Simulation-based Security and Joint State Theorems in the IITM Model

Ralf Küsters
University of Trier

Based on: Datta, K., Mitchell, Ramanathan, TCC 2005
K., CSFW 2006
K., Tuengerthal, CSF 2008
K., Tuengerthal, CSF 2009
K., Tuengerthal, new result
Overview

- Subtleties in Simulation-based Models
- The IITM Model
- Motivating Joint State
- General Joint State Theorem
- Applications: new functionalities with joint state
  * Public-key encryption and digital signatures
  * Symmetric encryption
    - Several applications, including a new computational soundness result
- Conclusion
Models for Simulation-Based Security

- **UC model**
  [Canetti 2001]

- **Reactive Simulatability**
  [Pfitzmann, Waidner 2001; Backes, Pfitzmann, Waidner, 2004]

- **(Sequential) Probabilistic Process Calculus**
  [Lincoln, Mitchell, Mitchell, Scedrov, 1998; Datta, K., Mitchell, Ramanathan, 2005]

- **Task PIOA**
  [Canetti, Cheung, Kaynar, Liskov, Lynch, Peireira, Segala, 2006]

- **IITM model**
  [K., 2006]

- [Hofheinz, Unruh, Müller-Quade, 2009]
Universal Composability

\[ \mathcal{P} \text{ and } \mathcal{F} \text{ are UC if } \forall A \ \exists I \ \forall \varepsilon : \]

\[ \begin{array}{c}
\mathcal{E} \\
\mathcal{A} \\
\mathcal{P}
\end{array} = \begin{array}{c}
\mathcal{E} \\
\mathcal{I} \\
\mathcal{F}
\end{array} \]
Universal Composability

\( \mathcal{P} \) and \( \mathcal{F} \) are UC if \( \forall A \ \exists I \ \forall \varepsilon: \)

\[
\begin{align*}
\mathcal{E} & \quad \mathcal{A} \quad \mathcal{P} \\
\text{I/O interface} & \quad \text{network interface} \\
\mathcal{I} \quad \mathcal{F} & \quad \mathcal{E} \\
\text{I/O interface} & \quad \text{network interface}
\end{align*}
\]
(Strong) Black-box Simulatability

\[ \mathcal{P} \text{ and } \mathcal{F} \text{ are SBB if } \exists S \forall A \forall \mathcal{E}: \]

\[ \equiv \]

\[ \mathcal{E} \]

\[ \mathcal{A} \]

\[ \mathcal{P} \]

\[ \mathcal{E} \]

\[ \mathcal{A} \]

\[ \mathcal{S} \]

\[ \mathcal{F} \]

\[ \equiv I \]
Strong Simulatability

$\mathcal{P}$ and $\mathcal{F}$ are SS if $\exists S \forall \mathcal{E}$:

$\equiv \mathcal{E} \parallel \mathcal{A}$ in UC and BB
Subtelties in Simulation-Based Models

So, several models, security notions, and assumptions in different papers ...

... confusing at first sight

Let’s look at two issues more closely

1. Master Process

2. Runtime of ITMs
Master process: Is triggered if no other process can go.

Who should play the role of the master process?

The literature provides different answers yielding different variants of security notions.
Relationships Between Security Notions

SS ≡ SBB

[PW 2001, BPW 2004]

Master: no restrictions

All of the following notions are equivalent:

<table>
<thead>
<tr>
<th>SBB</th>
<th>( \mathcal{A} )</th>
<th>S</th>
<th>( \mathcal{E} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pfitzmann, Waidner 2001</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Backes, Pfitzmann, Waidner 2004</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SS</th>
<th>S</th>
<th>( \mathcal{E} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Relationships Between Security Notions

\[ \text{SS} \equiv \text{SBB} \]

[PW 2001, ...

\textbf{Master: no notion} ...

\textbf{All of the following notions are equivalent:}

<table>
<thead>
<tr>
<th></th>
<th>( \mathcal{A} )</th>
<th>( \mathcal{I} )</th>
<th>( \varepsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC</td>
<td></td>
<td>( \times )</td>
<td></td>
</tr>
<tr>
<td>Canetti 2001</td>
<td></td>
<td></td>
<td>( \times )</td>
</tr>
<tr>
<td>Backes, Pfitzmann, Waidner 2004</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>( \mathcal{A} )</th>
<th>( \mathcal{S} )</th>
<th>( \varepsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>WBB</td>
<td>( \times )</td>
<td>( \times )</td>
<td>( \times )</td>
</tr>
<tr>
<td></td>
<td>( \times )</td>
<td>( \times )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \times )</td>
<td>( \times )</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( \times )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( \times )</td>
</tr>
</tbody>
</table>

\[ \text{UC} \equiv \text{WBB} \]

