A Simpler Variant of Universally Composable Security for Standard Multi Party Computation

Chloé Hébant

Ecole Normale Supérieure

February 22, 2018

Context

Protocol

Context

Protocol

Proof of security

Context

Protocol

Proof of security

Adversary model
$\rightarrow$ who?
$\rightarrow$ capabilities?
$\rightarrow$ goals?

Context

Protocol

Proof of security

Adversary model
$\rightarrow$ who?
$\rightarrow$ capabilities?
$\rightarrow$ goals?

Security model

Context

Protocol

Proof of security

Adversary model
→ who?
→ capabilities?
→ goals?

Security model

Indistinguishability
→ Find-then-Guess
→ Real-or-Random

Context

Protocol

Proof of security

Adversary model
→ who?
→ capabilities?
→ goals?

Security model

Indistinguishability
→ Find-then-Guess
→ Real-or-Random

Simulation
→ Classical Simulation
→ Universal Composability

Definition

Universal Composability model is a security model

- for **Multi Party Computation**

Definition

Universal Composability model is a security model

- for **Multi Party Computation**: n players $\mathcal{P}_i$ owning x_i, n-variable function f, Compute $f(x_1, \cdots, x_n) = (y_1, \cdots, y_n)$ s.t. each $\mathcal{P}_i$ learns y_i and nothing more

Definition

Universal Composability model is a security model

- for **Multi Party Computation**: n players $\mathcal{P}_i$ owning x_i, n-variable function f, Compute $f(x_1, \cdots, x_n) = (y_1, \cdots, y_n)$ s.t. each $\mathcal{P}_i$ learns y_i and nothing more
- based on a simulation between a **Real World** and an **Ideal World**

Definition

Universal Composability model is a security model

- for **Multi Party Computation**: n players $\mathcal{P}_i$ owning x_i, n-variable function f, Compute $f(x_1, \cdots, x_n) = (y_1, \cdots, y_n)$ s.t. each $\mathcal{P}_i$ learns y_i and nothing more
- based on a simulation between a **Real World** and an **Ideal World**
  - **Real World**: protocol, players, adversary
  - **Ideal World**: ideal protocol, virtual players, ideal adversary

Definition

Universal Composability model is a security model

- for **Multi Party Computation**: n players $\mathcal{P}_i$ owning x_i, n-variable function f, Compute $f(x_1, \cdots, x_n) = (y_1, \cdots, y_n)$ s.t. each $\mathcal{P}_i$ learns y_i and nothing more
- based on a simulation between a **Real World** and an **Ideal World**
  - **Real World**: protocol, players, adversary
  - **Ideal World**: ideal functionality, virtual players, ideal adversary

Definition

Universal Composability model is a security model

- for **Multi Party Computation**: n players $\mathcal{P}_i$ owning x_i, n-variable function f, Compute $f(x_1, \cdots, x_n) = (y_1, \cdots, y_n)$ s.t. each $\mathcal{P}_i$ learns y_i and nothing more
- based on a simulation between a **Real World** and an **Ideal World**
  - **Real World**: protocol, players, adversary
  - **Ideal World**: ideal functionality, virtual players, simulation of the adversary

Definition

Universal Composability model is a security model

- for **Multi Party Computation**: n players $\mathcal{P}_i$ owning x_i, n-variable function f, Compute $f(x_1, \cdots, x_n) = (y_1, \cdots, y_n)$ s.t. each $\mathcal{P}_i$ learns y_i and nothing more
- based on a simulation between a **Real World** and an **Ideal World**
  - **Real World**: protocol, players, adversary
  - **Ideal World**: ideal functionality, virtual players, simulation of the adversary

Ensure that an **environment** $\mathcal{Z}$ can't distinguish between both worlds

Definition

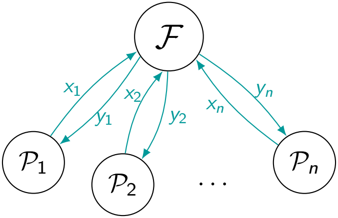


Figure 1: Ideal World

Definition

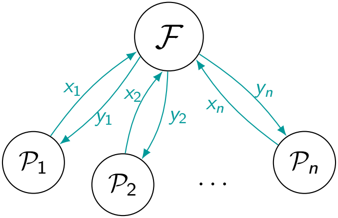


Figure 1: Ideal World

Construction of UC protocols:

- Define the ideal Functionality $\mathcal{F}$
- Construct a protocol Π that realises $\mathcal{F}$
- Make the proof: construct a simulator $\mathcal{S}$

Interest 1: $\mathcal{A}$ can choose a distribution for the inputs

In the UC model, no description of:

- what are the possible actions of the adversary
- the order of the requests
- the number of requests

Interest 1: $\mathcal{A}$ can choose a distribution for the inputs

In the UC model, no description of:

- what are the possible actions of the adversary
- the order of the requests
- the number of requests

The execution is taken as a whole: $\mathcal{Z}$ chooses the inputs of $\mathcal{P}_i$ and $\mathcal{A}$

