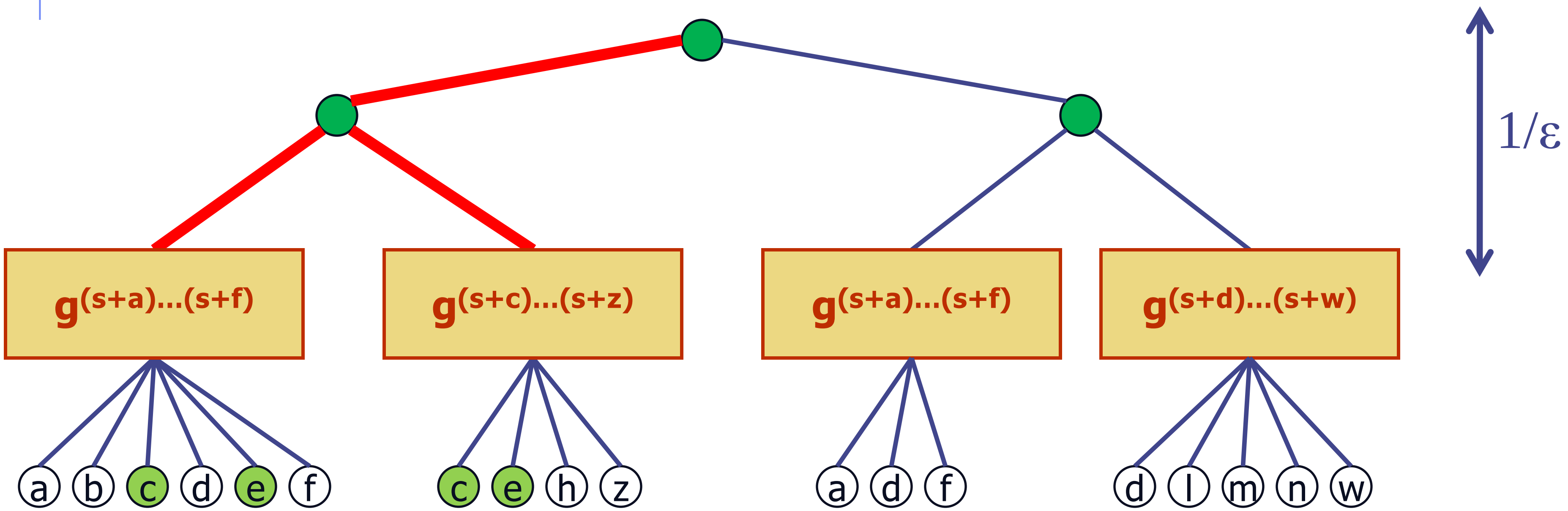
Proof of intersection $I = S_1 \cap S_2$

- **Proof of accumulation values** A_1 and A_2
- Let Π_1 and Π_2 be such proofs



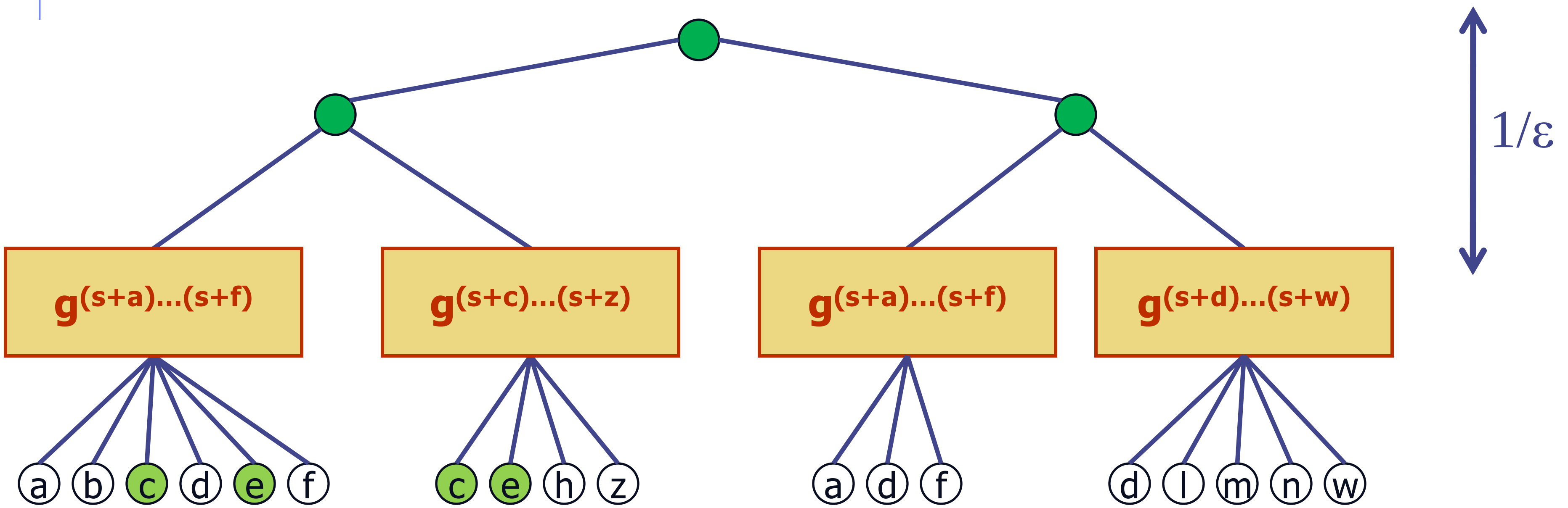
Proof of intersection $I = S_1 \cap S_2$

