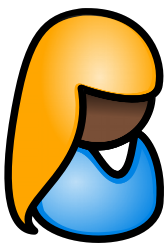
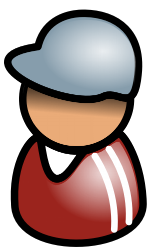
Cryptographic Commitment Schemes

Aniket Kate, CS 555 Cryptography, Fall 2016

Coin Tossing via Mail: Problem

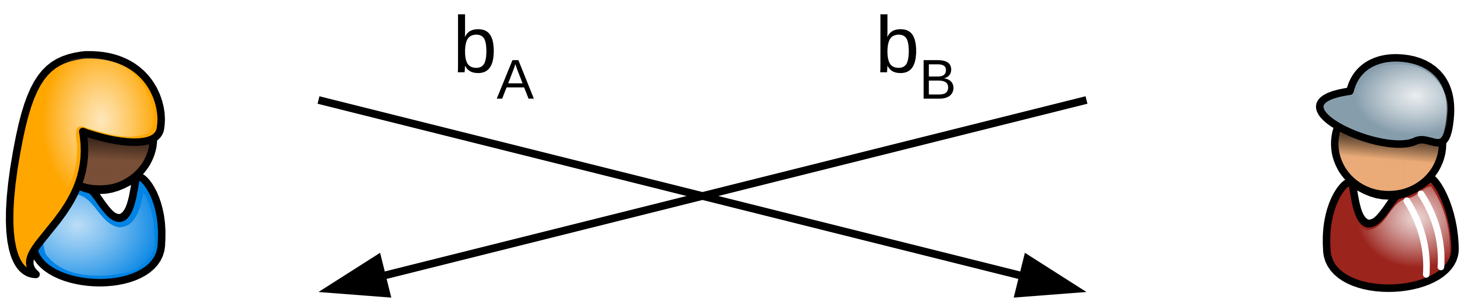
- Alice and Bob would like to flip a coin (or draw a random bit).
- They are in different locations and can only send (physical) mails. The postal service is trusted.
- They have different interests and don't trust each other.





