

Universally Composable Security: A New Paradigm for Cryptographic Protocols



Canetti. Ran

IBM TJ Watson Research Center

Lab Seminar, July 28 - 2009

Presented by: Rifki Sadikin

Outline

Introduction

Basic Concepts Underlying UC

Defining security of protocols

Universal Composition

UC formulations of some Computational Models

Backgrounds and Motivation

- **Rigorously** demonstrating security of a protocol design.
- **Robustness** to the execution environment
 - Traditional approach : stand alone → complex environments.
- UC provides framework and method for composing protocol.
 - Stand-alone with *secure composition*

Universal Composable

- Tools for **modular design** and analysis of **complex protocols**.
 - Complex task is partitioned into simpler sub-tasks
 - Design protocols securely realized the sub-tasks
 - Use the composition theorem to argue protocol composed from designed-sub protocols securely realizes the given task

Interactive Turing Machine

Definition : An Interactive Turing Machine

An Interactive Turing machine (ITM) M is a Turing Machine with the following tapes:

- an **External Writable** identity tape, an EW security parameter tape, an EW input tape, an EW incoming communication tape, an EW subroutine output tape,
- an output tape, a random tape, a read and write one-bit activation tape, and a read and write working tape

Interactive Turing Machine

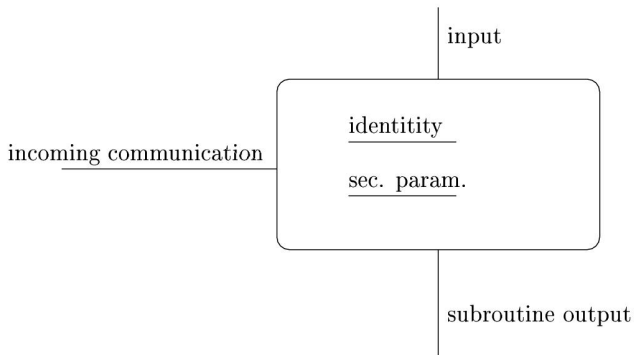
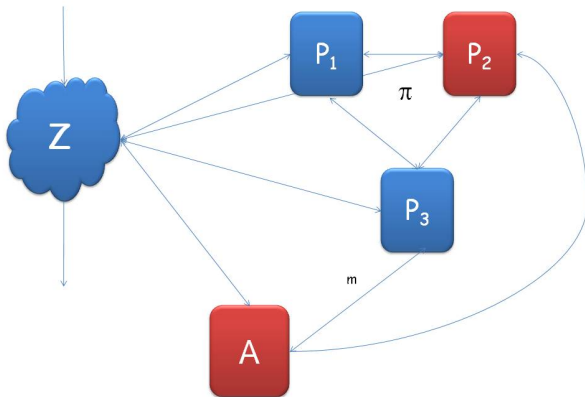


Figure: Interactive Turing Machine

The model of protocol execution

- Parameterized by three ITMs :
 - the protocol π to be executed
 - the environment \mathcal{Z}
 - the adversary \mathcal{A}
- The model for executing π is the extended system of PPT ITMs $(\mathcal{Z}, \mathcal{C}_{\text{EXEC}}^{\pi, \mathcal{A}})$

Execution of protocol π with environment \mathcal{Z} and adversary \mathcal{A}



Protocol Emulation

Definition (UC-emulates definition)

Let π and ϕ be PPT protocols. We say that π **UC-emulates** ϕ if for any PPT adversary \mathcal{A} there exists a PPT adversary \mathcal{S} such that for any PPT environment \mathcal{Z} we have:

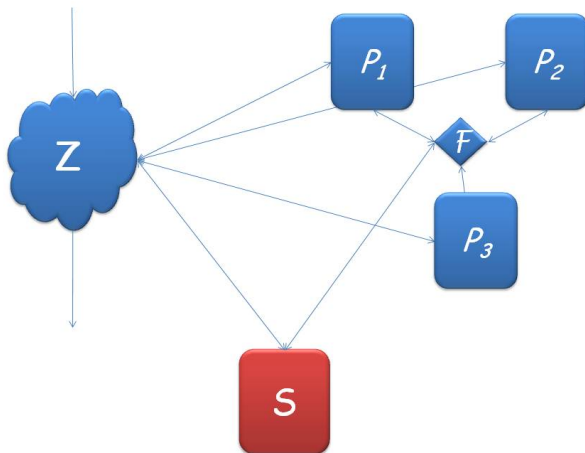
$$\text{EXEC}_{\pi, \mathcal{S}, \mathcal{Z}} \approx \text{EXEC}_{\phi, \mathcal{A}, \mathcal{Z}}$$

