RSA accumulators

Verifiable Computation (VC)

client

digest δ

function f

result = f(data) +

server

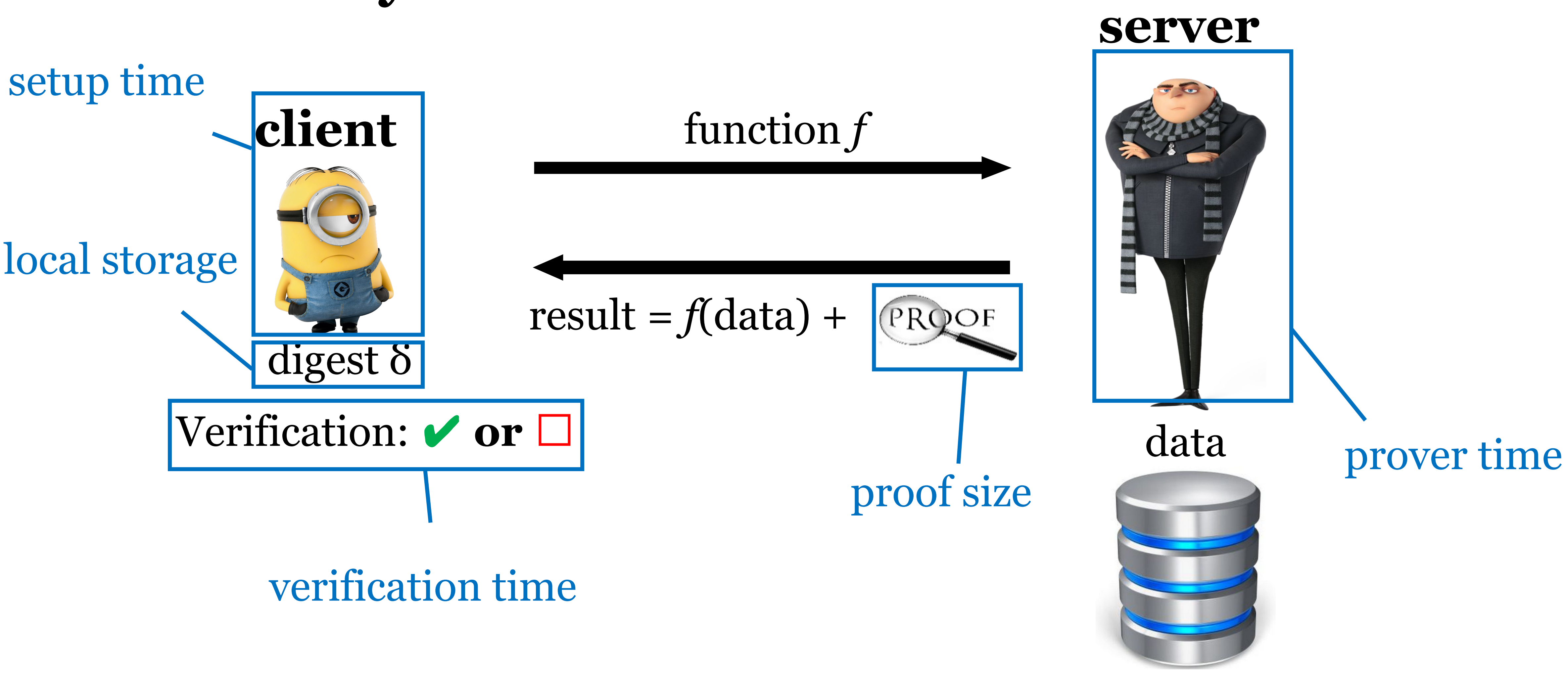
data

Verification: ✔ **or** ☐

Correctness/completeness: Pr[result = f(data) and proof is honest and verification is ✓] = 1

Soundness/security: Pr [result ≠ f (data) and verification is ✓] ≤ $\frac{1}{2^{100}}$

Efficiency measures



Group and field

Group: under 1 operation •

1. Closure: For all a, b in G, the result of the operation, a • b, is also in G
2. Associativity: For all a, b and c in G, (a • b) • c = a • (b • c)
3. Identity element: There exists an element e in G such that, for every element a in G, the equation e • a = a • e = a holds. Such an element is unique
4. Inverse element: For each a in G, there exists an element b in G, commonly denoted a−1 (or −a, if the operation is denoted "+"), such that a • b = b • a = e, where e is the identity element

Examples: integer with +

integer mod n with +

integer mod n with × that are relatively prime to n

Group and field

Group: under 1 operation •

1. Closure: For all a, b in G, the result of the operation, a • b, is also in G
2. Associativity: For all a, b and c in G, (a • b) • c = a • (b • c)
3. Identity element: There exists an element e in G such that, for every element a in G, the equation e • a = a • e = a holds. Such an element is unique
4. Inverse element: For each a in G, there exists an element b in G, commonly denoted a–1 (or –a, if the operation is denoted "+"), such that a • b = b • a = e, where e is the identity element

Field: under 2 operations + and ×

1. F is an abelian group under + (abelian or commutative: a • b = b • a)
2. F – {0} (the set F without the additive identity 0) is an abelian group under ×.