[C 2001, BPW 2004]

\textbf{Master: environment + other entities}
Relationships Between Security Notions

\[ SS \equiv SBB \]

[PW 2001, BPW 2004]

**Master:** no restrictions

\[ UC \equiv WBB \]

[C 2001, BPW 2004]

**Master:** environment + other entities

\[ \begin{array}{ccc}
WBB & A & S & E \\
\hline
X & & & \\
\end{array} \]

**Master:** only adversary
Relationships Between Security Notions

SS $\equiv$ SBB

[PW 2001, BPW 2004]

Master: no restrictions

UC $\equiv$ WBB

[PW 2001]

Master: not environment

All of the following notions are equivalent:

<table>
<thead>
<tr>
<th>UC</th>
<th>$\mathcal{A}$</th>
<th>$\mathcal{I}$</th>
<th>$\mathcal{E}$</th>
<th>WBB</th>
<th>$\mathcal{A}$</th>
<th>$\mathcal{S}$</th>
<th>$\mathcal{E}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pfitzmann, Waidner 2001</td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
<td></td>
<td>$\times$</td>
<td>$\times$</td>
<td></td>
</tr>
</tbody>
</table>

[C 2001, BPW 2004]

Master: environment + other entities

Master: only adversary
Relationships Between Security Notions

SS $\equiv$ SBB

[PW 2001, BPW 2004]
Master: no restrictions

iff FORWARDER

UC $\equiv$ WBB

[PW 2001]
Master: not environment

UC $\equiv$ WBB

[C 2001, BPW 2004]
Master: environment + other entities

WBB

Master: only adversary
**Relationships in BPW Model**

- \( SS \equiv SBB \)
  - **Master:** no restrictions

- **Disproves a theorem claimed in [BPW 2004].**

- \( UC \not\equiv WBB \)
  - **Master:** environment + other entities

- **Due to the buffering mechanism.**

- **WBB**
  - **Master:** only adversary

**FORWARDER**
Subtelties in Simulation-Based Models

1. Master Process

2. Runtime of ITMs
Runtime of ITMs

- [Canetti 2001]
  - Interactive Turing machines (ITMs).
  - (dummy) Universal Composability (UC).

  **Total runtime of components (ITMs, PIOAs, processes) is polynomially bounded in security parameter alone and independent of external input.**

- Datta, Raskin, Mitchell, Ramanathan, 2003
  - Probabilistic Polynomial Time Process Calculi.
  - All three above notions and strong simulatability (SS).

exhaustible ITMs
Drawbacks of Models with Exhaustible ITMs

Example

Ideal protocol behaves the same as the real protocol except that before sending a message on the network the bit-wise complement is taken.

Surprisingly: Real protocol in general does not black-box realize the ideal protocol.

Problem is closely related to the FORWARDER property:

\[
P \parallel Q \equiv P \parallel D \parallel Q.
\]

[Datta, Küsters, Mitchell, Ramanathan 2005]

Unintuitive and unexpected behavior:

- Almost identical protocols are not black-box simulatable.
- Parallel composition of two or more protocols/process/machines cannot be simulated by one ITM (needed for the Joint State Theorem [Canetti, Rabin 2003]).
Drawbacks of Models with Exhaustible ITMs

If machines could be forced to stop (e.g., UC model):

- Almost identical protocols are not black-box simulatable.
- Parallel composition of two or more protocols/process/machines cannot be simulated by one ITM (needed for the Joint State Theorem [Canetti, Rabin 2003]).
Overview

- Subtleties in Simulation-based Models
- The IITM Model
- Motivating Joint State
- General Joint State Theorem
- Applications: new functionalities with joint state
  * Public-key encryption and digital signatures
  * Symmetric encryption
    - Several applications, including a new computational soundness result
- Conclusion
The IITM Model

- General computational model
  - Inexhaustible Interactive Turing Machines (IITMs)
  - System of IITMs

- Simulation-based security
  - Security notions
  - Composition theorems
Combining ITMs is not so easy ...

Non-terminating system

Imposing a global polynomial bound does not work.
The IITM Model

- General computational model
  - Inexhaustible Interactive Turing Machines (IITMs)
  - System of IITMs

- Simulation-based security
  - Security notions
  - Composition theorems
Inexhaustible Interactive Turing Machine (IITM)

- Generic addressing mechanism
  (no specific addressing mechanism, e.g., based on SIDs/PIDs, is fixed)

- Runtime may depend on length of input

- Can be activated an unbounded number of times

- Can perform ppt computation in every activation

  \[ \Rightarrow \text{no exhaustion} \]

If machines could be forced to stop (e.g., UC model):

- two machines

\[ \neq \]

- single machine

 yields more useful functionalities and more natural properties
Inexhaustible ITMs (IITMs)

Tapes have names.