Ideal Functionalities

Ideal Functionalities

An ideal functionality represent **the expected functionality** of a certain task or a protocol problem. This includes both **“correctness”**, namely the expected input-output relation of uncorrupted parties and **“secrecy”** or the acceptable leakage of information to the adversary.

Ideal Process

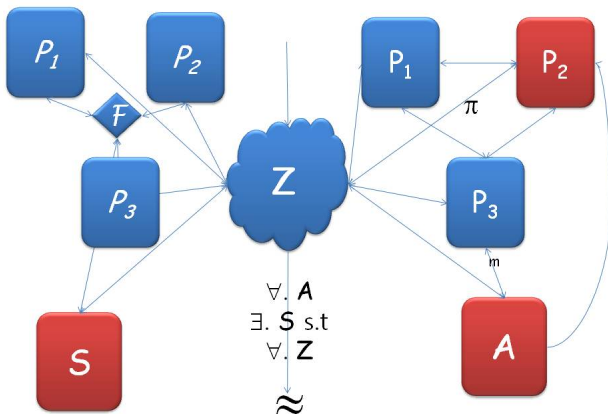


Realizing Ideal Functionality

Definition

Let \mathcal{F} be an ideal functionality and let π be an multi-party protocol. We say that π UC-realizes \mathcal{F} if π emulates the ideal protocol for \mathcal{F} .

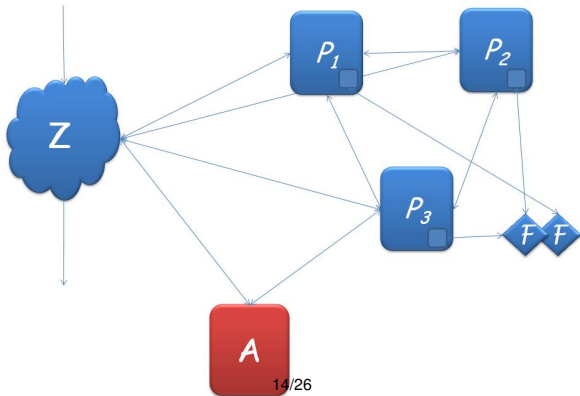
π UC-realizes \mathcal{F}



Hybrid Protocol

Definition

- \mathcal{F} -hybrid protocol π is a protocol that includes subroutine calls to $\text{IDEAL}_{\mathcal{F}}$, the ideal protocol for \mathcal{F}



Universal Composition Theorem

Definition (Universal Composition: General Statement)

Let π, ρ, ϕ be PPT multi-party protocols such that ρ UC-emulates ϕ ; ρ and ϕ are subroutine respecting. Then protocol $\pi^{\rho/\phi}$ UC-emulates protocol π .

Definition (Universal Composition: Realizing functionalities)

Let π be a subroutine respecting protocol that UC-realizes \mathcal{G} , and let ρ be a subroutine respecting protocol that securely realize \mathcal{F} . Then the composed protocol $\pi^{\rho/\mathcal{F}}$ securely realize \mathcal{G} .

Implications of the UC theorem

- Can design and analyze protocols in a **modular** way:
 - T is partitioned to T_1, \dots, T_k .
 - Construct protocols for realizing T_1, \dots, T_k .
 - Construct T assuming ideal access to T_1, \dots, T_k .
 - Use composition to obtain protocol T .
- Can deduce security of π in any **multi-execution protocol environment**:
 - Assume π UC-securely realize \mathcal{F} .

Authenticated Communication

Functionality $\mathcal{F}_{\text{AUTH}}$

1. Upon receiving an input (Send, sid, m) from party S , do: If $sid = (S, sid')$ for some R , then generate a public delayed output (Sent, sid, m) to R and halt. Else ignore the input.
2. Upon receiving an input ($\text{Corrupt} - \text{sender}, sid, m'$) from the adversary, and if (Sent, sid, m) output is not yet delivered to R , then output (Sent, sid, m') to R and halt.