- **Proof of accumulation values** A_1 and A_2
- Let Π_1 and Π_2 be such proofs
  - Values along the path of the tree
  - Construction of proofs: $O(m^{\varepsilon} \log m)$
  - Size of proofs: $O(1)$



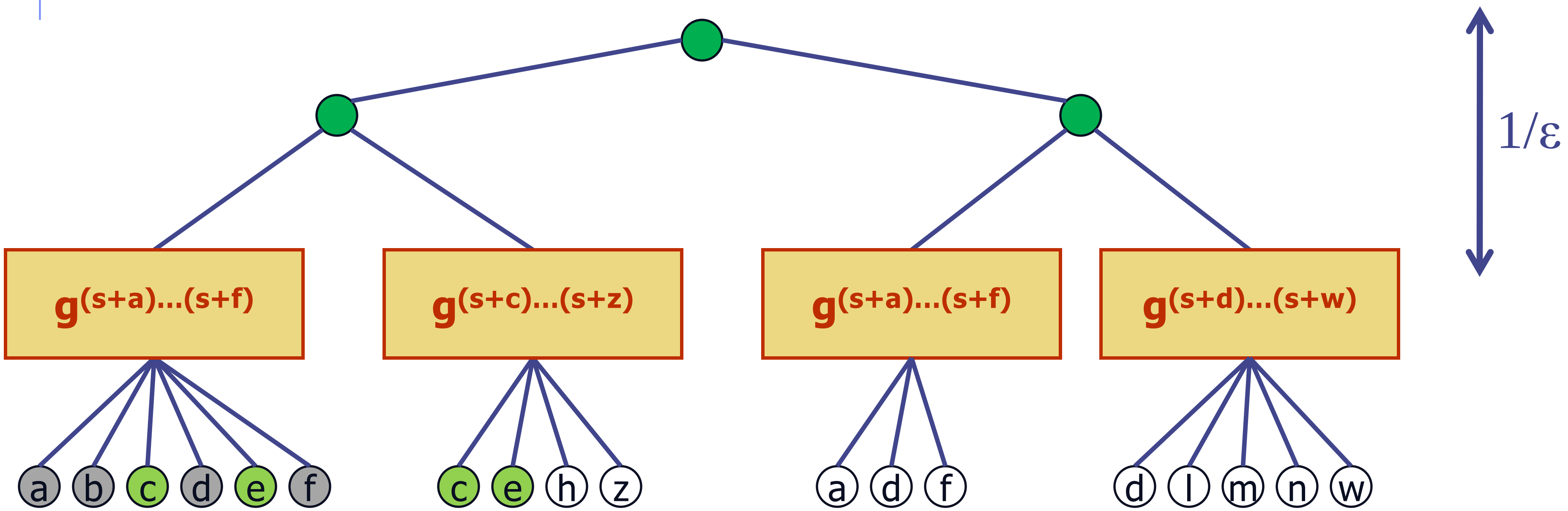
Proof of intersection $I = S_1 \cap S_2$

- **Subset condition**:



