Cryptography
CS 555

Topic 35: Multi-Party Computation

Recap

- Digital Signatures
- CCA-Secure Public Key Encryption
- SSL/TLS

Commitment Schemes

A commitment scheme allows one party to “commit” to a message **m** by sending a commitment **com** with the following security properties

- **Hiding:** the commitment doesn’t reveal anything about m
- **Binding:** it is infeasible for the committer to output a commitment **com** that can later be revealed as two different messages m and m’

Physical Analogy: Sealed envelope.

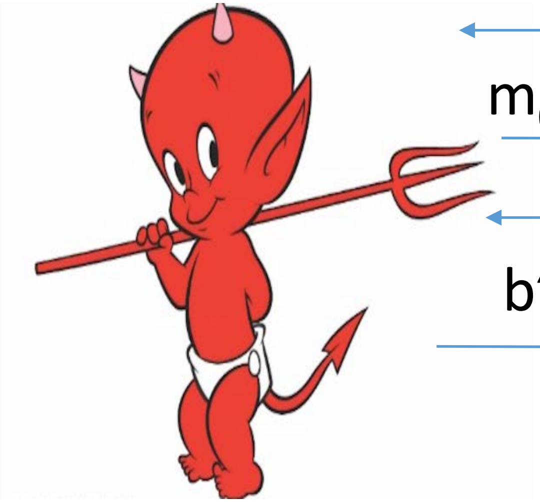
- **Hiding:** Receiver cannot see message inside the envelope
- **Binding:** Sender cannot change message inside the envelope

Commitment Scheme

- Three Algorithms
  - $\mathrm{Gen}(1^n)$ (Key-generation algorithm)
    - Input: Security parameter n
    - Output: public parameters **params** of commitment scheme
  - $\mathrm{Com}(params, m; r)$ (Commitment algorithm)
    - Input: parameters params, message $\mathrm{m} \in \mathcal{M}$ and random bits r
    - Output: commitment ***com***
  - Vrfy(params, com, m , r) (Verification Algorithm: Deterministic)
    - Input: parameters params, message $\mathrm{m} \in \mathcal{M}$ and random bits r
    - Output: 1/0 for "success" or "failure"

- To open a commitment **com** the committer can reveal m and r
- **Canonical Verification**: Check to see if **com** = Com(params, m; r)

Commitment Hiding Experiment ($\mathrm{Hiding}_{\mathrm{A,Com}}(n)$)



params

m_0, m_1

c = Com(params,m_b;r)

b’

$\mathrm{Hiding}_{\mathrm{A,Com}}(n) = 1 \text{ if } b = b'$

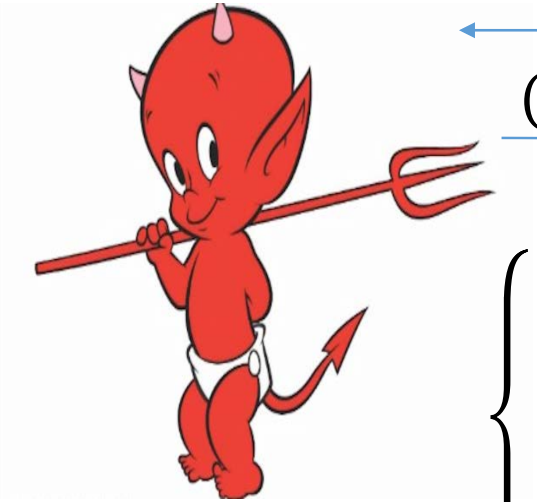
params = Gen(.)