- Per activation: polynomially bounded computation in
  * length of current input
  * length of current configuration
  * security parameter

  ⇒ ITM can read every message and can scan entire configuration in every activation.

  ⇒ No exhaustion.

- Length of output and configuration is polynomially bounded in security parameter plus length of input received on enriching tapes so far.
The IITM Model

- **General computational model**
  - Inexhaustible Interactive Turing Machines (IITMs)
  - System of IITMs

- **Simulation-based security**
  - Security notions
  - Composition theorems
Systems of IITMs

\[ S = M_1 \parallel \cdots \parallel M_n \parallel !M'_1 \parallel \cdots \parallel !M'_m \]

unbounded number of copies generated dynamically

This is not process calculus

single IITM

well-formed systems:
Graph of IITMs induced by enriching tapes is acyclic

Names of tapes determine how IITMs are connected.
Properties of Systems of IITMs

**Lemma:** Well-formed systems run in ppt.

**Lemma:** There exists a FORWARDER IITM $\mathcal{D}$:

$$\mathcal{P} \parallel \mathcal{Q} \equiv \mathcal{P} \parallel \mathcal{D} \parallel \mathcal{Q}$$

$\mathcal{D}$: independent of $\mathcal{P}$ and $\mathcal{Q}$, all tapes are enriching.

**Lemma:** Given systems $\mathcal{Q}_1$ and $\mathcal{Q}_2$ with $\mathcal{Q}_1 \parallel \mathcal{Q}_2$ well formed, then there exists an IITM $\mathcal{M}$ s.t.

$$\mathcal{Q}_1 \parallel \mathcal{Q}_2 \equiv \mathcal{Q}_1 \parallel \mathcal{M}$$

(Needed, e.g., in joint state theorem)
Copies of IITMs and the Generic Addressing Mechanism

IITMs run in one of two modes:

Check address  \{ Compute  \}  \quad \text{generic addressing mechanism}

Example:  \[ S = M_1 \parallel !M_2 \]
Copies of IITMs and the Generic Addressing Mechanism

ID version of $M$:  

\[ \langle id, x \rangle \xrightarrow{x} M \xrightarrow{y} \langle id, y \rangle = M \]

ID can be SID (session version) or PID (party version) of $M$
The IITM Model

- General computational model
  - Inexhaustible Interactive Turing Machines (IITMs)
  - System of IITMs

- Simulation-based security
  - Security notions
  - Composition theorems
Security Notions

Definition (Strong Simulatability)

\[ \mathcal{P} \leq \mathcal{F} \text{ iff } \exists \text{ Sim. } \mathcal{S} \ \forall \text{ Env. } \mathcal{E}: \ \mathcal{E} \parallel \mathcal{P} \equiv \mathcal{E} \parallel \mathcal{S} \parallel \mathcal{F} \]

- Similarly for UC and black-box simulatability
- No unnecessary details need to be fixed in the IITM model

\{ * Corruption \* Addressing \} Part of the description of protocols (⇒ flexible and expressive)
Composition Theorem in IITM Model

Composition Theorem:

\[ \mathcal{P}_1 \leq \mathcal{F}_1 \]
\[ \mathcal{P}_2 \leq \mathcal{F}_2 \]
\[ \mathcal{P}_1 \parallel \mathcal{P}_2 \]

\[ \mathcal{P}_1 \mathcal{P}_2 \leq \mathcal{F}_1 \mathcal{F}_2 \]

\[ \varepsilon \parallel \mathcal{P}_2 \parallel \mathcal{P}_1 \equiv [\varepsilon \parallel \mathcal{P}_2]_{\mathcal{P}_1} \parallel \mathcal{P}_1 \]
\[ \equiv [\varepsilon \parallel \mathcal{P}_2]_{\mathcal{P}_1} \parallel \mathcal{S}_1 \parallel \mathcal{F}_1 \]
\[ \equiv \varepsilon \parallel \mathcal{P}_2 \parallel \mathcal{S}_1 \parallel \mathcal{F}_1 \]
\[ \equiv \varepsilon \parallel \mathcal{S}_1 \parallel \mathcal{F}_1 \parallel \mathcal{P}_2 \]
\[ \equiv [\varepsilon \parallel \mathcal{S}_1 \parallel \mathcal{F}_1]_{\mathcal{P}_2} \parallel \mathcal{P}_2 \]
\[ \equiv [\varepsilon \parallel \mathcal{S}_1 \parallel \mathcal{F}_1]_{\mathcal{P}_2} \parallel \mathcal{S}_2 \parallel \mathcal{F}_2 \]
\[ \equiv \varepsilon \parallel \mathcal{S}_1 \parallel \mathcal{F}_1 \parallel \mathcal{S}_2 \parallel \mathcal{F}_2 \]
\[ \equiv \varepsilon \parallel \mathcal{S}_1 \parallel \mathcal{S}_2 \parallel \mathcal{F}_1 \parallel \mathcal{F}_2 \]
Composition Theorem in IITM Model