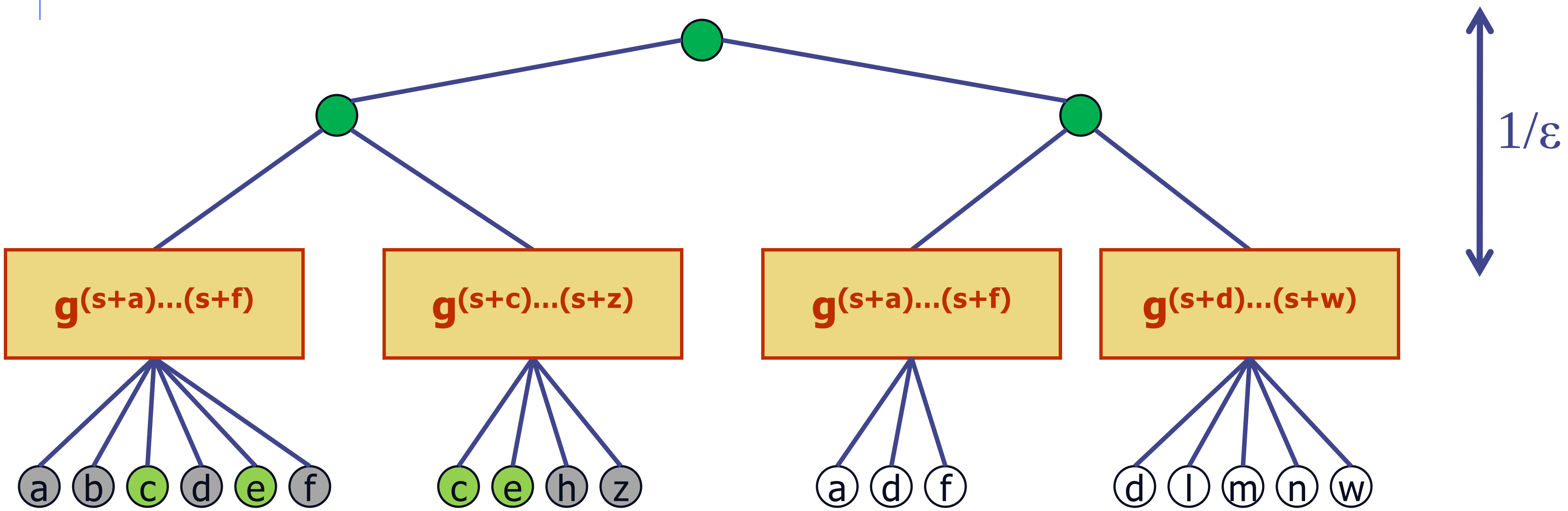
Proof of intersection $I = S_1 \cap S_2$

- **Subset condition**:
  - $I \subseteq S_1$: Subset witness $W_1 = g^{(s+a)(s+b)(s+d)(s+f)} = g^{P(s)}$



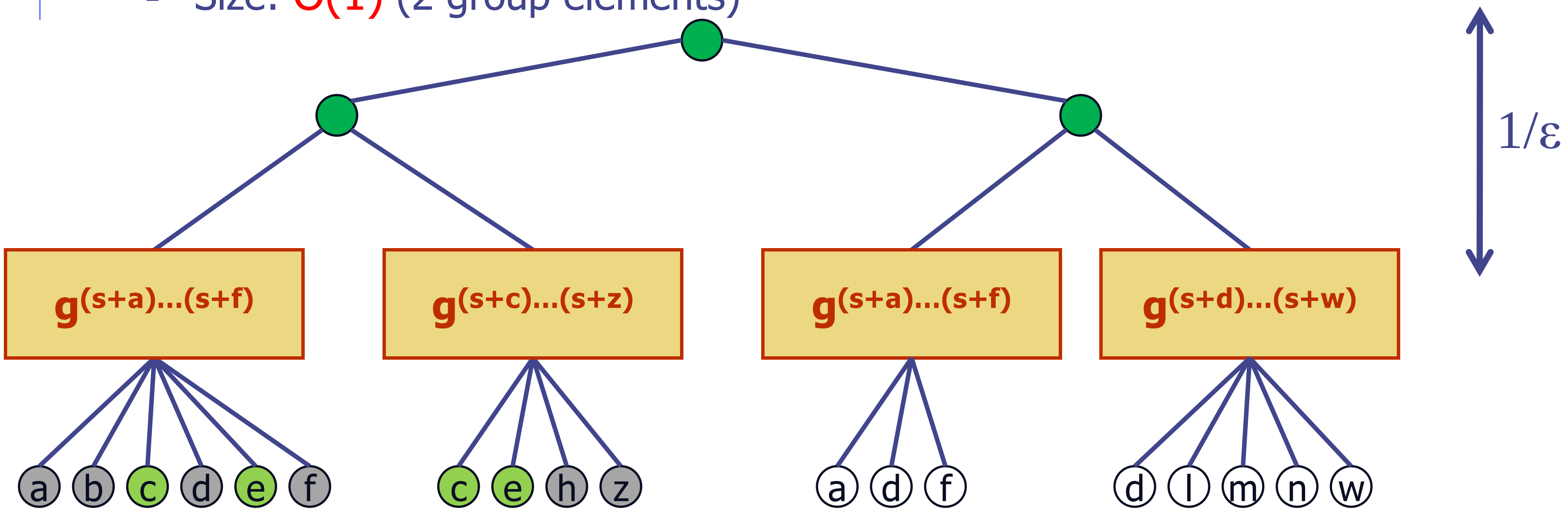
Proof of intersection $I = S_1 \cap S_2$