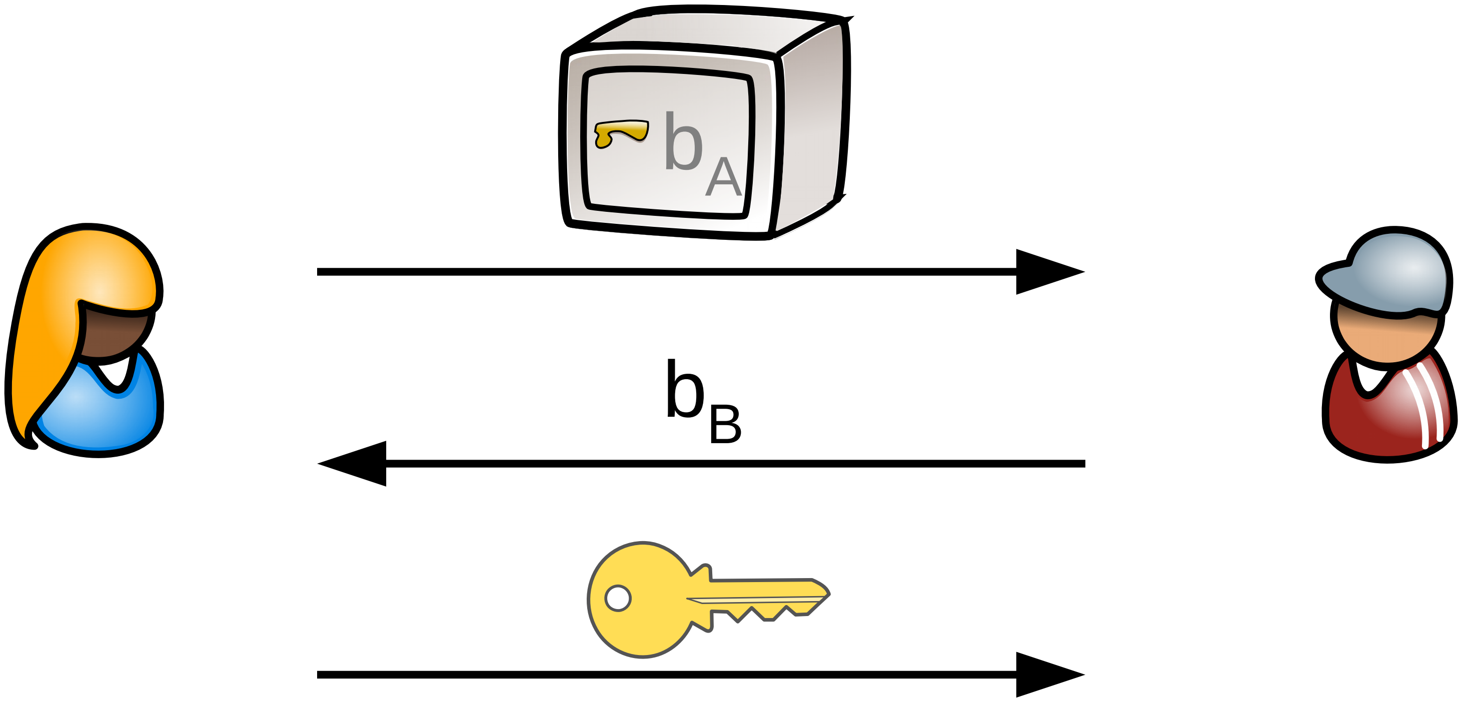
Coin Tossing over Mail: First Attempt

- Each of them draws a random a bit and sends it to the other party. Then they both take the XOR of the two bits.
- Is this secure?



