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

Chloé Hébant

Ecole Normale Supérieure

February 22, 2018
1 Introduction
- Definition
- Interest
- Difficulties

2 SUC Model
- Communication model and rules
- $\pi$ SUC-securely computes $\mathcal{F}$
- SUC composition theorem

3 Conclusion
Protocol
Context

Protocol

Proof of security
Context

Protocol

Proof of security

Adversary model
→ who?
→ capabilities?
→ goals?
Context

Protocol

Proof of security

Adversary model
→ who?
→ capabilities?
→ goals?

Security model

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

Simulation
→ Classical Simulation
→ Universal Composability
Context

Introduction

Definition

Protocol

Proof of security

Adversary model
→ who?
→ capabilities?
→ goals?

Security model

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

Chloé Hébant (ENS)

Working Group: SUC Security

February 22, 2018
**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 $P_i$ owning $x_i$, $n$-variable function $f$, Compute $f(x_1, \cdots, x_n) = (y_1, \cdots, y_n)$ s.t. each $P_i$ learns $y_i$ and nothing more
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 \( P_i \) owning \( x_i \), \( n \)-variable function \( f \), Compute \( f(x_1, \ldots, x_n) = (y_1, \ldots, y_n) \) s.t. each \( P_i \) 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 \( P_i \) owning \( x_i \), \( n \)-variable function \( f \), Compute \( f(x_1, \cdots, x_n) = (y_1, \cdots, y_n) \) s.t. each \( P_i \) 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 \( P_i \) owning \( x_i \), \( n \)-variable function \( f \), Compute \( f(x_1, \cdots, x_n) = (y_1, \cdots, y_n) \) s.t. each \( P_i \) 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

Construction of UC protocols:
- Define the ideal Functionality $\mathcal{F}$
- Construct a protocol $\Pi$ that realises $\mathcal{F}$
- Make the proof: construct a simulator $S$

Figure 1: Ideal World
Interest 1: $A$ 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: \( A \) 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: \( \mathcal{E} \) chooses the inputs of \( \mathcal{P}_i \) and \( A \)

\( \Rightarrow \) 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
Difficulties

Difficulty to define the ideal functionality

Ideal Functionality for Secure Message Transfer
Difficulties to define the ideal functionality

Ideal Functionality for Secure Message Transfer

$F_{STM}^l$ proceeds as follows:
parameterized by leakage function $l : \{0, 1\}^* \rightarrow \{0, 1\}^*$,

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

text = private content
Difficulties to define the ideal functionality

Ideal Functionality for Secure Message Transfer

$\mathcal{F}_{\text{STM}}^{l}$ proceeds as follows:

parameterized by leakage function $l : \{0, 1\}^* \rightarrow \{0, 1\}^*$,

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

$text = \text{private content}$

For example: leaking $l(m) = \text{length}(m)$ 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
1 Introduction
   - Definition
   - Interest
   - Difficulties

2 SUC Model
   - Communication model and rules
   - $\pi$ SUC-securely computes $\mathcal{F}$
   - SUC composition theorem

3 Conclusion
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
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
Definition

Let $\pi$ be a protocol for up to $m$ parties and let $F$ be an ideal functionality.

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

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

Let \( \pi \) be a protocol for the \( \mathcal{F} \)-hybrid model. Let \( \rho \) be a protocol that SUC-securely computes \( \mathcal{F} \) in the \( \mathcal{G} \)-hybrid model.

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

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

Corollary

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

By a drawing:
Corollary

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

By a drawing:

\[ \mathcal{H} \quad \mathcal{F} \quad \mathcal{F} \quad \pi \]
SUC composition theorem

Corollary

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

By a drawing:
Corollary

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

By a drawing:

\[ \mathcal{H} \quad \mathcal{F} \quad + \quad \mathcal{F} \quad \mathcal{G} \quad \Rightarrow \quad \mathcal{H} \quad \mathcal{G} \quad \mathcal{G} \]

\[ \pi \quad \rho \quad \pi^{\rho} \]
1 Introduction
   - Definition
   - Interest
   - Difficulties

2 SUC Model
   - Communication model and rules
   - $\pi$ SUC-securely computes $\mathcal{F}$
   - SUC composition theorem

3 Conclusion
In SUC, more rigid network model:

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

⇒ 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