- **Subset condition**:
  - $I \subseteq S_1$: Subset witness $W_1 = g^{(s+a)(s+b)(s+d)(s+f)} = g^{P(s)}$
  - $I \subseteq S_2$: Subset witness $W_2 = g^{(s+h)(s+z)} = g^{Q(s)}$



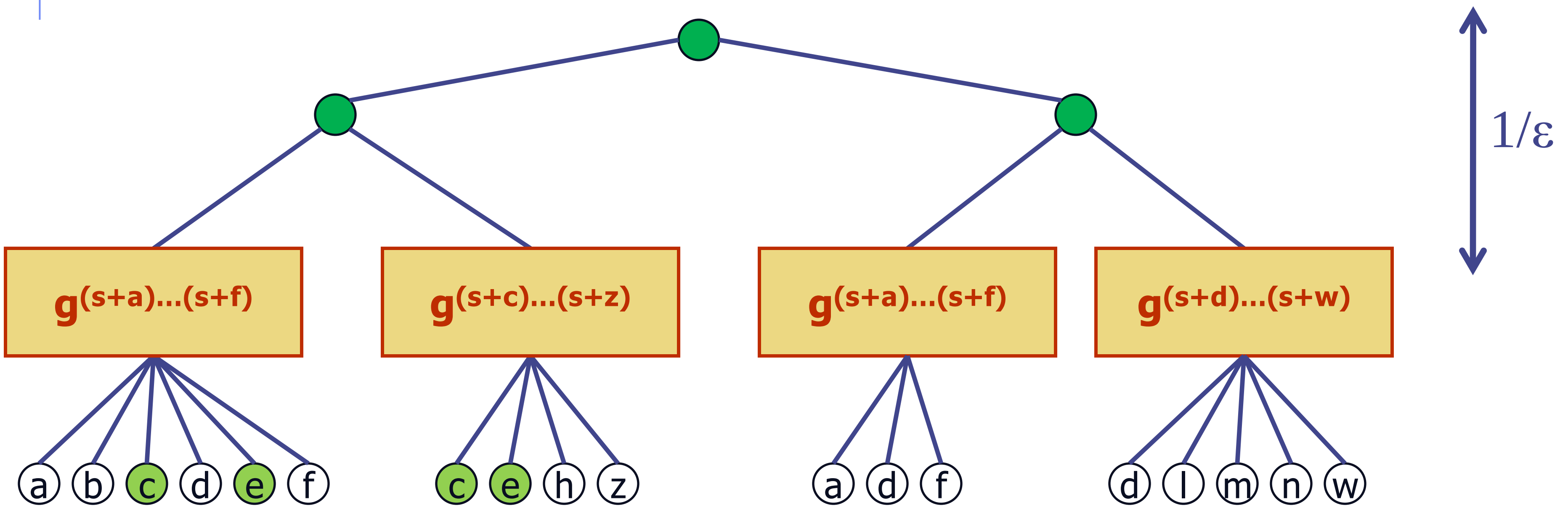
Proof of intersection $I = S_1 \cap S_2$

- **Subset condition**:
  - $I \subseteq S_1$: Subset witness $W_1 = g^{(s+a)(s+b)(s+d)(s+f)} = g^{P(s)}$
  - $I \subseteq S_2$: Subset witness $W_2 = g^{(s+h)(s+z)} = g^{Q(s)}$
- Complexity
  - Construction: O(nlog n) (polynomial interpolation)
  - Size: O(1) (2 group elements)



