Universal Composition of Cryptographic Protocols

Apoorva Deshpande

Joint work with: Raghav Bhaskar

27th June, 2013
Outline

- Trusted Party Paradigm
- Basic Security Framework
- Composition of Protocols
- Universal Composition
- Example: Commitment Protocol
- Further Work
Towards Secure Multi Party Computation

A secure protocol must satisfy:

- **SECRECY**
- **CORRECTNESS**

Trusted Party Paradigm
The Trusted Party Paradigm

\[ f(x_1, x_2, \ldots, x_n) \]

\[ f(x_n) \]

\[ x_1 \]

\[ x_2 \]

\[ x_n \]

\[ n \]

\[ 1 \]

\[ 2 \]

\[ n \]
Two Party Computation

\[ \pi \] is said to securely evaluate \( f \) if for any \( A \), there exists an \( S \) such that no environment can tell whether it is interacting with \( \pi \) and \( A \) or with Panda and \( S \).
Formalize the protocol execution experiment
Formalize the ideal functionality and process
Define what it means for the protocol to realize the functionality
BASIC SECURITY

Universally Composable Security: A New Paradigm for Cryptographic Protocols
R. Canetti. FOCS, 2001
**MODEL**: Interactive Turing Machine
System of ITMs

[Diagram showing a system of ITMs with 'INITIAL ITM' and 'Wait State' components and arrows indicating the flow of input, message, and output.]
**System of ITMs**

**Definition**
An ITM $M$ is PT if there exists a polynomial $p()$ such that at any point, overall steps taken by the ITM $M$ is $p(n)$

$n \rightarrow \text{(no. of bits on input tape of M)} - \text{(no. of bits written by M on other input tapes)}$
Formalize the protocol execution experiment

Formalize the ideal functionality and process

Define what it means for the protocol to realize the functionality
**Step 1: The Protocol Execution**

- **Party 1**: Input, Message, Output
- **ADVERSARY**: Input, Message, Output
- **ENVIRONMENT**: Input, Message, Output
- **Party 2**: Input, Message, Output
- **Party n**: Input, Message, Output
Note that..

• Each of the parties invoked by the environment must be running the protocol $\pi$.

• $E$ has no access to internal communication of the protocol.

• $A$ has no access to input/output of the parties invoked by $E$.

• Parties cannot directly send messages to each other, have to send only via $A$.

$A$ usually represents the network over which the parties communicate.
Formalize the protocol execution experiment
Formalize the ideal functionality and process
Define what it means for the protocol to realize the functionality
Step 2: The Ideal Process
Note that:

- $\mathcal{F}$ can repeatedly interact with $\mathcal{S}$ and the parties.
- The environment cannot directly communicate with $\mathcal{F}$.
- The simulator has no access to messages between the dummy parties and $\mathcal{F}$.

\[ I^f \] Ideal protocol for $\mathcal{F}$
**Abstract Concept of Trusted Party**

1. **STEP 1**
   - Formalize the protocol execution experiment

2. **STEP 2**
   - Formalize the ideal functionality and process

3. **STEP 3**
   - Define what it means for the protocol to realize the functionality

**Concrete Model**
Step 3:

\[ EXE_{\pi,A,E}(x) \] The 1 Bit output of Environment \( E \) on interaction with the protocol \( \pi \) and adversary \( A \) on input \( x \)

\[ \mathbb{E}XEC_{\pi,A,E} \] The ensemble of distributions on output of \( E \) on interaction with the protocol \( \pi \) and adversary \( A \), as \( x \) varies

**Definition** A protocol \( \pi \) emulates protocol \( \phi \) if for any PT adversary \( A \) there exists a PT adversary \( S \) such that for all PT environments \( E \) that output only one bit:

\[ EXE_{\phi,S,E} \approx EXE_{\pi,A,E} \]

**Definition** A protocol \( \pi \) realizes an ideal functionality \( F \) if \( \pi \) emulates \( I_f \)
COMPOSITION OF PROTOCOLS
How can protocols be composed?

**Timing Coordination**
- Sequential
- Non-concurrent
- Parallel
- Concurrent

**Input Coordination**
- Same Input
- Fixed Input
- Chosen Input

**Protocol Coordination**
- Self Composition
- General Composition

**State Coordination**
- Independent State
- Joint State

**Number of Executions**
- Fixed
- Bounded
- Unbounded
The Composition Operation

\[ \rho : \text{Invoking protocol} \]
\[ \pi, \phi : \text{Subroutine protocols} \]
Composability in Basic Security

**THEOREM:** If protocol $\pi$ emulates protocol $\phi$, then $\rho^{\pi/\phi}$ emulates $\rho$ for any protocol $\rho$ executing non-concurrently.

**Recall:** A protocol $\pi$ emulates protocol $\phi$ if for any PT adversary $A$ there exists a PT adversary $S$ such that for all PT environments $E$ that output only one bit:

$$EXEC_{\phi,S,E} \approx EXEC_{\pi,A,E}$$
UNIVERSAL COMPOSABILITY MODEL
Extensions to the Model
Step 1: The Protocol Execution
Note that..

- Environment $E$ can give input to the Adversary $A$ anytime. This input can also be in form of specific instructions or questions.

- Adversary $A$ can give output to the Environment $E$ anytime.

