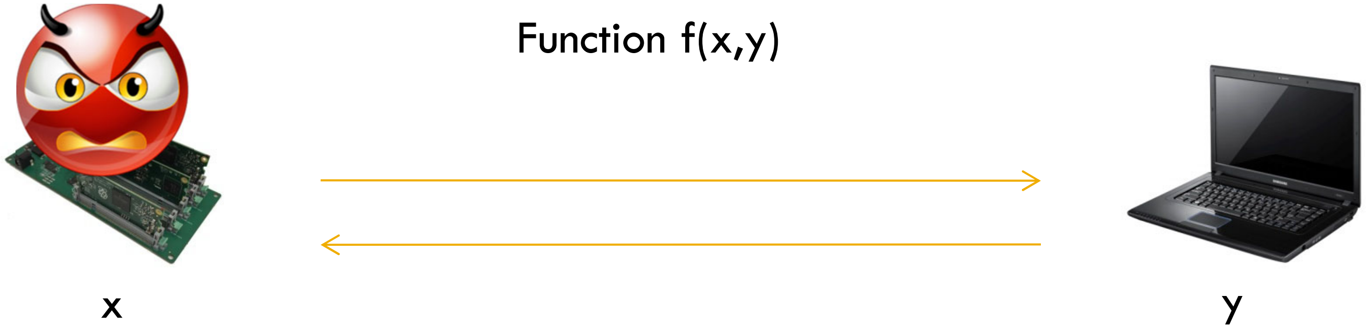
AN INTRODUCTION TO UNIVERSALLY COMPOSABLE SECURITY FRAMEWORK OF CANETTI

SEPIDEH AVIZHEH
SEPIDEH.AVIZHEH1@UCALGARY.CA
UNIVERSITY OF CALGARY, AB, CANADA

UNIVERSITY OF CALGARY

Information Security Talk Series- 21,May,2021

Security of cryptographic tasks



Function f(x,y)

x

y

Security properties: correctness, secrecy, fairness, integrity,…

Function f(x,y):

Commitment
Signature
Secret sharing
Key exchange
Oblivious transfer
…

Secure communication sessions
Secure remote storage
Auction
Private information retrieval
Electronic voting
Multi party computation

Security models

1) Game based security

- Consider a challenger and adversary
- Define the property as a randomized experiment
- Calculate the success probability of adversary

- Disadvantgaes:
    - Each game covers one property of interest
    - Do not guarantee security in the practice(real world)

EAV-security:

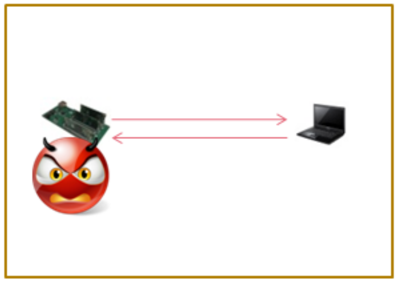
- Define a randomized exp't $\mathrm{PrivK}_{A,\Pi}(n)$:
    1. $A(1^n)$ outputs $m_0, m_1 \in \{0,1\}^*$ of equal length
    2. $k \leftarrow \mathrm{Gen}(1^n)$, $b \leftarrow \{0,1\}$, $c \leftarrow \mathrm{Enc}_k(m_b)$
    3. $b' \leftarrow A(c)$

    Adversary A *succeeds* if b = b', and we say the experiment evaluates to 1 in this case

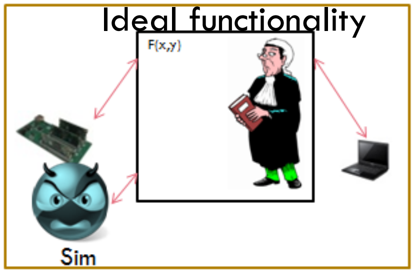
2) Simulation based security (real-ideal world paradigm)

- Standalone security
- Universally composable security

- Advantages of UC:
    - Ensures security in practice
    - Allows modular design in unpredictable environments



Real world



Ideal world

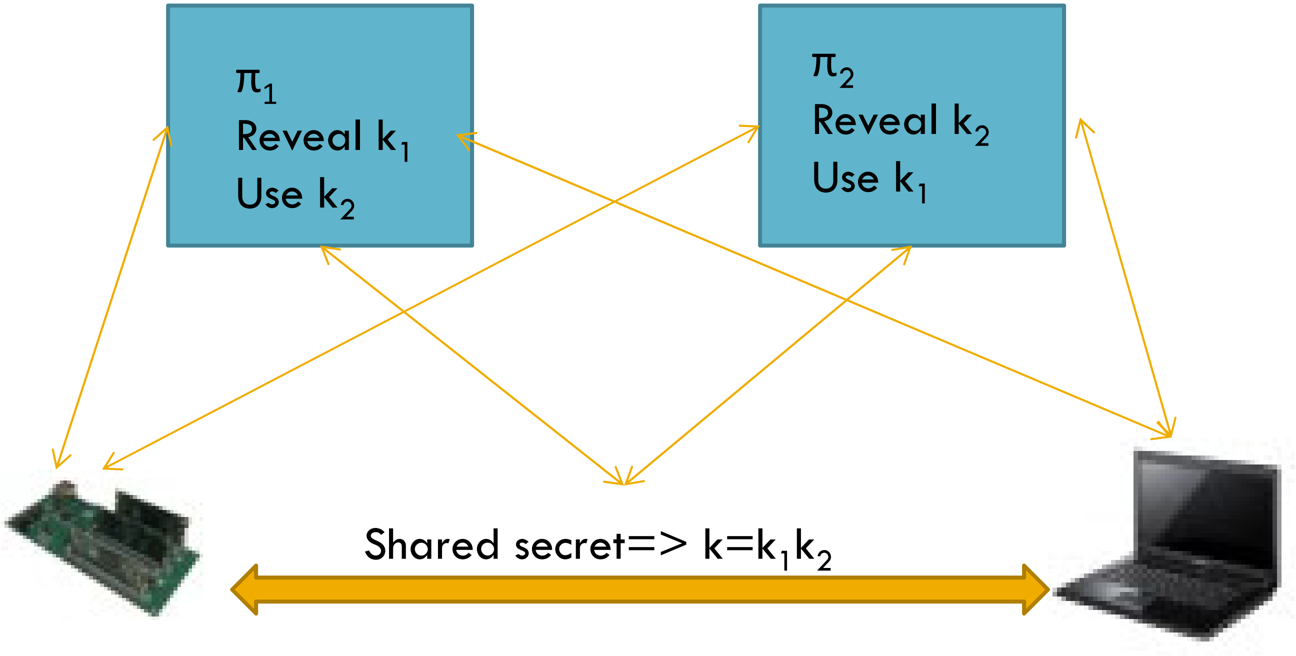
In this talk…

- Standalone security and its insufficiency
- UC components
- UC theorem
- Example
    - Impossibility of UC security for commitment in plain model
    - Random oracle model
    - UC secure commitment in random oracle model

Insufficiency of standalone security

Useful examples from [1]

- Example 1: a simple insecure protocol combination



π_1
Reveal k_1
Use k_2

π_2
Reveal k_2
Use k_1

Shared secret=> $k=k_1k_2$

[1] Ran Canetti, “How To Obtain and Assert Composable Security”, PPT

Insufficiency of Standalone security [1]

- Example 2 (more realistic scenario):

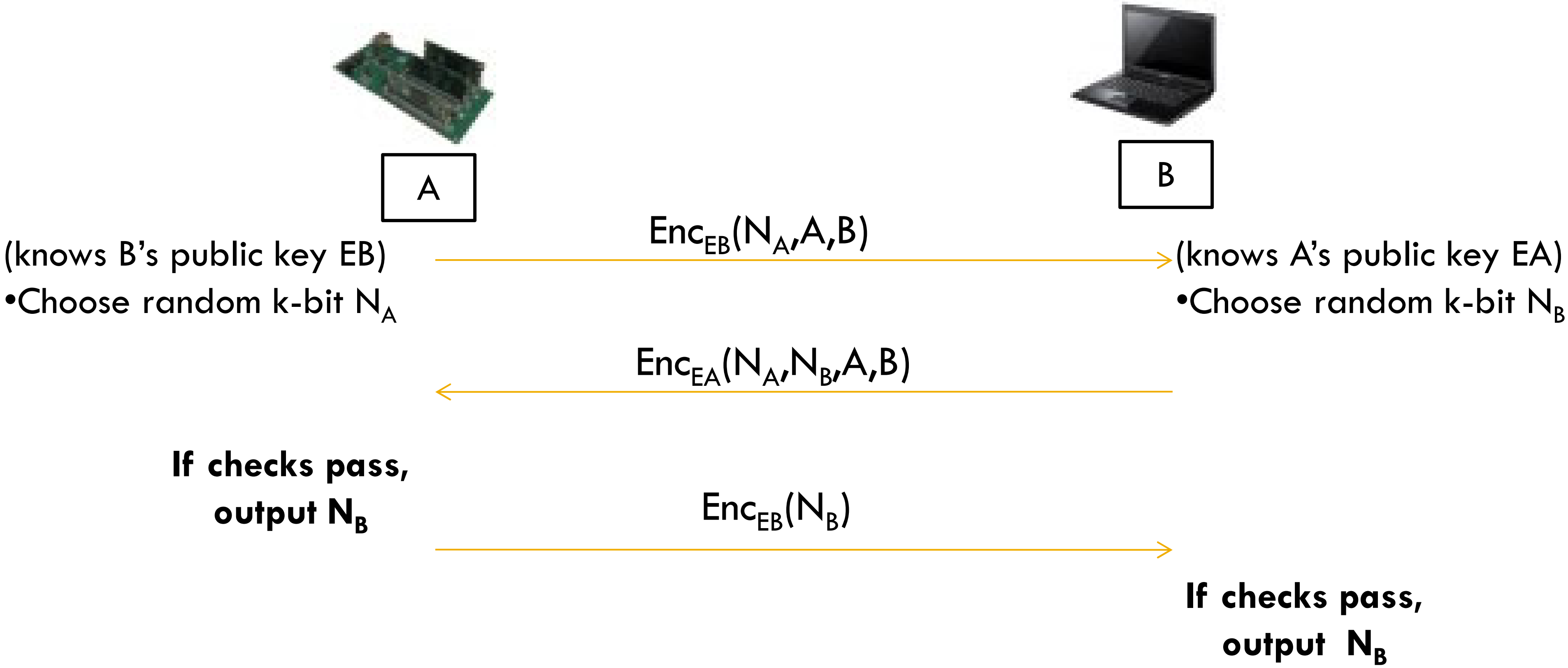
Two protocols use joint secret information in an "uncoordinated way".