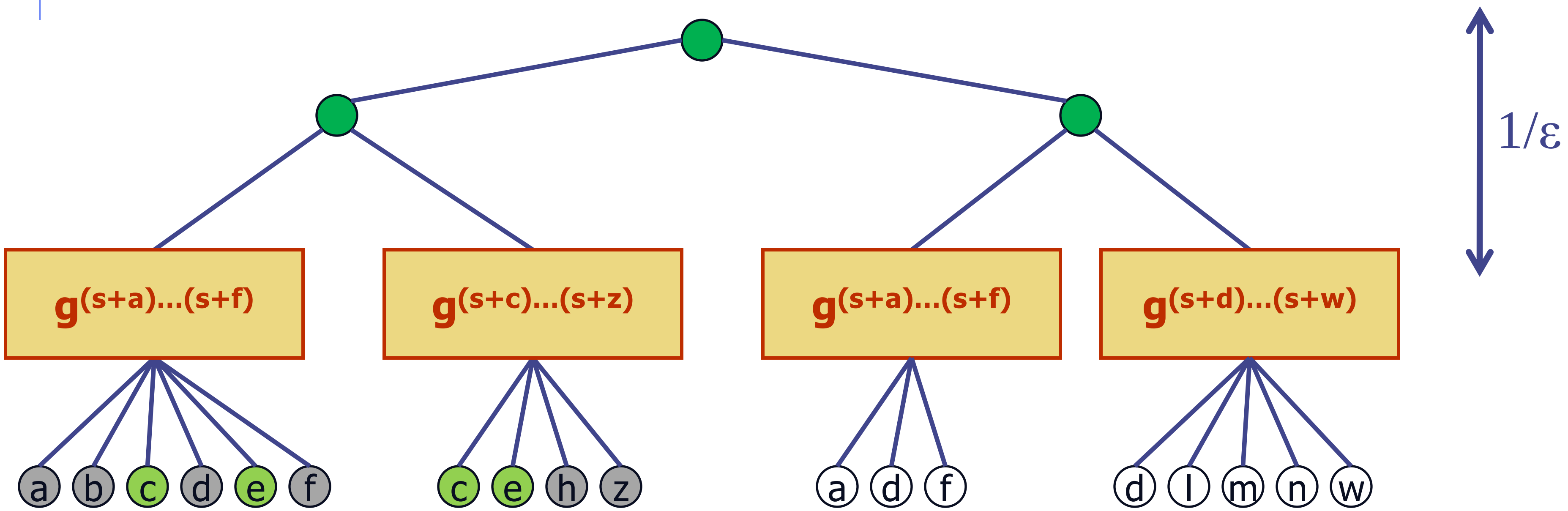
Proof of intersection $I = S_1 \cap S_2$

- **Completeness condition**:
  - $(S_1 - I) \cap (S_2 - I)$ is empty



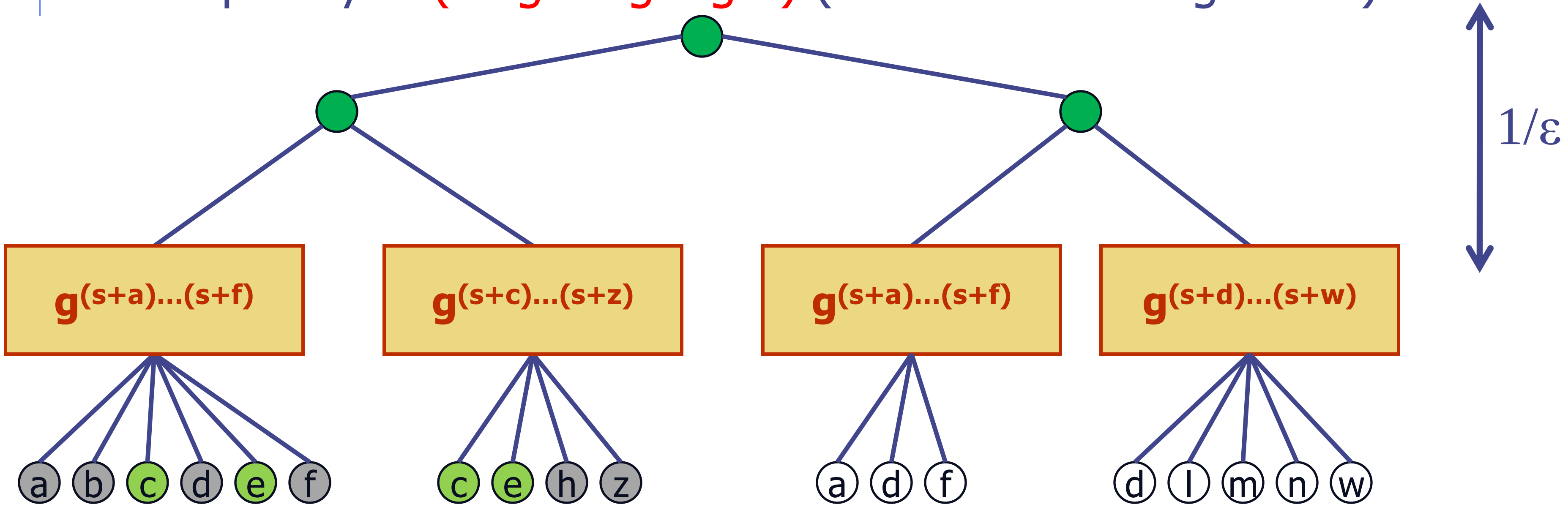
Proof of intersection $I = S_1 \cap S_2$