Interest 1: $\mathcal{A}$ can choose a distribution for the inputs

In the UC model, no description of:

- what are the possible actions of the adversary
- the order of the requests
- the number of requests

The execution is taken as a whole: $\mathcal{Z}$ chooses the inputs of $\mathcal{P}_i$ and $\mathcal{A}$

$\Rightarrow$ Model attacks where the **inputs are not uniform**

Interest 2: The composition theorem

Most important interest:

If a protocol is UC secure then it is secure for concurrent executions

Interest 2: The composition theorem

Most important interest:

If a protocol is UC secure then it is secure for concurrent executions

Example 1: UC-commitments $\rightarrow$ ZK

Example 2:

UC-secure authenticated key exchange + secure symmetric encryption
$\rightarrow$ Secure channels

Interest 2: The composition theorem

Most important interest:

If a protocol is UC secure then it is secure for concurrent executions

Example 1: UC-commitments $\rightarrow$ ZK

Example 2:

UC-secure authenticated key exchange + secure symmetric encryption
$\rightarrow$ Secure channels

$\Rightarrow$ Because of these 2 points, the **UC model is more secure** than the Find-then-Guess or Real-or-Random models

Difficulty to define the ideal functionality

Ideal Functionality for Secure Message Transfer

Difficulty to define the ideal functionality

Ideal Functionality for Secure Message Transfer

$\mathcal{F}^{l}_{\mathsf{STM}}$ proceeds as follows:
parameterized by leakage function $l : \{0,1\}^\star \to \{0,1\}^\star$,

Upon receiving an input $(\mathsf{Send}, \mathsf{sid}, m)$ from S, verify that $\mathsf{sid} = (S, R, \mathsf{sid}')$ for some R, else ignore the input. Next, send $(\mathsf{Sent}, \mathsf{sid}, l(m), m)$ to R.

text = private content

Difficulty to define the ideal functionality

Ideal Functionality for Secure Message Transfer

$\mathcal{F}^{l}_{\mathsf{STM}}$ proceeds as follows:
parameterized by leakage function $l : \{0,1\}^\star \to \{0,1\}^\star$,

Upon receiving an input $(\mathsf{Send}, \mathsf{sid}, m)$ from S, verify that $\mathsf{sid} = (S, R, \mathsf{sid}')$ for some R, else ignore the input. Next, send $(\mathsf{Sent}, \mathsf{sid}, l(m), m)$ to R.

text = private content

For example: leaking $l(m) = \mathsf{length}(m)$ is important because no cryptosystem can fully hide the size of the information being encrypted

Difficulties in proofs

In UC model, proofs more complex than in game based security:

- no rewind, need extractable inputs $\Rightarrow$ protocol more complex
- no end when the adversary wins $\Rightarrow$ proofs more complex

Communication model and rules

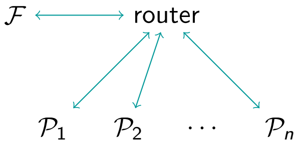


Figure 2: SUC communication model

Communication model and rules

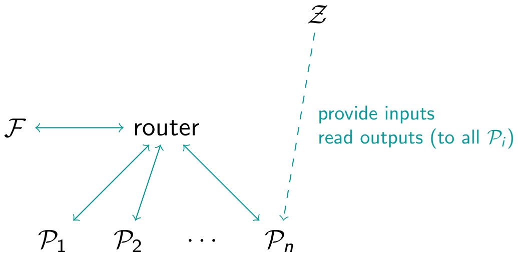


Figure 2: SUC communication model

Communication model and rules

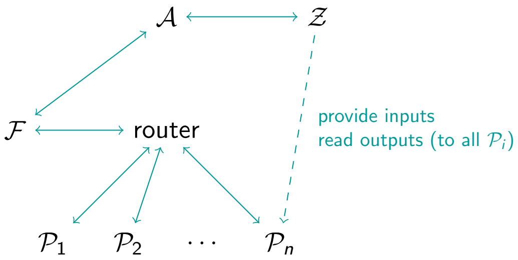


Figure 2: SUC communication model

Communication model and rules

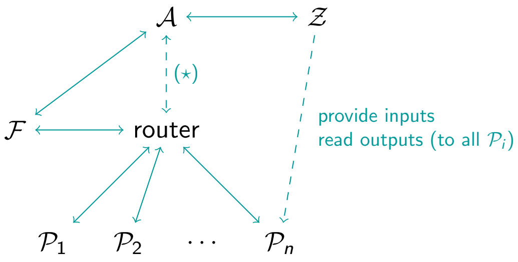


Figure 2: SUC communication model

$(\star)$ Router sends all messages to $\mathcal{A}$ and delivers them when instructed by $\mathcal{A}$

- Messages are of the format (sender,receiver;content)
- Router only sends public header of messages to and from $\mathcal{F}$ to $\mathcal{A}$ (so $\mathcal{A}$ does not see the private content)
- $\mathcal{A}$ notifies the router when to deliver messages but has no influence beyond that

π SUC-securely computes $\mathcal{F}$

Definition

Let π be a protocol for up to m parties and let $\mathcal{F}$ be an ideal functionality.