Composition Theorem:

\[ P_1 \leq F_1 \]

\[ P_2 \leq F_2 \]

\[ P_1 P_2 \leq F_1 F_2 \]

Corollary:

\[ P \leq F \]

\[ Q \leq F \]

\[ Q \leq F \]
Overview

- Subtleties in Simulation-based Models
- The IITM Model
- Motivating Joint State
- General Joint State Theorem

- Applications: new functionalities with joint state
  * Public-key encryption and digital signatures
  * Symmetric encryption
    - Several applications, including a new computational soundness result

- Conclusion
Models for Simulation-Based Security

- [Pfitzmann, Waidner 2001]
  [Backes, Pfitzmann, Waidner 2004]
  [Hofheinz, Müller-Quade, Unruh, 2005/2009]
- [Mitchell, Ramanathan, Scedrov, Teague, 2001]
  [Datta, Küsters, Mitchell, Ramanathan, 2005]
- [Canetti, Cheung, Kaynar, Liskov, Lynch, Pereira, Segala, 2006]

Comparison:
[Datta, Küsters, Mitchell, Ramanathan 2005/2008]

- UC model [Canetti 2001 – ]
- IITM model [Küsters, 2006]
Motivating Joint State

\[ \mathcal{P} \xrightarrow{\text{composition theorem}} \mathcal{F}_{PKE} \leq \mathcal{F}_{KE} \]

\[ \mathcal{P}_{PKE} \leq \mathcal{F}_{PKE} \]

Every copy uses different keys

Impractical!
Motivating Joint State

We rather want:

Joint State
Overview

- Subtleties in Simulation-based Models
- The IITM Model
- Motivating Joint State
- General Joint State Theorem
- Applications: new functionalities with joint state
  * Public-key encryption and digital signatures
  * Symmetric encryption
    - Several applications, including a new computational soundness result
- Conclusion
General Joint State Theorem – UC Model

Joint State Theorem in UC model: [Canetti and Rabin 2003]

\[ P \leq \widehat{F} \approx \underbrace{\overbrace{F F F F}}_{\ldots} \quad \rightarrow \quad Q[\widehat{P}] \leq \underbrace{Q F F F}_{\ldots} \]

single ITM → exhaustion problem

JUC operator

Conceptually a good idea, but technically the theorem is flawed
General Joint State Theorem – IITM Model

Joint State Theorem in IITM model: [K. and Tuengerthal 2008]

\[
\widehat{P} \leq !F
\]

\[
Q \parallel \widehat{P} \leq Q \parallel !F
\]

immediate consequence of composition theorem

no new operator needed and no shielding
Iterative Application of the Joint State Theorem
Overview

- Subtleties in Simulation-based Models
- The IITM Model
- Motivating Joint State
- General Joint State Theorem

- Applications: new functionalities with joint state
  * Public-key encryption and digital signatures
  * Symmetric encryption
    - Several applications, including a new computational soundness result

- Conclusion
Ideal Functionality for PKE: $F_{\text{PKE}}(l)$

$F_{\text{PKE}}$: used by one decryptor and arbitrary many encryptors.

$l: \{0, 1\}^* \rightarrow \{0, 1\}^*$ models leakage, e.g., $l_1: m \mapsto 1^{\mid m \mid}$

Main features:

- Has joint state realization
  - not true for other non-interactive formulations
- can be invoked
  - * an unbounded number of times
  - * with arbitrary long messages
  - * by an unbounded number of parties

```
if $l \neq 0$
  else
    $c := e(k'_e, m)$

$F_{\text{PKE}}$
```

$e$ and $d$ are provided by the simulator and are important for JS

missing in other formulations
Realizing $\mathcal{F}_{\text{PKE}}$ by CCA-Secure Encryption Schemes

**Theorem:**

$\Sigma$ is IND-CCA \quad $\quad \mathcal{P}_{\text{PKE}}(\Sigma) \leq \mathcal{F}_{\text{PKE}}(l)$