  - Key exchange and secure communication over untrusted network

Authenticated key exchange [1]

[based on Needham-Schroeder-Lowe,78+95]

- Encryption-based protocol



(knows B's public key EB)
•Choose random k-bit N_A

$\mathrm{Enc}_{EB}(N_A,A,B)$ →

(knows A's public key EA)
•Choose random k-bit N_B

← $\mathrm{Enc}_{EA}(N_A,N_B,A,B)$

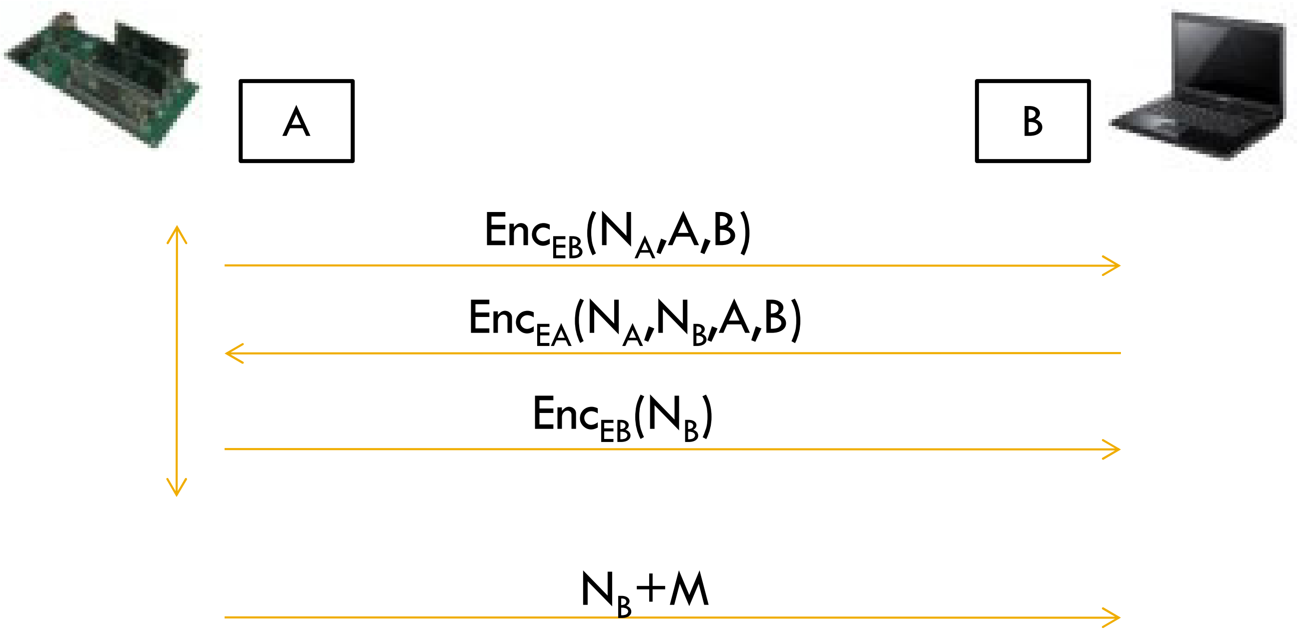
If checks pass, output N_B

$\mathrm{Enc}_{EB}(N_B)$ →

If checks pass, output N_B

Compose the key exchange with Encryption [1]

- The encryption protocol, Enc, is one-time-pad
- The message, M, is either “buy” or “sell”:



A

B

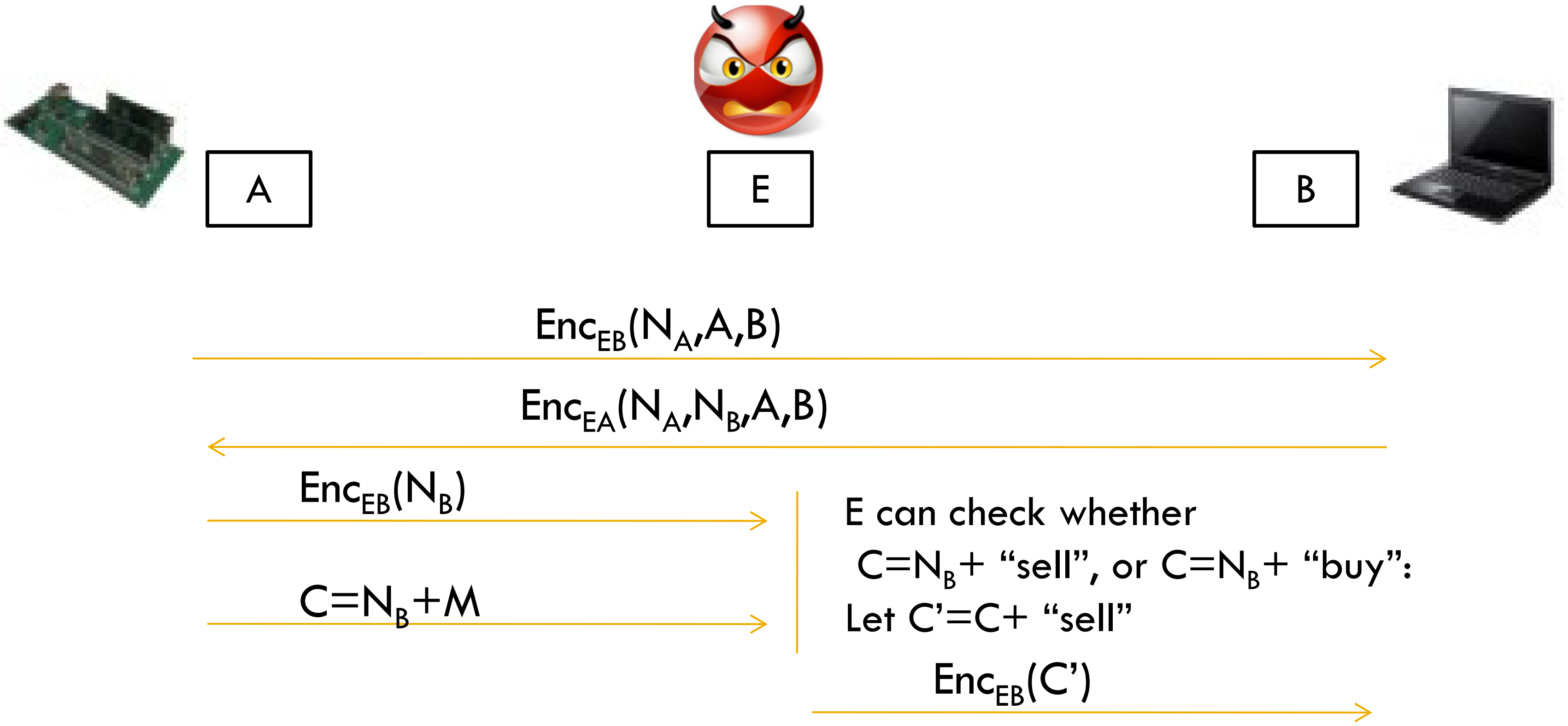
$Enc_{EB}(N_A,A,B)$

$Enc_{EA}(N_A,N_B,A,B)$

$Enc_{EB}(N_B)$

N_B+M

Attack on the composed protocol [1]



$Enc_{EB}(N_A,A,B)$

$Enc_{EA}(N_A,N_B,A,B)$

$Enc_{EB}(N_B)$

$C=N_B+M$

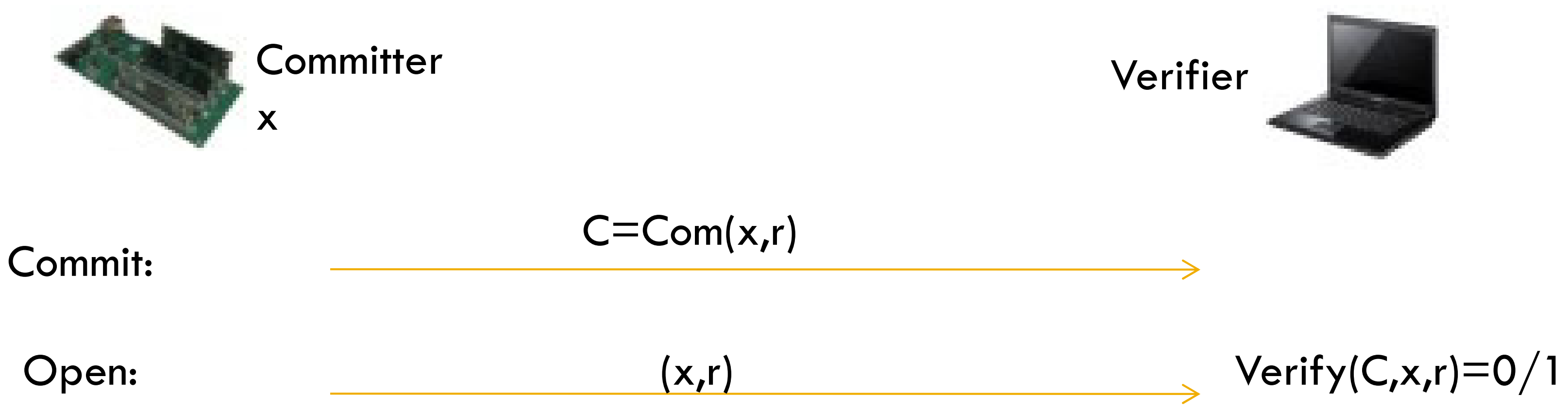
E can check whether
$C=N_B+$ “sell”, or $C=N_B+$ “buy”:
Let C’=C+ “sell”

$Enc_{EB}(C’)$

Note: if M=“sell” then C’=(N_B+ “sell”)+ “sell”=N_B. Else C’ != N_B.
B accepts if and only if M=“sell”

Insufficiency of Standalone security[1]

- Example 3: Malleability of Commitment



Committer

x

Verifier

Commit: C=Com(x,r)

Open: (x,r)