Examples: integer with + is not a field

rational numbers, real numbers, complex numbers

integer mode a prime p (finite field)

Generator

An element that generates all elements in the group by repeating the operation on itself (Cyclic group*)

Example: integer mod 4 with +

2+0=2; 2+2=0; 2+2+2=2; 2+2+2+2=0
3+0=3; 3+3=2; 3+3+3=1; 3+3+3+3=0

3 is a generator, 2 is not

Integer mod 7 with ×

$Z_7^* = \{1,2,3,4,5,6\}$

$2^0 = 1;\ 2^1 = 2;\ 2^2 = 4;\ 2^3 = 1$

$3^0=1;\ 3^1 = 3;\ 3^2 = 2;\ 3^3 = 6;\ 3^4 =4;\ 3^5 = 5;\ 3^6 = 1$

Discrete-log

- Z_p^* has an alternative representation as the powers of g: $\{g, g^2, g^3, \dots, g^{p-1}\}$

Discrete-log: given $a \in Z_p^*$, find k s.t. $g^k = a$

Euler Phi function $\phi(n)$

Positive integers up to n that are relatively prime to n

$\phi(p) = p - 1$ for prime p

$\phi(N) = (p - 1)(q - 1)$ for composite number N = pq where p and q are prime numbers

$$a^x \, mod \, n = a^{x \, mod \, \phi(n)} \, mod \, n$$

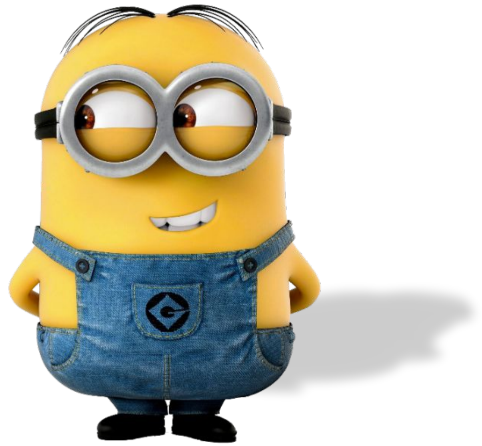
Diffie-Hellman

x | Z_p^* and generator g | y

Alice | | **Bob**

$$\xrightarrow{g^x}$$

$$\xleftarrow{g^y}$$



g^{xy} | | g^{xy}

Diffie-Hellman assumption: give Z_p^*, g, g^x, g^y, cannot compute g^{xy}

Diffie-Hellman problem is easier than discrete-log
Diffie-Hellman assumption is stronger than discrete-log

RSA

Publick key: N, e

Enc(m, pk): $c = m^e \ mod\ N$

Dec (c, sk): $m = c^d \ mod\ N$

Alice

1. Pick random large primes p,q, publish N = pq
2. Compute $\phi(N) = (p-1)(q-1)$
3. Pick random $e \in Z^*_{\phi(N)}$, publish e
4. Compute d as the inverse of e in $Z^*_{\phi(N)}$

$$e * d = 1\ mod\ \phi(N)$$

d is the private key

RSA assumption: given N,e, $c = m^e$, cannot find m

RSA problem is easier than factoring
RSA assumption is stronger than factoring assumption

Ronald Rivest Adi Shamir Leonard Adleman

Turing award 2002

Whitfield Diffie Martin Hellman

Turing award 2015

Accumulator

server

client

digest δ

Is element a in the set?

result +

Verification: ✔ **or** ☐

set

RSA accumulators

- Public: N, generator g
- Private: p, q

- Elements must be primes
- Accumulate set $\{x_1, x_2, \dots x_n\}$: digest $= \ g^{x_1 \cdot x_2 \cdot \dots \cdot x_n} \ mod\ N$
- Membership proof for x_i: $\pi_i = g^{x_1 \cdot \dots \cdot x_{i-1} \cdot x_{i+1} \cdot \ \dots \cdot x_n}$
- Verification: digest $= \ \pi^{x_i}$

Soundness/security

- Strong RSA assumption: given N and c in the group, cannot find m,e such that $c = m^e$ for 1<e<n

RSA assumption: given N,e, $c = m^e$, cannot find m

Strong RSA assumption ☐ RSA assumption ☐ factoring assumption

Complexity

- Local storage, size of accumulator: O(1)
- Setup: O(n)
- Prover time: O(1) with O(n) storage
- Proof size: O(1)
- Verification time: O(1)

Updates

- Update of the digest
  - Add new element x: new digest $\delta' = \delta^x$
  - Delete element x: using private key p,q, compute new digest $\delta' = \delta^{(x^{-1} \, mod \, \phi(n))}$

- Update of the proof: for x_i, $\pi_i = g^{x_1 \cdot \ldots \cdot x_{i-1} \cdot x_{i+1} \cdot \ldots \cdot x_n}$
  - Add: with new element x, set $\pi_i' = \pi_i^x$
  - Delete: with deleted element x and the new digest δ'
    - Extended Euclidean algorithm: find $ax_i + bx = 1$
    - Set $\pi_i' = \pi_i^b \delta'^a$

Complexity

- Local storage, size of accumulator: O(1)
- Setup: O(n)
- Prover time: O(1) with O(n) storage
- Proof size: O(1)
- Verification time: O(1)

- Update: add O(1), delete O(1) with secret key; O(n) without secret key
- Update proof: add O(1), delete O(1) with new digest

Non-membership proofs

- x is not in the set, then x and $u = x_1 \cdot \ldots \cdot x_n$ are co-prime
- Extended Euclidean algorithm: find $ax + bu = 1$
- Proof $a, d = g^b$
- Verification: $\delta^a d^x = g$

Complexity

- Local storage, size of accumulator: O(1)
- Setup: O(n)
- Prover time: O(1) with O(n) storage
- Proof size: O(1)
- Verification time: O(1)

- Update: add O(1), delete O(1) with secret key; O(n) without secret key
- Update proof: add O(1), delete O(1) with new digest

Compare to Merkle tree

- Local storage: O(1)
- Setup: O(n)
- Prover time: O(1) vs O(log n)
- Proof size: O(1) vs O(log n)
- Verification time: O(1) vs O(log n)

- Update: add O(1), delete O(1) with secret key; O(n) without secret key vs O(log n)

- Update proof: add O(1), delete O(1) with new digest vs O(log n) with the proof and the new digest