[K. and Tuengerthal 2008]

realization corresponding to $\Sigma$
Joint State Realization for PKE

Joint State Theorem for PKE: [K. and Tuengerthal 2008]

\[ \mathcal{P}_{\text{PKE}} \leq \mathcal{F}_{\text{PKE}}(l') \]

where \( l'(\langle \text{sid}, m \rangle) = \langle \text{sid}, l(m) \rangle \) (leakage of SID)

- Basic idea of \( \mathcal{P}_{\text{PKE}} \),
  similar to [Canetti, Rabin 2003] and [Canetti and Herzog 2006]
  - Encrypt \( \langle \text{sid}, m \rangle \) instead of \( m \)
  - Upon decryption check if plaintext is of shape \( \langle \text{sid}, m \rangle \)
    (else error)

- But proof has subtleties overlooked in other works
More Results

Joint State Theorem for replayable PKE:

\[ \mathcal{P}_{\text{PKE}} \xrightarrow{\leq} \mathcal{F}_{\text{RPKE}}(l') \]

where \( l'(\langle\text{sid}, m\rangle) = \langle\text{sid}, l(m)\rangle \) (leakage of SID)

plus realization for \( \mathcal{F}_{\text{RPKE}} \) for IND-RCCA secure \( \Sigma \)

(IND-RCCA introduced in [Canetti, Krawczyk and Nielsen 2003])

Joint State Theorem for Non-Interactive Digital Signatures:

\[ \mathcal{P}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]

\[ \mathcal{F}_{\text{SIG}} \xrightarrow{\leq} \mathcal{F}_{\text{SIG}} \]
Related Work

[Canetti, Rabin ’03]: – First to consider (general) JS theorem
  – JS theorem for interactive $\mathcal{F}_{\text{SIG}}$
  – but flawed

[Canetti ’05]: – Non-interactive $\mathcal{F}_{\text{PKE}}$ and $\mathcal{F}_{\text{SIG}}$
  – de facto interactive $\mathcal{F}_{\text{RPKE}}$
  – JS theorem claimed, without proof
  – but flawed

[Canetti, Herzog ’06]: – Non-interactive, parameterized $\mathcal{F}_{\text{PKE}}$
  – JS theorem claimed, without proof
  – but flawed

[CKN ’03]: Interactive $\mathcal{F}_{\text{RPKE}}$ (JS not considered)

[Pfitzmann, Waidner ’01], [Backes, Pfitzmann, Waidner ’03]:
  – Non-interactive parameterized $\mathcal{F}_{\text{PKE}}$ and $\mathcal{F}_{\text{SIG}}$
  – Unbounded number of copies of machines not considered
  – JS theorem not considered
Overview

- Subtleties in Simulation-based Models
- The IITM Model
- Motivating Joint State
- General Joint State Theorem

- Applications: new functionalities with joint state
  * Public-key encryption and digital signatures
  * Symmetric encryption
    - Several applications, including a new computational soundness result

- Conclusion
Functionalities have been developed for

- Digital signatures
- Public key encryption
- Key exchange
- Authentication
- Secure channel
- E-Voting
- Mix-Nets
- MPC
- . . .

But not for symmetric key encryption!

Except for a Dolev-Yao style functionality [Backes, Pfitzmann '04], see related work
Structure of the rest of the talk

1. Challenges for ideal symmetric key encryption

2. Our symmetric key encryption functionality

3. Applications of the functionality
   - Proving a protocol secure
   - Simplifying game-based proofs
   - Computational soundness for UC realization of key exchange

4. Related work
Challenges for Ideal Symmetric Key Encryption

- Symmetric keys may travel but must not be given to users
  - Users need **pointers to refer to keys**
  - The functionality needs to keep track of who knows which key (including the adversary)

- **Bootstrapping** required
  - e.g. from PKE or long-term (pre-shared) symmetric encryption

- Cryptographic challenges: **Key cycle** and **commitment** problems
Structure of the rest of the talk

1. Challenges for ideal symmetric key encryption

2. Our symmetric key encryption functionality

3. Applications of the functionality
   - Proving a protocol secure
   - Simplifying game-based proofs
   - Computational soundness for UC realization of key exchange

4. Related work
Our Symmetric Key Encryption Functionality

Two variants: **authenticated** and **unauthenticated encryption**

_bootstrapping component_

One instance per party

One instance per pair of parties
Our Symmetric Key Encryption Functionality

Long-term symmetric key encryption $\mathcal{F}_{\text{Itsenc}}$:

