Universal Composition

Hong-Sheng Zhou

CryptoDRM Lab
Computer Science and Engineering
University of Connecticut
Outline

• Review
• Key-Exchange
• Signature
Traditional Simulation-based Paradigm

- **REAL**
  - **adversary A**
  - $\rho$-protocol interaction

- **IDEAL**
  - **adversary S**
  - Trusted party implementing a functionality $F$
Alternative presentation

$\rho$-protocol interaction

REAL

IDEAL
UC Security

ρ-protocol interaction
UC Security

We say a protocol $\rho$ UC realizes functionality $F$ if for every real-world adversary $A$, there exists an ideal-world adversary $S$, such that no environment $Z$ can distinguish between a real execution with $A$ and an ideal execution with $S$. 
Hybrid vs. Ideal

Protocol $\rho$ in the $G$ hybrid world
UC realizes functionality $F$
Protocol $\rho$ in the $G$ hybrid world, UC realizes functionality $F$.

Protocol $\pi^p$ in the $G$ hybrid world, UC realizes functionality $H$.

Protocol $\pi$ in the $F$ hybrid world, UC realizes functionality $H$.
Composition Theorem

Let $\mathcal{G}$, $\mathcal{F}$ be ideal functionalities. Let $\pi$ be a multi-party protocol in the $\mathcal{F}$-hybrid world, and let $\rho$ be a multi-party protocol that UC realizes $\mathcal{F}$ in the $\mathcal{G}$-hybrid world. Then for any adversary $A$ in the $\mathcal{G}$-hybrid world there exists an adversary $S$ in the $\mathcal{F}$-hybrid world such that for any environment machine $Z$ we have: $\text{EXEC}^G_{\pi, A, Z} \approx \text{EXEC}^F_{\pi, S, Z}$
More on Corruption

Recall: UC Security

REAL

IDEAL

ρ-protocol interaction
More on Corruption

• Adaptive vs. Static
  – Adaptive: Parties could be corrupted throughout the computation;
  – Static: Parties are corrupted at the beginning;

• Active vs. Passive
  – Active: Corrupted parties operate arbitrarily; malicious;
  – Passive: Corrupted parties still need to follow the protocol specification; honest-but-curious;

• Non-erasure vs. Erasure
  – Non-erasure: once corrupted, the adversary is allowed to access to all pervious internals;
  – Erasure: once corrupted, the adversary is only allowed to access to the current internals; some of the internals could be erased by the parties when they are honest based on the protocol specification;
More on Corruption

- Adaptive vs. Static
- Active vs. Passive
- Non-erasure vs. Erasure

- Realistic Corruption = Adaptive + Active + Non-erasure
- In the UC Key-Exchange which will be investigated below, we consider a slight weaker corruption, i.e. Adaptive+Active+Erasure
Outline

• Review
• Key-Exchange
• Signature
Outline

• Review
• Key-Exchange
• Signature
The authenticated message transmission functionality, $F_{\text{AUTH}}$

<table>
<thead>
<tr>
<th>Functionality $F_{\text{AUTH}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Upon receiving an input $(\text{Send}, \text{sid}, m)$ from party $S$, do: If $\text{sid} = (S, R, \text{sid}')$ for some $R$, then generate a public delayed output $(\text{Sent}, \text{sid}, m)$ to $R$ and halt. Else ignore the input.</td>
</tr>
<tr>
<td>2. Upon receiving $(\text{Corrupt-sender}, \text{sid}, m')$ from the adversary, and if the $(\text{Sent}, \text{sid}, m)$ output is not yet delivered to $R$, then output $(\text{Sent}, \text{sid}, m')$ to $R$ and halt.</td>
</tr>
</tbody>
</table>
The Key-Exchange functionality, $F_{KE}$

1. Upon receiving an input $(\text{Establish-Key}, \text{sid})$ from party I, verify that $\text{sid} = (I, R, \text{sid'})$ for some identity $R$, record I as active, record $R$ as the responder, and send a public delayed output $(\text{Establish-Key}, \text{sid})$ to $R$.

2. Upon receiving $(\text{Establish-Key}, \text{sid})$ from party $R$, verify that $R$ is recorded as the responder, and record $R$ as active, then notify the adversary.
The Key-Exchange functionality, $F_{KE}$

Functionality $F_{KE}$

3. Upon receiving a message $(\text{Key}, \text{sid}, P, k')$ from the adversary, for $P$ in $\{I, R\}$ do:
   a) If $P$ is active and neither $I$, $R$ are corrupted, then do: if there is no recorded key $k$ then randomly choose $k$ and record $k$. Next output $(\text{Key}, \text{sid}, k)$ to $P$.
   b) Else, if $P$ is active and either of $I$, $R$ is corrupted then output $(\text{Key}, \text{sid}, k')$ to $P$.
   c) Else, $P$ is not active, do nothing.
Key-Exchange Protocols

Protocol 2DH

Assuming authentication channel

A
Choose \( x \),
Compute \( g^x \)
Compute \( g^{xy} \),
Erase \( x \),
Output \( g^{xy} \)

B
Choose \( y \),
Compute \( g^y \) and \( g^{xy} \),
Erase \( y \),
Output \( g^{xy} \)
**Key-Exchange Protocols**

**Protocol m2DH**

*Assuming authentication channel*

A
- Choose x,
- Compute $g^x$
- Compute $g^{xy}$,
- Erase x,
- Output $g^{xy}$

B
- Choose y,
- Compute $g^y$ and $g^{xy}$,
- Erase y
- Output $g^{xy}$,
Prove and Refute

• Prove: m2DH realizes $F_{KE}$ in the $F_{AUTH}$ hybrid world assuming erasure

• Refute: 2DH does not realize $F_{KE}$ in the $F_{AUTH}$ hybrid world assuming erasure
Prove

• Theorem: Key-exchange protocol m2DH in the $F_{\text{AUTH}}$ hybrid world UC-realizes functionality $F_{\text{KE}}$ under the DDH assumption and assuming secure erasure.