Bit b

$$\forall PPT\ A\ \exists \mu\ (\text{negligible})\ \mathrm{s.t}$$

$$\Pr\left[\mathrm{Hiding}_{\mathrm{A,Com}}(n) = 1\right] \leq \frac{1}{2} + \mu(n)$$

Commitment Hiding Experiment ($\mathrm{Binding}_{\mathrm{A,Com}}(n)$)



params

(COM, m, r, m', r')

$$\mathrm{Binding}_{\mathrm{A,Com}}(n) = \begin{cases} 0 & \textit{if } m = m' \\ 0 & \textit{if } \mathrm{Vrfy}(params, m, r) \neq 1 \\ 0 & \textit{if} \mathrm{Vrfy}(params, m', r') \neq 1 \\ 1 & \textit{otherwise} \end{cases}$$

params = Gen(.)
Bit b

$$\forall PPT\ A\ \exists \mu\ (\text{negligible})\ \text{s.t}$$
$$\Pr\left[\mathrm{Binding}_{\mathrm{A,Com}}(n) = 1\right] \leq \mu(n)$$

Secure commitment scheme

Definition: A commitment scheme Com is secure if for all PPT attackers A there is a negligible function $\mu(n)$ such that

$$\Pr\left[\mathrm{Hiding}_{\mathrm{A,Com}}(n) = 1\right] \leq \frac{1}{2} + \mu(n)$$

And

$$\Pr\left[\mathrm{Binding}_{\mathrm{A,Com}}(n) = 1\right] \leq \mu(n)$$

Application: Fair Coin Flipping

Com(params,b;r) →

← **b’**

$coin = b \oplus b'$ **and (b,r)** →

← Vrfy(params, com, b , r)

Bob;
params;
Bit b

Alice;
params

Security: Dishonest party cannot bias the coin

Secure Commitment Scheme with Random Oracle

$$\mathrm{Com}(params, m; r) = H(m \parallel r)$$

Theorem: In the random oracle model this is a secure commitment scheme.

Proof Hiding [sketch]: Any PPT attacker can make p(n) queries to RO.

- Case 1: Attacker never queries $H(*\parallel r)$
  - Attacker learns no information about m in an information theoretic sense
- Case 2: Attacker queries $H(*\parallel r)$
  - Happens with probability at most $\frac{p(n)}{2^n}$

Secure Commitment Scheme with Random Oracle

$$\mathrm{Com}(params, m; r) = H(m \parallel r)$$

Theorem: In the random oracle model this is a secure commitment scheme.

Proof Binding [sketch]: To win the binding game the attacker must find (m,m’,r,r’) such that

$$H(m \parallel r) = H(m' \parallel r')$$

If attacker makes p(n) queries to random oracle the probability of finding a collision is at most

$$\frac{p(n)^2}{2^n}$$

Application: Fair Coin Flipping

Com(params,b;r) →

← **b’**

$\boldsymbol{coin = b \oplus b'}$ **and (b,r)** →

← Vrfy(params, com, b , r)

Bob;
params;
Bit b

Alice;
params

Theorem: If the commitment scheme is secure and Bob is honest then Alice cannot bias the coin. If $|\Pr[Alice\ Responds]| \geq \frac{1}{p(n)}$ then $\left|\Pr[coin = 1|Respond] - \frac{1}{2}\right| \leq negl(n)$

Application: Fair Coin Flipping

Theorem: If the commitment scheme is hiding then a PPT Alice cannot bias the coin. If $|\Pr[Alice\ Responds]| \geq \frac{1}{p(n)}$ then $\left|\Pr[coin = 1|Respond] - \frac{1}{2}\right| \leq negl(n)$

Proof: Use Alice to break the commitment scheme. WLOG suppose that $\Pr[coin = 1] > \frac{1}{2} + \frac{1}{p(n)}$

1. Send m_0=0,m_1=1 to judge in hiding experiment $\mathrm{Hiding}_{\mathrm{A,Com}}(n)$
2. Receive **c = Com(params,m_b;r)** from judge.
3. Send c to Alice
4. Alice sends us b' $coin = b \oplus \boldsymbol{b}'$
5. **Ouput:** $b'' = \boldsymbol{b}' \oplus \mathbf{1}$

Application: Fair Coin Flipping

Theorem: If the commitment scheme is hiding and Bob is honest then a PPT Alice cannot bias the coin. $\left|\Pr[coin = 1] - \frac{1}{2}\right| \leq negl(n)$