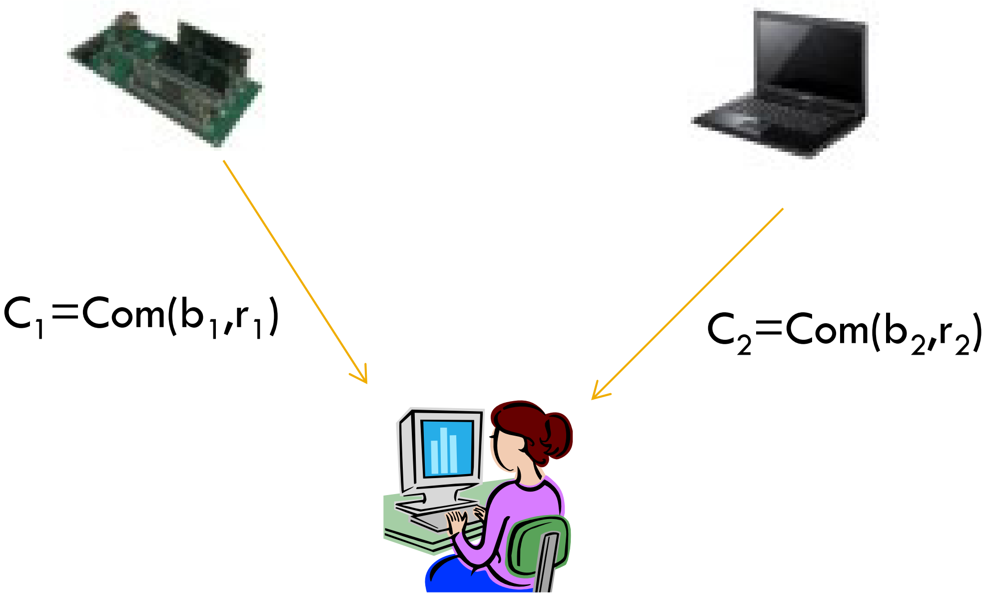
Verify(C,x,r)=0/1

Secrecy (hiding): Nothing is leaked about x

Binding: C can only be opened to a single value x

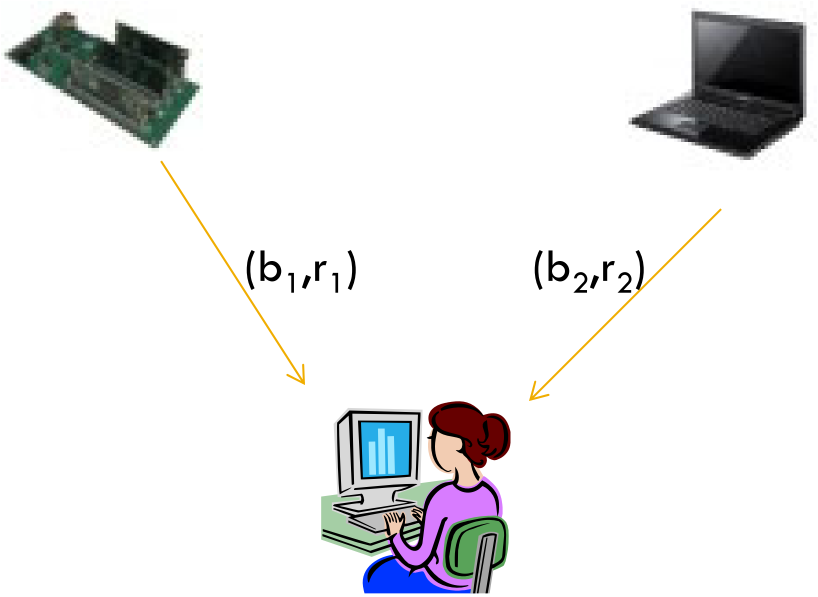
Auction protocol (based on commitments) [1]

- **Phase 1:** Each bidder publishes a commitment to its bid.
- **Phase 2:** Bidders open their commitments.



$C_1 = \mathrm{Com}(b_1, r_1)$

$C_2 = \mathrm{Com}(b_2, r_2)$

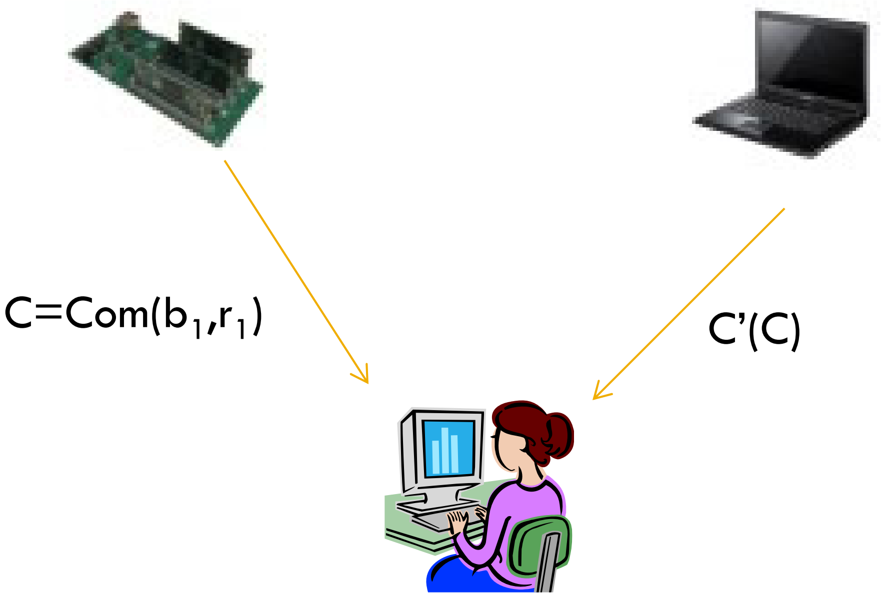


(b_1, r_1)

(b_2, r_2)

Attack on auction protocol [1]

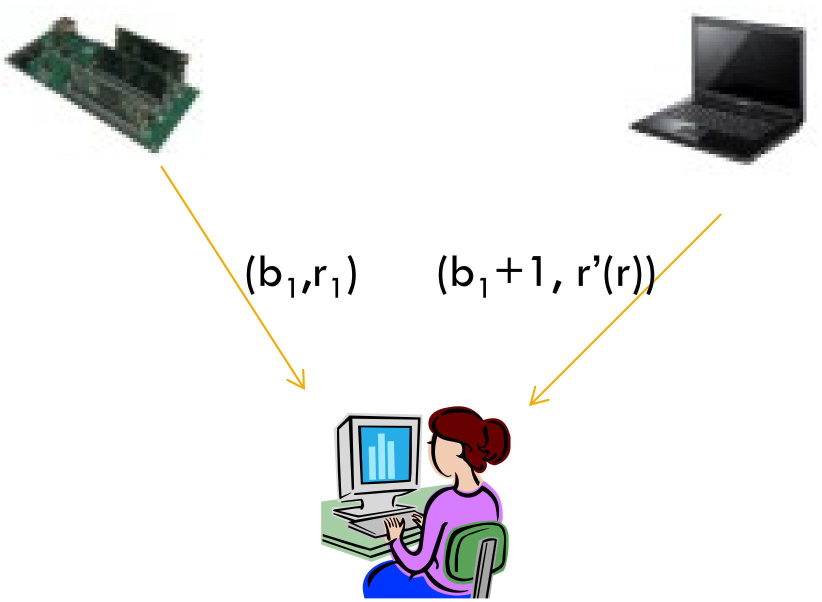
- **Phase 1:** Each bidder publishes a commitment to its bid.



$C=Com(b_1,r_1)$

$C'(C)$

- **Phase 2:** Bidders open their commitments.



(b_1,r_1)

$(b_1+1,\ r'(r))$

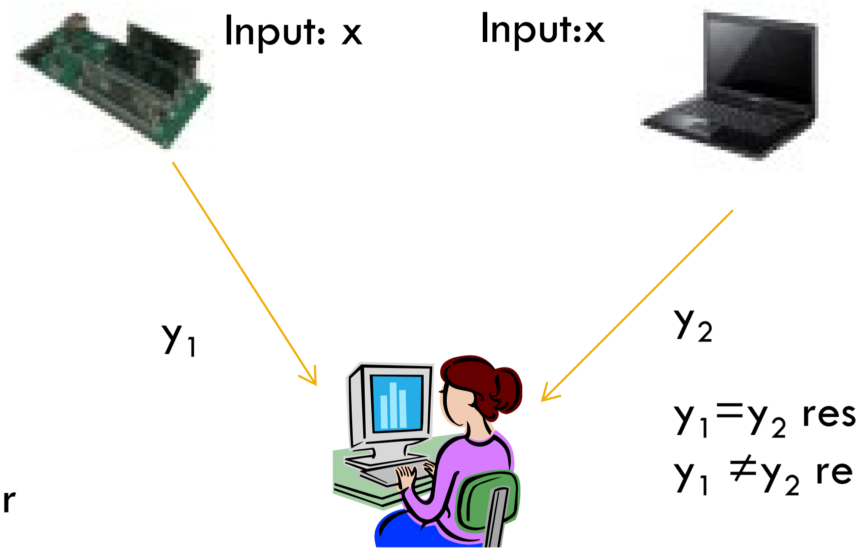
Insufficiency of standalone security [2]

- Example 4: Verifiable computation based on replication:

computation is delegated to two parties, if they both reveal the same result, the result is accepted

  - Correctness
  - Soundness

-- Parties can make a commitment to the result first and then open their commitment ➔ Dishonest party learns the result after opening