• Proof idea: construct a simulator for all environment the two worlds cannot be distinguished.
Prove

m2DH in the $F_{AUTH}$ hybrid world
Refute

• Theorem: Key-exchange protocol 2DH in the $F_{AUTH}$ hybrid world does not UC-realize functionality $F_{KE}$ under the DDH assumption and assuming secure erasure.

• Proof idea: construct an environment to distinguish the two worlds.
Refute 2DH in the $F_{AUTH}$ hybrid world
We may simulate the hybrid world as before; But, ...... If corrupt Alice when Bob just returns his output and Alice does not receive Bob’s second move, What will happen?

Hybrid =
\[
\begin{align*}
&\text{Alice’s internal } x \\
&\text{Bob’s transcript } g^y \\
&\text{Bob’s output } k = g^{xy}
\end{align*}
\]

Ideal =
\[
\begin{align*}
&\text{Alice’s internal } x \\
&\text{Bob’s transcript } g^y \\
&\text{Bob’s output random } k
\end{align*}
\]
Outline

• Review
• Key-Exchange
• Signature
Outline

- Review
- Key-Exchange
- Signature
Hierarchy
Functionality $\mathcal{F}_{\text{SIG}}$

**Key generation:** Upon receiving $(\text{KeyGen}, sid)$ from party $S$, verify that $sid = (S, sid')$ for some $sid'$. If not, then ignore the input. Else, forward $(\text{KeyGen}, sid)$ to the adversary $S$.

Upon receiving $(\text{Algorithms}, sid, s, v)$ from the adversary $S$, record $(s, v)$ and output $(\text{VerificationAlg}, sid, v)$ to party $S$, where $s$ is a signing algorithm, and $v$ is a verification algorithm.

**Signature generation:** Upon receiving $(\text{Sign}, sid, m)$ from party $S$ where $sid = (S, sid')$, let $\sigma = s(m)$, verify that $v(m, \sigma) = 1$. If so, then output $(\text{Signature}, sid, \sigma)$ to party $S$, and record $(m, \sigma)$. Else, halt.

**Signature verification:** Upon receiving $(\text{Verify}, sid, m, \sigma, v')$ from party $V$, where $sid = (S, sid')$, do: if $v' = v$, the signer $S$ is not corrupted, $v(m, \sigma) = 1$, and $m$ is not recorded, then halt. Else, output $(\text{Verified}, sid, v'(m, \sigma))$ to party $V$. 

Realization

Definition (EU-CMA Signature Schemes) A signature scheme $\Sigma = (\text{gen}, \text{sign}, \text{verify})$ is called EU-CMA if the following properties hold for any negligible function $\text{negl}(\cdot)$, and all large enough values of the security parameter $\lambda$,

Completeness: For any message $m \in \mathcal{M}$,

$$\Pr[(vk, sk) \leftarrow \text{gen}(1^\lambda); \sigma \leftarrow \text{sign}(vk, sk, m); 0 \leftarrow \text{verify}(vk, m, \sigma)] \leq \text{negl}(\lambda).$$

Consistency: For any $m \in \mathcal{M}$, the probability that $\text{gen}(1^\lambda)$ generates $\langle vk, sk \rangle$ and $\text{verify}(vk, m, \sigma)$ generates two different outputs in two independent invocations is smaller than $\text{negl}(\lambda)$.

Unforgeability: For any PPT forger $F$,

$$\Pr[(vk, sk) \leftarrow \text{gen}(1^\lambda); (m, \sigma) \leftarrow F^{\text{sign}(vk, sk, \cdot)}(vk);$$

$$1 \leftarrow \text{verify}(vk, m, \sigma) \text{ and } F \text{ never asked } \text{sign}(vk, sk, \cdot, \cdot) \text{ to sign } m \leq \text{negl}(\lambda).$$
By contradiction, assume $\Sigma$ is not EU-CMA, i.e. there exist either a successful completeness attacker, or a consistency attacker, or a forger, construct $Z$ based on them respectively to distinguish the two worlds.

By contradiction, assume $\pi_\Sigma$ does not realize $F_{\text{SIG}}$, i.e. we have a successful $Z$ in hand, and we need to show $\Sigma$ is not EU-CMA, i.e. we need to show $\Sigma$ is either not complete, or not consistent, or not unforgeable; assume $\Sigma$ is complete and consistent, we need to construct a successful forger based on the $Z$. 

**Theorem:** $\Sigma$ is EU-CMA $\iff \pi_\Sigma$ securely realizes $F_{\text{SIG}}$. 
Summary

• Framework vs. Primitives
• Prove vs. Disprove Techniques
• Concrete vs. General Constructions
References

• Google "Ran Canetti" & "DBLP" "slides" etc..

• Papers

• Slides
Thanks.
Universal Composition

Hong-Sheng Zhou

Computer Science and Engineering
University of Connecticut