Proof: Use Alice to break the commitment scheme. WLOG suppose that $\Pr[coin = 1] > \frac{1}{2} + \frac{1}{p(n)}$

- Alice sends us b' $\text{observe that } coin = b \oplus \boldsymbol{b'}$
- **Ouput:** $b'' = \boldsymbol{b'} \oplus \mathbf{1}$

$$\Pr[b'' = \boldsymbol{b}] = \Pr[\boldsymbol{b'} \oplus \mathbf{1} = coin \oplus \boldsymbol{b'}]$$
$$= \Pr[\mathbf{1} = coin] > \frac{1}{2} + \frac{1}{p(n)}$$

Application: Fair Coin Flipping

Com(params,b;r)

b’

$\boldsymbol{coin = b \oplus b'}$ **and (b,r)**

Vrfy(params, com, b , r)

Bob;
params;
Bit b

Alice;
params

Theorem: If the commitment scheme is secure, Alice is honest **and Bob never aborts** then Bob cannot bias the coin. $\left|\Pr[coin = 1] - \frac{1}{2}\right| \leq negl(n)$.

Fair Coin Flipping

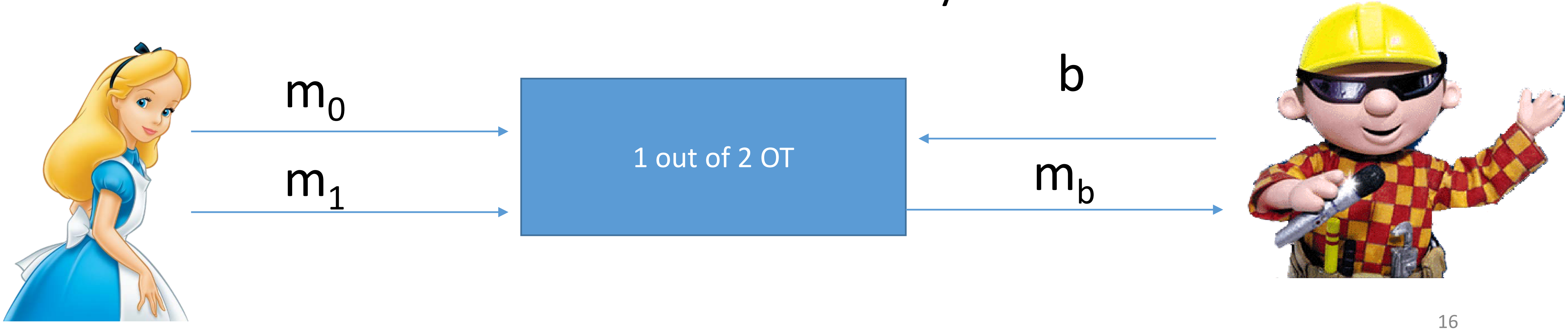
Theorem: If the commitment scheme is secure, Alice is honest **and Bob never aborts** then Bob cannot bias the coin. $\left|\Pr[coin = 1] - \frac{1}{2}\right| \le negl(n)$.

Proof: Use Bob to break **binding** property of commitment scheme. WLOG suppose that $\Pr[coin = 1] > \frac{1}{2} + \frac{1}{p(n)}$.

1. Simulate Bob who sends c=**Com(params,b;r)**
2. Select b' uniformly at random and send b' to Bob
3. Receive b",r" from Bob, if Vrfy(b'',r'')$\neq$ 1 then **abort**
4. **Rewind** Bob to step 2 and send (1-b') to Bob
5. Receive b''',r''' from Bob, if Vrfy(b'',r'')$\neq$ 1 then **abort**
6. **Output (Com,b'',r'',b''',r''') to win Binding game**

Oblivious Transfer (OT)

- 1 out of 2 OT
  - Alice has two messages m_0 and m_1
  - At the end of the protocol
    - Bob gets exactly one of m_0 and m_1
    - Alice does not know which one
- Oblivious Transfer with a Trusted Third Party