Input: x

Input:x

y_1

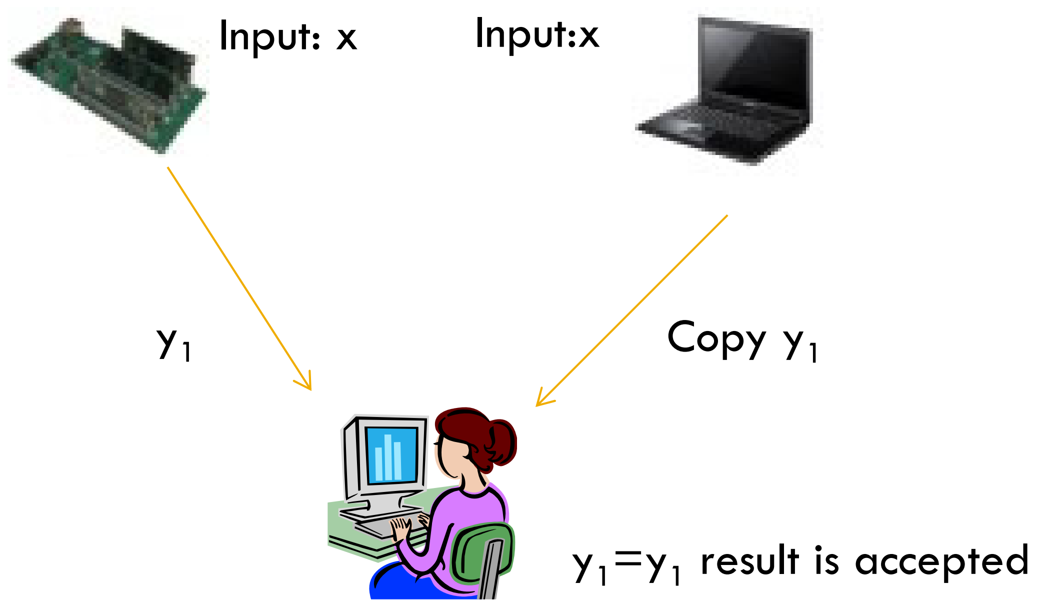
y_2

$y_1 = y_2$ result is correct

$y_1 \neq y_2$ reject

Attack on verifiable computtaion [2]

- In specific threat models and scenarios, correctness is not guaranteed



Input: x

Input:x

y_1

Copy y_1

$y_1=y_1$ result is accepted

[2] Avizheh, S., Nabi, M., Safavi-Naini, R., & Venkateswarlu K, M. (2019, November). Verifiable Computation using Smart Contracts. In *Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop* (pp. 17-28), and a followup paper.

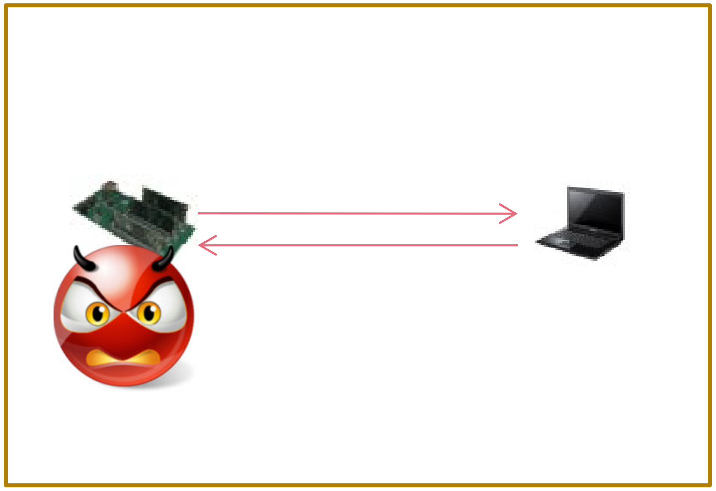
Universally Composable Security (UC) [3]

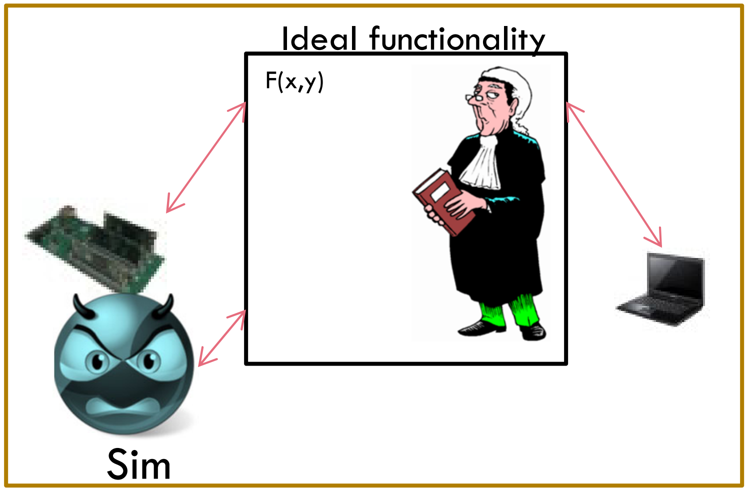
A closer look into simulation based security

- Entities
- Ideal functionality
- Environment

[3] Canetti, R. (2001, October). Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science* (pp. 136-145). IEEE.

Simulation-based security





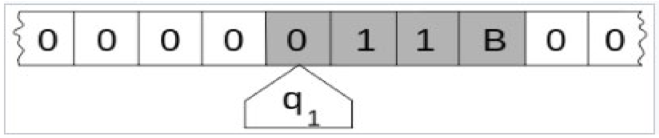
Real world ≈ Ideal world

- Whatever can be achieved in the real world can also be achieved in the ideal world, therefore real world is as secure as ideal world

Entities

TM

- a mathematical model of computation

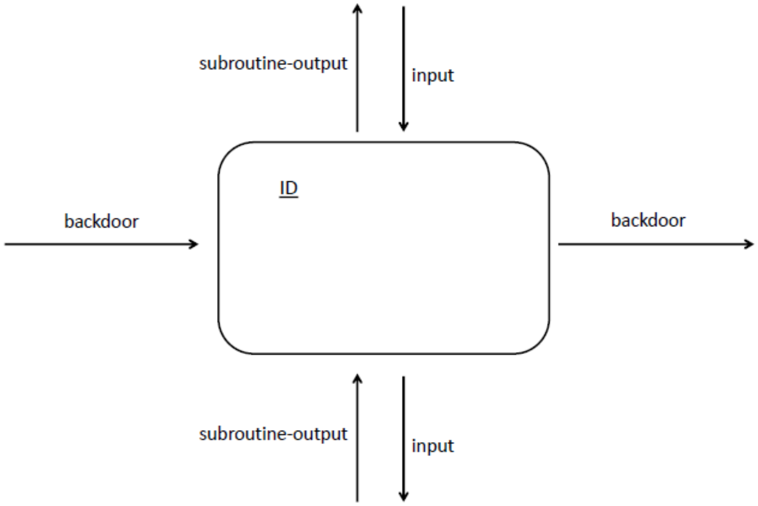


- Tape, head, state register, table of instructions

•Entities are dummy ITMs in ideal world

ITM

- ITM: has special tapes for communicationg with other ITMs
- All entities are modeled as Interacive Turing Machines (ITM)

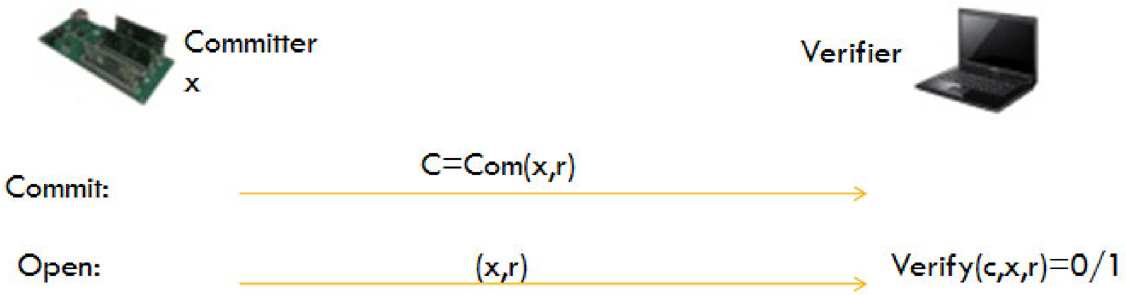


Ideal functionality

- An ideal functionality is an ITM
- Ideal functionality is fully trusted
- Captures the properties required by the scheme
- It interacts with protocol parties and simulator
  - Receives inputs from parties
  - Performs the task at hand
  - Interacts with Sim
  - Returns the result to parties

Ideal functionality for commitment

Real world



Secrecy: Nothing is leaked about x

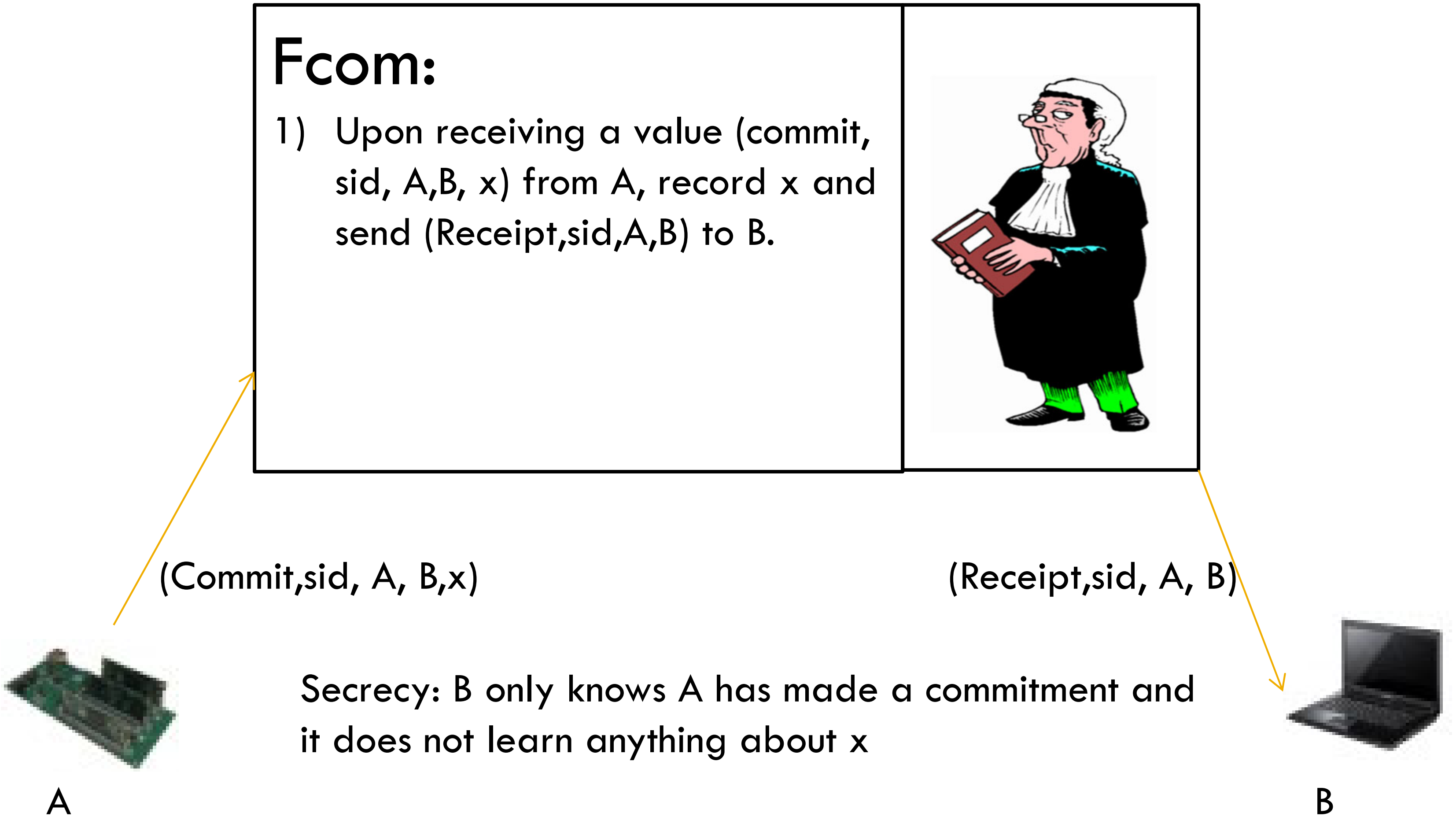
Binding: C can only be opened to a single value x

Ideal world

Fcom: running with parties A and B

1) Upon receiving a value (commit, sid, A,B, x) from A, record x and send (Receipt,sid,A,B) to B.
2) Upon receiving a value (Open, sid,A,B) from A, send (Open, sid, A,B,x) to B and halt. If no such message exist halt.