- **Completeness condition**:
  - $(S_1 - I) \cap (S_2 - I)$ is empty
  - Recall $W_1 = g^{P(s)}$ and $W_2 = g^{Q(s)}$



Proof of intersection $I = S_1 \cap S_2$

- **Completeness condition**:
  - $(S_1 - I) \cap (S_2 - I)$ is empty
  - Recall $W_1 = g^{P(s)}$ and $W_2 = g^{Q(s)}$
  - Completeness witness $F_1 = g^{A(s)}$ and $F_2 = g^{B(s)}$
  - **$A(s)P(s)+B(s)Q(s) = 1$**
- Complexity: $O(n\log^2 n \log\log n)$ (ext. Euclidean algorithm)



Recap

- **t** sets are intersected and **δ** is the size of the answer
- **N** is the sum of sizes of intersected sets

element of the proof	complexity	size
Intersection elements	N	δ

Recap

- **t** sets are intersected and **δ** is the size of the answer
- **N** is the sum of sizes of intersected sets

element of the proof	complexity	size
Intersection elements	N	δ
Accumulation values proofs	$tm^{\varepsilon}\log m$	t

Recap

- **t** sets are intersected and **δ** is the size of the answer
- **N** is the sum of sizes of intersected sets

element of the proof	complexity	size
Intersection elements	N	δ
Accumulation values proofs	$tm^{\varepsilon}\log m$	t
Subset witnesses	$N\log N$	t

Recap

- **t** sets are intersected and **δ** is the size of the answer
- **N** is the sum of sizes of intersected sets

element of the proof	complexity	size
Intersection elements	N	δ
Accumulation values proofs	$tm^{\varepsilon}\log m$	t
Subset witnesses	Nlog N	t
Completeness witnesses	$N\log^2 N \log\log N$	t

Recap

- **t** sets are intersected and **δ** is the size of the answer
- **N** is the sum of sizes of intersected sets

element of the proof	complexity	size
Intersection elements	N	δ
Accumulation values proofs	$tm^{\varepsilon}\log m$	t
Subset witnesses	Nlog N	t
Completeness witnesses	$N\log^2 N \log\log N$	t
TOTAL	$N\log^2 N \log\log N$ + $tm^{\varepsilon}\log m$	t+δ

Recap

- **t** sets are intersected and **δ** is the size of the answer
- **N** is the sum of sizes of intersected sets

element of the proof	complexity	size
Intersection elements	N	δ
Accumulation values proofs	$tm^{\varepsilon}\log m$	t
Subset witnesses	$N\log N$	t
Completeness witnesses	$N\log^2 N\log\log N$	t
TOTAL **almost optimal**	$N\log^2 N\log\log N$ + $tm^{\varepsilon}\log m$	t+δ