- **User commands (I/O):**
  - **Key exchange:** Ask for a key to be exchanged with other party.
  - **Encrypt $m$:** If corrupt: Return $c = \text{enc}(m)$
    Else: Return $c = \text{enc}(L(m))$ and record $(m, c)$
  - **Decrypt $c$:** If corrupt: Return $m = \text{dec}(c)$
    Else: Return $m = \begin{cases} m & \text{if recorded } (m, c) \\ \text{dec}(c) & \text{if variant } \mathcal{F}_{\text{unauth}_{\text{Itsenc}}} \text{ and } c \text{ not recorded} \\ \perp & \text{otherwise} \end{cases}$

- **Adversarial commands (network):**
  - **Provide:** encryption and decryption algorithms $\text{enc}(\cdot), \text{dec}(\cdot)$
  - **Corrupt:** static corruption

Leakage, e.g. $L(m) = 0_{|m|}$

No assumptions
Hence, abstract from algorithms
Our Symmetric Key Encryption Functionality

Long-term symmetric key encryption $\mathcal{F}_{\text{Itsenc}}$:

- $\Sigma$ symmetric encryption scheme
- Induces the obvious realization $\mathcal{P}(\Sigma)$ (no extra randomness or tagging)

**Theorem:**

- $\Sigma$ IND-CCA secure $\iff \mathcal{P}(\Sigma) \leq \mathcal{F}_{\text{unauth Itsenc}}$
- $\Sigma$ authenticated encryption scheme (IND-CPA + INT-CTXT secure) $\iff \mathcal{P}(\Sigma) \leq \mathcal{F}_{\text{auth Itsenc}}$
Our Symmetric Key Encryption Functionality

Long-term symmetric key encryption $\mathcal{F}_{\text{Itsenc}}$:

**Multi-session case:** Composition theorem yields

$\mathcal{P}(\Sigma) \leq \mathcal{F}_{\text{Itsenc}}$

**Impractical:**
Different key for each session

**Joint State Theorem:**

Encrypts $\langle \text{sid}, m \rangle$ instead of $m$

$\mathcal{P}_{\text{Itsenc}} \leq \mathcal{F}_{\text{Itsenc}}$

One instances per pair of parties

Holds for both variants **unauthenticated** and **authenticated** encryption
Our Symmetric Key Encryption Functionality

\[ \mathcal{F}_{\text{senc}} \]

\[ \mathcal{F}_{\text{pke}} \]

\[ \mathcal{F}_{\text{Itsenc}} \]

bootstrapping component
Our Symmetric Key Encryption Functionality

Short-term symmetric key encryption $\mathcal{F}_{senc}$:

**Pointer management:**

- For each party, mapping from pointers $(N)$ to keys (bit strings)
  
  $p \mapsto k_p$

- Plaintexts $m$ are arbitrary bit strings, may contain “$(\text{KeyPtr}, p)$”
  
  Replace “$(\text{KeyPtr}, p)$” by “$(\text{Key}, k_p)$” before encryption

  Replace “$(\text{Key}, k_p)$” by “$(\text{KeyPtr}, p)$” after decryption
  (create new pointers if necessary)

- Keep track of keys “(un)known” to adversary
Our Symmetric Key Encryption Functionality

Short-term symmetric key encryption $\mathcal{F}_{\text{senc}}$:

- User commands (I/O):
  - **Generate Key**: Adversary provides key $k$ (bit string)
    New pointer $p$ is returned to user
  - **Encrypt/Decrypt**: Encrypt $L(m)$ if key is "unknown",
    else encrypt $m$

A key is "unknown" if:

(a) it has been provided by the adversary,
(b) it is **not corrupt**, and
(c) it has never been encrypted by
  - a not "unknown" short-term key, or
  - a **corrupt** long-term or public key

- Adversarial commands (network):
  - **Provide algorithms**: $\text{enc}(\cdot, \cdot), \text{dec}(\cdot, \cdot)$
  - **Corrupt**: Static corruption (keys corruptible upon generation)
Our Symmetric Key Encryption Functionality

Realizing $\mathcal{F}_{\mathrm{senc}}$:
- $\Sigma$ symmetric encryption scheme
- Induces the obvious realization $\mathcal{P}(\Sigma)$
  (no extra randomness or tagging)

We would like to prove:

Impossible because of key cycle and commitment problems
Our Symmetric Key Encryption Functionality

Realizing $\mathcal{F}_{\text{senc}}$:

- Restricting the environment:
  1. Used order respecting: [Backes, Pfitzmann ’04]
     Keys ordered by first use
     Keys may only encrypt later keys
  2. Non-committing: An unknown used key must not become known

Main Theorem:

$\Sigma$ IND-CCA (resp., authenticated encryption) $\rightarrow \mathcal{P}(\Sigma) \overset{\prec}{\rightarrow} \mathcal{F}_{\text{senc}}$

\begin{align*}
\mathcal{F}^* & \quad \mathcal{F}^* \\
\mathcal{P}(\Sigma) & \quad \mathcal{F}_{\text{senc}} \\
\mathcal{F}_{\text{pke}} & \quad \mathcal{F}_{\text{itsenc}} \\
\mathcal{F}_{\text{pke}} & \quad \mathcal{F}_{\text{itsenc}}
\end{align*}
Our Symmetric Key Encryption Functionality

Realizing $\mathcal{F}_\text{senc}$:

Typically, a protocol $\mathcal{Q}$ enforces these restrictions: (e.g. Kerberos)

Corollary:

$\Sigma$ IND-CCA (resp., authenticated encryption)

$\mathcal{Q} \downarrow \mathcal{F}^* \downarrow \mathcal{P}(\Sigma) \downarrow \mathcal{F}_\text{senc}$

$\mathcal{Q} \downarrow \mathcal{F}^* \downarrow \mathcal{P}(\Sigma) \downarrow \mathcal{F}_\text{senc}$
Our Symmetric Key Encryption Functionality

Short summary:

1. **Broad application**, since low level:
   - Plaintexts are arbitrary bit strings
   - Only pointers are interpreted
   - Real ciphertexts are returned to users

2. **Natural realization and standard cryptographic assumptions**

3. **Modular design**
   - Realization of $\mathcal{F}_{\text{senc}}$ independent of realization of $\mathcal{F}_{\text{pke}}$ and $\mathcal{F}_{\text{senc}}$
   - Joint state theorems for $\mathcal{F}_{\text{pke}}$ and $\mathcal{F}_{\text{itsenc}}$ can be used
   - Modular proofs
Structure of the rest of the talk

1. Challenges for ideal symmetric key encryption

2. Our symmetric key encryption functionality

3. Applications of the functionality
   - Proving a protocol secure
   - Simplifying game-based proofs
   - Computational soundness for UC realization of key exchange

4. Related work
reduction proofs/
hybrid arguments
Applications: Proving a Protocol Secure

A variant of the Amended NSSK protocol:

1. \( B \rightarrow A : \{A, k_B\}k_{BS} \)
2. \( A \rightarrow S : A, B, n_A, \{A, k_B\}k_{BS} \)
3. \( S \rightarrow A : \{n_A, B, k_{AB}, \{k_{AB}, A\}k_B\}k_{AS} \)
4. \( A \rightarrow B : \{k_{AB}, A\}k_B \)

- Formulate as protocol system \( P_{NSSK} \) which uses \( F_{senc} \)

**Theorem:**

\[ P_{NSSK} \xleftarrow{\text{Authenticated}} F_{senc} < F_{ke} \]

Standard key exchange functionality
Applications: Proving a Protocol Secure

Proof of theorem:

• By joint state and composition theorem
  we only need to consider a **single instance** between $A$, $B$ and $S$

• If somebody is corrupt, simulator can corrupt $F_{ke}$ → done

• If all are uncorrupted:
  1. $B \rightarrow A$: $\{A, k_B\}_{k_{BS}}$
  2. $A \rightarrow S$: $A, B, n_A, \{A, k_B\}_{k_{BS}}$
  3. $S \rightarrow A$: $\{n_A, B, k_{AB}, \{k_{AB}, A\}_{k_B}\}_{k_{AS}}$
  4. $A \rightarrow B$: $\{k_{AB}, A\}_{k_B}$

1. $S$ uncorrupt  →  $k_{AB} = k'_{AB}$

2. All keys are unknown
   →  $k_{AB}$ encrypted ideally
   →  indistinguishable from random

Only possible plaintext in the functionality:

- $(A, k_B)$ created by $B$
- $(k_{AB}, A)$ created by $S$
- $(n_A, B, k'_{AB}, c)$ created by $S$
Applications: Simplifying Game-based Proofs

Secretive protocols: [Roy, Datta, Derek, Mitchell ’07]

- A protocol $P$ is **secretive** w.r.t. a key $k$ if $k$ is only sent “properly” encrypted
- $P$ secretive w.r.t. $k$
  - $\Rightarrow$ **key usability** for $k$
  - $\Rightarrow$ **key indistinguishability** for $k$ if $k$ is not used in protocol

Using $F_{senc}$:

- Definition: $P$ is **secretive** w.r.t. a short-term key $k$ if $k$ is always “unknown” and non-committing, used order respecting
Applications: Simplifying Game-based Proofs