Commitment: ideal world



Fcom:

1) Upon receiving a value (commit, sid, A,B, x) from A, record x and send (Receipt,sid,A,B) to B.

(Commit,sid, A, B,x)

(Receipt,sid, A, B)

Secrecy: B only knows A has made a commitment and it does not learn anything about x

Commitment: ideal world

Fcom:

2) Upon receiving a value (Open, sid,A,B) from A, send (Open, sid, A,B,x) to B and halt. If no such message exist halt.

(Open,sid, A, B)

(Open,sid, A, B,x)

A

Binding: A cannot open C to a different value x’ ≠ x

B

Other ideal functionalities

Multi party computation

1. Receive (Input, sid,x) from party A
2. Receive (Input,sid,y) from party B
3. Compute z=F(x,y) ➔ Output (Result,sid,z)

- Privacy of inputs
- Correctness of result
- Inputs are independent

Authenticated communication

1. Receive (Send, sid,B,m) from party A, do:
   If this is the first (Send…) input then record (B;m) and send (Sent,sid,A,B,m) to the adversary; else do nothing.
   When receiving (ok) from the adversary, output (Sent,sid,A,B,m) to B and halt.

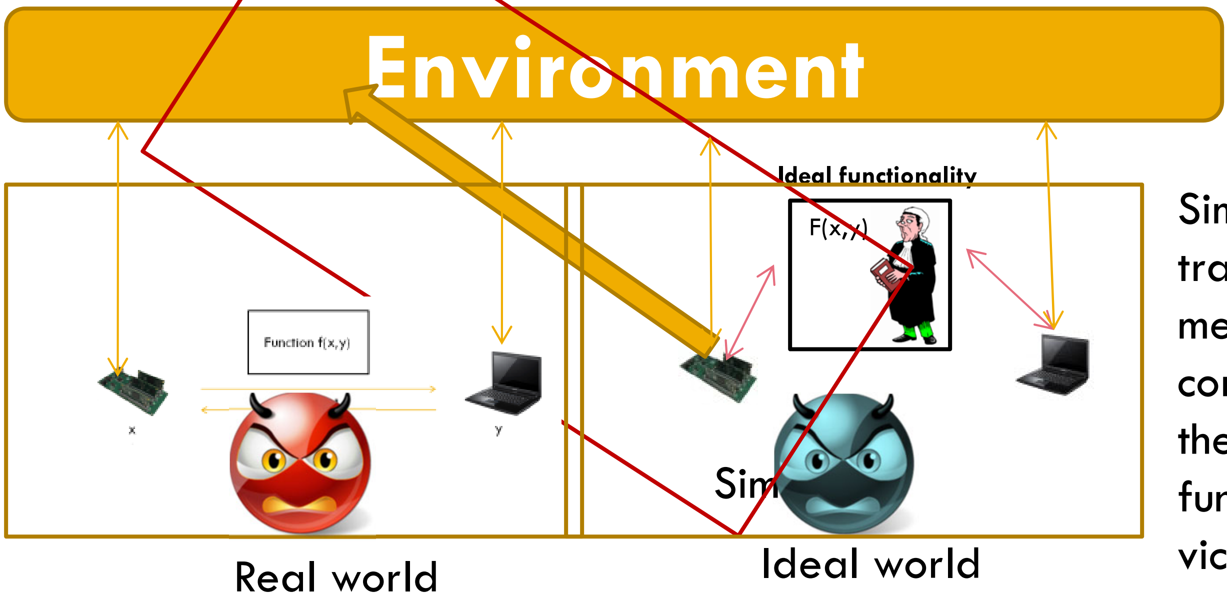
- Non-transferable authentication
- No secrecy for message and parties' identities

Environment

- An ITM which provides input to parties and receive outputs from them
- Captures everything that is external to the protocol of interest
- In UC environment interact with adversary during the protocol

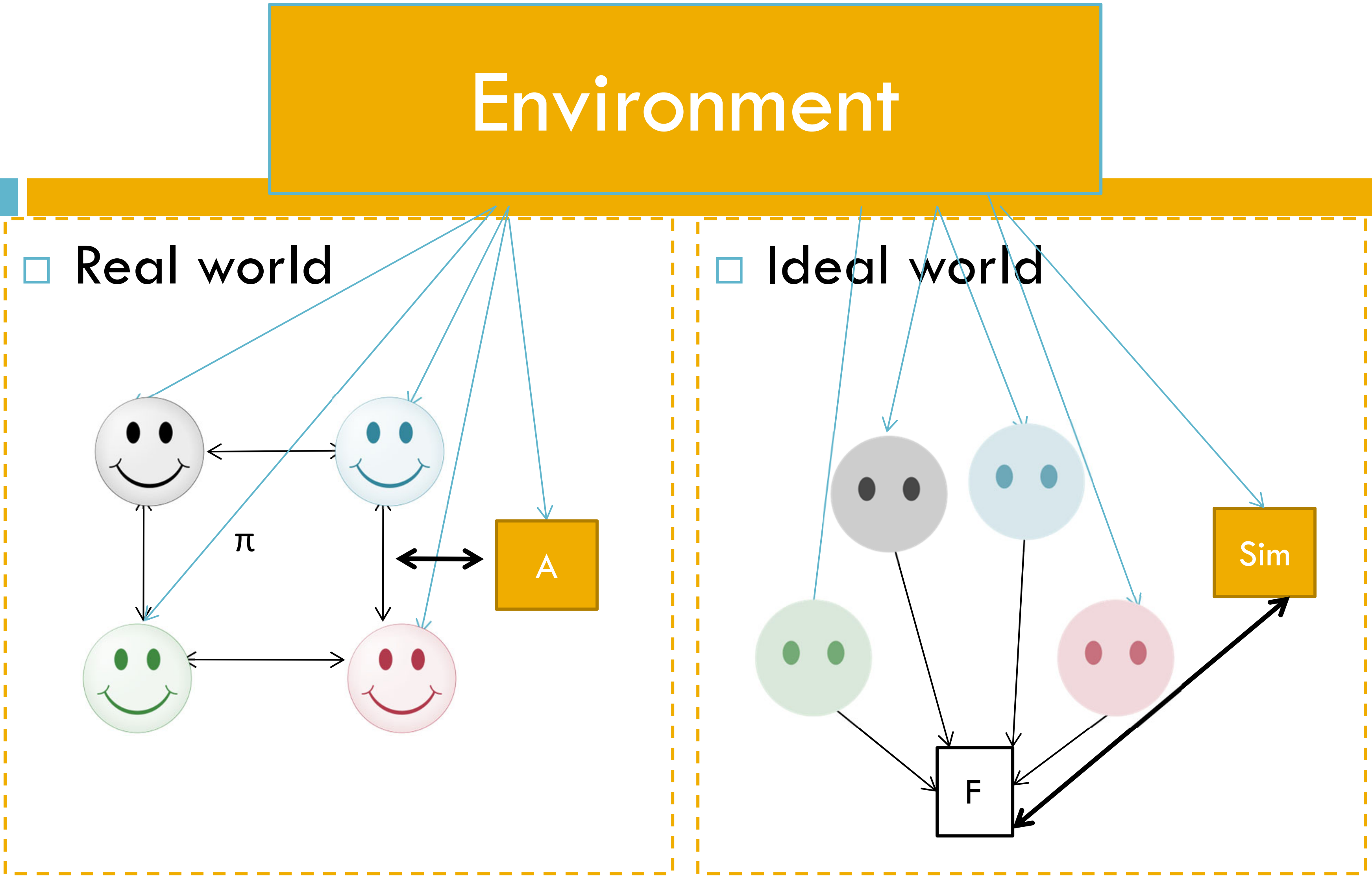
π realizes F

π emulates ρ_F



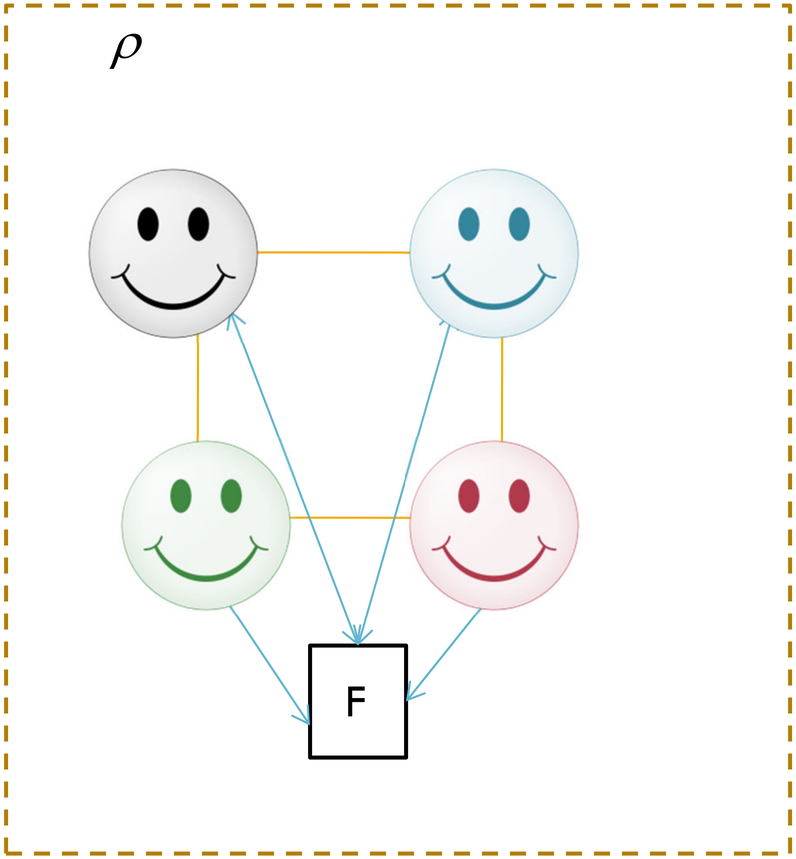
Simulator has to translate the messages of corrupted party to the ideal functionality and vice versa