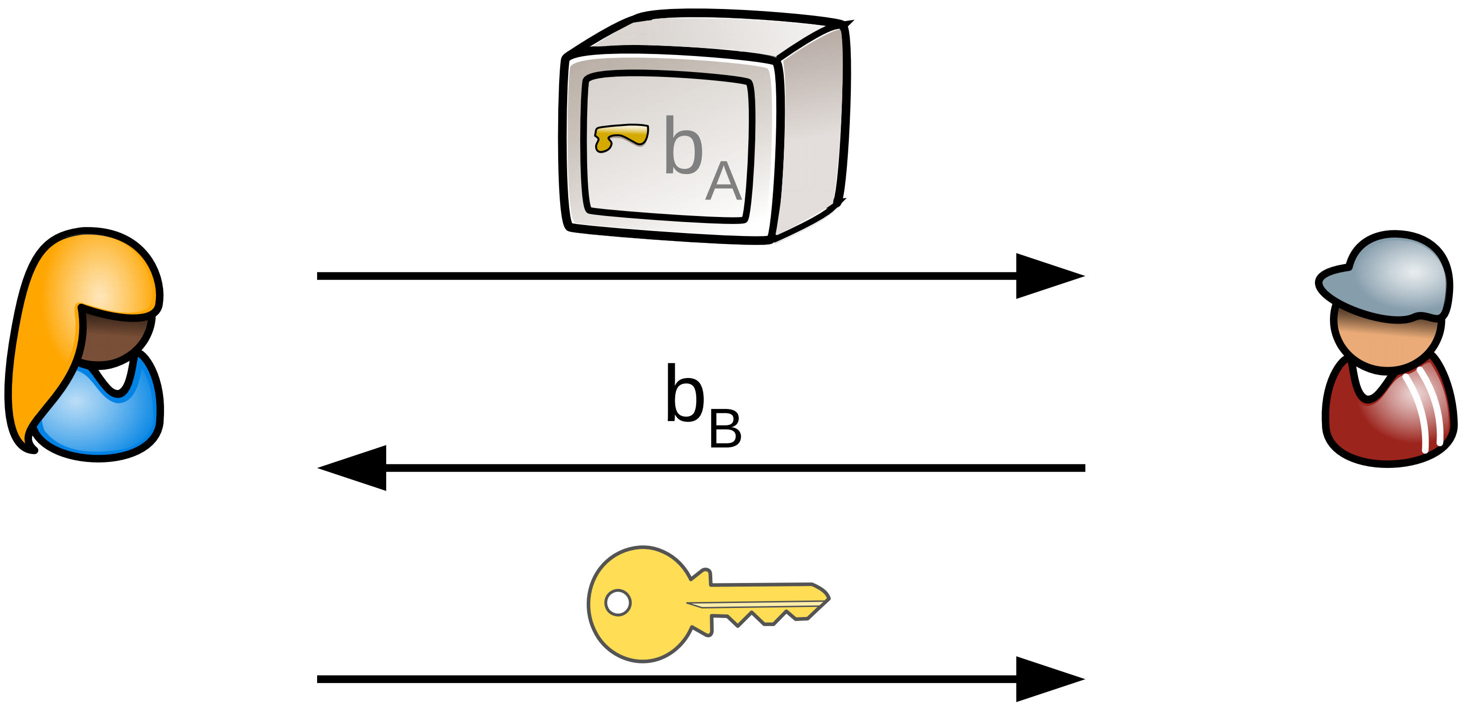
Coin Tossing over Mail: Second Attempt

- Alice puts his bit in a safe and sends it Bob
- Bobs sends his bit
- Alice opens the safe by revealing the key
- Again, take XOR of the two bits