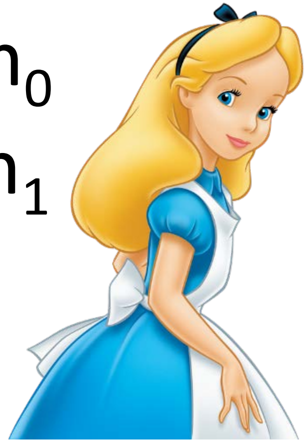


Bellare-Micali 1-out-of-2-OT protocol

- Oblivious Transfer without a Trusted Third Party
  - g is a generator for a prime order group G_q in which CDH problem is hard

m_0

m_1

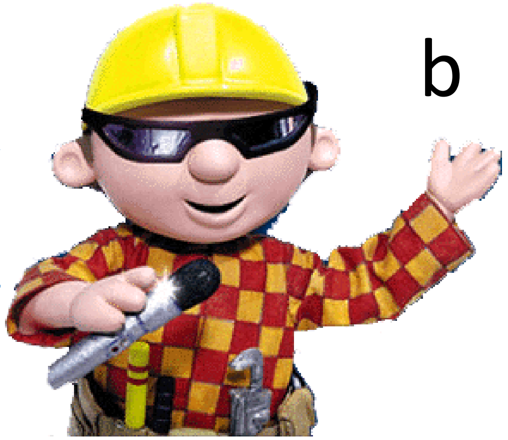


$c \leftarrow_R G_q$

c

z_0, z_1

$$C_0 = \left[g^{r_0}, H\left(z_0^{r_0}\right) \oplus m_0\right]$$
$$C_1 = \left[g^{r_1}, H\left(z_1^{r_1}\right) \oplus m_1\right]$$



b

$k \leftarrow_R Z_q$

$$z_b = g^k, z_{1-b} = cg^{-k}$$

Bob can decrypt C_b

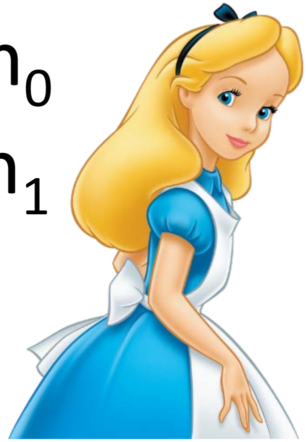
$$z_b^{r_b} = g^{K r_b}$$

Bellare-Micali 1-out-of-2-OT protocol

- Oblivious Transfer without a Trusted Third Party
  - g is a generator for a prime order group G_q in which CDH is Hard

m_0

m_1



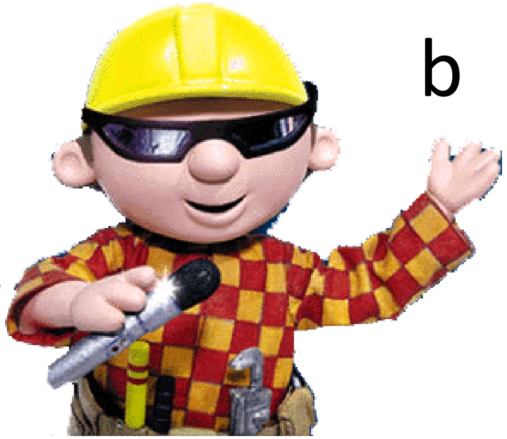
$c \leftarrow_R G_q$

c

z_0, z_1

$$C_0 = \left[g^{r_0}, H\left(z_0^{r_0}\right) \oplus m_0\right]$$
$$C_1 = \left[g^{r_1}, H\left(z_1^{r_1}\right) \oplus m_1\right]$$