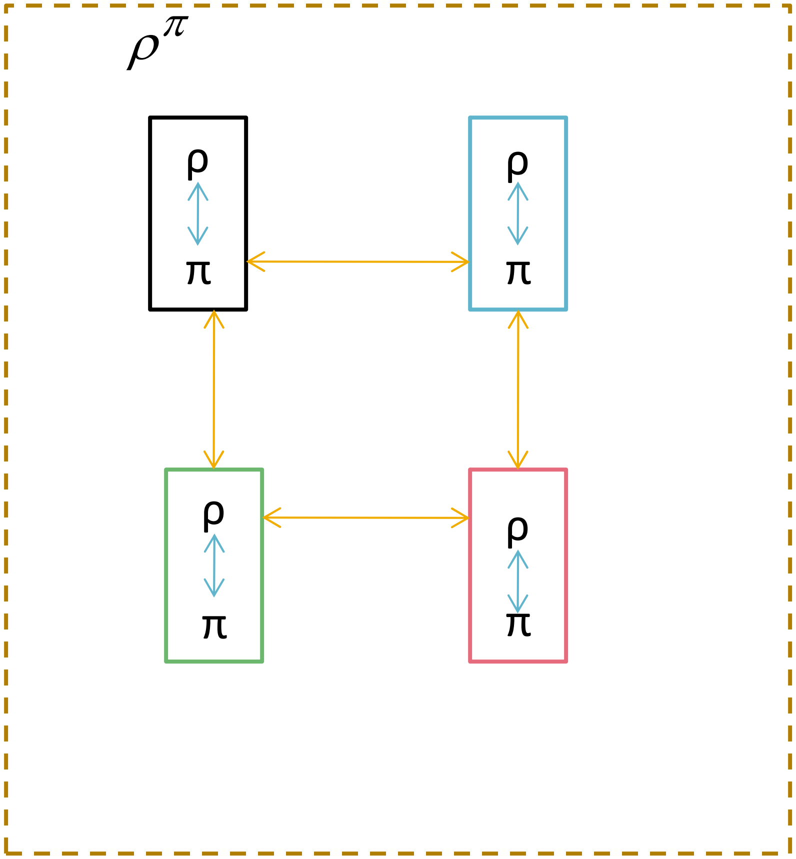
UC Theorem [3]



- Real world
- Ideal world

Universal composition theorem

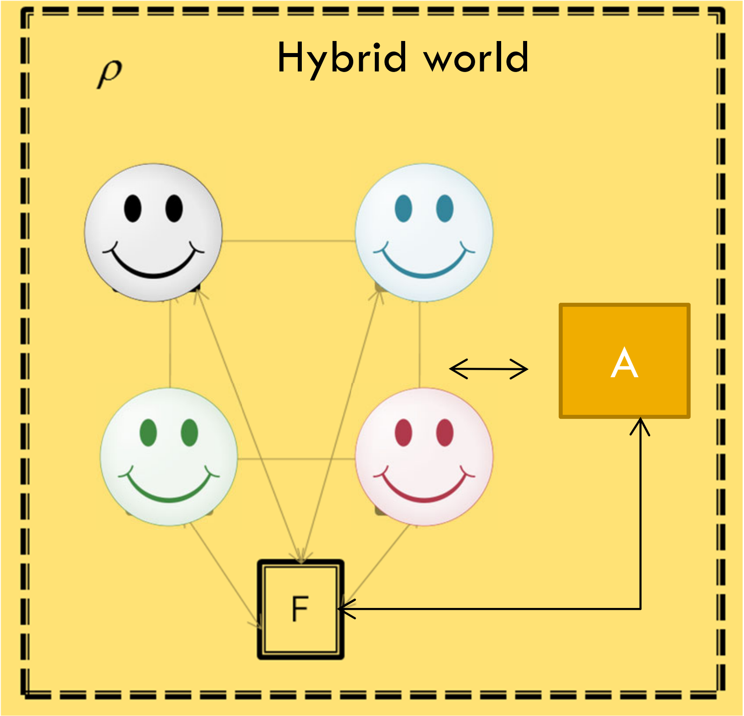




Protocol ρ^{π} emulates protocol ρ.

What is obtained?

1. Decompose the protocol to smaller modules
2. For each subroutines, formalize the specifications of the protocol using ideal functionality F in the presence of simulator Sim
3. Replace subroutines with ideal functionalities (hybrid world)
4. Build the ideal model, and show that Sim is able to simulate the protocol transcript



An example

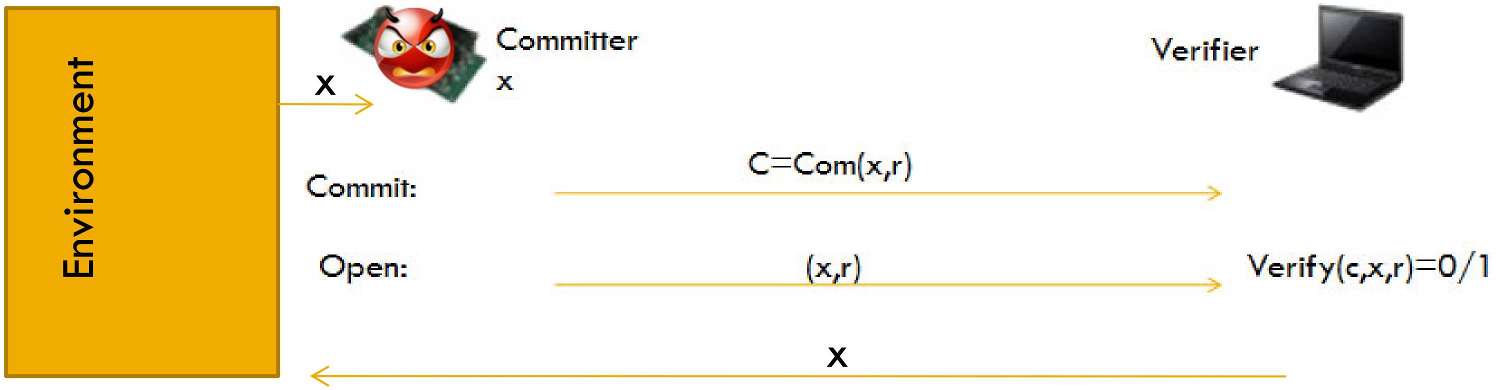
Commitment scheme

- Impossibility results [4]
- UC secure commitment with set up assumption
- How simulation is done

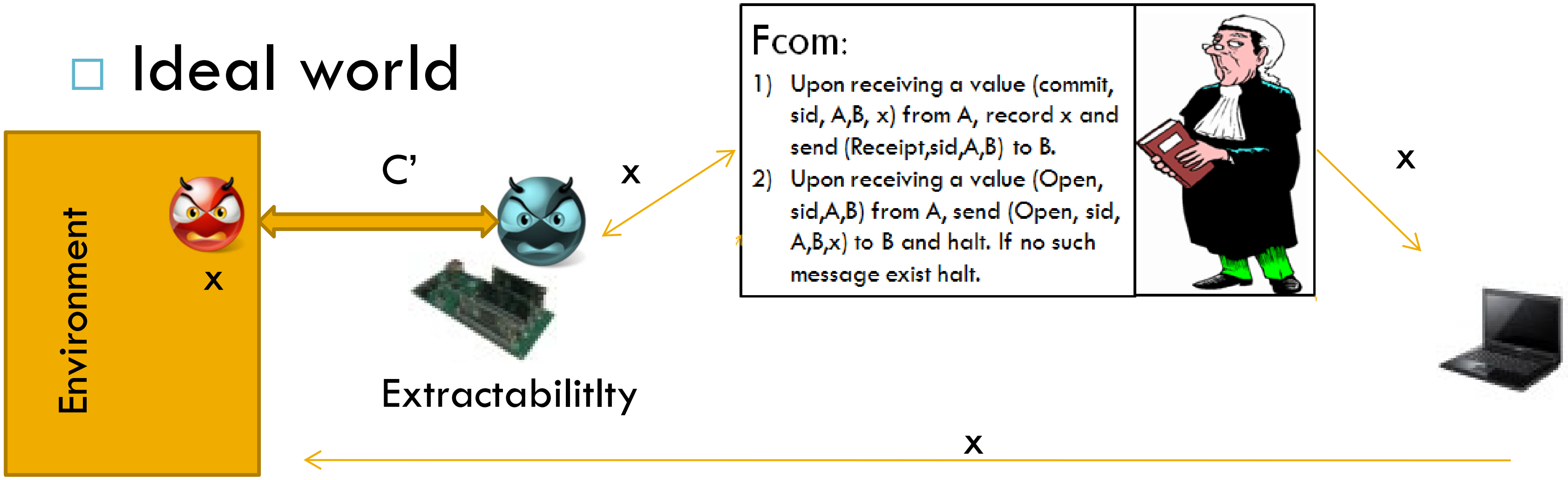
[4] Canetti, R., & Fischlin, M. (2001, August). Universally composable commitments. In *Annual International Cryptology Conference* (pp. 19-40). Springer, Berlin, Heidelberg.