On realizing $\mathcal{F}_{\text{AUTH}}$

- There exist no useful protocols that UC-realize $\mathcal{F}_{\text{AUTH}}$ in the bare model.
- If there exist signature schemes secure against chosen message attacks there exist \mathcal{F}_{REG} -hybrid protocol that UC-realizes $\mathcal{F}_{\text{AUTH}}$.

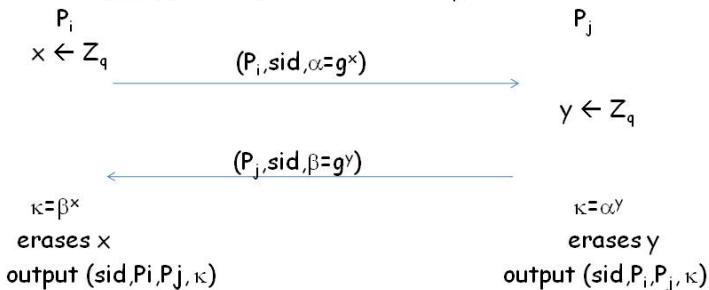
Key Exchange

Functionality \mathcal{F}_{KE}

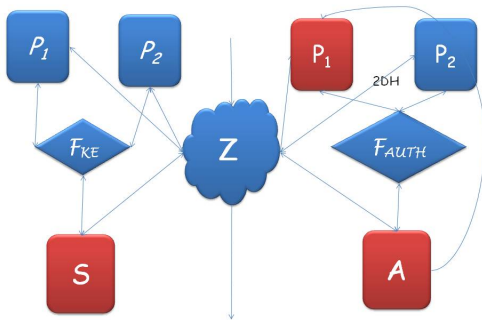
1. Upon receiving an input ($\text{Establish} - \text{Key}, \text{sid}$) from party I , verify that $\text{sid} = (I, R, \text{sid}')$ for some R , record I as active, record R as the responder, and send a public delayed output ($\text{Establish} - \text{Key}, \text{sid}$) to R .
2. Upon receiving ($\text{Establish} - \text{Key}, \text{sid}$) from party R , verify that R is recorded as the responder and record R as active.
3. Upon receiving a value $(\text{Key}, \text{sid}, m, \tilde{k})$ from the adversary, for party $P \in \{I, R\}$ do:
 - 3.1 If P is active and neither I, R are corrupted then do: If there is no recorder key $\kappa \xleftarrow{R} \{0, 1\}^k$ and record κ . Next output $(\text{Key}, \text{sid}, m, \kappa)$ to P
 - 3.2 Else, if P is active and either of I, R is corrupted then output $(\text{Key}, \text{sid}, m, \tilde{k})$ to P .
 - 3.3 Else (P is not active), do nothing.

Protocol 2DH

Crs : Primes $p, q, q/p-1$ and g of order q in Z_p^*



Protocol 2DH does not securely realize \mathcal{F}_{KE}



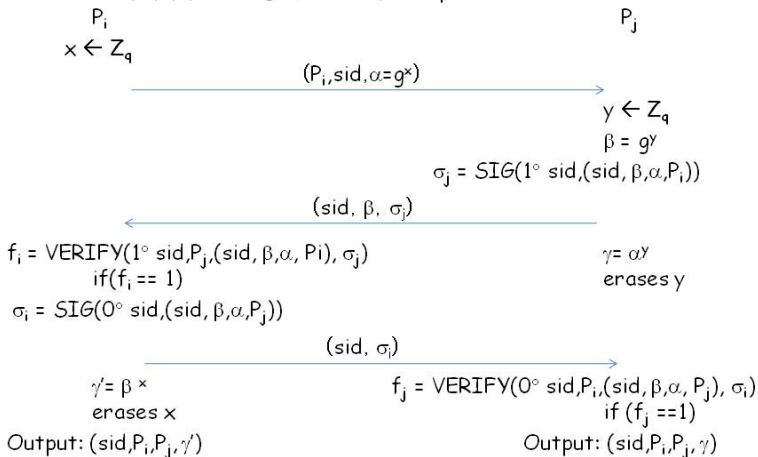
- \mathcal{Z} can distinguish $\text{IDEAL}_{\mathcal{F}_{KE}, S}$ from $\text{EXEC}_{\pi_{2DH}, A}$
 - In $\text{IDEAL}_{\mathcal{F}_{KE}, S}$, \mathcal{Z} has (x, g^y, κ) , while in $\text{EXEC}_{\pi_{2DH}, A}$ \mathcal{Z} has (x, g^y, g^{xy})

