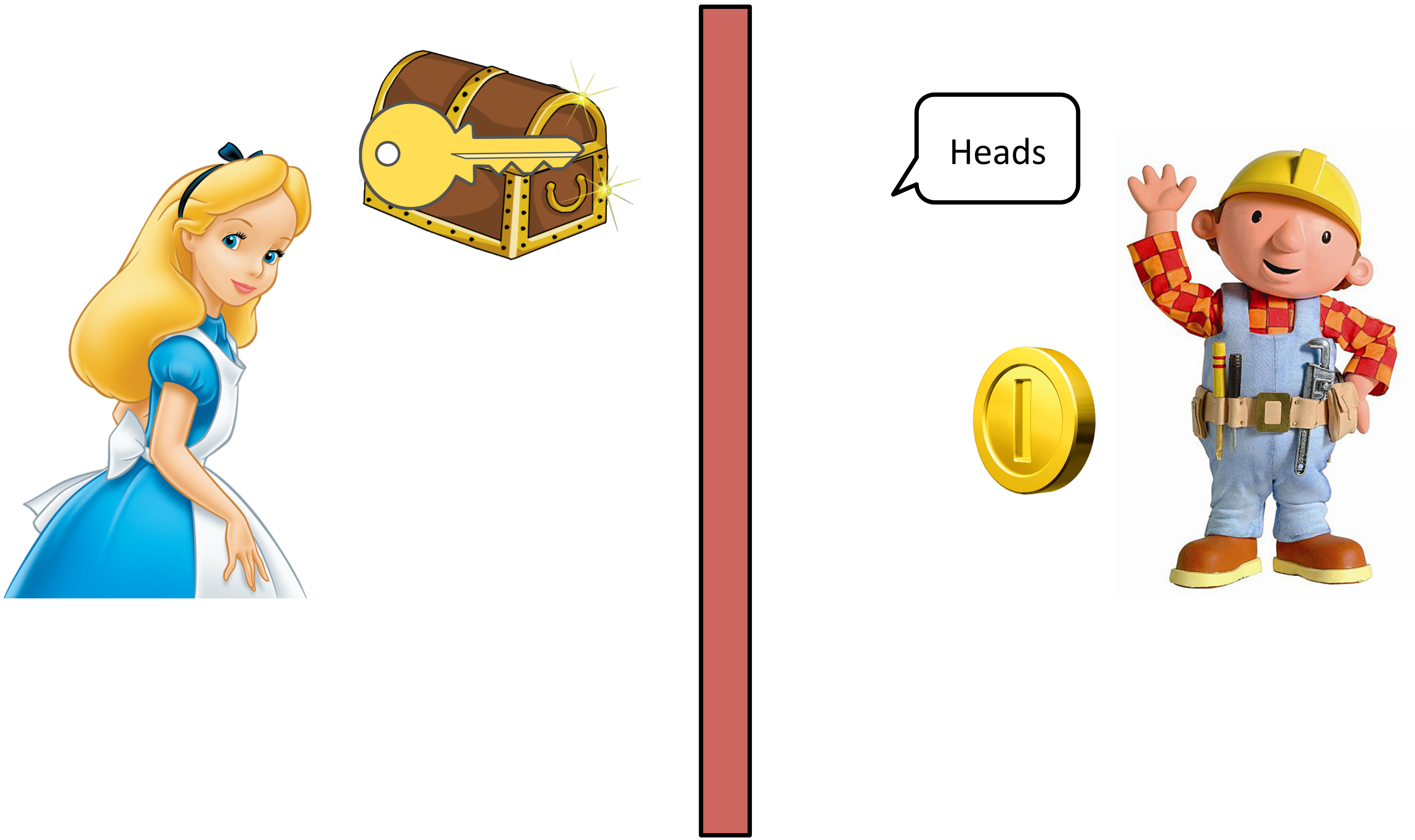
Commitment Schemes

Amit Datta

Coin Toss Example

Coin Toss in different places



Commitment

- Temporarily hide a value, but ensure that it cannot be changed later

- 1st stage: commit
  - Sender electronically "locks" a message in a box and sends the box to the Receiver
- 2nd stage: reveal
  - Sender proves to the Receiver that a certain message is contained in the box

Properties of Commitment Schemes

- Commitment must be hiding
  - At the end of the 1st stage, no adversarial receiver learns information about the committed value
  - If receiver is probabilistic polynomial-time, then <u>computationally</u> hiding; if receiver has unlimited computational power, then <u>perfectly</u> hiding
- Commitment must be binding
  - At the end of the 2nd stage, there is only one value that an adversarial sender can successfully "reveal"
  - Perfectly binding vs. computationally binding

  Can a scheme be perfectly hiding and binding?

Discrete Logarithm Problem

- Intuitively: given g^x mod p where p is a large prime, it is "difficult" to find x
  - Difficult = there is no known polynomial-time algorithm
- g is a generator of a multiplicative group Z_p*
  - g^0, g^1 … g^{p-2} mod p is a sequence of distinct numbers, in which every integer between 1 and p-1 occurs once
    - For any number y $\in$ [1 .. p-1], $\exists$ x s.t. g^x = y mod p
  - Fermat's Little Theorem
    - For any integer a and any prime p, a^{p-1}=1 mod p.
  - If g^q=1 for some q>0, then g is a generator of Z_q, an order-q subgroup of Z_p*

Pedersen Commitment Scheme

- Setup: receiver chooses…
  - Large primes p and q such that q divides p-1
  - Generator g of the order-q subgroup of Z_p*
  - Random secret a from Z_q
  - $h=g^a$ mod p
    - Values p,q,g,h are public, a is secret
- Commit: to commit to some $x \in Z_q$, sender chooses random $r \in Z_q$ and sends $c=g^x h^r$ mod p to receiver
  - This is simply $g^x(g^a)^r=g^{x+ar}$ mod p
- Reveal: to open the commitment, sender reveals x and r, receiver verifies that $c=g^x h^r$ mod p

Security of Pedersen Commitments

- Perfectly hiding
  - Given commitment c, every value x is equally likely to be the value commited in c
  - Given x, r and any x', <u>there exists</u> r' such that $g^x h^r = g^{x'} h^{r'}$

    $r' = (x-x')a^{-1} + r \mod q$ (but must know a to <u>compute</u> r')
- Computationally binding
  - If sender can find different x and x' both of which open commitment $c=g^x h^r$, then he can solve discrete log
    - Suppose sender knows x,r,x',r' s.t. $g^x h^r = g^{x'} h^{r'} \mod p$
    - Because $h=g^a \mod p$, this means $x+ar = x'+ar' \mod q$
    - Sender can compute a as $(x'-x)(r-r')^{-1}$
    - But this means sender computed discrete logarithm of h!

CFN90 scheme

$$B_i = r_i^e f(x_i, y_i) \pmod{n}$$
$$1 <= i <= k \text{ where}$$
$$x_i = g(a_i, c_i)$$
$$y_i = g(a_i \oplus (u || (v+i)), d_i)$$

Account#: u
Counter: v

Random
a_i, c_i, d_i, r_i
$1 <= i <= k$

- B_i is a blinded message: does not reveal information about f(x,y) to bank
- f(x,y) is a commitment to (x, y)
- x, y are constructed to reveal u in case Alice tries to spend the same coin twice

CFN90 scheme

R = random subset of k/2 indices

Reveal a_i, c_i, d_i, r_i for i in R

Check blinded candidates in R

- Ensure Alice following protocol
- Assume R = {k/2+1,….,k} to simplify notation