We say that π **SUC-securely computes** $\mathcal{F}$ if for every PPT real model adversary $\mathcal{A}$ there exists a PPT ideal-model adversary $\mathcal{S}$ such that for every PPT balanced environment $\mathcal{Z}$ and every constant $d \in \mathbb{N}$, there exists a negligible function $\mu(\cdot)$ such that for every $n \in \mathbb{N}$ and every $z \in \{0,1\}^\star$ of length at most n^d,

$$|\Pr[\text{SUC-IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}(n,z) = 1] - \Pr[\text{SUC-REAL}_{\pi,\mathcal{A},\mathcal{Z}}(n,z) = 1]| \leqslant \mu(n)$$

SUC composition theorem

Theorem

Let π be a protocol for the $\mathcal{F}$-hybrid model.
Let ρ be a protocol that SUC-securely computes $\mathcal{F}$ in the $\mathcal{G}$-hybrid model.

Then, for every PPT real model adversary $\mathcal{A}$ there exists a PPT ideal-model adversary $\mathcal{S}$ such that for every PPT environment $\mathcal{Z}$ there exists a negligible function $\mu(\cdot)$ such that for every $z \in \{0,1\}^\star$ and every $n \in \mathbb{N}$,

$$\left|\Pr[\text{SUC-HYBRID}^{\mathcal{G}}_{\pi^\rho,\mathcal{S},\mathcal{Z}}(n,z)=1] - \Pr[\text{SUC-HYBRID}^{\mathcal{F}}_{\pi,\mathcal{A},\mathcal{Z}}(n,z)=1]\right| \leqslant \mu(n)$$

SUC composition theorem

Corollary

Let π be a protocol that SUC-securely computes a functionality $\mathcal{H}$ in the $\mathcal{F}$-hybrid model. If protocol ρ SUC-securely computes $\mathcal{F}$ in the $\mathcal{G}$-hybrid (resp. real) model, then π^ρ SUC-securely computes $\mathcal{H}$ in the $\mathcal{G}$-hybrid (resp. real) model.

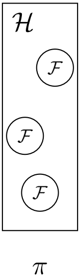
By a drawing:

SUC composition theorem

Corollary

Let π be a protocol that SUC-securely computes a functionality $\mathcal{H}$ in the $\mathcal{F}$-hybrid model. If protocol ρ SUC-securely computes $\mathcal{F}$ in the $\mathcal{G}$-hybrid (resp. real) model, then π^ρ SUC-securely computes $\mathcal{H}$ in the $\mathcal{G}$-hybrid (resp. real) model.

By a drawing:

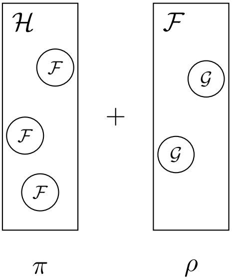


SUC composition theorem

Corollary

Let π be a protocol that SUC-securely computes a functionality $\mathcal{H}$ in the $\mathcal{F}$-hybrid model. If protocol ρ SUC-securely computes $\mathcal{F}$ in the $\mathcal{G}$-hybrid (resp. real) model, then π^ρ SUC-securely computes $\mathcal{H}$ in the $\mathcal{G}$-hybrid (resp. real) model.

By a drawing:

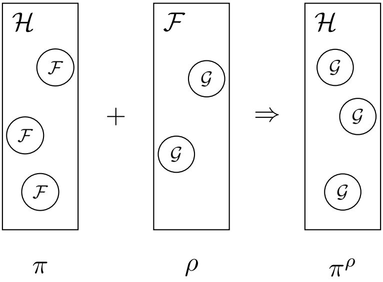


SUC composition theorem

Corollary

Let π be a protocol that SUC-securely computes a functionality $\mathcal{H}$ in the $\mathcal{F}$-hybrid model. If protocol ρ SUC-securely computes $\mathcal{F}$ in the $\mathcal{G}$-hybrid (resp. real) model, then π^ρ SUC-securely computes $\mathcal{H}$ in the $\mathcal{G}$-hybrid (resp. real) model.

By a drawing:



Bonus: Differences SUC - UC

In SUC, more rigid network model:

- build-in authenticated channel
- no subroutines
- set of parties a priori fixed

$\Rightarrow$ No digital signatures in SUC because no a priori polynomial bound on the number of interactions ($=$ number of signatures)

Conclusion

UC: Security model based on simulation to obtain Composition Theorem

Composition Theorem: If a protocol is UC secure then it is secure for concurrent executions

SUC: Simpler formalism for some protocols such that SUC-secure $\Rightarrow$ UC secure

$\Rightarrow$ Simpler proofs without loss of security guarantees

References

- CCL15 - A Simpler Variant of UC Security for Standard Multiparty Computation
- Che09 - Etude de protocoles cryptographiques à base de mots de passe
- Can01 - Universally Composable Security: A New Paradigm for Cryptographic Protocols