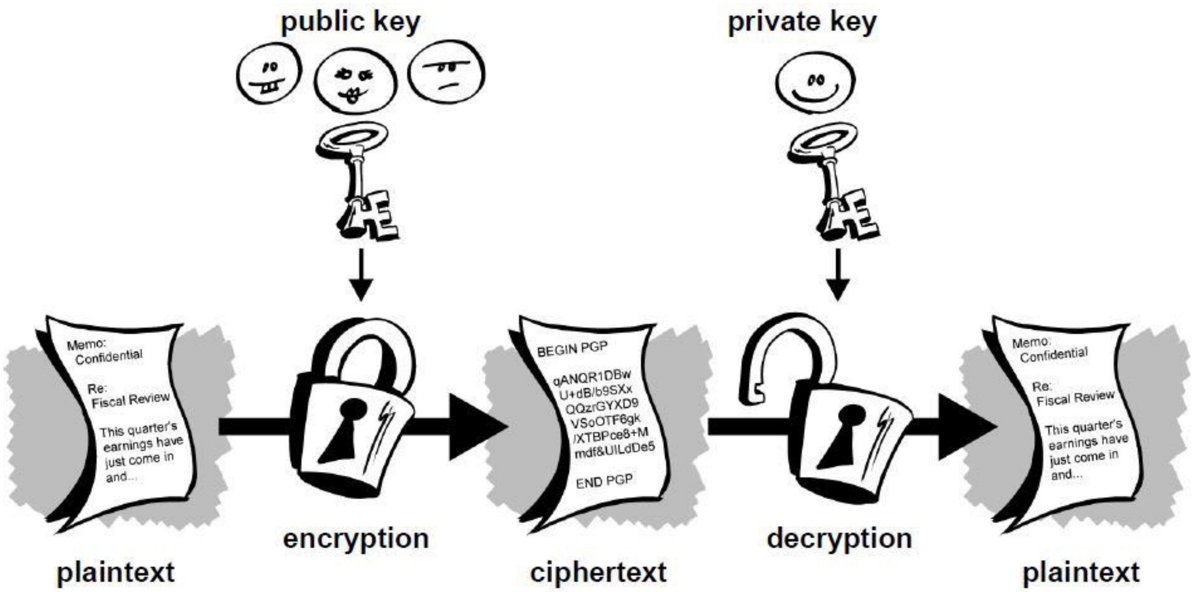
Coin Flipping Over the Internet

By

Vipul Goyal



Bad News

Flip a coin?

Alice and Bob are getting divorced

They can't even be in the same room together

Who keeps the car?

Coin Flipping over Telephone

Can we flip a fair coin over the phone (or internet)?

Applications: cryptographic protocols, online gaming, ….

Blum (1981)

Commitment Schemes

- Commitment like a note placed in a safe
- Two properties: hiding and binding
- Electronic equivalent of such a safe

Building Commitment Schemes

m

Committer

Enc(k, *m*)

k

Receiver

- Does this work?

Using one-time pads?

m

Committer

$C = k \oplus m$

k'

Receiver

- Can choose any m’ s.t. $C = k' \oplus m'$

ElGamal Commitment Scheme

- DDH assumption: given (g, g^a, g^b), any information about g^{ab} is hard to compute (looks random)

Generate a,b randomly

m

Committer

$$g,\ g^a,\ g^b,\ m.g^{ab} \longrightarrow$$

$$a,\ b \longrightarrow$$

Receiver

- After commitment phase: m hidden
- Binding: *a*, *b* unique given commitment phase, hence *m* unique

Coin Flipping Attempt 1

b_0 = 0/1 $\longrightarrow$

$\longleftarrow$ b_1 = 0/1

$$b = b_0 \oplus b_1$$

- If both parties honest: b is random
- If Bob dishonest: can dictate output
  - Suppose Bob wants output = 1
  - If Alice chooses 0, Bob chooses 1
  - If Alice chooses 1, Bob chooses 0

Coin Flipping Attempt 1

$b_0 = 0/1$

$b_1 = 0/1$

$$b = b_0 \oplus b_1$$

- What if Alice and Bob send messages simultaneously?
  - Protocol is secure even if Bob malicious

Fact: even if one bit random (and other bit doesn't depend upon it), XOR is random

Coin Flipping Protocol

$$\xrightarrow{\quad Com(b_0) \quad}$$

$$\xleftarrow{\quad b_1 \quad}$$

$$\xrightarrow{\quad Open\ b_0 \quad}$$

$$b = b_0 \oplus b_1$$

- If both honest: both bits random and independent. XOR random
- Alice dishonest, Bob honest: After round 1, b_0 fixed (binding). Hence b_0 independent of b_1. Now b_1 is random (and independent of b_0 if Bob honest). Hence XOR random.
- Alice honest, Bob dishonest: b_0 random, b_1 still independent of b_0 (hiding). Hence XOR random.

Multi-Party Coin Flipping

We have *n* parties, want to flip a single coin

- Example: choosing a leader in a group of n parties (leader election), choosing who goes first in a multi-player game, etc.
- Even if *n*-1 parties cheat and collude with each other, they can't bias outcome. Honest party is protected.

Issues: similar to two party.

- If an adv can choose its bit based on bits of other parties, it can control the output

Multi-Party Coin Flipping

Candidate Protocol

- Parties choose b_1, b_2, …., b_n resp
- Day 1: Parties send com(b_1), com(b_2), …, com(b_n) (no particular order, free to go)
- Day 2: open b_1, b_2, …, b_n. $b = b_1 \oplus b_2 \oplus \ldots \oplus b_n$

Idea: Say P_1 honest, others dishonest.

b_2,…, b_n fixed in stage 1. b_1 hidden.

So b_2,…, b_n can't depend upon b_1

Question

Is the candidate multi-party coin-flipping protocol secure?

Ans: Surprisingly, NO!

Answers

- Parties choose $b_1, b_2, \ldots., b_n$ resp
- Day 1: Parties send $com(b_1), com(b_2), \ldots, com(b_n)$ (no particular order, free to go)
- Day 2: open $b_1, b_2, \ldots, b_n$. $b = b_1 \oplus b_2 \oplus \ldots \oplus b_n$

Say P_1 honest, others dishonest.

Day 1: P_2 waits for P_1. Sets $com(b_2) = com(b_1)$

Day 2: P_1 sends opening of $com(b_1)$. P_2 replays the same message.

$$b = b_1 \oplus b_2 \oplus \ldots \oplus b_n = b_3 \oplus \ldots \oplus b_n$$

No easy fixes!

Committed values may be correlated

$g^a, g^b, m.g^{ab}$ $\longrightarrow$

a, b $\longrightarrow$

$g^a, g^b, 2.m.g^{ab}$ $\longrightarrow$

a, b $\longrightarrow$

A Secure Protocol

Parties choose b_1, b_2, …., b_n resp

- Day 1: P_1 send com(b_1),
- …
- Day n: P_n send com(b_n)
- Day n+1: P_n opens com(b_n)
- …
- Day n+n: P_1 opens com(b_1)

Idea: suppose P_2 "copied" from P_1

P_2 has to open before P_1

Hard for P_2: hiding

Questions?