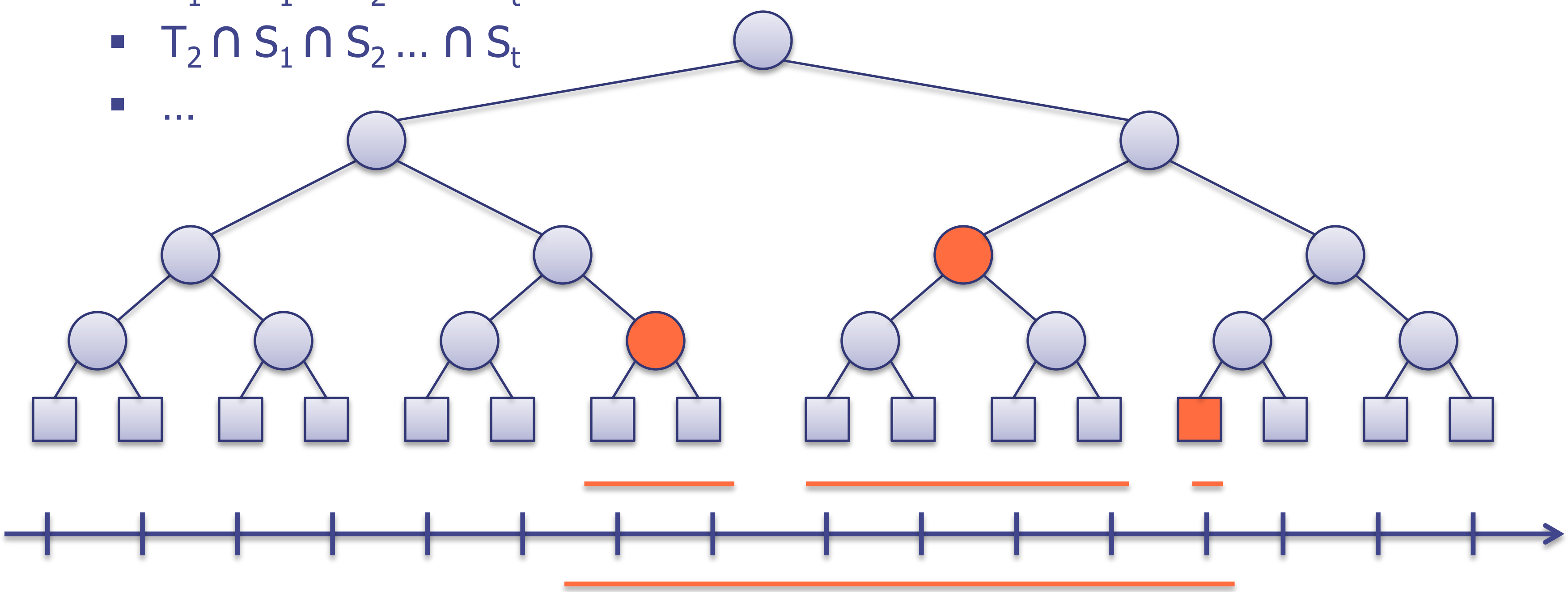
Size of proof for X ∩ Y in practice

|X|	|Y|	|X ∩ Y|	KBytes [M+ 04]	KBytes this work
1000	1000	10	3.34	**1.73**
1000	100	1	1.68	**1.55**
1000	10	0	**1.01**	1.53
1000	1	0	**0.46**	1.53
10000	10000	100	26.88	**3.53**
10000	1000	10	12.15	**1.73**
10000	100	1	6.86	**1.55**
10000	10	0	3.08	**1.53**
100000	100000	1000	263.25	**21.53**
100000	10000	100	116.13	**3.53**
100000	1000	10	63.18	**1.73**
100000	100	1	26.29	**1.55**

Thank you!

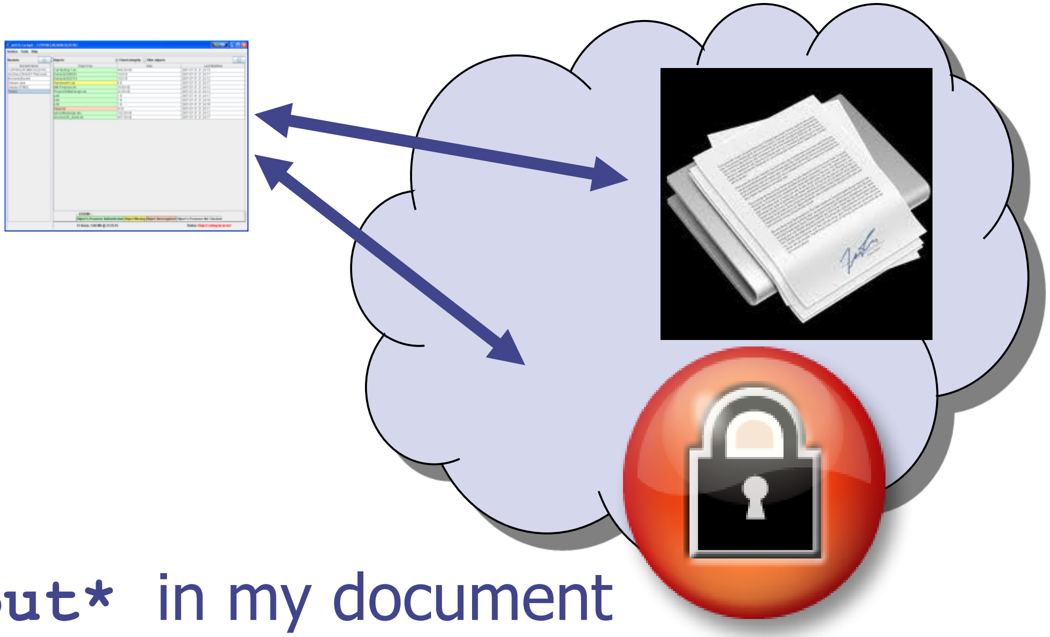
Application: Supporting timestamps

- For timestamped documents, use segment tree over the time dimension (N timestamps)
- Search interval covered by O(log N) canonical intervals in the segment tree, each corresponding to a set of documents T_j
- Timestamped keyword search equivalent to O(log N) set intersections
  - $T_1 \cap S_1 \cap S_2 \ldots \cap S_t$
  - $T_2 \cap S_1 \cap S_2 \ldots \cap S_t$
  - …