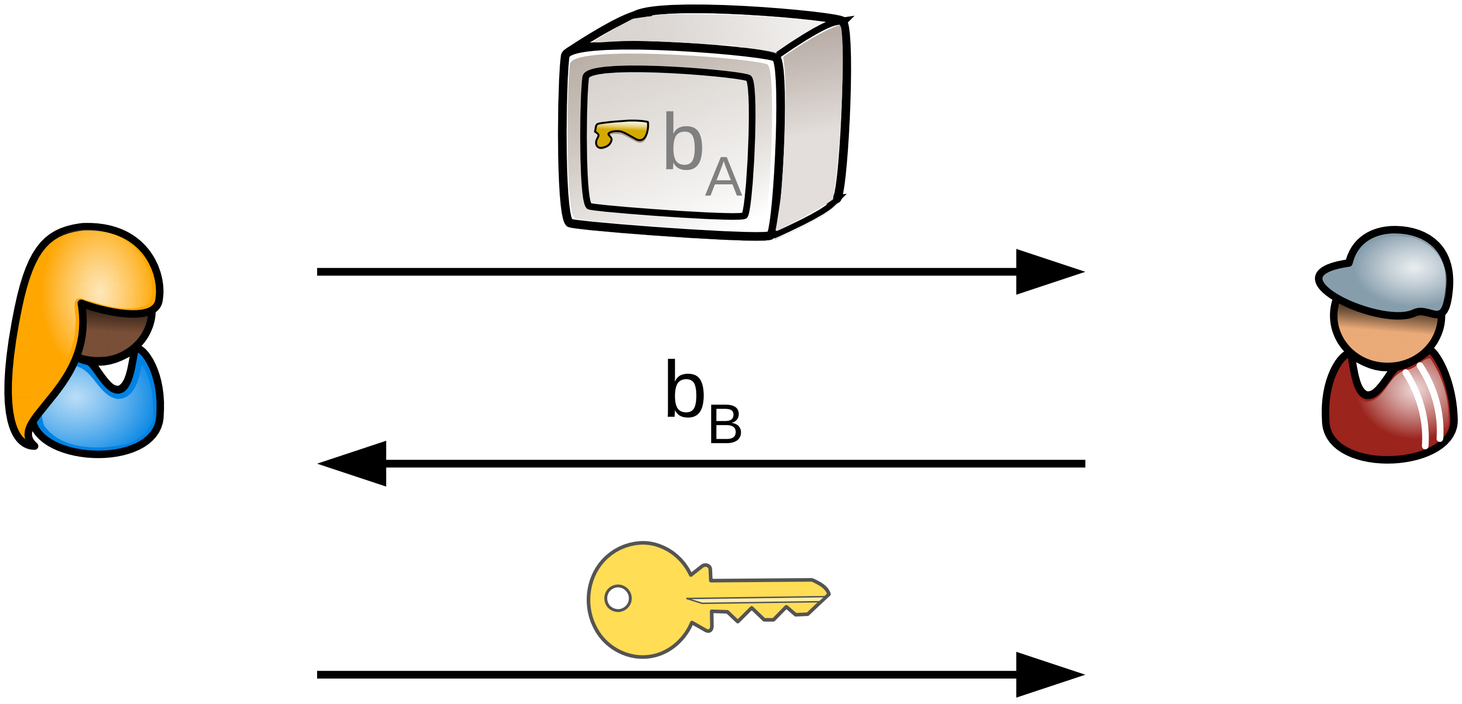
Commitment Schemes

- Cryptographic safes
- Two phases: *Commit* (put in safe) and *Open*



Security

- *Hiding*: Bob should not see the message before the *Open* phase.
- *Binding:* Alice should not be able to open to a different bit than $b_{A.}$



(Somewhat) Formal Definition

- *CS* = (**Setup**(*k*), **Commit**(*pp, m*), **Verify**(*pp, m*))
- **Setup**(*k*) outputs public parameters *pp*
- **Commit**(*pp, m*) outputs (*c*, *d*), where *c* is a "commitment"
  and *d* is a "decommitment" (or "opening information")
- **Verify**(*pp, m, c, d*) checks if *d* is a correct decommitment for *c* and outputs the message *m* that has been committed to

<u>Correctness</u>:
For all public parameters *pp* generated by **Setup**(*k*) and all messages *m,* **Verify**(**Commit**(*pp*, *m*)) = *m.*

Formal Definition (Hiding)

Information-theoretically hiding:

For all messages *m, m'* of the length *k*, and all public parameters *pp*,

{**Commit**(*pp*, *m* ; *r*)} = {**Commit**(*pp*, *m'* ; *r*)}, where *r* is a uniform random bitstring of length *k*.

Formal Definition (Binding)

Computationally binding:

For all ppt adversaries **A**,

Pr[$\mathbf{A}(pp) = (c, m, m', d, d')$: $\mathbf{Verify}(c,d) = m \wedge \mathbf{Verify}(c,d') = m' \wedge m \neq m'$] is negligible.

Remarks:

- In this model, *pp* can be generated by a trusted party or by the recipient. (Why?)
- The variety of definitions of large.
- Computationally hiding, information-theoretically biding schemes do exist as well.
- Interactive schemes exist as well.

Pedersen Commitments

- Setting: A large prime-order (sub)group G of order |G| =q, where Dlog is assumed to be hard.
- **Setup**(k):

  Select two random generators g and h.
  Return $pp := (g, h)$
- **Commit**($(g, h), m$):

  Select a random exponent r
  $c := g^m h^r$
  Return $(c, (r, m))$
- **Verify**($c, (r, m)$):
  check if $c = g^m h^r$, abort otherwise
  Return m

Security of Pedersen Commitments

- Information-theoritcally hiding:
  $g^m h^r$ is a random element if r is random
- Computationally binding:
  If $\mathbf{A}(pp) = (c, m, m', d, d')$ s. t.
  $\mathbf{Verify}(c,d')=m \wedge \mathbf{Verify}(c,d')=m' \wedge \mathrm{m} \neq \mathrm{m}'$,
  we can compute the discrete logarithm of h with respect to g.

(on the board)