b



$k \leftarrow_R Z_q$

$$z_b = g^k, z_{1-b} = cg^{-k}$$
$$= c(z_b)^{-1}$$

Alice must check that

$$z_1 = c(z_0)^{-1}$$

Bob can decrypt C_b

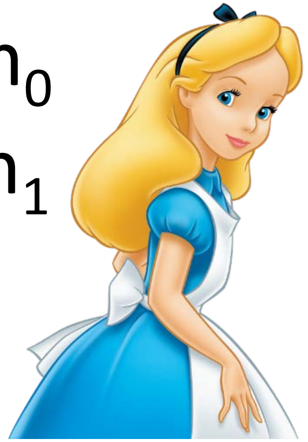
$$z_b^{r_b} = g^{Kr_b}$$

Bellare-Micali 1-out-of-2-OT protocol

- Oblivious Transfer withou
  - g is a generator for a prime

m_0

m_1



$c \leftarrow_R G_q$

$C_0 =$

$C_1 =$

Alice does not learn b because

- $z_1 = c(z_0)^{-1}$ and
- $z_0 = c(z_1)^{-1}$ and
- z_1, z_0 are distributed uniformly at random subject to these condition.

This is an information theoretic guarantee!

$z_b = g^k, z_{1-b} = cg^{-k}$