Example: Impossibility of UC secure commitment scheme in plain model [4]

- Real world



- Ideal world



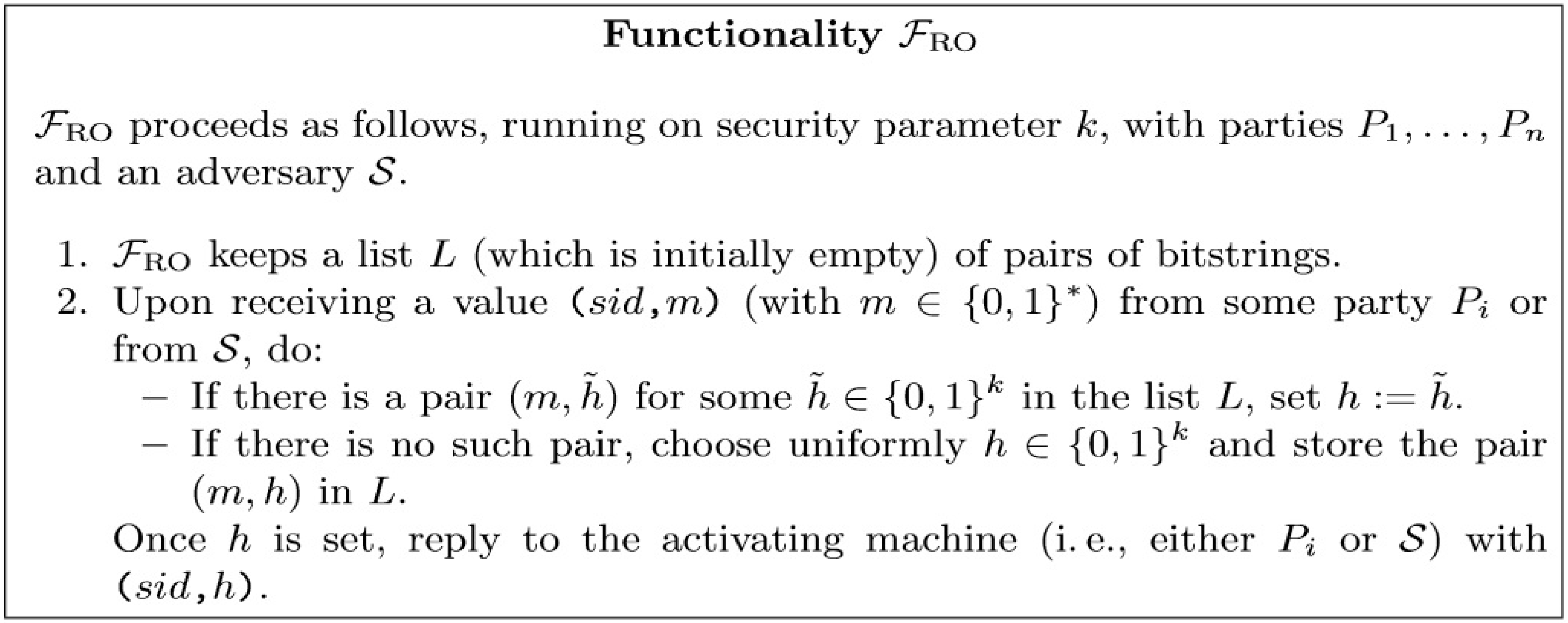
Fcom:

1) Upon receiving a value (commit, sid, A,B, x) from A, record x and send (Receipt,sid,A,B) to B.
2) Upon receiving a value (Open, sid,A,B) from A, send (Open, sid, A,B,x) to B and halt. If no such message exist halt.

Random oracle functionality [5]

Random oracle:

- For any message outputs a random value chosen from Uniform distribution
  - For each message there is a single random value (collision-resistance)
  - It is not possible to find m from h (pre-image resistance)



Functionality $\mathcal{F}_{\mathrm{RO}}$

$\mathcal{F}_{\mathrm{RO}}$ proceeds as follows, running on security parameter k, with parties $P_1, \ldots, P_n$ and an adversary $\mathcal{S}$.

1. $\mathcal{F}_{\mathrm{RO}}$ keeps a list L (which is initially empty) of pairs of bitstrings.
2. Upon receiving a value (sid, m) (with $m \in \{0,1\}^*$) from some party P_i or from $\mathcal{S}$, do:
   - If there is a pair $(m, \tilde{h})$ for some $\tilde{h} \in \{0,1\}^k$ in the list L, set $h := \tilde{h}$.
   - If there is no such pair, choose uniformly $h \in \{0,1\}^k$ and store the pair (m, h) in L.

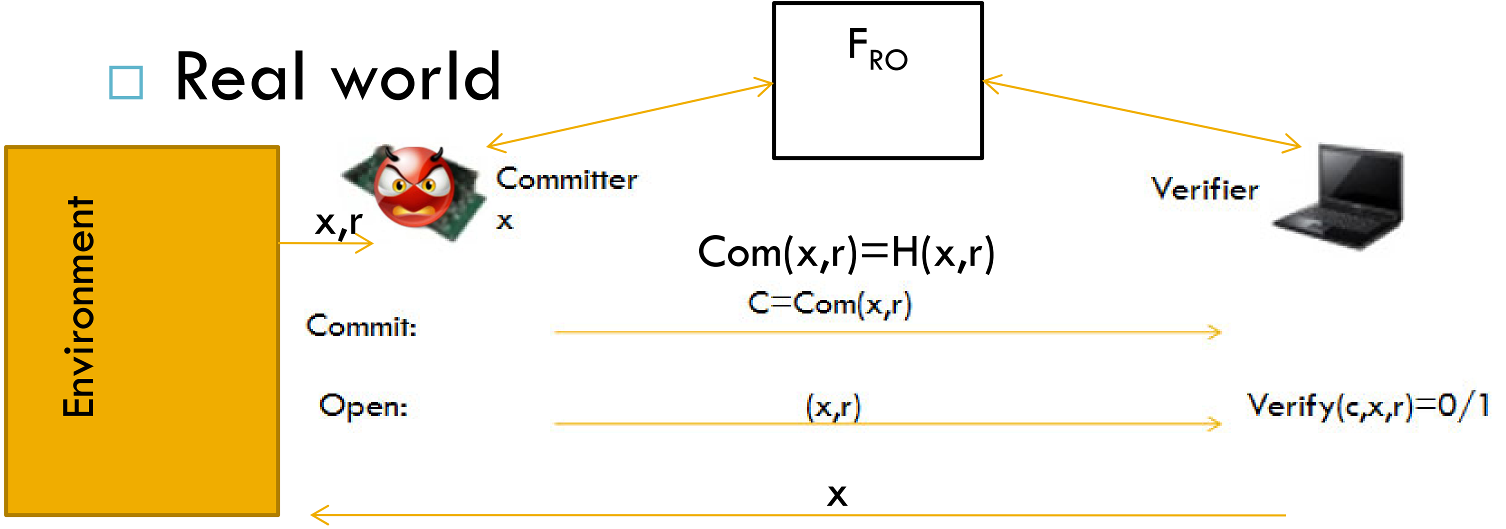
   Once h is set, reply to the activating machine (i.e., either P_i or $\mathcal{S}$) with (sid, h).

[5] Hofheinz, D., & Müller-Quade, J. (2004, February). Universally composable commitments using random oracles. In *Theory of Cryptography Conference* (pp. 58-76). Springer, Berlin, Heidelberg.

UC secure commitment in RO model (Extractability) [6]

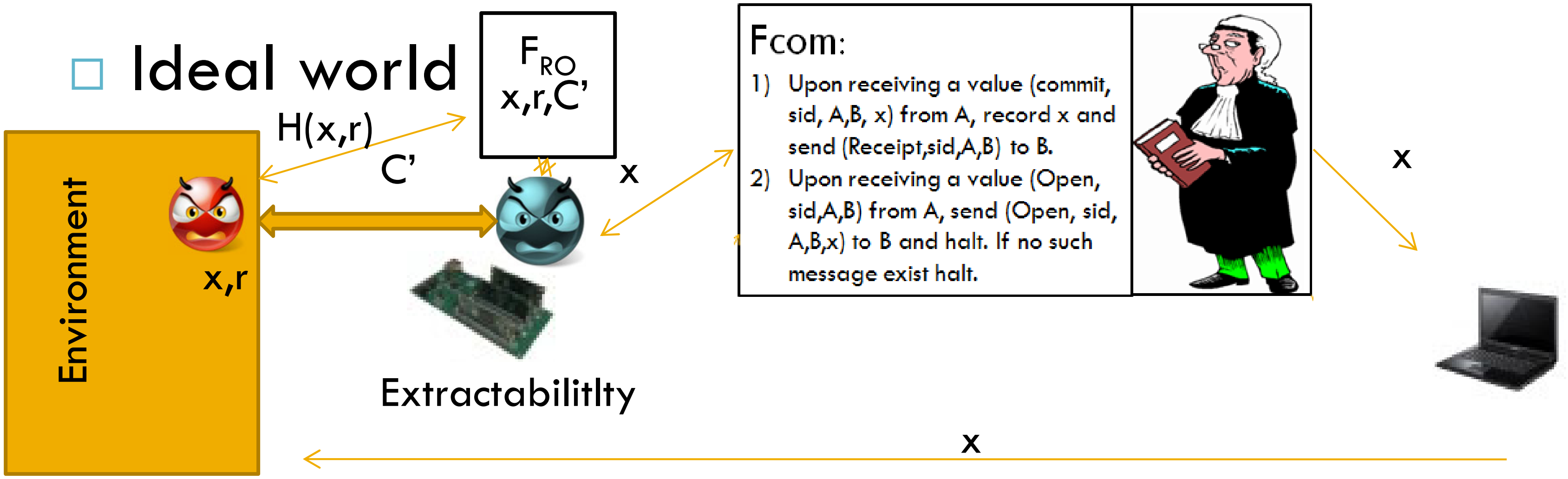
[6] Dziembowski, S., Eckey, L., & Faust, S. (2018, October). Fairswap: How to fairly exchange digital goods. *ACM CCS* (pp. 967-984).