Verifying outsourced computation

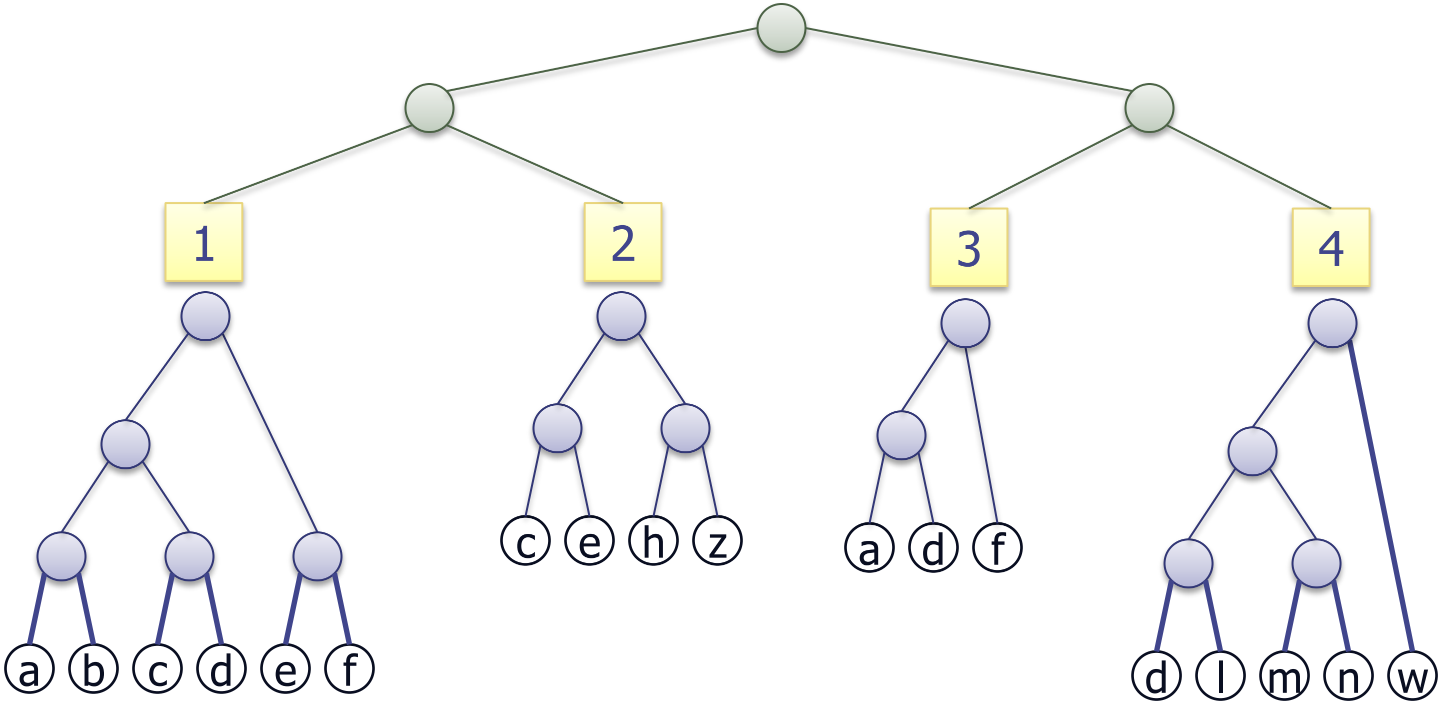
- Computation "on demand"
  - E.g., Google docs
  - …



- Find the pattern `comput*` in my document
- Is the result correct?
- Need for efficient computations

First solution: hashing

- [Devanbu et al., Algorithmica 2004; Yang and Papadias, SIGMOD 2009]
- Two-level tree structure and hierarchical cryptographic hashing
- **Space**: O(m + M), **update**: O(log m + log n)
- Intersection of two sets: O(n + log m) proof size and verification time
- **Security**: Cryptographic hashing
- **Same complexities**: Morselli et al., INFOCOM 2004



Second solution: precomputation

- [Pang and Tan, ICDE 2004]
- Sign the answer to every possible query
- **Space**: $O(m^2 + M)$ for a 2-intersection
- For any possible intersection space is
  - $O(2^m)$
- Proof size and verification: $O(\delta)$
- **Update**: $O(m^2)$ for a 2-intersection
- **Security**: discrete log

Signatures of

$S_1 \cap S_2$

$S_1 \cap S_3$

$S_1 \cap S_4$

$S_2 \cap S_3$

$S_2 \cap S_4$

$S_3 \cap S_4$

…