$= c(z_b)^{-1}$

Alice must check that

$z_1 = c(z_0)^{-1}$

Bob can decrypt C_b

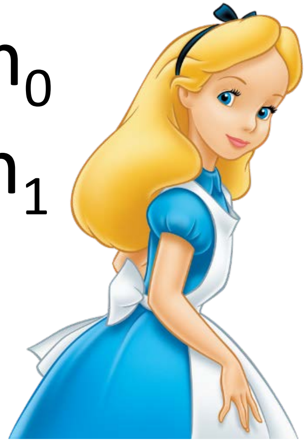
$z_b^{r_b} = g^{Kr_b}$

Bellare-Micali 1-out-of-2-OT protocol

- Oblivious Transfer withou
  - g is a generator for a prime

m_0

m_1



$c \leftarrow_R G_q$

$C_0 =$

$C_1 =$

Bob cannot decrypt C_{1-b}

Unless he queries random oracle at

- $c^{r_{1-b}} g^{-Kr_{1-b}}$
- Given this value we can obtain $c^{r_{1-b}}$
- Thus, we can break CDH assumption

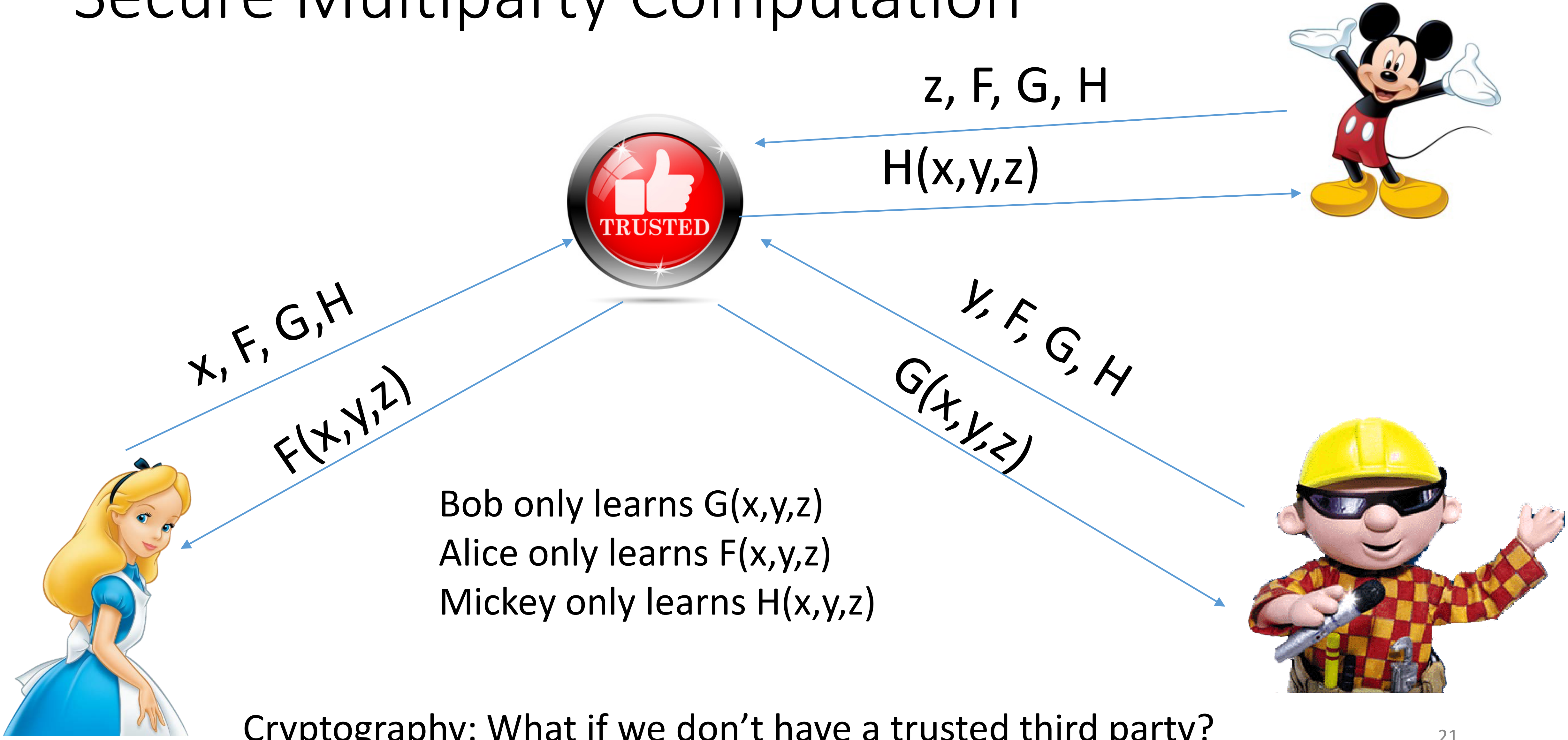
given random $c = g^m$ and $g^{r_{1-b}}$ it is hard to find $c^{r_{1-b}} = g^{mr_{1-b}}$