- ☐ Real world



Remark: Identities can also be used as input to H

- ☐ Ideal world

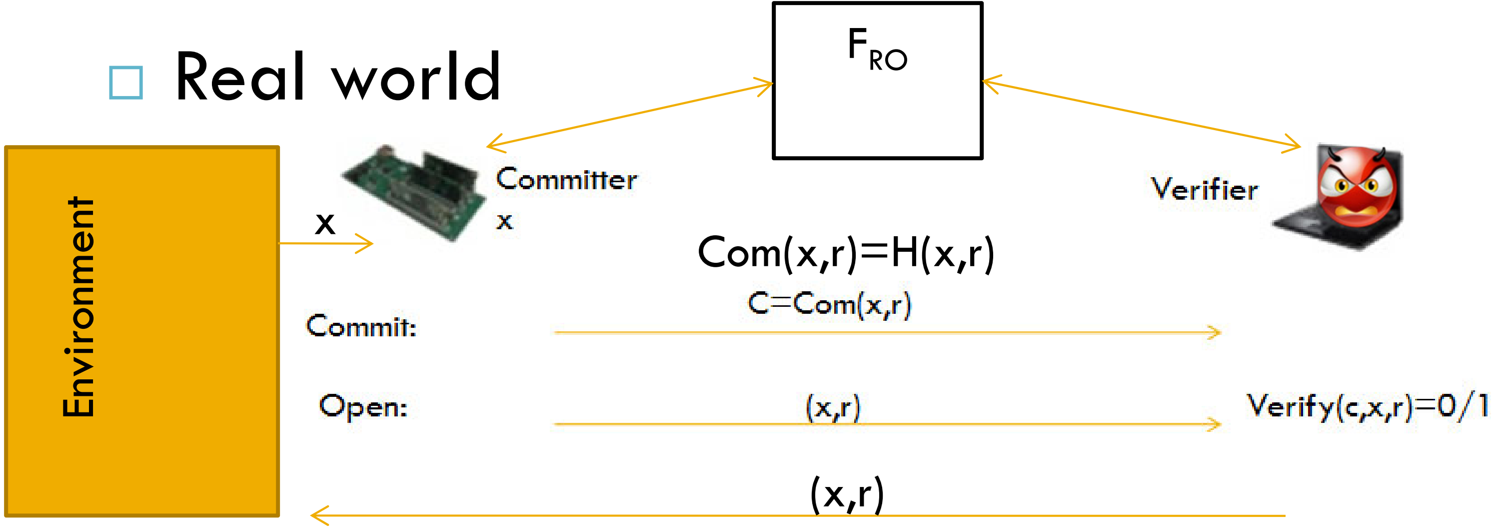


Fcom:

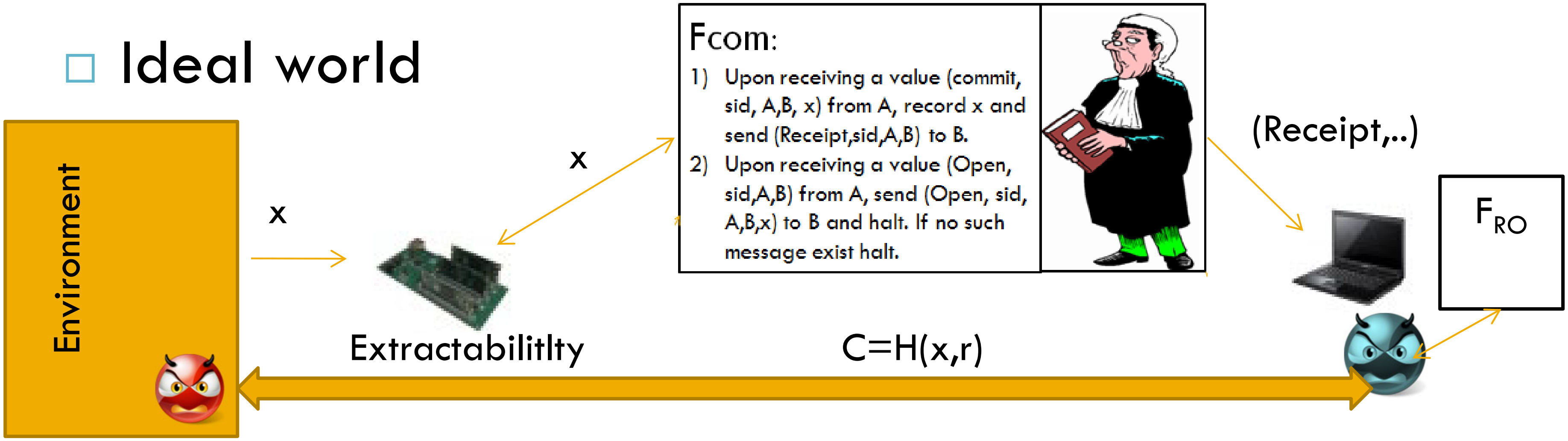
1) Upon receiving a value (commit, sid, A,B, x) from A, record x and send (Receipt,sid,A,B) to B.
2) Upon receiving a value (Open, sid,A,B) from A, send (Open, sid, A,B,x) to B and halt. If no such message exist halt.

UC secure commitment in RO model (Simulatibility)

- ☐ Real world



- ☐ Ideal world

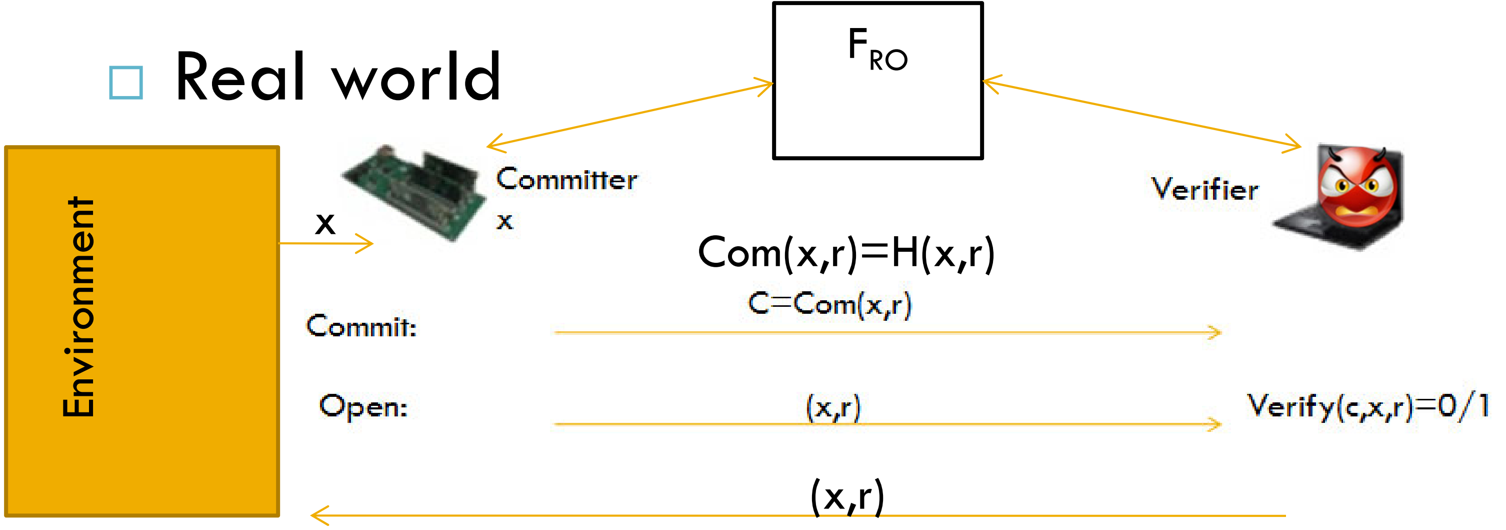


Fcom:

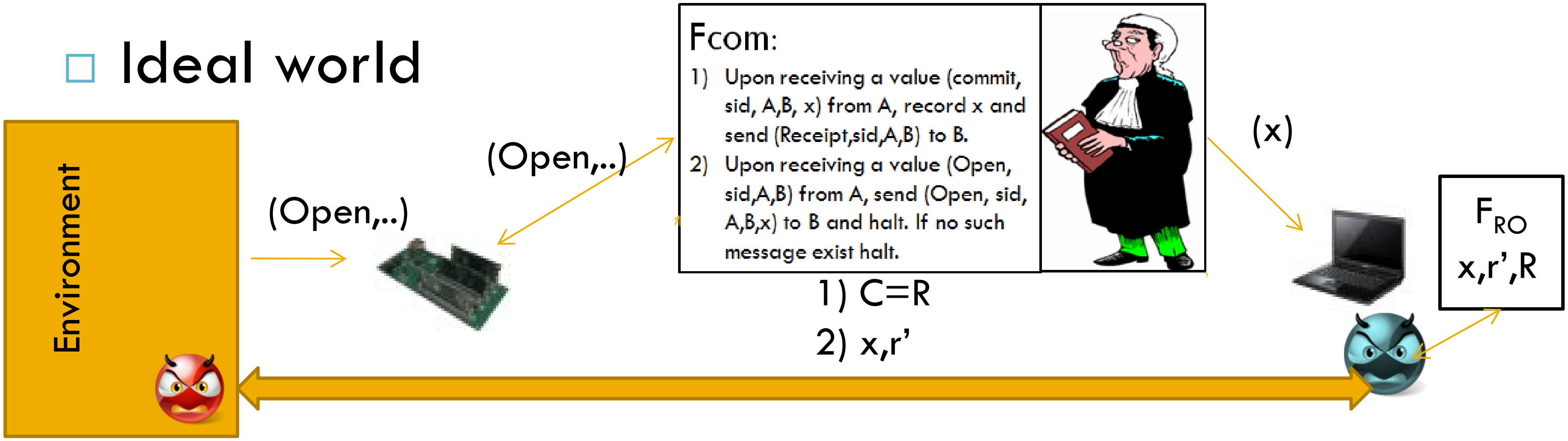
1) Upon receiving a value (commit, sid, A,B, x) from A, record x and send (Receipt,sid,A,B) to B.
2) Upon receiving a value (Open, sid,A,B) from A, send (Open, sid, A,B,x) to B and halt. If no such message exist halt.

UC secure commitment in RO model (Simulatibility)

- ☐ Real world



- ☐ Ideal world



Fcom:

1) Upon receiving a value (commit, sid, A,B, x) from A, record x and send (Receipt,sid,A,B) to B.
2) Upon receiving a value (Open, sid,A,B) from A, send (Open, sid, A,B,x) to B and halt. If no such message exist halt.

Concluding remarks

- Standalone security is not sufficient in practice
- UC security ensures that a protocol maintains its security in an unpredictable environment
- There are variants of UC security:
  - JUC: Joint state UC framework
  - GUC generalized UC framework
  - UC with non-monolithic adversaries
  - …
- There are lots of impossibility results

Thank you!