Modelling Corruption

- Environment instructs $A$ to corrupt some party $P$ of its choice. On getting corrupt:
  - $P$ forwards all its inputs to $A$
  - $P$ reveals its entire state to $A$ whenever activated
  - $A$ assumes write privileges on behalf of $P$
Step 2: The Ideal Process
Step 3:

\[ UC - EXEC_{\pi, A,E}(x) \] The 1 Bit output of Environment \( E \) on interaction with the protocol \( \pi \) and adversary \( A \) in UC-setting.

\[ UC - EXEC_{\pi, A,E} \] The ensemble of distributions on output of Environment \( E \) on interaction with the protocol \( \pi \) and adversary \( A \).

Definition

A protocol \( \pi \) UC-emulates protocol \( \phi \) if for any PT adversary \( A \) there exists a PT adversary \( S \) such that for all PT environments \( E \) that output only one bit:

\[ UC - EXEC_{\phi, S,E} \approx UC - EXEC_{\pi, A,E} \]

**THEOREM:** Let \( \pi \) and \( \phi \) be PT protocols such that \( \pi \) UC-emulates \( \phi \) then \( \rho_{\pi/\phi} \) emulates \( \rho \) for any PT protocol \( \rho \).
Advantages of the UC Model

- Enables modular protocol analysis

UC Security of protocol $\Rightarrow$ Complete Security of protocol

- Enables modular protocol design

$p_{\{p_1/t_1, \ldots, p_n/t_n\}}$
EXAMPLE: COMMITMENT PROTOCOL

Universally Composable Commitments.
The One Bit Commitment

Phase 1: Commitment

Phase 2: Opening
Ideal Functionality for Commitment

\[ F_{\text{COM}} \]

- Record b
  - <Commit, SID, Bob, Alice, b>
  - Check for record of b
  - <Open, SID, Bob, Alice>
  - <Open, SID, Bob, Alice, b>

- <Commit, SID, Bob, Alice>
Impossibility Result for the Plain Model

**THEOREM:** There exists no bilateral, terminating protocol $\Pi$ that securely realizes functionality $F_{\text{COM}}$ in the plain model.

---

**CRS Model**
Protocol UCC: Setup

Pseudorandom generator with trapdoor property

\[ G_{pk_0}, G_{pk_1} : \{0, 1\}^n \rightarrow \{0, 1\}^{4n} \]

Given \( td \) for \( pk \), it is possible to find if \( y \in \{0, 1\}^{4n} \) is in range of \( G_{pk} \)

Public String

\[ \sigma \in \{0, 1\}^{4n} \]

\( pk_0, pk_1 \) : keys for generators \( G_{pk_0}, G_{pk_1} \)
THEOREM: Protocol UCC securely realizes functionality $F_{\text{COM}}$ in the CRS model.
Construction of the Simulator

Choose $r \in \{0, 1\}^n$

$G_{pk_0}, G_{pk_1}$

$pk_0, pk_1$

$td_0, td_1$

Create Fake $\sigma$

$\sigma = G_{pk_0}(r) \oplus G_{pk_1}(r)$

ENVIRONMENT

$F_{com}$

ADVERSARY

Black Box Access

SIMULATOR
Construction of the Simulator: Honest Participants

\[ \sigma = G_{pk_0}(r) \oplus G_{pk_1}(r') \]

\[ y = G_{pk_0}(r) \]

\[ \langle \text{Commit, SID, } y, b \rangle \]

\[ \langle \text{Open, SID, Bob, Alice} \rangle \]

\[ F_{\text{COM}} \]

ENVIRONMENT

\[ \langle \text{Commit, SID, Bob, Alice} \rangle \]

\[ \langle \text{Open, SID, Bob, Alice, b} \rangle \]
Construction of the Simulator: Corrupt Sender

$\langle \text{Commit, SID, } b = 0 \rangle$

$\langle \text{Commit, SID, Bob, Alice, 0} \rangle$

$\langle \text{Commit, SID, Bob, Alice, 0} \rangle$

$\langle \text{Bob Corrupted} \rangle$

$F_{\text{COM}}$

$t_{\text{do}}$

$\langle \text{Commit, SID, } b = 0 \rangle$

$\langle \text{Commit, SID, Bob, Alice, 0} \rangle$

$\langle \text{Commit, SID, Bob, Alice, 0} \rangle$

$\langle \text{Bob Corrupted} \rangle$

$F_{\text{COM}}$
Proof Idea

Define three random variables:

**Real/Genuine**  
The output of the environment in a real life execution with Alice and Bob running $\pi$. $\sigma$ is uniformly distributed and $pk_0, pk_1$ and $r$ are truly random.

**Real/Fake**  
The output of the environment on interaction with chosen $pk_0, pk_1$ and $\sigma = G_{pk_0}(r) \oplus G_{pk_1}(r)$

**Ideal/Fake**  
The output of the environment in an execution in the ideal process with the simulator $F_{COM}$.
FURTHER WORK: PASS SCHEMES

Practical Yet Universally Composable Two-Sever Password-Authenticated Secret Sharing
J. Camenisch and A. Lysyanskaya and G. Neven. CCS, 2012
PASS: Password Authenticated Secret Sharing

**TASK:** Outsourcing the storage of key $K$

**Setup Phase**
- Splits Key $K$ into $K_1$ and $K_2$
- Remembers password $p$

**Retrieve Phase**
- Uses password $p$
- Retrieves back $K$

ZK Protocol to check $p_1 * p_2 = p_1' * p_2'$
PASS: Password Authenticated Secret Sharing

**Phase 1**

\[ F_{2\text{PASS}} \text{ - Setup} \]

Downward arrow

Setup Protocol

**Phase 2**

\[ F_{2\text{PASS}} \text{ - Retrieve} \]

Downward arrow

Retrieve Protocol
Any Questions?

Thank You