On realizing \mathcal{F}_{KE}

- UC-securely realize \mathcal{F}_{KE} strictly stronger than other known definition for Key Exchange protocols (SK-secure).
- Key exchange protocol that has ACK property and is SK-secure UC-securely realizes \mathcal{F}_{KE} .
- \mathcal{F}_{KE} can be relaxed so that securely realizing it becomes *equivalent* to the definition SK-secure.

Protocol SIG-DH

Crs: Primes $p, q, q/p-1$ and g of order q in Z_p^*



Digital Signature

Functionality \mathcal{F}_{SIG}

- **Key Generation:** Upon receiving a value $(\text{KeyGen}, \text{sid})$ from some party S , verify that $\text{sid} = (S, \text{sid}')$ for some sid' . If not then ignore the request. Else, hand $(\text{KeyGen}, \text{sid})$ to the adversary. Upon receiving $(\text{Algorithm}, \text{sid}, m, v)$ from the adversary, where s is a description of a PPT ITM and v is a description of a *deterministic* PT ITM, output $(\text{Verification}, \text{Algorithm}, \text{sid}, v)$ to S .
- **Signature Generation:** Upon receiving a value $(\text{Sign}, \text{sid}, m)$ from S , let $\sigma = s(m)$, and verify that $v(m, s) = 1$. If so. then output $(\text{Signature}, \text{sid}, m, \tau)$ to S and record the entry (m, σ) . Else output an error message to S and halt.
- **Signature Verification:** Upon receiving a value $(\text{Verify}, \text{sid}, m, \sigma, v')$ from some party V . do: If $v' = v$, the signer not corrupted, $v(m, \sigma) = 1$ and no entry (m, σ') for any σ' is recorded then output an error message to S and halt. Else, output $(\text{Verified}, \text{sid}, m, v'(m, \sigma))$ to V .

On realizing \mathcal{F}_{SIG}

Claim

Let $\Sigma = (\text{gen}, \text{sig}, \text{ver})$ be a signature scheme. Then π_{Σ} securely realize \mathcal{F}_{SIG} iff Σ is EU-CMA.

EU-CMA signature scheme

A signature scheme $\Sigma = (\text{gen}, \text{sig}, \text{ver})$ is called EU-CMA if:

- **Completeness:** $\text{Prob}[(\bar{s}, \bar{v}) \leftarrow \text{gen}(1^k); \sigma \leftarrow \text{sig}(\bar{s}, m); 0 \leftarrow \text{ver}(m, \sigma, \bar{v})] < \nu(k)$
- **Consistency (Non repudiation):** $\text{Prob}[(\bar{s}, \bar{v}) \leftarrow \text{gen}(1^k); \text{ver}(m, \sigma, \bar{v})$
generate different outputs in two different invocations] $< \nu(k)$
- **Unforgeability:** $\text{Prob}[(\bar{s}, \bar{v}) \leftarrow \text{gen}(1^k); (m, \sigma) \leftarrow F^{\text{sig}(\bar{s}, \cdot)}(\bar{v}); 1 \leftarrow \text{ver}(m, \sigma, \bar{v}); F$
never asked sig to sign m] $< \nu(k)$

Summary

- Overview of the universal composable framework.
- Reviewed a basic and general notion of security for cryptographic protocols.
- Final words from [2]:
 - Security analysis of protocols is only as good as the security notion used — and subtleties abound.
 - It is imperative to use a security notion that is appropriate for the relevant setting.

THANK YOU

For Further Reading



R. Canetti.

Universally Composable Security: A New Paradigm for Cryptographic Protocols.

FOCS '01: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science , :136, 2001.



R. Canetti.

Selected Topics in Cryptographic Protocols.

Available at <http://courses.csail.mit.edu/6.897/spring04/>.