$z_b = g^k, z_{1-b} = cg^{-k}$
$= c(z_b)^{-1}$

Alice must check that
$z_1 = c(z_0)^{-1}$

Bob can decrypt C_b
$z_b^{r_b} = g^{Kr_b}$

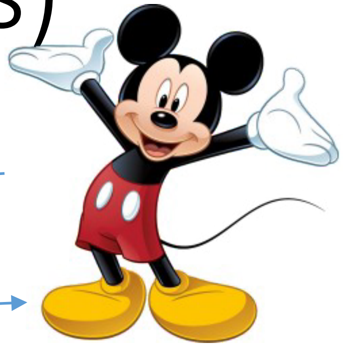
Secure Multiparty Computation



Bob only learns G(x,y,z)
Alice only learns F(x,y,z)
Mickey only learns H(x,y,z)

Cryptography: What if we don't have a trusted third party?

Secure Multiparty Computation (Crushes)



Z="Alice", F, G, H

H(x,y,z)="no match"

X="Bob", F, G,H

F(x,y,z)="match"

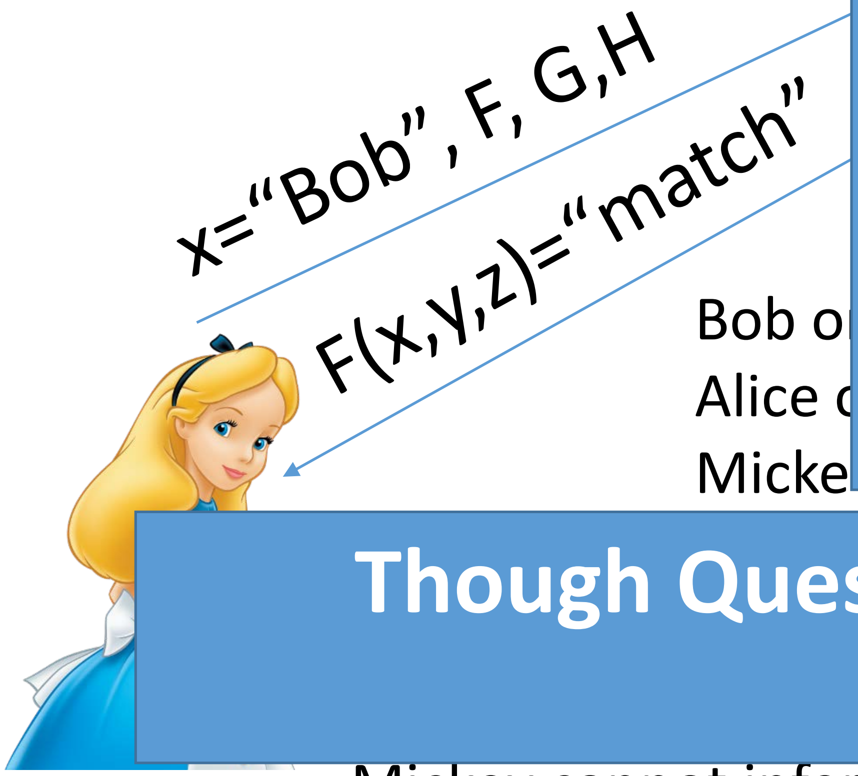
Y="Alice", F, G, H

G(x,y,z)="match"

Bob only learns G(x,y,z)
Alice only learns F(x,y,z)
Mickey only learns H(x,y,z)

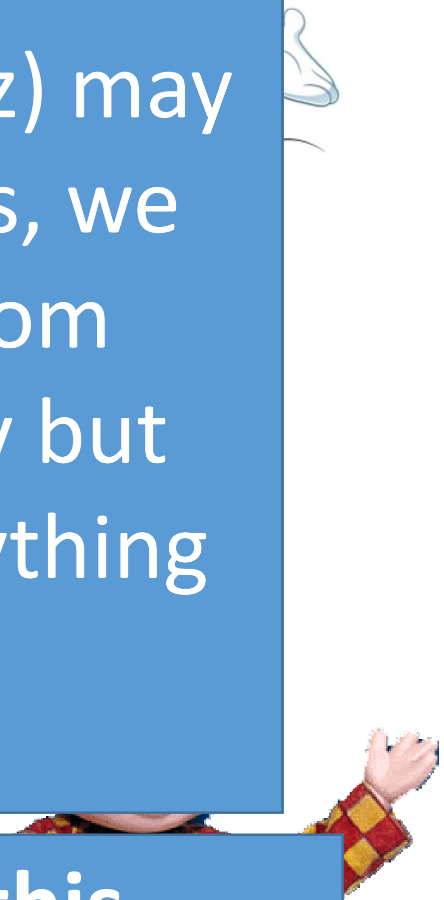
Alice can infer Y from F(x,y,z) and Bob can infer X from H(x,y,z).
But Alice/Bob cannot infer anything about Z.
Mickey cannot infer y, and learns that x$\neq$ "Mickey"

Secure Multiparty Computation (Crushes)



Bob o
Alice c
Micke

Key Point: The output H(x,y,z) may leak info about inputs. Thus, we cannot prevent Mickey from learning anything about x,y but Mickey should not learn anything else besides H(x,y,z)!



Though Question: How can we formalize this property?

Mickey cannot infer y, and learns that x≠ "Mickey"

Adversary Models

- Semi-Honest ("honest, but curious")
  - All parties follow protocol instructions, but…
  - dishonest parties may be curious to violate privacy of others when possible
- Fully Malicious Model
  - Adversarial Parties may deviate from the protocol arbitrarily
    - Quit unexpectedly
    - Send different messages
  - It is much harder to achieve security in the fully malicious model
- Convert Secure Semi-Honest Protocol into Secure Protocol in Fully Malicious Mode?
  - Tool: Zero-Knowledge Proofs

Next Class: Zero-Knowledge Proofs

- Read Wikipedia entry on Zero-Knowledge Proofs
- https://en.wikipedia.org/wiki/Zero-knowledge_proof