**Theorem:**

\( \mathcal{P} \) is secretive w.r.t. \( k \)

\[ \implies \textbf{key usability for } k \]

\[ \implies \textbf{key indistinguishability for } k \text{ if } k \text{ is not used in protocol} \]

**Proof:** (for key indistinguishability)

In the \textbf{ideal} world:

- \( k \) is never used
- \( k \) is only encrypted ideally

\[ \implies k \text{ is indistinguishable from random} \]

\[ \implies k \text{ is indistinguishable from random in the \textbf{real} world} \]

Proof for key usability: Similarly simple
Using $\mathcal{F}_{\text{senC}}$ for Computational Soundness

New Result:

Computational soundness for realizing key exchange protocols with symmetric encryption in a universally composable way
(dishonestly generated keys are allowed)

related work [Canetti, Herzog, 2006]
Using $\mathcal{F}_{\text{senC}}$ for Computational Soundness

Class of protocols we consider:

- Symbolic protocol with symmetric encryption, pairing, and nonces, similar to Comon-Lundh, Cortier, 2008
  (extension with public key encryption should be easy)

- But with branching (if-then-else) and mild tagging in the realization

- Protocol describes single session
  (use joint state theorem for multiple sessions)

- Protocols have to satisfy the following symbolic criterion:
  If all parties involved in the session are uncorrupted, then all short term keys used by the parties are secret (i.e., cannot be derived in the symbolic sense).
Using $\mathcal{F}_{\text{senc}}$ for Computational Soundness

**Theorem.** For a protocol $\mathcal{P}$ as above. If

$$\mathcal{P} \sim_o \text{rand} (\mathcal{P})$$

then

- **Information theoretic proof**
- **Observational equivalence**
  can be checked automatically

**Standard key exchange functionality**

- **Authenticated encryption version**
Using $\mathcal{F}_{\text{sen}}$ for Computational Soundness

**Corollary.** For a protocol $\mathcal{P}$ as above. If

$\mathcal{P} \sim_0 \text{rand}(\mathcal{P})$ and used order respecting (no key cycles)

then

Immediate consequence of composition theorem
Using $\mathcal{F}_{\text{senC}}$ for Computational Soundness

**Corollary.** For a protocol $\mathcal{P}$ as above. If $\mathcal{P} \sim_o \text{rand}(\mathcal{P})$ and used order respecting then

![Diagram](diagram.png)

Immediate consequence of composition theorem

**But impractical:** new long-term keys are used for every session

**Practical realization:** replace multi session version of $\mathcal{F}_{\text{itsenc}}/\mathcal{P}_{\text{itsenc}}$ by joint state realization
Structure of the rest of the talk

1. Challenges for ideal symmetric key encryption

2. Our symmetric key encryption functionality

3. Applications of the functionality
   - Proving a protocol secure
   - Simplifying game-based proofs
   - Computational soundness for UC realization of key exchange

4. Related work
Related Work

Backes, Pfitzmann '04:
- Cryptographic Library
- Abstract Dolev-Yao style interface

More abstract reasoning
- Have to consider multi session case
- Limited to operations in the library
- Realizable only by non-standard encryption schemes
  - Extra randomness
  - Adding identifiers for symmetric keys

- Only works for honestly generated keys
  (restricted class of adversaries) [Cortier, Comon-Lundh, 2008]

Other (not simulation-based):
- Soundness of Dolev-Yao style reasoning: [Abadi, Rogaway '00] [Laud '04] [Comon-Lundh, Cortier '08]
- Formal logic for reasoning: [Datta, Derek, Mitchell, Warinschi '06]
Structure of the rest of the talk

1. Challenges for ideal symmetric key encryption

2. Our symmetric key encryption functionality

3. Applications of the functionality
   - Proving a protocol secure
   - Simplifying game-based proofs
   - Computational soundness for UC realization of key exchange

4. Related work
Overview

- Subtleties in Simulation-based Models
- The IITM Model
- Motivating Joint State
- General Joint State Theorem
- Applications: new functionalities with joint state
  * Public-key encryption and digital signatures
  * Symmetric encryption
    - Several applications, including a new computational soundness result
- Conclusion
Conclusion

- Models for simulation-based security do not have to be complicated

- Simulation-based security is very useful
  - Modular design
  - Simpler analysis

- ... also in game-based settings
FCC 2009: Call for Papers

Workshop on Formal and Computational Cryptography

July 11-12, 2009, Port Jefferson, New York, USA

affiliated with CSF 2009

Deadline for submission: April 30, 2009

Submissions: extended abstract (1 page)
- Papers published elsewhere
- New ideas and work in progress